Universitat Autónoma de Barcelona

Trustworthy Task Allocation for Human Teams

Athina Georgara

Barcelona, 2023

In collaboration with:

Trustworthy Task Allocation for Human Teams

A dissertation submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy in

Computer Science *awarded by the* **Universitat Autónoma de Barcelona**

In collaboration with:

Artificial Intelligence Research Institute, CSIC *and*

Enzyme Advising Group

Barcelona, November 2023

Athina Georgara (Αθηνά Γεωργαρά)

Supervisors

Professor Carles Alberto Sierra Garcia
Professor Juan Antonio Rodríguez Aguilar
Ph.D. Luís Manuel Artiles Martínez

Tutor

Professor Carles Alberto Sierra Garcia

UAB
Universitat Autònoma
de Barcelona

CSIC
Consejo Superior de Investigaciones Científicas

IIIA
Institut d'Investigació en
Intel·ligència Artificial

enzyme

Abstract

Task allocation for human teams is of paramount importance in a plethora of real-world settings. Teams bring together individuals with different competencies, interests and perspectives, enabling them to tackle complex challenges that a single person cannot handle due to lack of resources (e.g., knowledge and skills) or time. Effective teamwork fosters a sense of belonging, shared purpose, and commitment among team members, driving them to put in extra effort, remain focused on their goals, and ultimately reach high-quality outcomes. From workplaces to educational settings and community activities, forming and allocating teams is crucial for achieving success. In this dissertation, we tackle the problem of *trustworthy task allocation for human teams*. Specifically, we contribute by putting forward tools to aid the process towards effective teamwork.

First, we review the literature regarding teams and team formation across several scientific domains, including Computer Science, Organisational Psychology, Motivational Psychology and Social Sciences. We study on which bases teams are formed in the different scientific areas, and we explore which human characteristics influence teamwork and team performance.

Second, we use the findings from the literature and we put together important human characteristics that benefit teamwork. We propose metrics that allow us to evaluate a team across these characteristics. In particular, we discuss how to aggregate from an individual level to a team level several human characteristics such as competencies, personality, gender, preferences and interpersonal relations. We propose four such aggregating metrics, namely the *competence affinity*, *congeniality*, *motivation* and *social cohesion*. We also introduce *collegiality*, a metric that considers the beneficial-to-teamwork individual characteristics and can be used as a predictor for team performance.

Third, we study the problem of forming teams. In particular, we focus on settings that involve multiple tasks and require teams that each team works on a different task, while each individual can participate in at most one team. Hence, we define the *Non Overlapping Many Teams to Many Tasks Allocation Problem (NOMTMT-AP)*. We show that the NOMTMT-AP is $\mathcal{NP}$-complete, and we put forward two algorithms for solving the problem: an optimal solver and Edu2Com, an anytime heuristic solver. We conduct a manifold empirical evaluation. Our evaluation allowed us to study *(i)* the quality, runtime and anytime behaviour of Edu2Com when pitched against the optimal solver, *(ii)* the solubility of Edu2Com along with the limitations of the optimal solver, and *(iii)* the team performance when the teams are formed considering the individuals' competencies, personality, gender, preferences and interpersonal relations.

Fourth, towards trustworthiness, we address the problem of explaining why a team formation algorithm formed the teams it outputs and not others. In this direction, we identify a collection of questions that are intuitive and meaningful and cover the main points of interest regarding team formation scenarios. Then, we introduce a general explanatory algorithm that can wrap an existing team formation algorithm without modify-

ing it and build contrastive explanations. We conduct an empirical evaluation and show that our algorithm builds contrastive explanations are easy to understand, requiring just the reading level of a high-school student. Along with explaining team formation scenarios, we turn our attention to a vital challenge regarding explanations. Specifically, we address the problem of preserving privacy upon providing explanations. In this light, we put forward a privacy breach detector that assesses whether an explanation is bound to reveal private information. Finally, we propose a general framework that describes the interactions between a team formation algorithm, an explanatory algorithm and a privacy breach detector.

Acknowledgements

The past four years have been an amazing journey, filled with moments of knowledge, growth, and discovery. I would like to express my deepest gratitude to those who have played a crucial role in completing my PhD thesis.

First and foremost, I want to thank my academic supervisors, Juan Antonio Rodríguez-Aguilar and Carles Sierra. I am forever grateful for giving me the opportunity to work by your side and learn from you and our collaboration. Thank you for putting your trust in me and guiding me through this rough journey of the PhD. I would never be able to reach the end without you and your support. Thank you, Juan Antonio, for always being there for me and helping me even with stupid bureaucratic things (due to my complete incapability of learning Spanish). Thank you, Carles, for always having at least one solution for every problem. You have been great mentors and a huge inspiration for me; thank you for helping me become the research I am today. I am deeply grateful for having both of you as my academic parents.

I would also like to thank my supervisor from Enzyme Advising Group, Luís Artiles Martínez, and all the people in Enzyme who accepted me and always made me feel like one of them. Luís, thank you for always being patient with me, worried about my well-being and super supportive of the research directions I took.

This dissertation would not have been completed without the help and support of many collaborators. I would like to thank our partners from the AI4EU project Ornella Mich, Raman Kazhamiakin, Alessio Palmero Aprosio from the Fondazione Bruno Kesler and Jean-Christoph Pazzaglia from SAP, who helped us conduct empirical evaluations in the hard time of COVID-19. I would like to thank Georgios Chalkiadakis from the Technical University of Crete for trusting us and offering one of his courses to run experiments. This evaluation would have never been realised without the help of Nikolaos Papas and Spyros Argyropoulos from the Technical University of Crete. I would also like to thank Mireia Montané Balagué and Antony Lewis Poole from EADA Business School, who organised for us an empirical evaluation with their students. From the Artificial Intelligence Research Institute (IIIA-CSIC), I would like to thank Bruno and Joan, who prepared the platforms for hosting my algorithms; and, of course, Lissette helped with courses and communicating my work.

I want to express my deepest gratitude to Georgios Chalkiadakis for trusting me ever since my undergraduate studies, encouraging me to pursue a PhD and introducing me to Carles and Juan Antonio. Thank you for being my mentor, my collaborator, and becoming my friend.

The last four years wouldn't have been the same without the old friends and the new ones I made. Far away from home, Dimitra, Roger, Thiago, Shuxien, and Jairo, you became my family. Thank you, Roger, for helping me visit all the kebab restaurants in Barcelona. Thank you, Dimitra and Thiago, for the interesting debates about life and society. Thank you, Jairo and Shuxien, for the long philosophical conversations during lunch and coffee breaks. Thank you, Thalia, Marina and Eleftheria, for always supporting me. Thank you, Errikos and Dimitris, for the collaboration and the discussions.

Finally, above all, I want to thank my family. To my mom, Lina, my dad, Christos, my sister, Victoria, and my brother-in-law Giannis, who endlessly supported me and encouraged me to pursue my dreams. And, last but not least, to my nephews Christos and Giorgos, and my godson, Nikolas, whom I am determined to turn into young researchers one day.

Athina Georgara

Research has been partially supported by an Industrial PhD scholarship from the Generalitat de Catalunya (2019DI17), and the European projects AI4EU, Yoma OR, and VALAWAI.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Human teams play a crucial role in modern societies. People's collaborations and teamwork are behind significant accomplishments. What makes teams so unique is their ability to bring together people with different skills and perspectives. When individuals join forces, they can achieve great things. That is, they can tackle complex problems that a single person cannot handle, due to lack of power or time. Moreover, working in teams is much more than carrying out a complex task. Working with others gives people a sense of belonging and a shared goal. This sense of belonging leads people to put in extra effort while working, stay focused on their goal, and not give up easily, especially when they comprise well-tuned teams. The collaborative nature of teams creates an environment that fosters fresh, innovative ideas and promotes continuous improvement both at an individual and a group level. Additionally, teamwork is instrumental in promoting inclusivity and diversity. Combining different backgrounds and perspectives within teams offers a fruitful field for creativity and promotes critical thinking, amplifying the sense of being included. In essence, human teams are the backbone of progress as a society, helping communities reach hard goals and great achievements.

Moreover, many real-world problems require allocating teams of individuals (not necessarily humans) to tasks. For instance, forming teams of robots for search and rescue missions [Capezzuto et al., 2020], forming teams of Unmanned Aerial Vehicles (UAVs) for surveillance [Ponda et al., 2015], building teams of people to perform projects in a company [Ballesteros-Perez et al., 2012, da Silva and Krohling, 2018], or grouping students to undertake school projects [Andrejczuk et al., 2019]. We illustrate the problem and our results in the domain of education. In schools and educational institutes, it is prevalent that students work in teams, i.e., students collaborate with their teammates towards some common goal, such as their homework, semester projects etc. For example, in primary and secondary schools, teachers usually need to divide their students into study groups (teams) to carry out some school projects. Similarly, in universities, students are usually requested to work in teams in order to carry out their semester projects. Moreover, educational authorities often need to form student teams and match them with internship programs, as it is more and more common for students to spend time with companies to gain experience in the industry. Current practices require teachers and education authorities form student teams mainly by hand. However, given the *combinatorial* nature of the problem, solving manually such a problem requires a large amount of work, especially as the number of agents and tasks grows.

Beyond classroom activities, the problem of allocating non-overlapping teams to tasks can also be found in events and competitions where participants need to work in teams and compete with each other, such as hackathons. Similarly, we find the non-overlapping allocation problem in events where different teams need to work in parallel, tackling perhaps complementary tasks—for example, in search and rescue missions where individuals cannot be in multiple teams simultaneously. Thus, forming human teams is vital for dealing with many every-day situations, however, currently the team formation process is done mainly by hand. Therefore, we see the necessity for developing artificial intelligence tools that aid the team formation process.

One step further, using AI-based tools to support ever-day life decisions often make people to be more reluctant and show a feeling of 'distrust' towards the decisions made. In other words, people tend to have doubts concerning the validity of a decision made by an AI system. Explainable artificial intelligence tries to settle such doubts by providing the users with the rationale and the causes that led to certain decisions. Notably, deciding how to form teams with the help of AI algorithms is not an exception in this matter, especially when teams consist of humans. That is, when humans are involved in a team formed with some AI tool, they need to be sure that their team is suitable for them in order to accept participating in the team. Hence, providing explanations on why an AI-based team formation algorithm outputs some team instead of another is necessary for engaging people to collaborate with their teammates.

We structure this chapter as follows, first, we thoroughly discuss the motivation for this dissertation, exploring the open challenges found in the literature. Then, we put forward the research questions that we will tackle in this thesis and elaborate on our contributions. Finally, we provide the road map for the rest of the thesis and the conference and journal publications made from this research.

1.1 Motivation

In this section, we discuss the reasons motivating this research. This thesis explores how to allocate tasks to human teams. This line of research derives from three pillars of motivation. The first pillar regards real-world application areas where we must form human teams and assign working tasks. The second pillar of motivation focuses on the open challenges in the existing literature concerning task allocation and human team formation. Moving to the third pillar of motivation, we come across an essential challenge in the team formation problem that needs to be addressed: providing explanations regarding the teams formed by a team formation algorithm.

1.1.1 Human Teams in Real-World Domains

In a plethora of real-world settings, people join forces when working on some project or task. There are many reasons why people start collaborating with each other. For example, an individual alone may not have the power, resources or time to complete a given task. Another reason is related to *quality* and *efficiency*. That is, a group of people jointly working on the same task results in high-quality outcomes faster than an individual working alone. Moreover, teamwork is associated with a positive potential for personal growth. When people collaborate, they share knowledge and exchange ideas, allowing them to *learn* from each other [Bruffee, 1993] and advance communicating and socialising skills. Given these benefits from collaboration and teamwork, many real-world domains tend to form human teams to work on projects or tasks.

An application area that widely employs teamwork is the *educational domain*. In all educational levels (primary, secondary and higher education), students are put in teams to do homework, study or carry out school projects. For example, in primary and secondary schools, teachers usually need to divide their students into study groups (teams) to carry out some school projects, activities or homework. Similarly, in higher-level education (both in pre-graduation and post-graduation), students work in teams to carry out semester projects or research work. Moreover, it is common for students (either in a secondary or higher level) to spend time in companies to gain industry experience. Recently, the practice of forming student teams to work on internship programs has gained much attention. Teachers and educational authorities invest in teamwork to promote youngsters' personal growth through collaborative learning. The current practice in schools is to form student teams by hand, with educational personnel spending many working hours in this activity. Forming teams manually often results in "poor-quality" teams since there is a huge number of different teams to choose from.

Beyond education, another application area prying in teamwork is *industry*, and mainly project-based companies. Over the past decades, the private sector has focused on promoting teamwork. Industry has been aiming to form efficient teams to work on company projects or even to staff their departments/branches/stores. Even from the very foundation of a company, building the company's core team efficiently is of utmost importance [Foss et al., 2008]. For example, the core team of a startup company can be a key factor in the company's prosperity [Thirasak, 2020]. More and more companies adopt a team-based orientation and promote collective effort instead of individual achievements [Levi, 2001]. In fact, according to PwC's HR Technology Survey 2020,[1] 40% of companies think they need tools that allow them to create collaborative work environments. Teamwork within the workplace boosts efficiency and pro-

[1] `https://www.hrmanagementapp.com/pwc-hr-technology-survey-2020/`

ductivity [Richter et al., 2011, Khawam et al., 2017]. By utilising teamwork, companies achieve better quality in their products or services (by combining their employees' skills and expertise), reduce projects' duration and delivery time (by sharing workload among the team members), and train their employees via collaborative learning.

In the information age, organisations (both in the public and private sectors) seek external ideas, knowledge and expertise from a community further than the organisation's own resources. *Open innovation* events and challenges aim to bring together people towards sharing and receiving information, ideas, and knowledge. Working in teams is essential for innovation[Kirschbaum, 2005]. People can build upon each other's ideas, leading to more innovative and refined concepts. Collaborative environments foster creativity and the emergence of novel approaches. Teamwork enables participants to receive constructive feedback from one another. This feedback loop allows for continuous iteration and improvement of ideas, increasing the likelihood of generating high-quality solutions.

Similarly, *crowdsourcing* events address a crowd of experts or even the general public intending to achieve a specific goal or solve a particular problem. Crowdsourcing refers to the collective effort and collaboration of a diverse group of individuals working jointly towards a common objective. Individuals from different backgrounds, experiences, and expertise come together as a team, either voluntarily or through organised platforms, to tackle a wide range of tasks. Such tasks include content creation contests, idea generation challenges, market research and surveys, software development challenges, innovation and technology challenges, and social and humanitarian projects. In a similar direction, *volunteering* and *social impact tasks* depend on groups' collaborating towards the public good.

This dissertation considers education as its primary application domain. Nonetheless, we believe that the generality of the approach proposed in this disseration might also be valuable for a wide range of application domains.

1.1.2 Open Challenges in the Team Formation Literature

In the field of artificial intelligence and multi-agent systems, the problem of team formation has attracted much attention. The team formation problem is about building one or more teams so they can work towards one or more common tasks. Literature offers a plethora of algorithms tackling the team formation problem [Juárez et al., 2021]. The majority of algorithms focus on selecting one group of individuals (out of a larger population) who are to jointly work towards a common goal [Lappas et al., 2009, Wi et al., 2009, Anagnostopoulos et al., 2010, Li and Shan, 2010, Kargar and An, 2011, Anagnostopoulos et al., 2012]—i.e., these algorithms form a single team for a single task. We can also find algorithms for building a single team to work on many tasks [Crawford et al., 2016]—i.e., the very same group of people shall work on a series of different tasks. There are also algorithms forming many different teams to work on a single task [Andrejczuk et al., 2019]—i.e., several groups of individuals with each group working on the same task jointly. One step further, there is also a handful of research on how to form multiple teams to resolve multiple tasks. There are algorithms for forming *overlapping* teams—i.e., teams where an individual can participate in more than one team—to work on different tasks [Capezzuto et al., 2020], and algorithms for forming teams where different teams can work on the very same task [Bachrach et al., 2010].

However, there needs to be more attention given to the problem of how to form teams to work on different tasks without overlaps, i.e., teams that share no common members and tasks that are uniquely tackled by one team. In many real-world applications, overlaps are not permitted. Considering, for example, the educational domain—which is the application domain of this thesis—each student is part of precisely one team, working on exactly one project/task, and each project/task is tackled by exactly one team. Moreover, in many real-life cases, we need to form several teams which are to work on a different task each, *in parallel*. For instance, in social impact task events, all the different teams need to work on their tasks at the same time (e.g., due to deadline constraints). In such cases, each individual cannot, in practice, participate in more than one team. Despite limited existing research on this topic [Czatnecki and Dutta, 2019, Präntare and Heintz, 2018], we cannot rely on prior approaches to solve the problem of forming non-overlapping teams for many tasks due to several shortcomings. On the one hand, [Präntare and Heintz, 2018] uses brute force and branch-and-bound techniques to form teams. Given, though, the combinatorial nature of the problem, the brute force technique limits the number of agents and tasks that the approach can handle. On the other hand, [Czatnecki and Dutta, 2019] handles settings where there exist Nash stable solutions. However, in the general case, the existence of Nash stable solutions is not guaranteed. As such, we need to devise new approaches to efficiently build multiple human teams to tackle many different tasks that overcome the aforementioned shortcomings.

In most works tackling the team formation problem, regardless of whether the corresponding algorithm forms a single or multiple teams for a single or multiple tasks, the final teams to be formed are decided upon the teams' competencies. That is, the majority of the existing team formation algorithms decide which team(s) to form according to individuals' expertise, knowledge or skills. As noted in [Andrejczuk et al., 2019], the literature on team composition and formation considers either a Boolean model of competencies (an agent has or has not a competence) [Lappas et al., 2009, Anagnostopoulos et al., 2010, Anagnostopoulos et al., 2012, Czatnecki and Dutta, 2019]; or a graded model (an agent has a competence up to some degree) [Chalkiadakis and Boutilier, 2012, Andrejczuk et al., 2018, Andrejczuk et al., 2019]. Common to all these models is the assumption that a team assigned to a task must possess the competencies exactly as required by the task. This is rather limiting to cope with real-world problems. Instead, it might be the case that it is sufficient to acquire some similar competence to handle a specific required competence. For instance, consider the example from the educational domain below. It is common for students to participate in *internship programs*, i.e., spend some time in companies to earn working experience. A student may not possess all the required competencies precisely as requested by some internship program. However, they may be adequate for the internship as long as the student's competencies are similar enough to the required ones. So far, the *semantic relationship* between competencies has been disregarded when forming teams. This prevents, for instance, that a team is formed to work on a task requiring competencies similar to those offered by the team.

The majority of existing algorithms concern building teams of agents, where agents can be humans, or robots or software agents. However, when we form human teams, in particular, we need to take into consideration the human features, including characteristics that have either a positive or a negative influence on people's collaborations. Research in psychology and social sciences shows that many of characteristics, such as acquired and endogenous characteristics, desires, and beliefs, drive people's behaviour during teamwork. In this way, the performance of a team depends on its members, and specifically on its team members' human features. The majority of the existing algorithms tend to neglect the human nature of individuals. Instead, as we previously discussed, state-of-the-art algorithms solely focus on people's skills, qualifications, and expertise. With the exception of the work in [Andrejczuk et al., 2018, Andrejczuk et al., 2019], which considers people's intelligences and personalities, state-of-the-art team formation algorithms usually consider either some arbitrary *value function* [Präntare and Heintz, 2018, Capezzuto et al., 2020]—i.e., a function that determines the value of a team working on a task without specifying how this value has been obtained—or a *skill-based function* [Lappas et al., 2009, Anagnostopoulos et al., 2010, Bachrach et al., 2010, Li and Shan, 2010, Kargar and An, 2011, Anagnostopoulos et al., 2012, Czatnecki and Dutta, 2019], usually accompanied with some communication and/or transportation cost.

1.1.3 The Right to Explanation

Artificial intelligence is used to solve hard, complex and time-consuming problems. As humans' decisions depend more and more on AI-assisting tools, people are becoming curious about the rationale and the methodology of these tools. Thus, we observe a new surge of interest in explaining how an AI system reaches a specific decision. Providing such explanations is necessary in order to earn the users' trust regarding the AI tool at hand [Miller, 2018]. Over the recent years, the EU has taken several actions and put forward legislation highlighting the *right to exaplanations* [Goodman and Flaxman, 2017]. As [Ramchurn et al., 2021] discusses, developing trustworthy-by-design systems to facilitate human-AI partnerships is a necessity. When people use AI-assisted tools to resolve a complex task and provide personal data to such an AI tool, it is their right to know why AI reaches the solutions provided and how their data has been used.

Forming human teams to allocate tasks is a complex and time-consuming problem, making AI-driven team formation algorithms very useful in several domains, as we discussed in Section 1.1.1. In the context of team formation, we can discern two main roles: *(i)* the role of the *team maker* and *(ii)* the role of a *team member*. On the one hand, the team maker needs to trust that the proposed team(s) by the AI tool is suitable for the team maker's needs. The team member needs to understand why they are participating in a certain team, why they have to collaborate with specific people, and why the AI assigned them a particular task. On the other hand, the team member needs to understand why they are participating in a certain team, why they have to collaborate with specific people, and why the AI assigned them a particular task. The team maker (who is responsible for the teams) needs to understand the criteria and the rationale with which the team formation algorithm formed the teams. Consider, for example, a classroom scenario. Here, the teacher corresponds to the team maker, the students correspond to the team members, and say that the team maker uses an AI-based team formation algortihm for distributing their students into study groups. Then each of the students may question the AI algorithm's result and ask why they have been put in a certain team or why they should work on a specific task (e.g., a school project). At the same time, the teacher needs to be sure that the proposed teams serve the students' best interest, and therefore they may also question the result by asking, for example, why a certain team was formed or why two particular students are not in the same team. Being able to explain why the AI team formation algorithm reached certain decisions is crucial towards earning the team maker's and the team members' trust.

[Goodman and Flaxman, 2017] notes that people have the right to know how their personal data is used by an AI algorithm, and therefore we need transparent algorithms that provide explanations regarding their decisions. Recently, [Kraus et al., 2020] have raised awareness on the need for explanations in multiagent environments (xMASE), and they have identified the key challenges towards xMASE. Among other challenges—such as the development of appropriate algorithms for generating explanations and user modelling to appropriately tailor explanations and increase user satisfaction—Kraus et al. refer to the issue of non-disclosing private data and information. In the context of team formation, the corresponding algorithms require access to team members' personal data that will be used in order to build the desired team(s), and use this data to explain how and why they formed specific teams. This data may include private information such as people's skills, educational level, interests, desires, personality, religion, age, nationality, current location, etc. Therefore, it is essential for team formation algorithms to preserve privacy upon providing explanations[2] and ensure that team members' private information is not disclosed to third parties.

1.2 Research Questions

The problem we address in this thesis opens questions across three directions. These three directions encompass *(A)* how to model human agents, *(B)* how to form multiple teams for multiple tasks, and *(C)* how to provide explanations concerning the teams formed. In the following three

[2] Here, we make the assumption that during the process of forming teams, there is no reason for the algorithm to provide sensitive information regarding the participants to anyone, including the team maker.

subsections, we unfold the research questions we tackle in this thesis with respect to the aforementioned three directions.

1.2.1 Modeling human agents

This thesis focuses on forming human teams, i.e., working teams consisting of human beings. However, we must *model* the individuals before forming any team. As noted in [Andrejczuk, 2018], computer science and organisational psychology have followed separate paths regarding team formation. On the one hand, computer science, which offers several team formation algorithms, focuses on automating the team formation process and usually overlooks findings related to organisational psychology. On the other hand, organisational psychology mainly focuses on analysing human behaviour during teamwork and ignores the computational challenges of building teams. Following the work of [Andrejczuk, 2018], we argue that other domains, such as motivational psychology and social sciences, offer valuable research paths regarding teamwork and team building. As such, we see the potential to be brought to human team composition by bridging the gap between computer science, organisational and motivational psychology, and social sciences. Our first research question is:

> **Question Q.A1 :** Which human aspects identified in organisational psychology, motivational psychology and social sciences should be considered when building teams?

Our research question Q.A1 will provide us with a list of human characteristics that influence people's behaviour during teamwork and, therefore, should be considered when one forms teams. After that, we need to model human agents in a way that captures all the identified characteristics in order to consider them while forming teams. Hence, the next research question arises:

> **Question Q.A2 :** How can we model the identified beneficial-to-teamwork human characteristics and, therefore, human agents?

On top of that, one characteristic has been acknowledged as highly important and considered by most of the existing team formation algorithms. This characteristic is no other than people's skills and competencies. However, as discussed in Section 1.1.2, the existing competence models are somewhat limiting. Currently, a team is considered adequate for tackling a task (either in Boolean "adequate or non-adequate" mode or adequate up to a degree) if and only if the team offers the exact same competencies as the ones required by the task. We argue that a team can be adequate for a task as soon as the team offers *similar enough* competencies to the ones required by the task. This observation leads to the following research question:

Question Q.A3 : How can we define semantic similarities between competencies to characterise a team's competency for tackling a task?

1.2.2 Forming multiple teams for multiple tasks

This thesis studies how to build human teams, and specifically, we focus on how to form many teams to tackle many tasks, disallowing overlaps. This is a real-life problem we find in several domains (education, industry, crowdsourcing, etc.). To solve our team formation problem, first, we need to determine the complexity of the problem and the computational cost of forming multiple teams to work on various tasks. We claim that the problem is hard to solve, especially as the number of agents and tasks increases.

Question Q.B1 : What is the computational cost of forming multiple teams for multiple tasks with no overlaps?

Next, we need to find methods to form several human teams while, at the same time, exploiting the beneficial-to-teamwork characteristics that we identified in reply to research question Q.A1. As already discussed, we can hardly rely on the existing approaches in the computer science literature since they either do not scale up (e.g. because they employ brute force search) or place strong assumptions that cannot always be met (e.g. seeking Nash-stable solutions). Hence, we must investigate *how* we can form many promising, non-overlapping teams to work on various tasks and devise efficient methods for that.

Question Q.B2 : Can we efficiently form multiple promising teams for multiple tasks with no overlaps?

1.2.3 Explaining teams and task allocations

Last but not least, this thesis explores an important and challenging problem related to team formation that must be addressed. In the last part of this thesis, we turn our attention to explaining why our team formation algorithm formed the teams it outputs and not others. Providing explanations helps people to understand and trust the team formation algorithm that produced the teams and consequently to embrace the teams. Recently, [Kraus et al., 2020] highlighted the necessity of providing explanations in environments involving many parties (i.e., multi-agent environments), such as team formation scenarios. Despite the need for explanations pointed out by Kraus et al., to the best of our knowledge, the problem of explaining team formation decisions has not been addressed yet. Hence, in this thesis, we try to make headway in this matter. To begin with, we must first identify what questions one may ask within the context of team formation (and task allocation).

Question Q.C1 : What are the typical queries that team members and team makers will pose?

After that, we need to investigate how to build explanations. According to [Miller, 2018], people prefer to give and receive *contrastive* explanations. That is, people understand and accept answers to questions of the form "Why X instead of Y?" or "Why X and not Y?" more easily. Acknowledging this observation, we study how to build such contrastive explanations, considering the questions relevant to team formation. On top of that, we remind the reader that the explanations target to enlighten an explainee (i.e., a team member or a team maker) about the rationale followed during team formation. Thus, the explanation should be easy to read and comprehend by explainees. As such, the following research question arises.

Question Q.C2 : How can we build contrastive and comprehensive explanations?

Building an explanation is not trivial. We argue that computing some queires' explanations may be more challenging than others. That is, depending on the query posed, we may require more computational effort and/or time in order to build a contrastive explanation. Thus, this thesis aims to determine the computational cost of building contrastive explanations for team formation scenarios. This leads to the following research question:

Question Q.C3 : What is the computational cost of explanations?

As already discussed, many team formation algorithms exist in the literature. The several algorithms solve the team formation problem differently. For instance, some form a single team, while others form multiple teams. Regardless of each algorithm's different approaches, providing explanations is essential to all team formation algorithms. Consequently, the necessity of developing a general methodology to build explanations that *any* team formation algorithm can follow arises. We argue that our proposed method for building contrastive explanations is general and can accompany any team formation algorithm.

Question Q.C4 : Is there a general-purpose framework for building explanations for team formation algorithms?

Finally, [Kraus et al., 2020] points out several challenges towards explanations within multi-agent systems. Among others, the issue of preserving privacy is raised, especially since many agents are involved in providing their private data. In addition, legislation and actions similar to the GDPR within the EU lead to the *right to explanation*, which in turn leads to the right of *data protection* and *privacy preservation*. In the context of team formation, and specifically upon providing explanations, we argue that team members' privacy must not be breached. Instead, an AI system should only offer explanations that are guaranteed not to disclose private data. As such, our final research question regards whether we can preserve people's private information.

Question Q.C5 : Can an explanatory algorithm preserve team members' private information?

1.3 Contributions

Given the research questions presented above, this thesis addresses these questions and tries to make headway towards providing answers and solutions. In what follows, we discuss our contributions with respect to the research questions.

Addressing the research questions regarding the modelling of humans, we thoroughly review the existing literature in organisational psychology, motivational psychology and social sciences. In response to questions Q.A1 and Q.A2, we discern several characteristics that have been identified to promote teamwork and team performance, and we formally define those characteristics in order to consider them during the team formation process. In fact, we exploit characteristics such as people's competencies, personality, gender, preferences and interpersonal relations as a compass to determine a *team's collegiality*, i.e., to measure the companionship and cooperation between colleagues within a working team. In some detail, we formally define how to model humans as agents consisting of their several characteristics, we formally define tasks, and we formally define a *human team* as a group of people who are inextricably linked to a task. Regarding research question Q.A3, we embrace competence ontologies and propose a method for computing the semantic similarity between two competencies in the given ontology. In addition, we put forward a novel method to characterise a team's *competence affinity* to a task based on the team's collectively offered competencies and the ones required by the task. In a nutshell, by tackling the research questions regarding modelling human agents, we make the following contributions:

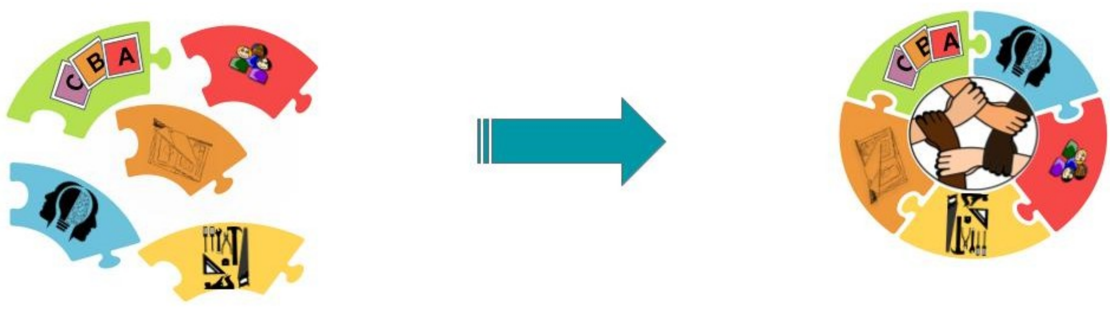


Figure 1.1: Beneficial human characteristics for teamwork and team performance.

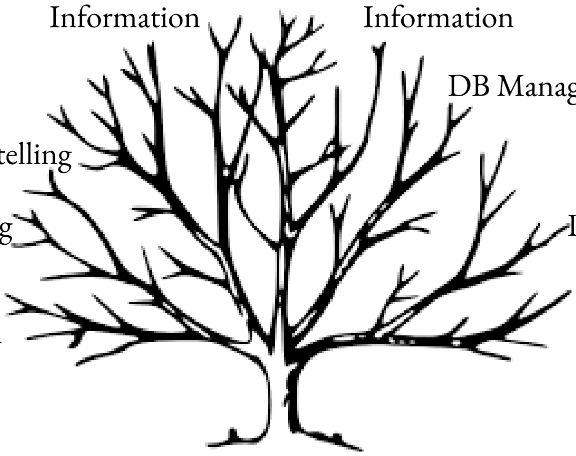


Figure 1.2: Scemantic similarities among competencies.

1. We identify key human characteristics that influence teamwork.

2. We formally define *(i)* human agents with respect to their characteristics (competencies, personality, gender, preferences and interpersonal relations); *(ii)* tasks with respect to their requirements in terms of competencies and desired team size; and *(iii)* human teams as a group of people who are inextricably linked to a task.

3. We develop metrics to evaluate the quality of a team in terms of their collective competencies, personality diversity, motivation, social cohesion and, ultimately, collegiality.

4. We adopt the concept of a competence ontology and use it to measure semantic similarities between different competencies.

Towards research questions Q.B1 and Q.B2, we formalise the *non-overlapping many teams to many tasks allocation problem* (NOMTMT-AP) as an optimisation problem with constraints. We study the complexity of the problem, and we characterise the vastness of the search space. Therefore we put forward two solvers: an *exact solver* and a *heuristic solver*. Specifically, on the one hand, we show how to solve the problem with integer linear programming (ILP). On the other hand, we extend the work of [Andrejczuk, 2018] to propose Edu2Com, an anytime heuristic solver that iteratively improves a solution by swapping agents between pairs of teams using different strategies. Furthermore, we conduct a systematic empirical evaluation of the proposed solvers. First, we pitched Edu2Com against the state-of-the-art solver CPLEX using

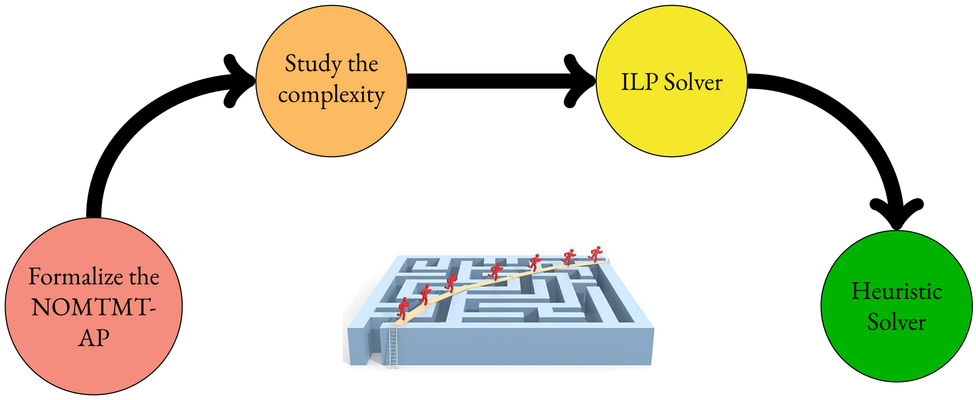


Figure 1.3: Two solvers for the non-overlapping many teams to many tasks allocation problem.

synthetic data to explore *(i)* the limits of CPLEX and *(ii)* analyse the solutions quality of our proposed algorithm (compared to the optimal solutions computed by CPLEX). Then, we tasked Edu2Com to solve large real-world problem instances (using real-world data). As we show, the optimal solver cannot cope with such large problem instances since the search space grows rapidly. Therefore, for dealing with large real-world instances like the scenarios we present in this thesis, our heuristic is the algorithm of choice. In a nutshell, we make the contributions below with respect to forming many teams to many tasks:

5. We formalise the non-overlapping many teams to many tasks allocation problem.
6. We study the complexity of the NOMTMT allocation problem and characterise the search space.
7. We propose two solvers for the NOMTMT allocation problem:
   a) an exact solver, and
   b) an anytime heuristic solver.

In the third part of this thesis, we address the research questions Q.C1 to Q.C5. First, we identify a collection of queries that are intuitive and meaningful and cover, in our opinion, the main points of interest regarding team formation scenarios. As such, we provide a collection of query templates that allow us to challenge the decisions of a team formation algorithm. Then, we introduce a novel, general algorithm for building contrastive explanations in the context of team formation (see Figure 1.4). Importantly, our general methodology wraps the team formation algorithm at hand and uses it as a service that provdes team formation solutions. We detail how our explanatory algorithm processes the query posed by the questioner, and how the general explanatory algorithm wraps the team formation algorithm at hand to build contrastive explanations. Moreover, we analyse and determine the cost of explanations depending on the various queries. Notably, to built explanations we consider different *tailoring* techniques. In particular, each tailoring technique highlights a different point of view of the explanation. Additionally, we introduce a novel evaluation metric that allows us to empirically evaluate the *quality of explanations*. The results show that all the explanations generated by our algorithm are easy to understand, requiring only the reading level of a high-school student.

Finally, we address the challenge of preserving privacy by providing explanation within team formation scenarios. Specifically, we propose a privacy breach detector capable of finding whether a given explanation is bound to lead to privacy breaches. As we detail later, we model the reasoning triggered by explanations in the explainee using a theory of mind [Frith and Frith, 2005], which allows our detector to capture explanations bound to cause breaches. In addition, we propose a general framework that describes how our privacy breach detector interacts with a team formation algorithm (AI system) and an explanatory algorithm (XAI system) to approve or disapprove explanations. That is, we argue that an explanatory algorithm shall only provide explanations that are guaranteed to disclose no private information. As such, our privacy breach detector assesses an explanation and notifies the explanatory algorithm as to whether the explanation is safe or bound to cause a privacy breach.

In an nutshell, we make the following contributions towards explainable team formation:

8. We identify a collection of query templates that allow to challenge the decisions of a team formation algorithm.

9. We introduce a novel, general algorithm for building contrastive explanations in the context of team formation; without modifying the team formation algorithm at hand.

10. We put forward a privacy breach detector to accompany an explanatory algorithm and assess whether an explanation is bound to disclose private information.

11. We propose a general framework that shows the interactions among a team formation algorithm, an explanatory algorithm, and a privacy breach detector.

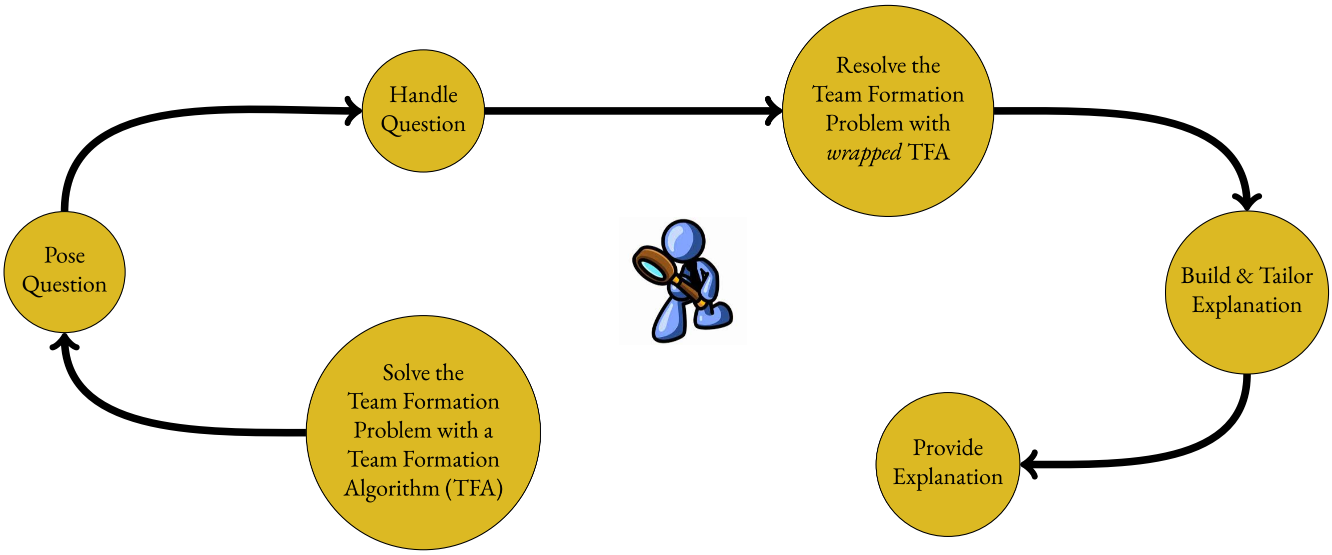


Figure 1.4: A general algorithm for building contrastive and comprehensive explanations in team formation scenarios.

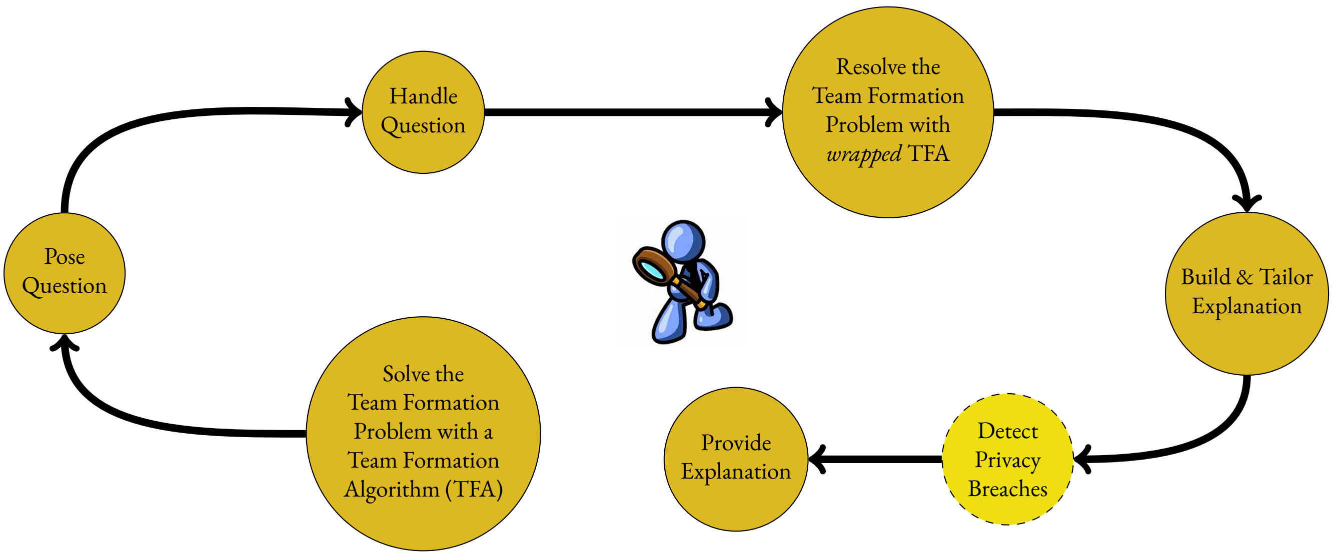


Figure 1.5: Privacy-aware explanatory algorithm for team formation scenarios.

1.4 Road Map

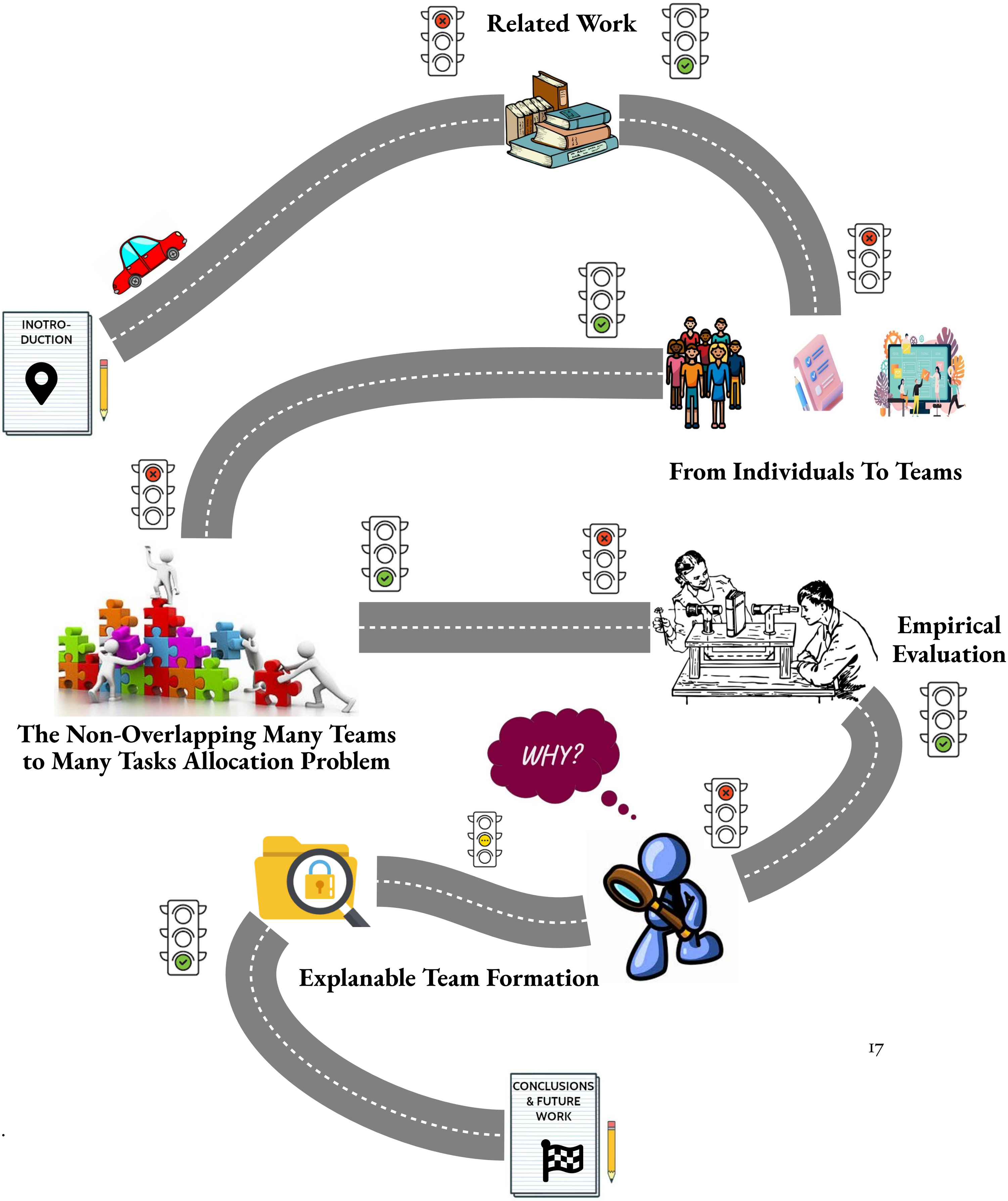
The rest of the thesis is structured as follows. Chapter 2 discusses the relevant literature to this dissertation. We investigate the concepts of team and task allocation and the problem of team formation as it is studied from the point of view of *computer science*, *psychology* and *social sciences*. In Chapter 3, we, specifically, discuss the concepts of agents and tasks, the different elements that comprise each, and how to model them. Additionally, in this chapter, we put forward the notion of a team and introduce several metrics for evaluating a team. In some detail, our metrics consider the several agents' characteristics that affect teamwork (competencies, personality, and preferences) and aggregate them from the individual level to the team level.

Next, in Chapter 4, we present the *Non-Overlapping Many Teams to Many Tasks Allocation Problem* (NOMTMT AP). Here, we formalize the problem and cast it as an optimisation problem with constraints. We study its complexity and characterise the vastness of its search space. Thereafter, we propose two solvers for the NOMTMT AP. The first is an *optimal solver*, using Integer Linear Programming, while the second is an *anytime heuristic solver*, which iteratively improves the current solution via agents' swaps. Chapter 5 contains the empirical evaluation of our proposed algorithms. Specifically, we conducted a series of empirical evaluations involving synthetic and real-world data in order to study *(i)* the behaviour of our heuristic algorithm when solving the NOMTMT AP compared to the optimal solvers, *(ii)* the scalability of our heuristic approach and the limitations of the optimal solver, and *(iii)* the performance of the teams formed with our proposed algorithm.

Chapter 6 delves into the world of *explainable artificial intelligence (XAI)* and tackles the challenge of explaining the decisions of a team formation algorithm. In particular, we put forward a novel general explanatory algorithm that can wrap existing team formation algorithms without modifying them. We identify a collection of intuitive and meaningful questions that cover, in our opinion, the main points of interest regarding team formation scenarios and show that our generated explanations are comprehensive and easy to read. One step further, we address the challenge of preserving privacy upon providing explanations, and we propose a general framework describing the interactions among the team formation algorithm, the explanatory algorithm and a privacy breach detector along with the detector's functionality.

Finally, Chapter 7 concludes the work of this thesis, highlights the lessons learned and discusses future work.

Road Map


Related Work
INOTRO-
DUCTION
From Individuals To Teams
The Non-Overlapping Many Teams
to Many Tasks Allocation Problem
Empirical
Evaluation
WHY?
Explanable Team Formation
CONCLUSIONS
& FUTURE
WORK

1.5 Thesis Software Tools & Publications

This thesis resulted in several software tools, and conference and journal publications.

1. Our team formation algorithm is available in the AI4EU marketplace:
   `https://www.ai4europe.eu/research/ai-catalog/edu2comapi`

2. The team formation algorithm and the explanatory technique developed in the thesis are intergraded into the eduteams platform hosted by IIIA.
   `https://eduteams.testing.iiia.csic.es/`

Our work on the team formation problem we tackle in this thesis and the algorithms to solve it is presented in the following:

Conference Papers

1. Athina Georgara, Juan A. Rodríguez-Aguilar, and Carles Sierra. 2021. Towards a Competence Based Approach to Allocate Teams to Tasks. In Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS '21). International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 1504–1506.

2. Athina Georgara, Juan A. Rodríguez-Aguilar, Carles Sierra, Ornella Mich, Raman Kazhamiakin, Alessio Palmero Aprosio, and Jean-Christophe Pazzaglia. 2022. An Anytime Heuristic Algorithm for Allocating Many Teams to Many Tasks. In Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS '22). International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 1598–1600.

3. Athina Georgara, Juan A. Rodríguez-Aguilar, abd Carles Sierra. 2022. Allocating Teams to Tasks: An Anytime Heuristic Competence-Based Approach. In: Baumeister, D., Rothe, J. (eds) Multi-Agent Systems. EUMAS 2022. Lecture Notes in Computer Science(), vol 13442. Springer, Cham.

Journals

1. Athina Georgara, Raman Kazhamiakin, Ornella Mich, Alessio Palmero Aprosio, Jean-Christophe Pazzaglia, Juan A. Rodríguez-Aguilar, and Carles Sierra. The AI4Citizen pilot: Pipelining AI-based technologies to support school-work alternation programmes. Appl Intell (2023). https://doi.org/10.1007/s10489-023-04758-3

Our work regarding explaining team formation scenarios, and preserving privacy upon providing explanations is presented in:

Conference Papers

1. Athina Georgara, Juan A. Rodríguez Aguilar, and Carles Sierra. 2022. Building Contrastive Explanations for Multi-agent Team Formation. In Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS '22). International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 516–524.

2. Athina Georgara, Juan A. Rodríguez-Aguilar, and Carles Sierra. 2022. Privacy-Aware Explanations for Team Formation. In PRIMA 2022: Principles and Practice of Multi-Agent Systems: 24th International Conference, Valencia, Spain, November 16–18, 2022, Proceedings. Springer-Verlag, Berlin, Heidelberg, 543–552.

Finally, we also presented our work in the following workshops that do not have formal procceedings:

WorkShop Papers

1. Athina Georgara, Carles Sierra, and Juan A. Rodríguez-Aguilar. 2020. TAIP: an anytime algorithm for allocating student teams to internship programs. In 11th International Workshop on Optimization and Learning in Multiagent Systems (OptMas'20), May 2020, Auckland, New Zealand.

2. Athina Georgara, Carles Sierra, and Juan A. Rodríguez-Aguilar. 2020. Edu2Com: an anytime algorithm to form student teams in companies. In Harvard CRCS Workshop on AI for Social Good (AI4SG), July 2020.

Chapter 2

Related Work

In this chapter, we discuss the related work for this dissertation. On the one hand, we review how the concepts of teams, team formation, and task allocation have been addressed within the computer science, psychology and social sciences literature. On the other hand, we delve into explainable artificial intelligence and go through the relevant literature focusing on research regarding explaning multiagent environments.

2.1 Teams and Team Formation in Computer Science

Team formation and task allocation for teams have received much attention within the Computer Science community, particularly in the light of Multiagent Systems. Multiagent systems model settings involving many agents (for example, software entities, robots, humans, or a combination) who need to interact and cooperate. Organising and coordinating the different agents in teams that combine their expertise and jointly act towards a common goal is a crucial problem for multiagent systems, which is increasingly complex as the multiagent environment grows. As such, within the computer science domain, the relevant team formation literature focuses on identifying and modelling the problem and its several parties (e.g., agents, tasks, skills, etc.), studying the hardness of forming teams, and proposing algorithms for solving the team formation problem.

We can identify three main categories in the computer science literature regarding team formation. First, we discern research on how to form a single team that shall tackle a single task. This category holds the majority of research as it describes the simplest version of the team formation problem. Considering more complex versions of the problem, we find research regarding many teams and many tasks (instead of a single one). Literature includes research on how to form a single team to work on multiple tasks, multiple teams to work on the same single task or multiple teams who shall resolve multiple tasks. In this line of research, we can identify two more categories regarding *overlaps*, i.e., whether the problem studied allows agents to participate in multiple teams (agent-teams overlaps), teams to tackle multiple tasks (team-tasks overlaps), or tasks being solved by multiple teams (task-teams overlaps). In what follows, we discuss the related work produced by the computer science literature.

Begining with the simplest version of the team formation problem, [Lappas et al., 2009] tackles the problem of finding a single team of experts to work on a single task. To do so, the authors consider a social network over the experts (agents) and form teams to minimize the communication cost among team members. Particularly, Lappas et al. explore two ways of computing a team's *communication cost* based on how the agents are connected in a given social network. After that, they propose several algorithms for solving the problem depending on the cost function at hand.

[Anagnostopoulos et al., 2010] studies the problem of team formation in large-scale communities. Specifically, the authors study the problem of forming teams to undertake tasks considering the task's requirements in terms of skills, agents' expertise, and agents' maximum workload (i.e., the maximum amount of duties an agent has). In their work, Anagnostopoulos et al. focus on the *online* version of the problem, where tasks arrive sequentially, and a single team must be formed each time. The authors propose several heuristics concerning the cost function they try to minimise —team size, the maximum workload within the team, the overall workload of the team members, and the overloaded members in the community—and analyse the competitive ratio (with respect to the optimal cost) achieved by these heuristics. Notably, although the problem here involves many teams and many tasks, the fact that the authors consider the online version of the problem and propose algorithms that form a single team at a time to work on the task at hand allows us to consider this work as research for a single team for a single task.

In the same direction, the authors in [Anagnostopoulos et al., 2012] put forward the *online* team formation problem—i.e., forming a team to work on each task as a stream of tasks sequentially arrives—considering the task's requirements in terms of skills along with a social network over the agents. The social network allows the authors to compute a team's coordination/communication cost depending on how the team members are connected within the network and, therefore, form teams that meet a maximum coordination threshold. Specifically, the authors propose a *general online* team formation algorithm that can handle different coordination cost functions and provide approximation algorithms with provable theoretical guarantees for the different coordination cost functions. Notably, in [Anagnostopoulos et al., 2012], agents can be 'reused' in several teams depending on their workload.

[Kurtan et al., 2020] addresses the problem of building teams for query answering in multi-agent systems. The authors suggest that the team's performance differs from the individual performance of the team members, especially as the different *subtasks* are interdependent. As such, the authors study the dependencies between subtasks of a given task and propose algorithms for building a single team for a single task (consisting of interdependent subtasks) considering some desired qualities, such as preserving privacy.

Moving to more complex versions of the team formation problems, [Crawford et al., 2016] studies the *robustness* of a team. The term robustness refers to the capability of a team to complete a given task, even if some members of the team are unavailable —i.e., a team is k-robust if the team can successfully carry out their task when k members are incapacitated. In their work, the authors present a suite of approximation algorithms for finding a single k-robust team with minimum cost for tackling a number of different tasks. Note that since a single team shall tackle many different tasks, the team formation problem in this work permits team-tasks overlaps.

[Capezzuto et al., 2020] consider a number different tasks, and aims to form *coalitions* to tackle all these tasks on time. Specifically, the problem of interest in this work is to allocate teams of agents to tackle tasks considering temporal and spatial constraints. At each timestamp, agents and tasks are positioned in space, with the agents needing to move towards the allocated task's location to work on, while tasks must be successfully completed before their due time. The authors propose two algorithms for solving the problem to maximise the number of completed tasks. Note that the proposed algorithms allow agent-teams and task-teams overlaps, i.e., agents can participate in multiple teams, and multiple teams may work on the same task as time progresses.

[Andrejczuk et al., 2019] studies the problem of partitioning a set of agents into equally-sized teams, i.e., forming a team of similar size where each agent can participate in exactly one team. Each formed team works separately on the same task; that is, there is a common task with specific requirements (in terms of competencies), and each team needs to tackle this common task separately. Note that since many teams tackle the same problem, the team formation problem in [Andrejczuk et al., 2019] allows task-teams overlaps. The authors introduce two algorithms for solving the problem: an exact solver and a heuristic one. The proposed algorithms aim in *fair partitions*[1], where each team is *(i)* proficient in terms of competencies to tackle the task, and *(ii)* diverse in terms of team members' personalities.

Here we would like to highlight that, despite [Andrejczuk et al., 2019] works towards forming multiple teams with no overlaps (since they consider partitioning a set of agents), considering that all teams shall be of a similar size and work on the very same task is a limitation. As such, in this dissertation, we extend their work to consider *(i)* multiple tasks, *(ii)* multiple teams of several (independent) sizes, and *(iii)* permitting no overlaps.

Regarding the problem of forming many teams for many tasks with no overlaps, we find the work of [Präntare and Heintz, 2018]. In their work, Prentater and Heitz study the problem of coalition structure generation and assigning an independent task to each coalition. The authors propose an anytime search algorithm that uses branch-and-bound techniques to determine the optimal team size for each task and apply brute force to find the optimal coalition structure and task assignment. A search space representation based on multiset permutations of integer

[1] The authors consider the Bernoulli-Nash product to promote fair partitions, i.e., partitions containing teams of roughly equal value.

partitions allows the proposed algorithm to always reach optimal solutions and outperform the state-of-the-art optimal solver CPLEX. Nevertheless, applying brute force search limits the size of the problem instances that the algorithm can solve.

In the same direction, [Czatnecki and Dutta, 2019] tackles the problem of coalition formation for task allocation. In more detail, the authors consider heterogeneous robots equipped with different sensors and actuators, while the tasks require different sets of sensors. [Czatnecki and Dutta, 2019] proposes a hedonic coalition formation framework and an algorithm that forms Nash stable partitions where each coalition works in a different task. Notably, in the proposed framework, a coalition consists of robots and a single task. The authors put forward a utility function that encodes how well an agent fits in their coalition and, more precisely, how well an agent matches the coalition's task, considering the task's required skills and those offered by each agent. This utility function indicates agents' willingness to remain or abandon their current coalition, forming the agents' hedonic preferences. Strikingly, the way of forming the agents' hedonic preferences, which in fact correspond to agents' preferences over the different tasks, allows the existence of Nash stable partitions. However, in the general case where agents have preferences over the coalitions based on the members of the coalitions, including other agents (not just the tasks), Nash stable partitions may not exist. In their later work [Czarnecki and Dutta, 2021], the authors showed that their proposed approach scales up concerning the number of agents (up to 2000 robots) and tasks (up to 400).

Team formation and task allocation are thoroughly studied in computer science literature. Given the discussion above, we can see that research explores several variations of the team formation problem (considering forming a single vs. many teams and/or allocating a single vs. many tasks), proposes different objectives to target while forming teams (e.g., minimise communication cost, maximise robustness, meet workload threshold, etc.), and devises a suit of algorithms for forming teams. Notably, in all the works discussed above, regardless of the objective considered during team formation, we see that fulfilling skill or competence requirements is a prerequisite (either as part of the objective or as a constraint). However, despite the extensive research on the topic, we still discern open challenges regarding the formation of many teams for many tasks with no overlaps. To be more precise, the existing work for tackling the team formation problem considering multiple teams and multiple tasks with no overlaps exhibits limitations regarding *(i)* scalability (proposed algorithms can handle a small number of agents and tasks) and *(ii)* employability to general instances of the problem.

2.2 Teams and Team Formation in Psychology and Social Sciences

The concepts of team and teamwork are well-studied topics in Psychology and Social Sciences, Research in these scientific fields explores how individuals come together to collaborate, communicate, and achieve common goals. The primary focus is to understand the different factors that drive human behaviour when they participate in a team, elicit the relation between human behaviour and work performance (either at an individual or a team level), and propose guidelines on what are the "ingredients" of an efficient team. Below, we distinguish three main scientific areas that study the relations between human team composition and team efficiency, namely the *Organisational Psychology*, the *Motivational Psychology* and the *Social Sciences*.

2.2.1 Organisational Psychology

Organisational psychology studies human behaviour within organizations and workplaces [APA, 2023]. This scientific field aims to identify and formalise individual and group behaviour principles and use this knowledge to improve the working environment and performance. Going through the relevant literature in organisational psychology, we find that the main focus lies in exploring the relations between *personality* and *team performance*. In particular, most of the research concerns user studies investigating how team members' personality affects team performance. The key ingredient in organisational psychology is how to describe people's personalities, where we can discern different models such as the Big Five-Factor Model, Belbin's Team Roles Model and the Myers-Briggs Type Indicator Model. The different personality models share similarities and exhibit differences that have been studied in psychology [Furnham, 1996, Higgs, 1996]. In what follows, we discuss the observations and findings of the literature with respect to the relations between personality and performance.

[Mount et al., 1998] conducts a study to investigate the relations of personality—and specifically the Big Five-Factor Model [Norman, 1963]—with job performance. In some detail, the analysis focuses on jobs involving interpersonal interactions, either between employee and client or among co-workers, and studies how personality affects employees' performance in such jobs. The results showed that *emotional stability* and *agreeableness* are highly correlated with performance in jobs involving teamwork. The factors of *extraversion* and *openness* exhibited a lower influence on job performance. Nonetheless, all of the five factors have non-zero relations with performance.

[Barrick et al., 2001] examines and summarises the results of prior studies regarding the personality-performance relation. The authors considered 15 studies investigating whether the five-factor personality model influences job performance. The meta-analysis of the prior studies showed that the results of these studies are relatively consistent. In particular their analysis showed that the factors of *emotional stability* and *aggreableness* are highly influential for teamwork (the influence to teamwork is more significant than to other criteria);[2] the factor of *conscientiousness* is observed to be influential to all criteria including teamwork; while the factors of *extraversion* and *openess* exhibit lower influence to teamwork.

[Driskell et al., 2006] studies the effects of the team members' personalities on teamwork and team effectiveness. First, the authors provide a fine-grained hierarchical model of personality traits (based on the Big Five-Factor Model) that defines specific facets of personality related to team or performance. Then, the authors present a classification scheme over the teamwork dimensions relevant to team effectiveness. Finally, the authors provide linkages between the personality and team effectiveness dimensions and present which facets can predict each dimension based on existing literature.

[Higgs et al., 2005] investigates the relation between team composition and team performance considering the task's complexity that the team needs to work on. The authors considered diverse teams in terms of personality roles, according to the Belbin Team Role model [Belbin, 1993], and tasks of different complexity[3]. The findings showed a clear relation among team composition, task complexity, and performance. Specifically, the results showed that team diversity positively affects team performance when a high-complexity task is in place. In contrast, team diversity negatively affects performance when a low-complexity task is in place.

Given the evidence that team members' personality influences team performance, we discern a handful of research that exploits this evidence to devise general rules and methodologies on composing efficient teams. Towards this direction, [Wilde, 2011] focus on Jung's qualitative personality theory and investigates the relations with the Myers-Briggs Type Indicator (MBTI) [Myers et al., 1998] personality model. The author proposes converting the qualitative concepts described in Jung's theory into a quantitative one. More precisely, the proposed theory represents Jung's personality types with numerical data and manages to link these personality types with the MBTI types. Later, in [Wilde, 2013], the author puts forward the *Post-Jungian Personality Theory* and advances a methodology for composing efficient teams combining people with different personality traits. The methodology described in [Wilde, 2013] is supported by evidence showing that the approach increased the fraction of student teams in Stanford who received national prizes by the Lincoln Foundation [Wilde, 2009].

Following the work of Dougles J. Wilde, [Andrejczuk, 2018] utilises the findings from Dougles' prior work and puts forward artificial intelligence tools for computing team formation. First, in [Andrejczuk et al., 2018], the authors provide a thorough review of team formation literature in the fields of organisational psychology and computer science identifying the similarities and differences. Then, in [Andrejczuk et al., 2019], the authors, based on the findings of [Wilde, 2009], introduce a novel way to measure a team's *congeniality* (i.e., the diversity

---

[2] This result supports the findings of the prior work of [Mount et al., 1998], which is among the studies reviewed in [Barrick et al., 2001].

[3] The tasks were assessed by a panel of experts (individually) to determine their level of complexity.

of team members in terms of personality), and propose two algorithms for partitioning a set of agents in similar size teams.

Considering the findings and observations discussed above, we can see that it is generally accepted that team composition in terms of personality affects team performance. Knowledge regarding team members' personalities can be used as an indicator of team performance and, therefore, can aid in forming highly performing teams.

2.2.2 Motivational Psychology

Motivational psychology aims to understand and study the factors that drive human behaviour, focusing on the motives, incentives, desires, needs, and goals underlying people's actions. It seeks to explore why individuals behave in specific ways, make particular choices, and persist in pursuing specific targets. In task allocation, individuals' motivation to work on some task (e.g., doing a particular job) may significantly impact the individuals' performance and, therefore, the quality of the desired outcomes. Research in motivational psychology seeks to identify those motivational factors related to people's performance. Towards this path, [Deci and Ryan, 1985] put forward the *Self-Determination Theory* (STD). The self-determination theory focuses on the different motives that drive people's behaviour and investigates to what extent the behaviour is self-motivated or self-determined. According to the STD, there are three motivation types (namely, the *amotivation*, the *extrinsic motivation* and the *intrinsic motivation*). Each type of motivation affects job performance to a different degree, with intrinsic motivation being related to greater performance [Deci et al., 2017]. That is, self-determined people tend to perform better in their jobs.

Notably, the relevant literature in motivation psychology consists solely of case studies across different areas (including sports, the medical sector, the educational domain and private companies) investigating how motivation affects performance. Below, we present the related studies and their findings regarding the relationship between motivation and team performance.

[Mallett, 2005] investigates the effects of motivation on sports performance. The author proposes an autonomy-supportive motivational climate for coaching elite athletes following the principles of the self-determination theory. In some detail, the proposed coaching program provided the athletes with the perception of choice (for instance, allowing them to choose among the training tasks), promoting, in this way, the athletes' self-determination. The proposed approach was employed during the two-year preparation period of the Australian relay teams participating in the Olympic Games of 2004 in Athens. By analysing the performance of the teams, the author observed that the teams *(i)* outperformed themselves in terms of time compared to previous years, *(ii)* achieved their seasonal best performance during the Olympic Games—it is very rare to do so during the Olympics—and *(iii)* improved their team ranking by eight places.

[Suliman and Al-Sabri, 2009] conducted a survey investigating relations between motivation, job satisfaction and employee performance in public hospitals in the United Arabic Emirates. The study involved 450 public hospital employees, including doctors, administrative staff, and nurses. The authors observed a strong relationship between motivation and job satisfaction (i.e., individuals being satisfied with their job) with job performance. Therefore, motivation and job satisfaction can be good performance predictors.

[McLean and Mallett, 2012] investigates the motivation of athletes' coaches. Specifically, the authors surveyed 13 elite coaches in Australia, with all coaches awarded accreditations. The study aims to identify the types of motivation that affect coaches' job performance (depicted in their athletes' performance). The results showed that intrinsic and extrinsic motivation are crucial in the participants' coaching.

[Abdulsalam and Mawoli, 2012] studies the relationship between motivation and job performance in the educational domain. Specifically, the authors considered the performance of academics in the tasks of *(i)* teaching and *(ii)* research. Abdulsalam and Mawoli surveyed 219 academic staff members across 15 different departments at Ibrahim Badamasi Babangida University. The results showed that motivation significantly influences teaching performance, observing that 23% of the variance in teaching performance was caused by motivation. On the other hand, no significant relation between motivation performance in research was observed.

[Tabassi et al., 2012] studies the effects of motivation during employees training in *(i)* teamwork and *(ii)* task performance. The authors conducted a survey involving 107 large construction companies in Iran, and they investigated how motivating parameters such as training assignment, perceived importance of training, hygiene factors, and motivating environments affect the training practices and, therefore, teamwork and task performance. The findings showed that companies that applied motivators such as training assignment, perceived importance of training, and motivating environments during their training programs achieved improvements in their employees' teamwork activities. Regarding task efficiency, the authors observed that companies that considered motivators in their training achieved more significant performance improvements than companies that did not.

Based on the above observations, there is much evidence across different domains of application that motivated teams exhibit better performance than non-motivated ones. Therefore, using motivation as a predictor for team performance can help us compose efficient teams. Interestingly, to the best of our knowledge, no team formation algorithm exploits observations regarding motivation and performance.

2.2.3 Social Sciences

Social sciences focus on *society* (or community), i.e., a structure of a group of people who coexist and interact within the same spatial or social territory. Social sciences study the relationships and social interactions among the individuals within a society. Teams can be considered small societies (consisting of a few individuals). Hence, the relations among team members affect a team's prosperity and, therefore, the team's performance. In what follows, we go through the relevant literature in social sciences. Notably, this line of research is limited and follows two paths. The first path regards works that conduct empirical studies in an attempt to determine the relation between *social cohesion* (i.e., a measure indicating whether a team is likely to stick together [Moreno and Jennings, 1938]) and team performance. The second path regards works exploring ways to measure social interactions in order to use this information while forming effective teams.

Regarding the first research path, [Lucius and Kuhnert, 1997] studies the relationship between team members' social interactions and team performance. The authors surveyed 29 military squads at the college level to study how the social relations among team members affect the squad's performance on typical military procedures such as field exercises, presenting arms marching, flag duty, etc. During the survey, the participants were administered a *sociometric questionnaire*[4] to quantify the social relations among the participants and investigate whether teams with strong social relations outperform. The authors observed that *socially coherent* teams, i.e., teams with members that mutually prefer working together, exhibit better performance, suggesting that social cohesion can be used as a predictor for team performance.

[Carron et al., 2002] studied the impact of social cohesion within sports teams. In this work, the authors conducted a meta-analysis of previously published surveys on the relationship between social cohesion and performance. The analysis considered 46 prior studies involving 9988 athletes and 1044 sports teams. The results of the analysis showed social cohesion and performance are highly correlated. According to the findings, teams engaged in coactive sports (e.g., bowling or golf) exhibit a relationship between social cohesion and performance that is slightly stronger than teams in interactive sports. Moreover, female teams' performance is influenced more by the team's social cohesion compared to male teams. Similarly, the less the experience level of the team (e.g., intercollegiate vs professional clubs), the stronger the relation between social cohesion and performance. However, in all experience levels, the relation is significant. Notably, the authors question whether social cohesion improves performance or vice versa.

The observations above are rather valuable since they support the claim that teams who share strong social bonds tend to exhibit high team performance. As such, when forming human teams, it can be very beneficial to take into consideration people's social interactions. Now, regarding the second research path, we find studies that acknowledge the influence of social

[4] Sociometry is a quantitative method for measuring relations within a social group. The method was developed by J.L. Moreno and H.H. Jennings.

interactions on teamwork and team performance. Therefore, in this path, researchers work towards measuring people's social interactions by proposing methods to do so and thereafter exploit the measured interactions to build efficient teams.

In this light, [Ballesteros-Perez et al., 2012] investigates the influence of social interaction within working groups. The authors propose using sociometry to capture social relations among co-workers and exploit the measured social interactions to form effective teams. In some details, the authors propose a method for computing the *efficiency* of a group in terms of social interactions and form teams that maximise efficiency and respect the project's requirements.[5] Using a detailed example, the authors illustrated that even considering a small number of individuals, efficiency may exhibit significant fluctuations. The proposed method was also applied to a construction company that focuses on designing, constructing, exploiting and maintaining big Waste Water Treatment Plants to handle the company's human resources for 12 such projects. The findings indicated that using sociometry to form socially coherent teams can be used as a predictive tool that allows project managers to put together working teams that achieve greater performance.

[da Silva and Krohling, 2018] builds on [Ballesteros-Perez et al., 2012] and puts forward a fuzzy socimetric technique. Specifically, the authors introduce a model to capture social interactions between individuals using fuzzy logic, proposing the *fuzzy social cohesion* of a group. Thereafter, [da Silva and Krohling, 2018] proposes an algorithm to form teams to work on several projects so that the fuzzy social cohesion of all teams is maximised.

Research in social sciences indicates that relations between people have an impact on teamwork and team performance. Evidence supports that socially coherent teams, i.e., teams whose members have positive social interactions, outperform non-coherent teams. As such, social relations among team members can be a useful predictor of team performance and, therefore, a valuable asset when forming teams. Notably, capturing and measuring social interactions is a non-trivial problem that has received some attention.

2.3 Explanable Artificial Intelligence (XAI)

Over the past decade, research in artificial intelligence has witnessed a resurgence of *explainable AI (XAI)* as a weapon to earn trust and alleviate users' concerns regarding 'black-box' AI systems. In 2018, Tim Miller published his work on explainable AI, where the author studies how explanations are and should be given within AI systems. [Miller, 2018] reviews research from social sciences (e.g., philosophy, psychology and cognitive science) regarding explanations and explores the relation between explanations in social sciences and explainable artificial intelli-

[5] In the case study presented in [Ballesteros-Perez et al., 2012], project requirements referred to the number of people belonging to a particular department, while all individuals within the same department were considered equally competent.

gence. The paper aims to shed light on how artificial intelligent systems should use social science observations to provide explanations that target non-expert people (e.g., users and the general public) instead of scientists, researchers and AI professionals. The author focuses on the *everyday explanations*, that is, on explaining why particular events happened. Among the paper's major findings is that humans tend to give and seek *contrastive explanations*, i.e., explanations that answer why a particular event occurred instead of another event.

Given the great attention in XAI, particularly on machine learning models and recommender systems, [Kraus et al., 2020] discusses the importance of providing explanations within multiagent environments (xMASE)—an area that has received little attention so far. In multiagent environments, decisions concerning multiple agents must be made, and therefore, these decisions are made based on all agents' goals and preferences. Kraus et al. argue that in such environments, explanations shall serve as a tool to increase user satisfaction. The authors claim that providing explanations within multiagent systems is rather challenging, especially as the explanations should consider the AI system's decisions and functionality along with the users' and other agents' preferences. The authors identify the key challenges towards explaining multiagent environments, review the state-of-the-art on xMASE, and discuss the open questions and research directions towards xMASE.

In the context of providing explanations within multiagent systems, we find a handful of research. To begin with, [Boixel and Endriss, 2020] works towards justifying collective decisions. In particular, the authors focus on multiagent environments where individuals express their preferences and, collectively, they need to make some decision(s) following a voting process. Boixel and Endriss put forward a justification method for explaining under which voting axioms the decision at hand was selected. The generated explanation includes a minimal subset of axioms (out of any finite set of voting axioms). Therefore, by considering this minimal subset of axioms and the agents' preferences, one can justify why the decision at hand was selected as the winner of the voting process.

[Mosca and Such, 2021] addresses the problem of *multiuser privacy conclict*, i.e., the problem of aligning privacy preferences among multiple agents upon sharing co-owned content. Specifically, the paper focuses on finding optimal sharing policies in online social networks that respect all involved parties' privacy preferences. The authors introduce *ELVIRA*, an agent that *(i)* recommends sharing policies and *(ii)* is able to justify its recommendations by generating contrastive explanations. In some details, for the agent to build a policy, it needs to answer three critical questions, and then the agent selects the policy that exhibits only negative answers. To generate an explanation, the agent examines an alternative sharing policy (provided by the user), presents the critical questions to which the alternative policy exhibits positive answers and highlights the differences between the recommended policy and the alternative one.

[Pozanco et al., 2022] studies the problem of explaining *preference-driven schedules*. The authors consider settings where multiple agents must be matched with timeslots and resources while the agents declare their preferences of different types (e.g., preferences over timeslots). The agents' preferences are prioritised concerning their type (some types of preferences are more important to the agent than others). At the same time, a set of constraints shall be met (e.g., each agent must be matched with at least one timeslot). Pozanco et al. propose the *EXPRESS* framework to provide explanations in such settings and specifically to tackle questions of the kind "Why a specific preference was not satisfied?". In some detail, given an optimal solution and an unsatisfied preference, the proposed framework computes the best scheduling to satisfy the preference at hand. After that, *as reasons* identifies preferences no longer satisfied in the alternative solution. The generated explanation contains reasons concerning more important preferences (according to the agents' preference prioritisation) than the preference at hand.

[Nizri et al., 2022] focuses on cooperative game theory and works towards explaining payoff allocations. In cooperative game theoretic environments, agents collaborate with one another and jointly (as a coalition) achieve a particular utility that shall be shared among the agents. In the paper, the authors consider cooperative games where utility is distributed according to the Shapley values; then they put forward *X-SHAP*, an algorithm that decomposes the game in several *easy-to-explain* sub-games and generates a brief verbal explanation for each sub-game. The authors conducted a survey involving 210 people. They report that the explanations generated with X-SHAP significantly outperformed explanations that state the benefits of Shapley values in justifying whether the payoff allocations were fair.

Explaining decisions made by an AI system is a key factor towards understanding and trusting such systems. It is rather important that explanations target to address users and the general public instead of explaining the AI systems to AI experts. As such, explainable AI should embrace existing findings regarding providing and accepting explanations by humans. As highlighted, providing explanations within multiagent environments is a challenging problem that has received little attention so far. To the best of our knowledge, there is no research explaining AI algorithms for the team formation problem.

2.4 Conclusions

In conclusion, we find literature relevant to our work in this thesis in computer science, psychology and social sciences. In more detail, first, we see there are open challenges in solving the team formation problem. That is, despite the extensive interest in the team formation problem, existing research and algorithms for forming many teams with many tasks exhibit major limitations regarding scalability and employability to general instances. As such, in this dissertation, we work towards this direction. In our work, we focus on the problem of forming multiple teams to be allocated to multiple tasks with no overlaps, and we aim to lift the limitations observed in existing related work.

Next, we reviewed how human teams are viewed through psychology and social sciences. We discerned three scientific fields that study teamwork and went through their findings and observations. In the field of organisational psychology, we find that personality is associated with performance and, in fact, that a team's composition in terms of team members' personalities can be a useful predictor of team performance. Observations from the field of motivational psychology support that motivated teams outperform non-motivated ones. As such, team members' motivation can also be used as an indicator of team performance. Finally, in the field of social sciences, we see that social interactions among the members of a team can affect team performance, as well. Specifically, the existing literature supports the claim that teams with strong bonds among their members (i.e., socially coherent teams) exhibit better performance. Against this background, our work embraces the observations from psychology and social sciences in order to form high-performance teams. Thus, in this thesis, we put forward AI algorithms to form teams that are diverse in terms of personality, motivated and socially coherent.

Finally, we delve into explainable artificial intelligence. AI systems are usually seen as 'black boxes' that often cultivate users' distrust. Explainability is a tool towards earning the users' trust. However, as highlighted by Miller, explanations should incorporate findings concerning how humans explain things in order to address non-experts such as the users. Moreover, we see that, despite the great interest in explainable AI, little research exists on explaining decisions in multiagent environments. As Kraus et al. points out, providing explanations in a multiagent system is rather challenging. Notably, to the best of our knowledge, there is no research regarding team formation. Therefore, in this dissertation, we make the headway towards explaining team formation algorithms.

Chapter 3

From Individuals to Human Teams

"United we stand, divided we fall." This phrase symbolises the power that people have when they collaborate and fight together towards the same goal, as there is plenty of evidence that individuals can achieve better results when working together. Nevertheless, what is a team? What characteristics transform a group of individuals into a well-tuned team? What does it mean to be a member of a team? Moreover, how does each team member affect the performance and the well-being of the team and, ultimately, the quality of the final outcome?

Being part of a team implies *communication*, *coordination* and *cooperation* among the team members. Team members shall be able to interact with each other, share /exchange information related to the task they are working on and the progress of their work, and help one another overcome problems and difficulties that arise during teamwork [Kozlowski and Ilgen, 2006]. Being a team member means that one is not just responsible for their own progress and success; instead, they are also (partially) responsible for the progress and success of the whole team. In other words, when an individual becomes a member of a team, then a special connection is created between this individual and the rest of the team: if everyone prospers, then the team prospers, and if the team prospers then everyone prospers —i.e., all for one and one for all. This chapter focuses on three key concepts for this dissertation, namely the agents, teams, and tasks. In more detail, we discuss how to model people as agents, considering the features that characterise humans while we provide the formalisation for both human agents and tasks. Then, we turn to the concept of a *human team*. That is, we detail how to assemble individual agents to build a single team. Finally, we discuss evaluating team adequacy for a task. That is, when a team and a task are in place, we introduce metrics to assess the suitability of the team for working together on the task at hand.

In summary, in this chapter, we have a threefold discussion: *(a)* how to *model* human agents and tasks, *(b)* how to *form* a single human team, and *(c)* how to *evaluate* a single team. In what follows, in Sections 3.1 and 3.2, we model human agents and tasks, respectively. In Section 3.3, we discuss what a human team is and how to form a single team; in Section 3.4, we discuss how to evaluate a human team.

3.1 Modelling Agents

An agent is an individual (human) available to work on some tasks. Let $\mathcal{A} = \{a_1, a_2, \cdots, a_n\}$ denote a set of available (human) agents. We describe each agent, $a \in \mathcal{A}$, through their *profile*, which characterises the agent across some *features* of interest. That is, an agent's profile holds information regarding the agent and comprises the agent's acquired or endogenous characteristics, desires, beliefs, and ethical values that drive their behaviour. Since we aim to form teams to work on some task, an agent's profile should be such that it will allow us to estimate the agent's behaviour during teamwork. In other words, the features of interest that we include in the agents' profiles will allow us to determine how and to what extent an agent will contribute to teamwork and estimate the agent's behaviour throughout the collaboration. In order to do so, we adopt features of interest that influence performance according to findings from psychology and social sciences [Andrejczuk, 2018, Lucius and Kuhnert, 1997, Abdulsalam and Mawoli, 2012]. We have singled out four features that, either at the agent or team level, can positively or negatively influence performance:

1. competencies;
2. personality and gender;
3. preferences over tasks;
4. preferences over potential teammates.

The first feature corresponds to acquired characteristics such as knowledge, skills, and capabilities. Naturally, since the aim is to work on some task (as part of a team), if an agent is equipped with the appropriate skill set to carry out the task, it will significantly influence the agent's and team's performance. Each agent, $a \in \mathcal{A}$, has a *competence profile*. According to the Oxford Learner's Dictionary,[1] competence refers to the ability to perform well, the authority or power in handling a particular situation, or a necessary skill for performing a particular job or task. In this work, by adopting the latter part of the definition and similarly to [Hager and Gonczi, 1996] with the term *competence*, we refer to knowledge, skills, attributes, and related experience which is necessary when performing a job or a task. We assume that there is a pre-defined and finite set of competencies, denoted as $\mathcal{C}$. Thus, we formally define an agent's competence profile as follows:

[1] https://www.oxfordlearnersdictionaries.com/

Definition 1 (Agent's Competence Profile). *Given a set og agents $\mathcal{A}$, a finite set of competencies $\mathcal{C}$, and a set of competence levels Q_{Level}, we define the* competence profile *of agent $a \in \mathcal{A}$ as a pair $\mathcal{P}_a^{\text{c}} = \langle C_a, l_a \rangle$, where $C_a \subseteq \mathcal{C}$, and $l_a : C_a \to Q_{\text{Level}}$.*

With Q_{Level}, we denote a pre-defined domain of level of expertise. This domain can be *quantitative*, e.g., a value in $[0, 1]$, or *qualitative*, e.g., the five levels of expertise in the Dreyfus model [Dreyfus and Dreyfus, 1980], Q_{Level} = {Novice, Advanced Beginner, Competent, Proficient, Expert}.

Apart from competencies, other important features that have an impact on an agent's behaviour are endogenous properties. Specifically, the agent's personality and gender [Mount et al., 1998, West, 2012, Andrejczuk et al., 2018]. According to the *American Psychological Association*,[2] the term *personality* refers to the enduring characteristics and behaviour that determine an individual's unique adjustment to life, including major traits, interests, drives, values, self-concept, abilities, and emotional patterns. Even though there is not a unique, universally accepted definition of what personality is among the scientific community of psychology [Bergner, 2020], psychologists tend to agree that personality is a psychological system composed of a group of parts (personal characteristics, traits, interests, derivatives, and more) that interact, develop, and impact a person's behavioural expression [Mayer, 2007]. The Myers-Briggs Type Indicator (MBTI) [Briggs et al., 1995] is a well-received model of describing personality. This indicator evaluates an individual across four dimensions: Extraversion-Introversion(E-I), Sensing-Intuition (S-N), Thinking-Feeling (T-F) and Judging-Perceiving (J-P). The different combinations of the four dimensions result in sixteen different personality types, as shown in Figure 3.1.

As such, each agent $a \in \mathcal{A}$ has a *personality profile*, which we formally define as:

Definition 2 (Agent's Personality Profile). *The* personality profile *of agent $a \in \mathcal{A}$ is a tuple $\mathcal{P}_a^{\text{p}} = \langle SN_a, TF_a, EI_a, JP_a \rangle$, where $SN_a \in \{S, N\}$, $TF_a \in \{T, F\}$, $EI_a \in \{E, I\}$ and $JP_a \in \{J, P\}$ indicate the evaluation of agent a across the dimensions Sensing-Intuition (S-N), Thinking-Feeling (T-F), Extraversion-Introversion(E-I) and Judging-Perceiving (J-P), respectively.*

[2] https://www.apa.org/topics/personality/

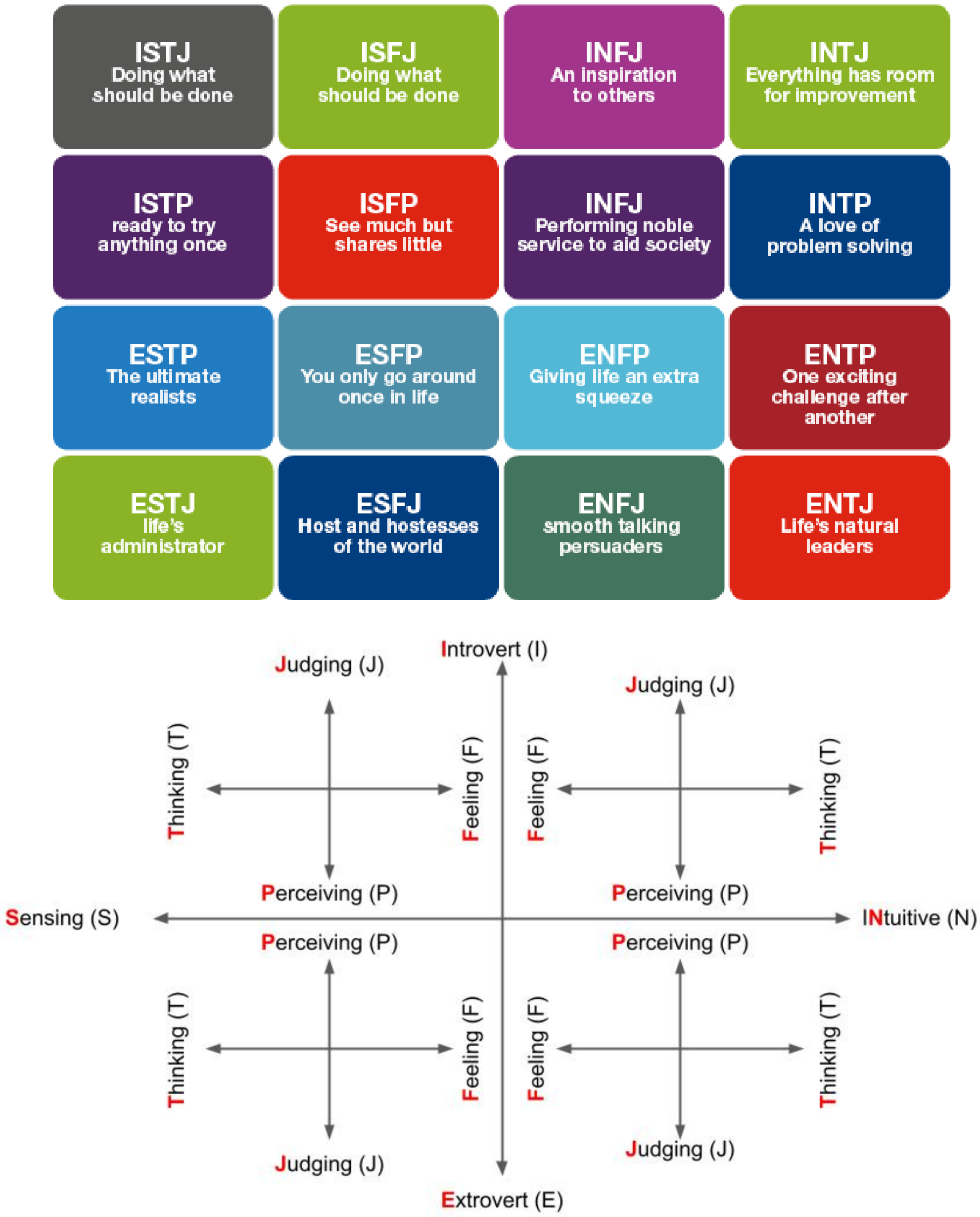


Figure 3.1: Sixteen MBTI personalities. (Schemes come from [Leadership Centre,] and [Wideman, 2023].)

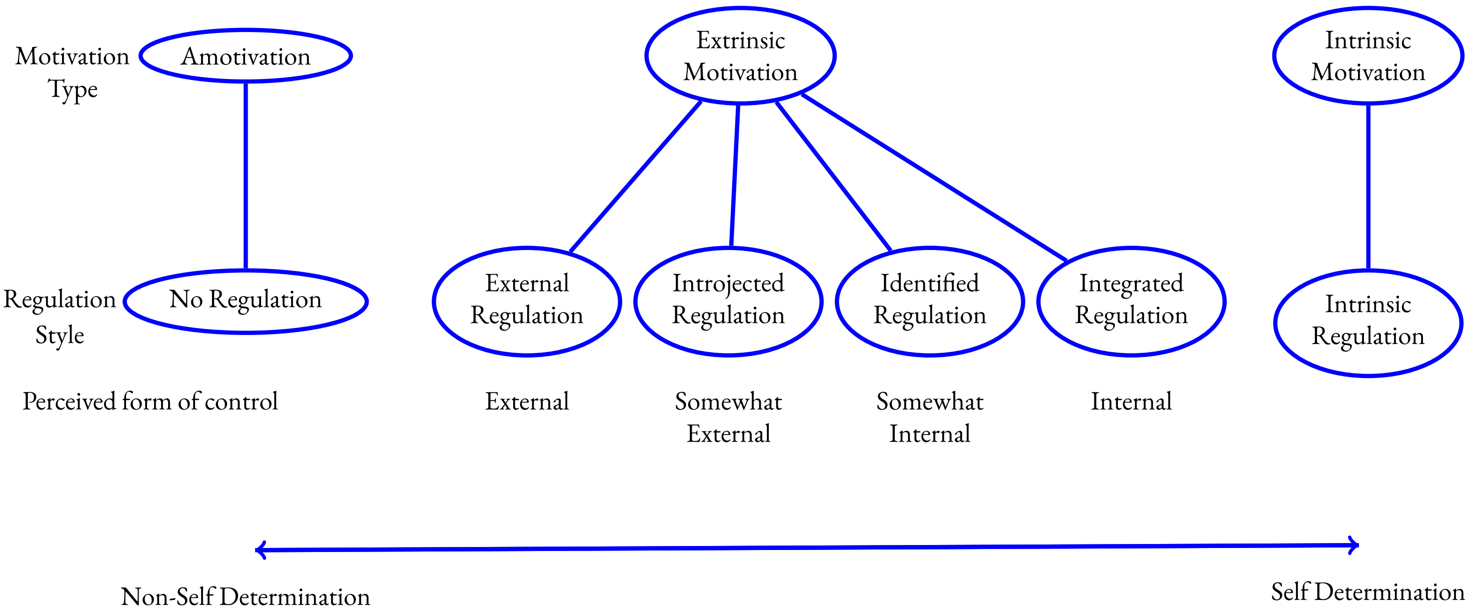


Figure 3.2: Self Determination Continuum: Amotivation, Extrinsic Motivation, Intrinsic Motivation and Regulations (Scheme comes from [Ryan and Deci, 2000]).

The third feature we use to describe an agent is *preferences over tasks*. In the motivational psychology literature, it is observed that people who are happy with their job's conditions (purpose, social status, monetary rewards, etc.) tend to be more productive and reach their job's goals [Deci et al., 2017]. Contrarily, people who feel unconfident with their work environment and need more inspiration towards performing their job tend to produce negative consequences such as poor quality outcomes [Tremblay et al., 2009, Howard et al., 2016]. Notably, individuals lacking motivation are more likely to resign from their jobs to look for a more satisfying workplace [Deci et al., 2017]. According to Self-determination theory [Deci and Ryan, 1985], three main categories of motivation drive people's behaviour, namely *amotivation*, *extrinsic motivation*, and *intrinsic motivation*, that form the self-determination continuum as shown in Figure 3.2. Studies showed that different types of motivation influence people's performance differently, with intrinsic motivation (and motives close to that extreme) enabling high-quality performance [Baard et al., 2004, Manganelli et al., 2018]. When individuals work on exciting or enjoyable tasks, they tend to perform well. With this in mind, we assume that each agent expresses their *preferences* over the different tasks based on their motivation, i.e., more interesting and enjoyable tasks that intrinsically motivate the agent are preferable to less intriguing or boring tasks. Thus, each agent has a *task preference profile*, defined as:

Definition 3 (Agent's Task Preference Profile)**.** *Given a set of agents $\mathcal{A}$ and a set of tasks $\mathcal{T} = \{\tau_1, \cdots, \tau_m\}$, agent $a \in \mathcal{A}$* task preference *profile is represented as $\succsim_a^{\mathcal{T}} \subseteq \mathcal{T} \times \mathcal{T}$, where $\tau \succsim_a^{\mathcal{T}} \tau'$ means that agent a is more or equally motivated to work on task τ than on task τ' for $\tau' \neq \tau$.*

Last but not least, we consider *preferences over potential teammates* to completely characterise an agent. When people work together in teams, they form a social system, and therefore, they naturally develop social relations with one another and establish social norms and agreements. In the social sciences, *social cohesion*, is considered the "glue" that sticks a team together [Friedkin, 2004]. Through the years, researchers have described social cohesion differently. For example, [Moreno and Jennings, 1938, Festinger et al., 1950] relate social cohesion to the temporal duration of a group of individuals being members of the same team. In contrast, [Back, 1951] initiated a shift in the notion of social cohesion to describe the tendency to join or remain a team member due to its members. The latter interpretation has prevailed. In this work, we consider social cohesion as a measure of how much each member is accepted by the team and how much each member accepts the team. We define a teammate preference profile for each agent to capture which teammates the agent prefers. The teammate preference profiles of the other agents will tell who likes this agent. Formally, the teammate preference profile is defined as:

Definition 4 (Agent's Teammates Preference Profile)**.** *Given a set of agents $\mathcal{A}$, and agent $a \in \mathcal{A}$, a's* agent preference *profile is represented as $\succsim_a^{\mathcal{A}} \subseteq \mathcal{A} \times \mathcal{A}$, the expression $a' \succsim_a^{\mathcal{A}} a''$ means that agent a prefers wot work with agent a' at least as much as agent a''. $\succsim_a^{\mathcal{A}}$ must be a preorder.*

There is a plethora of research studying the relations between social cohesion within a team and the team's performance; the observations support that socially coherent teams outperform non-socially coherent teams. [Lucius and Kuhnert, 1997, Hoegl and Gemuenden, 2001, Beal and Cohen, 2003, Høigaard et al., 2006, Mathieu et al., 2015]. Finally, we formally describe a *human* agent as a combination of competence, personality, task preference and teammate preference profile:

Definition 5 (Human Agent)**.** *Given a set of agents $\mathcal{A}$, a set of tasks $\mathcal{T}$, and a finite set of gender symbols $\mathcal{X}$, a human agent $a \in \mathcal{A}$ is a tuple $a = \langle \mathcal{P}_a^{\mathrm{c}}, \mathcal{P}_a^{\mathrm{p}}, \mathcal{X}_a, \succsim_a^{\mathcal{T}}, \succsim_a^{\mathcal{A}} \rangle$, where $\mathcal{P}_a^{\mathrm{c}}$ is a competence profile, $\mathcal{P}_a^{\mathrm{p}}$ is a personality profile, $\mathcal{X}_a \in \mathcal{X}$ is the human self-declared gender, $\succsim_a^{\mathcal{T}}$ is a task preference profile, and $\succsim_a^{\mathcal{A}}$ is a teammate preference profile.*

3.2 Modelling Tasks

A task corresponds to some activity that someone (an agent or a team) must perform. Let $\mathcal{T} = \{\tau_1, \tau_2, \cdots, \tau_m\}$ denote a set of tasks. Each task $\tau \in \mathcal{T}$ is associated with some *goal* that it aims to achieve. A task description includes the procedure that must be followed to achieve the associated goal. For example, suppose a high school student wants to write an essay on the French Revolution for their history course. To do so, they need first to study and gather information about the French Revolution and then write down the information they identify as most relevant regarding events and people of that period. In this example, the student's desired goal is to complete their homework, which is to write an essay about a particular historical event, while the task's goal is to make students do research on that specific historical moment and write down their findings and conclusions.[3]

Also, a task's description includes its knowledge requirements, that is, the necessary competencies that someone (an agent or a team) needs to possess (or acquire) to *successfully* perform the task. For example, for the French Revolution essay, someone would need to have the competence "to process information, ideas and concepts", the competence "to write analytical reports", the competence "searching information on the web", the competence "to use text editors", and the competence "time management".

Even though every required competence is necessary for successfully completing a task, not all competencies are equally determinant, and not all competencies are needed at the same expertise level. In this work, we allow each required competence to be accompanied by a *required expertise level* and a *relevant importance weight*. In our example, for instance, having the competence to write analytical reports is more important than using text editors since the quality of the essay depends more on its content than on its looks. Also, it is sufficient for the student to be an advanced beginner with respect to the competence "searching information on the web". However, it is necessary that the student is at least proficient concerning the "processing information, ideas and concepts" competence. In summary, each task is defined through its required competencies along with the competencies' level of expertise and their importance.

Finally, each task has a requirement in terms of team size. That is, there is an optimal size for the team that will undertake each task. Intuitively, tasks differ in terms of workload or urgency; the size of a team for a task is set so that there are enough human resources to complete the task in time successfully. That is, a team smaller than the optimal team size will not be able to complete the task, while a team larger than the optimal team size will under-use the assigned human resources. Given all the above, we formally define a task as follows:

[3]Note that in this example, the task's goal is to "write the essay". In contrast, "complete the homework" or even "pass the history course" describe more abstract, indirect goals. When we refer to the goal achieved via a task, we refer to the imminent goal.

Definition 6 (Task). *A task τ is a tuple $\tau = \langle C_\tau, l_\tau, w_\tau, s_\tau \rangle$, where $C_\tau \subseteq C$ is the set of required competencies; $l_\tau : C_\tau \to Q_{\text{Level}}$ maps each competence $c \in C_\tau$ to the minimum expertise level; $w_\tau : C_\tau \to [0, 1]$ maps each competence $c \in C_\tau$ with a real number in range $[0, 1]$ representing the importance of the competence; and $s_\tau \in \mathbb{N}_+$ is a positive integer indicating the team size. We denote by $\mathcal{T}$ the set of all possible tasks.*

Note that for a competence $c \in C_\tau$, an importance weight $w_\tau(c)$ close to 1 indicates high relative importance, while an importance weight $w_\tau(c)$ close to 0 indicates low relative importance.

3.3 Forming a Human Team

As we mentioned at the beginning of this chapter, it is a common understanding that when people work in teams, they can achieve great things. According to [Cohen and Bailey, 1997], "a team is a collection of individuals who are interdependent in their tasks, who share responsibility for outcomes, who see themselves and who are seen by others as an intact social entity embedded in one (or more) larger social systems". Notably, working in teams has proven to have multiple beneficial outcomes [Manzoor et al., 2011]:

- boosts productivity [Cohen and Bailey, 1997, Gallie et al., 2009];
- improves creativity and innovative thinking [Hoegl and Parboteeah, 2007];
- helps individuals to grow personally:
  - improve/acquire new competencies [Laal and Ghodsi, 2012],
  - develop soft skills [Strom and Strom, 2011]; and
- positively affects the members' job satisfaction [Ogbonnaya et al., 2018].

A *team* is much more than just a group of people. A team is a social system in which its members share a *common goal*, *interact* with each other, and *jointly work* towards their goal. A team's common goal is associated with a task and vice versa, as discussed in the previous section; therefore, we need a task in place for a group of people to become a team, i.e., a team cannot exist unless they have a task to work on. The task steers the team members on how to coordinate their actions to carry out the task and reach their goal successfully.

An essential factor of a team is *sharing responsibilities* and *complement* each other, as noted in [Cohen and Bailey, 1997] and [Kozlowski and Ilgen, 2006]. In our modelling, a responsibility corresponds to handling one of the task's required competencies. Therefore, sharing responsibilities is interpreted as assigning the required competencies to the team members. In contrast, each team member is expected to contribute to the teamwork by covering their assigned competencies. [Kargar and An, 2011, Kargar et al., 2013] the authors consider one team member as an "expert" for each required skill. From a different point of view, the authors in [Andrejczuk et al., 2019] consider *competence assignment functions*. A competence assignment function assigns to each team member a set of competencies so that the team member is responsible for their assigned competencies. In this work, we also adopt the concept of competence assignment functions.

Competence Assignment Functions

A competence assignment function defines how the agents within a team shall share the task's responsibilities, i.e., for which competencies each agent will be responsible for covering. [Andrejczuk, 2018] provides a thorough list of properties that a competence assignment function may satisfy.

Let us denote with $\eta_{\tau\to K}$ a competence assignment of a team K for a task τ, which maps each agent $a \in K$ with a subset of required competencies $\Xi \subseteq C_\tau$. Moreover, let us denote with $\theta_{\tau\to K}$ a function that maps each competence $c \in C_\tau$ with a subset of team members $S \subseteq K$, given a competence assignment $\eta_{\tau\to K}$, i.e., $\theta_{\tau\to K}$ show which agents are allocated to be responsible for each required competencies. According to [Andrejczuk, 2018], every competence assignment should guarantee that every required competence is assigned to at least one agent in the team. Intuitively, if a required competence is not assigned to any agent within a team, then this competence is not covered. Therefore the task cannot be completed, regardless of the importance of the competence for the task. Formally, following the definition of *inclusive competence assignment* in [Andrejczuk, 2018] we define:

Definition 7 (Competence Assignment Function (CAF)—Adapted from [Andrejczuk, 2018]). *Let $K \subseteq \mathcal{A}$ and $\tau \in \mathcal{T}$. A* competence assignment *function (CAF), $\eta_{\tau\to K} : K \to 2^{C_\tau}$, maps each agent $a \in K$ with a subset of required competencies $\Xi \subseteq C_\tau$. Each agent, $a \in K$, is responsible for covering their assigned competencies $\eta_{\tau\to K}(a) = \Xi$. For any CAF $\eta_{\tau\to K}$ it must hold that all required competencies are assigned to at least one agent, i.e., $\bigcup_{a\in K} \eta_{\tau\to K}(a) = C_\tau$*

With $\mathcal{H}_{\tau\to K}$, we denote the family of all CAFs of team K to task τ. For every CAF $\eta_{\tau\to K}$, there is a reverse competence assignment function $\theta_{\tau\to K}$.

Definition 8 (Reverse Competence Assignment Function (r-CAF)—Adapted from [Andrejczuk, 2018]). *Given a competence assignment function $\eta_{\tau\to K}$ of team K to task τ, there is exactly one reverse competence assignment function (r-CAF) $\theta_{\tau\to K} : C_\tau \to 2^K$ that maps each required competence $c \in C_\tau$ with a subset of agents $S \subseteq K$ who are responsible for c according to $\eta_{\tau\to K}$. For the r-CAF $\theta_{\tau\to K}$ given an $\eta_{\tau\to K}$ it must hold that for each agent a allocated to be responsible for some competence c in the r-CAF, this competence c is assigned to agent a in the CAF, and vice versa; i.e.,*

1. *if $c \in \eta_{\tau\to K}(a)$ then $a \in \theta_{\tau\to K}(c)$, and*

2. *if* $a \in \theta_{\tau \to K}(c)$ *then* $c \in \eta_{\tau \to K}(a)$.

With $\Theta_{\tau \to K}$, we denote the family of all r-CAFs of team K to task τ.

Given a task τ and a set K, there is a large number of competence allocations. In particular $\mathcal{H}_{\tau \to K}$ contain $|\mathcal{C}_\tau| \cdot 2^{|K|}$ different CAFs (and, respectively, $\Theta_{\tau \to K}$ contains $|\mathcal{C}_\tau| \cdot 2^{|K|}$ r-CAFs, one per each CAF). However, not all CAFs are equivalent or equally desired. For instance, consider a competence assignment $\eta_{\tau \to K}$ of a team $K = \{a_1, a_2, a_3\}$ for a task τ that requires $\mathcal{C}_\tau = \{c_1, c_2, c_3, c_4, c_5, c_6\}$; and that according to this competence assignment, agent a_1 is responsible for every required competence, i.e., $\eta_{\tau \to K}(a_1) = \mathcal{C}_\tau$, and agents a_2 and a_3 are responsible for no required competencies, i.e., $\eta_{\tau \to K}(a_2) = \eta_{\tau \to K}(a_3) = \emptyset$. This particular CAF is unfair by charging agent a_1 with covering all six required competencies, while agents a_2 and a_3 have no responsibilities. On the contrary, a different competence assignment $\tilde{\eta}_{\tau \to K}$ according to which agent a_1 is responsible for competencies c_5 and c_6 ,i.e., $\tilde{\eta}_{\tau \to K}(a_1) = \{c_5, c_6\}$, agent a_2 is responsible for competencies c_1 and c_4 ,i.e., $\tilde{\eta}_{\tau \to K}(a_2) = \{c_1, c_4\}$, and agent a_3 is responsible for competencies c_2 and c_3 ,i.e., $\tilde{\eta}_{\tau \to K}(a_3) = \{c_2, c_3\}$, is more *fair* as $\tilde{\eta}_{\tau \to K}$ shares responsibilities among team members evenly. Hence, we adopt the concept of *fair competence allocation function (FCAF)*, enriching the *inclusive competence assignment* discussed in [Andrejczuk, 2018]. Formally, we define an FCAF as:

Definition 9 (Fair Competence Assignment Function (FCAF)). *Let be a subset of agents* $K \subseteq \mathcal{A}$ *and some task* $\tau \in \mathcal{T}$. *A competence assignment function* $\eta_{\tau \to K}$ *is a* fair competence assignment function (FCAF) *if for every* $a \in K$ *it holds that:*

$$1 \leq |\eta_{\tau \to K}| \leq \left\lceil \frac{|\mathcal{C}_\tau|}{|K|} \right\rceil.$$

We consider fairness in terms of the amount of responsibilities assigned to each agent. Specifically, an FCAF determines the minimum and the maximum amount of an agent's responsibilities. Each agent must be included by being in charge of at least one competence. At the same time, each agent is responsible for, at most, as many competencies as if competencies were evenly assigned across all team members. The lower bound on the number of competencies to be assigned to each team member—also present in the definition of inclusive assignments—guarantees that every team member actively participates towards fulfilling the task. On the other hand, by introducing this particular upper bound, we avoid overloading some agents with excessive responsibilities while guaranteeing that all required competencies can be assigned to at least one agent.

Now, we can formally define a human team:

Definition 10 (Human Team). *Given a set of agents* $\mathcal{A}$ *and a set of tasks* $\mathcal{T}$, *a human team is defined as* $\langle K, \tau, \eta_{\tau \to K} \rangle$ *where* $K \subseteq \mathcal{A}$, $\tau \in \mathcal{T}$, *and* $\eta_{\tau \to K}$ *is a fair competence assignment function.*

3.4 Evaluating a Human Team

In this section, we discuss how to evaluate a single team. We go through the features used to describe a human agent as discussed in Section 3.1, and we put forward metrics that evaluate a team across each feature. In this work, we remind the reader that the features that describe an agent are *competencies, personality and gender, preferences over tasks* and *preferences over teammates*. As such, we evaluate a team across the following four dimensions:

1. *competence affinity*: describes the competency of a team for tackling the assigned task;
2. *personality diversity* & *gender balance*: describes the compatibility of the team members in terms of personality traits and gender balance;
3. *motivation*: describes the motivation of a team for working on the assigned task captured by individuals' preferences over tasks;
4. *social cohesion*: describes the team members' cohesiveness in terms of social relations captured by individuals' preferences over teammates.

3.4.1 Competence Affinity

We start with the feature of *competencies*. As discussed in Section 3.3, team members jointly work on some tasks by sharing responsibilities and coordinating their actions towards the common goal. In terms of competency, how efficient a team is at tackling a specific task depends on *(a)* the task's required competencies and *(b)* the team's offered competencies. A team's offered competencies refer to each team member's competencies at their disposal according to their competence profile (see Definition 1). In this section, we put forward the metric of *competence affinity*. The competence affinity metric, given a team $K \subseteq \mathcal{A}$, a task $\tau \in \mathcal{T}$ and a competence assignment function $\eta_{\tau \to K}$, evaluates:

1. each member for *covering* their assigned responsibilities; and
2. the overall *covering* of each required competence of the target task.

Before introducing our competence affinity metric, we must discuss competence models [Deist and Winterton, 2005] and competence ontologies [Miranda et al., 2017].

Competence Models and Competence Ontology

To do a matching (or alignment) between a set of agents' offered competencies and a task's required ones, it is essential to have in place a *competence model*. Competence or competency model refers to a framework that identifies and specifies the necessary competencies for individuals to efficiently/successfully carry out tasks defined in a domain at hand [Vazirani, 2010]. A competence model shall provide not only *which* competencies are needed in a domain, but also *how* they are needed. As noted in the work of [Andrejczuk et al., 2019], regarding the behavioural description of a competence model, one can discern two key models: the *binary model*; and the *graded model*. In a binary competence model (also referred to as Boolean competence model), a task requires some competence, and for an agent to handle this task it is sufficient to determine whether the agent acquires the competence or not. That is, such models, assume that a task that requires a single competence can be carried out equally well by two different agents who both acquire this competence; while two different agents who neither hold this competence will perform the task equally badly.

However, as [Andrejczuk et al., 2019] points out, binary models are rather limiting, and therefore scarcely realistic. In real life, it is rarely the case that a person just *has* or *has not* a competence. Instead, people usually have a competence *up to some degree* or do not have the competence at all. Such situations are captured via the *graded competence models*. That is, in a graded competence model a task requires some competence at a degree or grade; and, similarly, an agent acquires some competence up to a degree or grade. In this work, we consider graded competence models; remember that each required competence is required at some expertise level (see Definition 6), and each agent acquires a competence at some expertise level (see Definition 1). Moving a step further, one can notice that even though the graded competence model is more realistic compared to the binary one, the model is yet limiting. In particular, both the binary and the graded competence models assume that an agent can adequately handle a task if and only if the agent acquires the *exact* competencies required by the task (and meet the required levels in the case of the graded model). In other words, an agent would be considered inadequate for a task if they did not acquire some competence exactly as required by the task. Such a restriction may become extremely hard to be met, especially in multidisciplinary environments. For example, graduate students are equipped with competencies defined within the educational domain, while job positions require competencies defined in the industry domain; yet the students are suitable for the job market since they hold *similar* competencies to the ones required by the available jobs. As such below we propose the use of a structure that captures *semantic similarities* between different but essentially similar competencies, a concept that has been neglected so far.

Every two competencies c, c' in the collection of competencies $\mathcal{C}$ are distinctively different, however, they might share some *semantic* similarities. For instance, the competence of "managing information" (let it be competence c) is different to the "processing information" one (let it be competence c'), however, they share some essential similarities. That is, the former competence refers to the ability to store, organise and retrieve information in a manual or digitalised way, while the latter refers to the ability to insert, record, and update data using electronic or

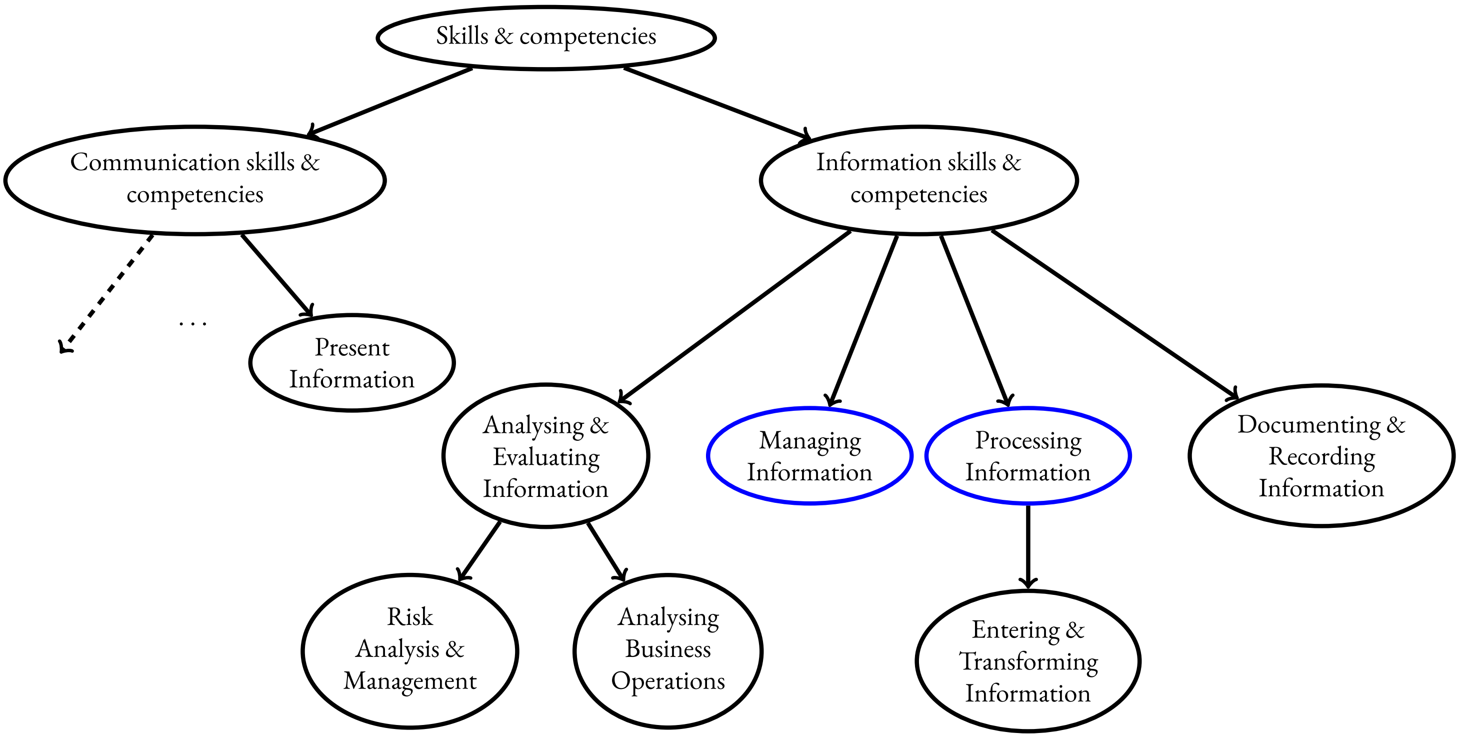


Figure 3.3: Example of graph-structured competence ontology (instance from ESCO).

manual systems.[4] Even though the competencies are different, it is natural to assume that an agent acquiring competence c would be somewhat adequate for handling competence c', and vice versa. As such, one could represent the two competencies in a graph structure where c and c' are nodes both semantically connected to a common ancestor which could be called "information skills and competencies". Figure 3.3 illustrates an example of such a graph structure. In this structure, nodes correspond to competencies and edges indicate intimate semantic relations between the nodes. Notice that the graph structure in Figure 3.3 contains *directed* edges. The direction of the edges indicates specialization or refinement; i.e., parent-nodes correspond to broader concepts while child-nodes correspond to narrower concepts. We refer to directed (acyclic) graph structures, similar to the one described above, as *competence ontologies* or *taxonomies*—in this work we adopt the term ontology. Notably, many countries are putting forward such structures in order to create a map of competencies across the different domains; with the primary goal to bridge the gap between the domain of education and the domain of industry. Indicatively, one can consider the existing ontologies listed below (we refer to [Popov et al., 2022] for further details on the existing competence ontologies):

[4] Example of competencies described in ESCO.

- ***O*Net***[5] is a competence ontology containing skills, abilities, work activities, training and job characteristics for occupations relative to the labour market of the United States. O*Net describes the competencies for almost 1,000 jobs in the US.

- ***ESCO***[6] is the European ontology of Skills, Competencies and Occupations, which describes the relations among skills defined with the EU labour market and education. ESCO contains 13,890 skills, providing high granularity, and is available in 28 Languages.

- ***Nesta***[7] provides a multi-level hierarchy of more than 10,500 competencies that were extracted from job advertisements in the United Kingdom. Nesta, like ESCO, is a competence ontology of high granularity.

- ***SFIA***[8] is a competence framework which maps skills and competencies required for digital occupations. SFIA lists skills required by Information & Communication Technology (ICT) roles defined by several organisations around the globe, such as the USA National Initiative for Cybersecurity Education (NICE), the Australian Public Service, the Ministry of ICT of the Kingdom of Saudi Arabia, etc.

Now given a competence ontology, the semantic similarity between two competencies can be quantified due to the graph structure of the ontology. That is, given a competence ontology represented as a graph, we can measure how semantically similar are two different competencies based on their position of the competencies in the graph. As we mentioned the nodes in an ontology-graph correspond to competencies; while a directed edge between two nodes, connects a broader concept with a relevant concept which is more specialised, i.e., with a more refined version of the broader concept. As such, the deeper in the ontology a competence is positioned, the more specialised concept it represents. Accordingly, when a competence lies in a higher position, then it corresponds to broader and more abstract concepts—compared to the competencies that exist deeper in the ontology. Competencies in leaf nodes, i.e., in nodes that have only incoming edges and no outgoing ones, are competencies that cannot be further specialised in the ontology at hand. On the other hand, competencies in nodes that contain solely outgoing edges (root nodes) correspond to competencies that cannot be further abstracted. In other words, the depth in which a competence lies within an ontology indicates the “abstractness” or “the specialisation” of the competence.

Therefore the semantic similarity between two competencies c and c' depends on the position of the competencies in the ontology. Here, we build on the similarity metric introduced

[5]`https://www.onetonline.org/`

[6]`https://esco.ec.europa.eu/en`

[7]`https://data-viz.nesta.org.uk/`

[8]`https://sfia-online.org/`

in [Li et al., 2003] for computing semantic similarities between words in an ISA lexical hierarchical structure, and we adjust this metric for computing semantic similarities between competencies within a competence ontology. As such, given a competence ontology over a set of competencies $\mathcal{C}$, we define a *semantic similarity function* $\text{sim} : \mathcal{C} \times \mathcal{C} \to [0, 1]$ that maps each pair of competencies $c, c' \in \mathcal{C}$ with a real number in the range $[0, 1]$. For computing the similarity of two competencies $c, c' \in \mathcal{C}$ we consider *(i)* the shortest path between c and c' (ignoring the direction of the edges in the graph), and *(ii)* the competencies' deepest common ancestor, i.e., the deepest competence subsuming both c and c'. Let l denote the shortest path between c and c', and h denote the depth of the deepest common ancestor of c and c'. Hence we compute the similarity between two competencies $c, c' \in \mathcal{C}$ as:

$$\text{sim}(c, c') = e^{-\lambda \cdot l} \cdot \frac{e^{\kappa \cdot h} - e^{-\kappa \cdot h}}{e^{\kappa \cdot h} + e^{-\kappa \cdot h}} \tag{3.1}$$

where λ, κ are parameters used to regulate the influence of l and h on the similarity metric, respectively. According to Eq (3.1), the closest the competencies are (i.e., the lower the value that l has) the greater the similarity between the competencies is, and the deeper the common ancestor is (i.e., the higher the value that h has) the greater the similarity is. The shortest path captures that two competencies positioned in close proximity in the ontology shall be more similar than two competencies positioned in farther proximity; that is, a pair of competencies $\langle c_1, c_2 \rangle$ with shortest path l is more semantically similar than another pair $\langle c_3, c_4 \rangle$ with shortest path $l' > l$. For example, considering the ontology in Figure 3.1, competence "Risk Analysis & Management" is more similar to competence "Analysing Business Operations" with shortest path $l = 2$, compared to competence "Entering & Transforming Information" with shortest path $l = 4$. Considering regulating parameters $\lambda = 1$ and $\kappa = 100$ we have:[9]

$$\text{sim}(\text{“Risk Analysis \& Management”}, \text{“Analysing Business Operations”}) = e^{\lambda \cdot 2} \frac{e^{\kappa \cdot 3} - e^{-\kappa \cdot 3}}{e^{\kappa \cdot 3} + e^{-\kappa \cdot 3}} \simeq 0.18$$

$$\text{sim}(\text{“Risk Analysis \& Management”}, \text{“Entering \& Transforming Information”}) = e^{\lambda \cdot 4} \frac{e^{\kappa \cdot 2} - e^{-\kappa \cdot 2}}{e^{\kappa \cdot 2} + e^{-\kappa \cdot 2}} \simeq 0.0003$$

Intuitively, an ontology shall structure the competencies based on conceptual characteristics—or on empirical observations such as how often one competence has substituted another one—hence the competencies that are conceptually close, shall be also positioned in close proximity in the ontology and vice versa. The deepest common ancestor captures the level of "abstractness" of the competence that connects the two competencies of interest. Intuitively, competencies in close proximity that lie deeper within the ontology shall exhibit a greater degree of semantic similarity than competencies (in similar proximity) that lie in less depth with the ontology; since the competencies in the former case are more specialised while in the latter case,

[9] We use here a large value for the parameter κ to "smather" the effect of the deepest common ancestor in the metric.

the competencies are more abstract. For example, the similarity between competencies "Risk Analysis & Management" and "Analysing Business Operations" shall be greater than the similarity between "Communication skills & competencies" and "Information skills & competencies" although the shortest path in both cases has the same length since in the latter case, we have two broad competencies that are node-parents to a large number of more refined competencies while in the former case, the competencies are already refined enough. Considering regulating parameters $\lambda = 0.3$ and $\kappa = 0.4$ we have:

$$\text{sim}(\text{“Risk Analysis \& Management”}, \text{“Analysing Business Operations”}) = e^{\lambda \cdot 2}\frac{e^{\kappa \cdot 3} - e^{-\kappa \cdot 3}}{e^{\kappa \cdot 3} + e^{-\kappa \cdot 3}} \simeq 0.457$$

$$\text{sim}(\text{“Risk Analysis \& Management”}, \text{“Entering \& Transforming Information”}) = e^{\lambda \cdot 4}\frac{e^{\kappa \cdot 2} - e^{-\kappa \cdot 2}}{e^{\kappa \cdot 2} + e^{-\kappa \cdot 2}} \simeq 0.364$$

As such, the semantic similarity function in Eq (3.1) offers the two desired properties we discussed above, and allows these two properties to be regulated to increase / decrease the influence of each property in the similarity. However, the function in Eq (3.1) exhibit some undesirable, obscure behaviour. Specifically, as [Osman et al., 2014] points out, a similarity function should range in $[0, 1]$, with values close to 0 indicating that the entities in comparison (here competencies) are not similar while values close to 1 indicating that the entities are very similar, and ultimately entities exhibit similarity equal to 1 if and only if the entities are *identical*. However, according to Eq (3.1) a similarity of value 1 between two competencies c and c' can be achieved when *(a)* the shortest path between c and c' is minimum i.e., $l = 0$; and *(b)* the depth of the deepest common ancestor tends to infinity, i.e., $h \rightarrow \infty$. Notably, the similarity of a competence c to itself $\text{sim}(c, c)$ at some depth h is lower than the similarity another competence c' to itself $\text{sim}(c', c')$ at a deeper depth $h' > h$, i.e., $\text{sim}(c, c) < \text{sim}(c', c')$. In other words, the similarity function in Eq (3.1) lacks the *reflexive property of similarity*, i.e., a competence is maximally similar to itself regardless of its depth with the ontology. For this reason, we use a variation of the metric proposed in [Li et al., 2003] so that we guarantee the reflexive property of similarity:

$$\text{sim}(c, c') = \begin{cases} e^{-\lambda \cdot l} \cdot \frac{e^{\kappa \cdot h} - e^{-\kappa \cdot h}}{e^{\kappa \cdot h} + e^{-\kappa \cdot h}} & \text{if } l > 0 \\ 1 & \text{if } l = 0 \end{cases} \tag{3.2}$$

At this point, we would like to notice that this similarity metric we are using is highly sensitive to the total depth of the ontology at hand. Hence, parameters λ and κ should be carefully tuned to properly serve the ontology used. Notably, the lower the λ is, the more influential the shortest path is, and the lower the κ is, the more influential the depth of the deepest common ancestor is.

Competence Coverage

To begin with, we remind the reader that each task $\tau \in \mathcal{T}$ requires a set of competencies, denoted as $\mathcal{C}_\tau$; and each competence $c \in \mathcal{C}_\tau$ is required at some expertise level, determined through a level function $l_\tau : \mathcal{C}_\tau \to Q_{\text{Level}}$; and is relatively important for the task with some importance weight, determined though a weight function $w_\tau : \mathcal{C}_\tau \to Q_{\text{Importance}}$. Accordingly, each agent $a \in \mathcal{A}$ acquires a set of competencies, denoted as $\mathcal{C}_a$; and they acquire each competence $c \in \mathcal{C}_a$ at some expertise level, determined through a level function $l_a : \mathcal{C}_a \to Q_{\text{Level}}$. Let us assume that agent $a \in \mathcal{A}$ is assigned to work on task $\tau \in \mathcal{T}$; and also let us assume, for now, that a needs to work on τ on their own. Therefore, all the task's required competencies shall be *covered* by this agent; i.e., agent a will be responsible for each one of the required competence $c \in \mathcal{C}_\tau$. In the case where a required competence $c \in \mathcal{C}_\tau$ is one of the competencies which the agent acquires, then the agent can cover this competence with a level of:

$$\text{cvg}(c, a) = l_a(c)$$

Nonetheless, it is a rather strong assumption that an agent is equipped with each and every one of the required competencies. That is, it is unrealistic to believe that each required competence in $\mathcal{C}_\tau$ is one of the agent's acquired competencies in $\mathcal{C}_a$, i.e., we cannot guarantee that it always holds that $\mathcal{C}_\tau \subseteq \mathcal{C}_a$. Instead, considering the discussion in the previous section, for each required competence $c \in \mathcal{C}_\tau$ there is some of the agent's acquired competence $c' \in \mathcal{C}_a$ that it is maximally similar to c. As such, in this case we say that a can cover a competence $c \in \mathcal{C}_\tau$ with a level of:

$$\text{cvg}(c, a) = \max_{c' \in \mathcal{C}_a} \left(l_a(c') \cdot \text{sim}(c, c') \right)$$

Thus, having in mind that an agent a might acquire or not a given competence c, we compute the *competence coverage* of the competence by the agent as:

$$\text{cvg}(c, a) = \begin{cases} l_a(c) & \text{if } c \in C_a \\ \max_{c' \in \mathcal{C}_a} \left(l_a(c') \cdot \text{sim}(c, c') \right) & \text{otherwise} \end{cases} \tag{3.3}$$

Therefore, given a task $\tau \in \mathcal{T}$ and a single agent $a \in \mathcal{A}$, we say that the agent with their acquired competencies can cover the required by the task ones as:

$$\text{cvg}(\tau, a) = \prod_{c \in \mathcal{C}_\tau} \text{cvg}(c, a) \tag{3.4}$$

In Eq 3.4, we use the product over the competence coverage of the agent across all the required competencies. In this way, the coverage of each competence is *equally* contributing to the determination of the coverage of the task by the agent.

Team's Competence Affinity

Moving now from a single agent to a team of agents, we need a different way to dignify the contribution of each team member to the teamwork, and the overall competency of the team as a whole for the assigned task. As highlighted in [Kurtan et al., 2020], an ideal team is not just a group of ideal agents. In other words, the competency of a single agent to tackle a task does not reflect the competency of that agent to tackle the task when the agent is part of the team. Instead, team members share responsibilities; therefore the competency of an agent for a task depends on the competency of the agent to cover their assigned responsibilities. As such, here we define a team's *competence affinity* for tackling the assigned task. Our competence affinity metric takes into consideration:

1. the agents' competence profile;
2. the agents' assigned competencies according to a competence assignment function $\eta_{\tau \to K}$; and
3. the relevant importance of each competence.

Moreover, we define the competence affinity metric so that it satisfies the following three requirements:

1. the higher the coverage of an assigned competence, the higher the competence affinity;
2. the lower the importance of an assigned competence, the higher the competence affinity; and
3. the competence affinity is at most equal to the coverage of any assigned competence with maximal importance.

Highly covered competencies should naturally be reflected in the overall competence affinity by high contribution. On the other hand, competencies which are covered poorly should contribute accordingly to their importance. That is, as some competencies are more important than others, intuitively a prioritisation is in place targeting to achieve higher coverage in the most critical competencies. Conversely, achieving lower coverage in less important competencies should not significantly impact the overall competence affinity.

As we mentioned before, a team's competency depends on the contribution of the team members to their assigned task, i.e., how well each team member can cover the responsibilities assigned to them according to the competence assignment function at hand. Thus, given a team $K \subseteq \mathcal{A}$, its assigned task $\tau \in \mathcal{T}$ and a competence assignment function $\eta_{\to}$, we first define an agent's competence affinity, as:

Definition 11 (Agent's Competence Affinity). *Given an agent $a \in \mathcal{A}$, a task $\tau \in \mathcal{T}$, and a competence assignment function $\eta_{\tau \to K}$, the competence affinity of a to τ is:*

$$\mathrm{aff}(a, \tau, \eta_{\tau \to K}) = \prod_{c \in \eta_{\tau \to K}(a)} \max\left\{(1 - w_\tau(c)), \mathrm{cvg}(c, a)\right\} \tag{3.5}$$

Therefore, the overall team's competence affinity is computed as the product of each agent's competence affinity. The product assigns a larger value to teams where all agents equally contribute to their assigned task, rather than to teams with unbalanced contributions.[10]

Definition 12 (Team's Competence Affinity). *Given a team of agents $K \subseteq \mathcal{A}$, its assigned task $\tau \in \mathcal{T}$, and a competence assignment function $\eta_{\tau \to K}$, the competence affinity of K to τ is:*

$$\mathrm{aff}(K, \tau, \eta_{\tau \to K}) = \prod_{a \in K} \mathrm{aff}(a, \tau, \eta_{\tau \to K}) \tag{3.6}$$

Observe that the competence affinity of a team to its assigned task depends on the competence assignment function. In other words, for a given set of agents K and a given task τ, different competence assignment functions result in different competence affinities. Finding the competence assignment function that yields the highest competence affinity is actually an optimisation problem. As such, in order to compute the team's competence affinity, we should consider the *optimal* competence assignment given by:

$$\eta^*_{\tau \to K} = \arg\max_{\mathcal{H}_{\tau \to K}} \mathrm{aff}(K, \tau, \eta) \tag{3.7}$$

where $\mathcal{H}_{\tau \to K}$ denotes the family of all CAFs for task τ and team K. Therefore, considering also fair competence assignments, we need to solve the following optimisation problem: let us us a binary decision variable x_a^c for each agent $a \in K$ and each requires competence $c \in C_\tau$ which indicates that competence c is assigned to agent a. Then the non-linear optimisation problem to solve is:

$$\max_{\eta_{\tau \to K} \in \mathcal{H}_{\tau \to K}} \left(\mathrm{aff}(K, \tau, \eta_{\tau \to K})\right)^{x_a^c} = \max_{\eta_{\tau \to K} \in \mathcal{H}_{\tau \to K}} \left(\prod_{a \in K} \prod_{c \in \eta_{\tau \to K}(a)} \mathrm{cvg}(a, c)\right)^{x_a^c} \tag{3.8}$$

subject to

$$\sum_{a \in K} x_a^c \geq 1 \qquad \forall\, c \in C_\tau \tag{3.8a}$$

$$1 \leq \sum_{c \in C_\tau} x_a^c \leq \left\lceil \frac{|C_\tau|}{|K|} \right\rceil \qquad \forall\, a \in K \tag{3.8b}$$

[10] Remember that the product promotes distributing evenly the agents' contributions.

The above optimisation problem can be easily solved using an out-of-the-self optimiser with the means of linear programming. Specifically, we can solve an equivalent linear optimisation problem:

$$\max_{\eta_{\tau\rightarrow K} \in \mathcal{H}_{\tau\rightarrow K}} \sum_{a\in K} \sum_{c\in\eta_{\tau\rightarrow K}(a)} x_a^c \cdot \log\left(1 + \mathrm{cvg}(c, a)\right) \tag{3.9}$$

subject to Eq (3.8a) and Eq (3.8b).

Note that the Eq (3.9) is an equivalent optimisation problem with Eq (3.8). That is, without affecting the monotonicity of the function *(i)* we use the $\log(\cdot)$ to convert the double product to double sum, and the powered factor into a product; and *(ii)* we change the function's domain to avoid $\log(0)$. Solving this problem optimally is rather inexpensive, considering that, in practice, for a task τ both team size s_τ and the number of required competencies $|C_\tau|$ are relatively small: usually, team size range in $[2, 5]$, while the required competencies are usually less than 10.

3.4.2 Personality Diversity & Gender Balance

The second feature across which we want to evaluate a team is that of personality and gender. According to Definition 2, an agent's personality profile is described by a quartet $\langle SN, TF, EI, JP\rangle$ that positions the agents in a four-dimensional personality space, as shown in Figure 3.1.

[Wilde, 2009] conducted a series of experiments to study student teams and reach some valuable conclusions regarding the combinations of personality types that compose efficient teams. [Andrejczuk, 2018, Andrejczuk et al., 2019] introduced a novel metric to measure *congeniality* of a team based on Wilde's findings. Specifically, the congeniality metric proposed in [Andrejczuk, 2018] follows four base rules for composing a team:

1. The team members should be as diverse as possible across the Sensing-Intuition (SN) and Thinking-Feeling (TF) dimensions.
2. There should be at least one member that inclines towards the extrovert (E), the thinking (T) and the judging (J) extreme at the same time.
3. There should be at least one introvert (I) member.
4. There should be gender balance within the team.

[Andrejczuk, 2018] used a numerical representation of personality, considering a four-element vector where each element corresponds to one of the personality dimensions and ranges from -1 to 1, indicating the two extremes of the dimension. For example, such a vector is $\mathbf{p} = [sn, tf, ei, jp]^\top$ where the first element sn represents the Sensing-Intuition dimension and when $sn = -1$ the individual stands in the sensing extreme, when $sn = 1$ the individual stands in the intuition extreme, and when $-1 < sn < 1$ the individual stands somewhere in-between.

As such, the personality profile of an agent according to Definition 2 shall be $\mathcal{P}_a^{\text{personality}} = \langle SN_a, TF_a, EI_a, JP_a\rangle = [SN_a, TF_a, EI_a, JP_a]^\top \in [-1, 1]^4$, i.e., each element in $\mathcal{P}^{\text{personality}}$ is represented by a real number in the range $[-1, 1]$. Then [Andrejczuk et al., 2018] put forward the congeniality metric as:

Definition 13 (Team's Congeniality [Andrejczuk et al., 2018]). *Given a team of agents $K \subseteq \mathcal{A}$, the congeniality of the team is computed as:*

$$\text{cong}(K) = \sigma_{SN}(K) \cdot \sigma_{TF}(K) + \max_{a \in K^{ETJ}} \big(\max(\mathbf{c} \cdot \mathbf{p}, 0)\big) + \max_{a \in K} \big(\max(\mathbf{d} \cdot \mathbf{p}, 0)\big) + e \cdot \sin\left(\frac{fem}{|K|} \cdot \pi\right)$$

where $\sigma_{SN}(K)$ and $\sigma_{TF}(K)$ stands for the standard deviation of the team members in the SN and the TF dimensions, respectively; $K^{ETJ} = \{a \in K : TF_a > 0, EI_a > 0, JP_a > 0\}$ is the subset of the agents in K that incline towards extrovert, thinking and judging extreme. $\mathbf{c} = [0, c, c, c]$ is a vector of importance weights for the TF, EI and JP personality dimensions; $\mathbf{d} = [0, 0, -d, 0]$ is a vector of importance weight for the EI dimension. Finally, fem is the number of agents of the female gender in the team K, and e weighs the influence of gender balance in the team's congeniality.

Andrejczuk argues that parameters c and d shall be such to weight the personality dimensions equally, and therefore the author opted that $c = 0.19$ and $d \leq 1$. Moreover, in order to allow the gender balance equally influence the team's congeniality with the other elements, an appropriate compromise is to use $e = 0.1$. Notably, Adrejczuk considers solely two genders, i.e., an agent can be either male or female. However, people may identify themselves differently as either male or female. Thus, we adjust the congeniality metric to consider more than two genders. Specifically, we propose a metric to measure the diversity of a team regarding a feature with categorical values. In our view, a diversity metric should satisfy the following properties:

1. diversity should be minimum when every team member holds the same value for the feature; and
2. diversity should be maximum when the feature values' within the team are distributed similarly to the feature values' within the total population.

Thus, we put forward a definition of the diversity of a team with respect to some feature with categorical values that satisfy the two properties mentioned above:

Definition 14 (Team's Diversity). *Given a set of agents $\mathcal{A}$, a team of agents $K \subseteq \mathcal{A}$ and a feature f with categorical values, we compute the diversity of K as:*

$$\text{diversity}_f(K) = \left(\prod_{v \in V} \exp\left(-\frac{\left|\frac{n_v}{|K|} - \mathrm{f}_v\right|}{1 - \frac{n_v}{|K|} + \varepsilon}\right)\right)^{\frac{1}{|V|}} \tag{3.10}$$

where $V = \{v_1, v_2, \cdots, v_k\}$ *is a set of categorical values for feature* f*;* n_v *stands for the number of agents in team* K *that has value* v*,* f_v *is the frequency with which we can find value* v *in the total population* $\mathcal{A}$*, and* ε *is a small positive number.*

Now considering the feature of *gender*, let f be a frequency distribution of genders on the total agents' population $\mathcal{A}$. For example, if agents identify themselves in 3 genders $V_{gender} = \{Male, Female, Non-Binary\}$ and 40% of the agents are male, 40% are female and 20% are non-binary,then the frequency distribution would be $\mathrm{f}_{Male} = 0.4$, $\mathrm{f}_{Female} = 0.4$ and $\mathrm{f}_{Non-Binary} = 0.2$. Then, the diversity of a team K concerning the gender feature is given by:

$$\text{diversity}_{gender}(K) = \left(\prod_{gn \in V_{gender}} \exp\left(-\frac{\left| \frac{n_{gn}}{|K|} - \mathrm{f}_{gn} \right|}{1 - \frac{n_{gn}}{|K|} + \varepsilon} \right) \right)^{\frac{1}{|V_{gender}|}} \tag{3.11}$$

Therefore, we adjust the congeniality metric proposed in [Andrejczuk, 2018] in order to consider more than two genders:

Definition 15 (Team's Congeniality)**.** *Given a team of agents* $K \subseteq \mathcal{A}$*, the congeniality of* K *is:*

$$\text{cong}(K) = \sigma_{SN}(K) \cdot \sigma_{TF}(K) + \max_{a \in K^{ETJ}} \big(\max(\mathbf{c} \cdot \mathbf{p}, 0) \big) + \max_{a \in K} \big(\max(\mathbf{d} \cdot \mathbf{p}, 0) \big) + e \cdot \text{diversity}_{gender}(K)$$

where $\sigma_{SN}(K)$ *and* $\sigma_{TF}(K)$ *stands for the standard deviation of the team members in the SN and the TF dimensions, respectively;* $K^{ETJ} = \{a \in K : TF_a > 0, EI_a > 0, JP_a > 0\}$ *is the subset of the agents in* K *that incline towards extrovert, thinking and judging extreme.* $\mathbf{c} = [0, c, c, c]$ *is a vector of importance weights for the TF, EI and JP personality dimensions;* $\mathbf{d} = [0, 0, -d, 0]$ *is a vector of importance weights for the EI dimension; and* e *weighs the influence of gender diversity in the team's congeniality.*

3.4.3 Motivation

The next metric we introduce is related to agents' motivation. Specifically, we consider the agents' preferences over tasks to measure the agents' satisfaction with their assigned tasks. How motivated a team is to work on their assigned task depends on how much each team member *is willing* to work on this task. Thus, here we put forward the *motivation* metric; which, given a team $K \subseteq \mathcal{A}$ and its assinged task $\tau \in \mathcal{T}$, it evaluates:

1. the *satisfaction* of each team member; and
2. the overall *motivation* of the team.

To begin with, according to Definition 3, each agent has a task preference profile, which corresponds to an ordering of the tasks per agent. That is, for an agent $a \in \mathcal{A}$ and any two tasks $\tau, \tau' \in \mathcal{T}$ it holds that:

$$\tau \succsim_a^{\mathcal{T}} \tau' \Leftrightarrow a \text{ prefers working on } \tau \text{ as least as much as working on } \tau'$$

The *satisfaction* of an agent $a \in \mathcal{A}$ towards task $\tau \in \mathcal{T}$ depends on the position of τ on a's preference ordering. Thus, we define the agent's satisfaction [11] as:

Definition 16 (Agent's Satisfaction). *Given an agent $a \in \mathcal{A}$ and a task $\tau \in \mathcal{T}$, a' satisfaction for working on τ is computed as:*

$$\mathrm{sat}(a, \tau) = \frac{|\mathcal{T}| - \mathrm{pos}(\succsim_a^{\mathcal{T}}, \tau) + 1}{|\mathcal{T}|} \tag{3.12}$$

where $|\mathcal{T}| = m$ is the number of tasks in $\mathcal{T}$, and $\mathrm{pos}(\succsim_a^{\mathcal{T}}, \tau)$ stands for the position of task τ in the preference ordering $\succsim_a^{\mathcal{T}}$.

When $\mathrm{pos}(\succsim_a^{\mathcal{T}}, \tau)$ is equal to 1, then τ is the *most* preferred task for agent a; while when $\mathrm{pos}(\succsim_a^{\mathcal{T}}, \tau)$ is equal to m, then the τ is the *least* preferred task for agent a. When an agent a is indifferent between two different tasks $\tau, \tau' \in \mathcal{T}$, denoted as $\tau \sim_a^{\mathcal{T}} \tau'$, we assume that tasks τ and τ' are in the same position in the preference over tasks, i.e., $\mathrm{pos}(\succsim_a^{\mathcal{T}}, \tau) = \mathrm{pos}(\succsim_a^{\mathcal{T}}, \tau')$.

We want to highlight that we can "extract" the agents' preference profile by explicitly defining the $\mathrm{sat}(\cdot)$ function. That is, each agent $a \in \mathcal{A}$ dignifies a function $\lambda_a^{\mathcal{T}} : \mathcal{T} \to [0, 1]$, and for which it holds that task $\tau \in \mathcal{T}$ is at least preferred as $\tau' \in \mathcal{T}$ if and only if the value of τ according to $\lambda_a^{\mathcal{T}}$ is greater or equal to the value of τ', i.e.,:

$$\tau \succsim_a^{\mathcal{T}} \tau' \Leftrightarrow \lambda_a^{\mathcal{T}}(\tau) \geq \lambda_a^{\mathcal{T}}(\tau')$$

Such a function can be easily obtained by asking each agent to *rate* each task in a five-Likert Scale, with 1 indicating the least preferred tasks and 5 indicating the most preferred tasks. In such a case, we have that the agent's satisfaction corresponds to the following:

$$\mathrm{sat}(a, \tau) = \lambda_a^{\mathcal{T}}(\tau) = \frac{a\text{'s rating for } \tau}{5} \tag{3.13}$$

Now, the motivation of a team depends on its members' satisfaction. The more satisfied each member is with the assigned task, the more motivated they are to work on it, and, importantly, the more likely they are to inspire non-motivated team members. In other words, more motivated members of a team are identified as "leaders" (see Proposition 8 in [Ellemers et al., 2004]), and leaders tend to positively motivate their "followers" (see Proposition 9 in [Ellemers et al., 2004]).

[11] Note that by agent satisfaction, we refer to satisfying and respecting the agent's preferences. This is not to be confused with the term "job satisfaction" [Tietjen and Myers, 1998] in Motivational Psychology.

Thus, our *motivation* metric shall satisfy the following properties:

1. motivation is maximum when all team members announce the task as their most preferred one;
2. motivation is minimum when all team members announce the task as their most preferred one;
3. the more team members who are maximally satisfied with the task, the more motivated the team is; and
4. the more team members who are minimally satisfied with the task, the less motivated the team is.

With the above in mind, we compute a team's motivation for working on its assigned task as the geometric mean of the team members' satisfaction. Formally:

Definition 17 (Team's Motivation)**.** *Given a team of agents $K \subseteq \mathcal{A}$ and its assigned task $\tau \in \mathcal{T}$, the motivation of K for working on τ is:*

$$\mathrm{mot}(K, \tau) = \left(\prod_{a \in K} \mathrm{sat}(a, \tau)\right)^{\frac{1}{|K|}} \tag{3.14}$$

We opted for the geometric mean because it is immune to the team's size and fluctuations.

3.4.4 Social Cohesion

Lastly, we introduce a metric related to the agents' acceptance for and being accepted by their team. More precisely, in this section, we introduce a variation of the group efficiency metric proposed in [Ballesteros-Perez et al., 2012]. We consider the agents' preferences over potential teammates to measure the team's acceptance by each member. Thus, we put forward a *social cohesion* metric, which given a team $K \subseteq \mathcal{A}$, evaluates:

1. the *fondness* of each member for the rest of the team; and
2. the overall *social cohesion* of the team.

According to Definition 4, each agent has a teammate preference profile, which corresponds to an ordering of all agents. That is, for an agent $a \in \mathcal{A}$ and any other two agents $b, c \in \mathcal{A}$ it holds that:

$$b \succsim_a^{\mathcal{A}} c \Leftrightarrow a \text{ prefers working with } b \text{ as least as much as working with } c$$

The *fondness* of an agent $a \in \mathcal{A}$ towards another agent $b \in \mathcal{A}$ depends on the position of b in a' preference ordering over teammates. Thus we define the agent's teammate fondness as:

Definition 18 (Agent's Teammate Fondness). *Given an agent $a \in \mathcal{A}$ and another agent $b \in \mathcal{A}$, a's fondness for working with b is computed as:*

$$\mathrm{fond}(a, b) = \frac{|\mathcal{A}| - \mathrm{pos}(\succsim_a^{\mathcal{A}}, b) + 1}{|\mathcal{A}|} \tag{3.15}$$

where $|\mathcal{A}| = n$ is the number of agents in $\mathcal{A}$, and $\mathrm{pos}(\succsim_a^{\mathcal{A}}, b)$ *stands for the position of agent b in the preference ordering $\succsim_a^{\mathcal{A}}$.*

Again, when $\mathrm{pos}(\succsim_a^{\mathcal{A}}, b)$ is equal to 1, then agent b is the most preferred teammate for agent a, while when $\mathrm{pos}(\succsim_a^{\mathcal{A}}, b)$ is equal to 0, then agent b is the least preferred teammate for agent a. When agent a is indifferent between two different teammates $b, c \in \mathcal{A}$, is denoted with $b \sim_a^{\mathcal{A}} c$, and we assume that agents b and c are in the same position in the preference over teammates, i.e., $\mathrm{pos}(\succsim_a^{\mathcal{A}}, b) = \mathrm{pos}(\succsim_a^{\mathcal{A}}, c)$. Notice that for the sake of completeness, we consider a preference relation over all agents in $\mathcal{A}$, including a themselves. That is, including a in the ordering of $\succsim_a$ $\mathcal{A}$ for every $a \in \mathcal{A}$, we are able to *compare* two preferences of two different agents if necessary. However, since any agent a cannot avoid being in a team with themselves, we assume that for any $a \in \mathcal{A}$ it stands $\mathrm{pos}(\succsim_a^{\mathcal{A}}, a) = 1$, i.e., a is among the most preferred teammates for a.

Similarly to the task preference profile, we can extract the agents' teammate preference profile by explicitly defining the $\mathrm{fond}(\cdot)$ function. That is, each agent $a \in \mathcal{A}$ dignifies a function $\lambda_a^{\mathcal{A}} : \mathcal{A} \longrightarrow [0, 1]$, and for which it hold that agent $b \in \mathcal{A}$ is at least preferred as $c \in \mathcal{A}$ if and only id the value of b according to $\lambda_a^{\mathcal{A}}$ is greater or equal to the value of c, i.e.,:

$$b \succsim_a^{\mathcal{A}} c \Leftrightarrow \lambda_a^{\mathcal{A}}(b) \geq \lambda_a^{\mathcal{A}}(c)$$

Such a function can be easily obtained by asking each agent to *rate* each task on a five-Likert Scale, with 1 indicating the least preferred teammate and 5 indicating the most preferred teammate. In such a case, we have that the agent's fondness for a teammate corresponds to the following:

$$\mathrm{fond}(a, b) = \lambda_a^{\mathcal{A}}(b) = \frac{a\text{'s rating for } b}{5} \tag{3.16}$$

Now, give a team $K \subseteq \mathcal{A}$ who is working on a task τ, we define an agent's $a \in K$ fondness for the team K as:

Definition 19 (Agent's Team Fondness). *Given a team of agents $K \subseteq \mathcal{A}$, the fondness of any agent $a \in K$ for the team K is computed as:*

$$\text{fond}(a, K) = \sum_{b \in K} \frac{\text{fond}(a, b)}{|K|} \tag{3.17}$$

where $|K|$ is the team size.

$$\text{soc}(K) = \left(\prod_{a \in K} \text{fond}(a, K) \right)^{\frac{1}{|K|}} \tag{3.18}$$

The *social cohesion* of a team depends on its members' fondness for the team. The more mutually accepted the team is by each member, the more socially coherent the team is. In other words, teams with reciprocated accepted members tend to stick together [Harwood and Thrower, 2020].

Thus, our *social cohesion* metric shall satisfy the following properties:

1. social cohesion is maximum when all team members maximally fond their team, and
2. Social cohesion is minimum when all team members minimally fond their team.

Therefore, we compute a team's social cohesion as the geometric mean of the members' fondness for the team. Formally:

Definition 20 (Team's Social Cohesion). *Given a team of agents $K \subseteq \mathcal{A}$ working on some task $\tau \in \mathcal{T}$, the social cohesion of K is:*

$$\text{soc}(K, \tau) = \left(\prod_{a \in K} \text{fond}(a, K) \right)^{\frac{1}{|K|}} = \left(\prod_{a \in K} \sum_{b \in K} \frac{\text{fond}(a, b)}{|K|} \right)^{\frac{1}{|K|}} \tag{3.19}$$

We opted for the geometric mean because it is immune to the team's size and fluctuations.

3.4.5 Collegiality

In the previous sections, we discussed how to evaluate a single team of agents across the four features of interest: competencies, personality and gender, preferences over tasks, and preferences over potential teammates. For this purpose, we introduced three novel metrics and adjusted an existing one. However, as we have discussed so far, a team's performance when they carry out

Figure 3.4: Team's Collegiality

some task and how effective the team is depends on all four features. Here, we introduce the *collegial value* of a team that balances the influence of each one of the metrics competence affinity, congeniality, motivation, and social cohesion to the teamwork. Formally, we define *collegiality* as:

Definition 21 (Team's Collegiality)**.** *Given a team of a gents $K \subseteq \mathcal{A}$ who works on a task $\tau \in \mathcal{T}$ the collegiality of K is computed as:*

$$\text{collegiality}(K, \tau) = \alpha \cdot \text{aff}\,(K, \tau, \eta^*_{\tau \to K}) + \beta \cdot \text{cong}(K) + \gamma \cdot \text{mot}(K, \tau) + \delta \cdot \text{soc}(K) \quad (3.20)$$

where α, β, γ and δ are positive real numbers used to regulate the influence of competence affinity, congeniality, motivation, and social cohesion on the collegial value, respectively.

3.5 Summary

This chapter explored the essential components for forming human teams associated with some task. We began by discussing the fundamental features that describe a human agent. These features include competencies, knowledge, and skills that allow individuals to complete tasks on their own or as part of a team. We then turned our attention to personality, which is an enduring characteristic that drives people's behaviour and adjustment to life in general. Humans have a unique personality that influences how they react to situations and interact with others. Additionally, we discussed how gender could impact an individual's behaviour as it is linked to personality traits. Next, we examined how human agents prioritise different options and express their preferences. We learned that people's preferences could affect their motivation and willingness to accept an option. In a team formation scenario, agents express their preferences for the different tasks and teammates to work with. Understanding these preferences can help form effective teams that can efficiently tackle their assigned tasks. Having specified the above features, we formally defined human agents as a combination of these features.

In addition, we talked about tasks and how to describe them formally. A task is simply an activity someone needs to do to achieve a goal. However, to successfully complete a task, the agent(s) must have the necessary skills and knowledge, i.e., the necessary competencies. Otherwise, failure is expected. Furthermore, we discussed that the difficulty of a task and the amount of work involved determine the ideal team size needed to complete it. So, we formally describe a task through its competencies and team size requirements. Next, we described a human team that associates a set of humans with a task by fully determining the responsibilities of each team member in order to complete the task. In particular, we discussed *competence assignments*, functions responsible for distributing duties among the team members.

Finally, we elaborated on how to evaluate a team, i.e., how suitable or efficient is a set of agents for working on a task. Specifically, we introduced three novel evaluation metrics (competence affinity, motivation, and social cohesion), we enriched an existing one (diversity), and we put forward *collegiality* to evaluate a team combining all four metrics mentioned above. *Competence affinity* evaluates how competent is a set of agents for the assigned tasks; *diversity* evaluates how diverse in personality and gender is a set of agents; *motivation* evaluates how motivated is a set of agents to work on the assigned tasks, and *social cohesion* evaluates how accepted is an agent by a set of agents and vice versa. Each of these metrics draws information from team members' individual profiles and aggregates this information to the team level.

Notably, with the above metrics, we can not only evaluate a single team but also compare different teams. In this chapter, we developed the tools that allow us to distinguish among teams and ultimately decide which team is better (compared to others). With this in mind, the following chapter introduces the team formation problem we tackle in this thesis, where solving the problem amounts to deciding which teams should be formed, i.e., which teams should be selected from many alternative solutions.

Chapter 4

The Non-Overlapping Many Teams To Many Tasks Allocation Problem

Many real-world problems require allocating teams of individuals to tasks. For instance, forming teams of robots for search and rescue missions [Capezzuto et al., 2020], forming teams of Unmanned Aerial Vehicles (UAVs) for surveillance [Ponda et al., 2015], building teams of people to perform projects in a company [Ballesteros-Perez et al., 2012, da Silva and Krohling, 2018], or grouping students to undertake school projects [Andrejczuk et al., 2019]. In this chapter, we discuss the problem of forming multiple human teams, i.e., the problem of assigning different tasks to different sets of agents. Specifically, we focus on problems where no overlaps are permitted; that is, each agent can be part of at most one team, and each task can be assigned to at most one set of agents. In other words, two teams share no common agents, and each team is allocated to work on a different task.

As we have discussed in Chapters 1 and 2, the multi-agent systems (MAS) literature has tackled the problem of allocating teams to tasks in several ways. The existing literature includes research on how to form a single team and allocate it to a single task [Anagnostopoulos et al., 2010, Lappas et al., 2009, Anagnostopoulos et al., 2012]; how to form a single team and match it with multiple tasks [Crawford et al., 2016]; and how to form multiple teams to solve the very same task [Andrejczuk et al., 2019]. There is a handful of research works on forming multiple teams to match with multiple tasks, either by allowing agent overlaps (agents participate in multiple teams [Capezzuto et al., 2020]), and/or task overlaps (different teams jointly solve a task [Bachrach et al., 2010]). However, the problem of distributing agents in non-overlapping teams, each to solve a different task, has deserved little attention, with the exception of [Czatnecki and Dutta, 2019, Präntare and Heintz, 2018, Czarnecki and Dutta, 2021]. This non-overlapping many teams to many tasks (NOMTMT) allocation problem is the one we address in this thesis.

As such, in this chapter, we make headway in the problem of allocating many non-overlapping human teams to many tasks through the following contributions. First, we provide the formalisation of the non-overlapping many teams to many tasks (NOMTMT) allocation problem, cast it as an optimisation problem, and discuss the complexity of the problem. Then we obtain two solutions of the problem:

1. an optimal solution, based on casting the problem as an integer linear program (ILP) that we can solve with off-the-self optimisation libraries, and

2. an approximate solution, obtained with a novel anytime hueristic algorithm.

4.1 Formalising the Non-Overlapping Many Teams To Many Tasks Allocation Problem

The non-overlapping many teams to many tasks allocation problem amounts to picking a size-compliant set of agents per task so that each agent belongs to at most one team. In the previous chapter, we discussed how to evaluate the matching of a group of agents with a task. To begin with, assume that we only have a single task at hand. Given a set of agents $\mathcal{A}$, the *best* team for this task corresponds to the size-compliant subset of agents with the maximum collegial value (Definition 21). That is, given a single task $\tau = \langle C_\tau, l_\tau, w_\tau, s_\tau \rangle$ (with $T = \{\tau\}$ and $|T| = 1$), the best team to resolve τ corresponds to the subset of agents $K^* \subseteq \mathcal{A}$ with $|K^*| = s_\tau$ and collegiality(K^*, τ) is maximum, i.e.,

$$K^* = \arg\max_{K \in \mathcal{K}_\tau} \text{collegiality}(K, \tau)$$

where $\mathcal{K}_\tau = \{K \subseteq \mathcal{A} : |K| = s_\tau\}$ is the set of all feasible candidate teams that can be matched with task τ.

For a collection of tasks T, with $|T| > 1$, we must maximise the collegiality of all candidate teams with the corresponding tasks, given that

- each agent can participate in at most one team,
- each team works on exacrtly one task (as in Definition 10), and
- each task can be assigned to at most one team.

First, we need to formally define what is a Feasible Team Allocation Function (FTAF), and then proceed on finding the optimum one, i.e., the one that maximises collegiality.

Definition 22 (*Feasible Team Allocation Function (FTAF)).* Given a set of tasks T, and a set of agents A, a feasible team allocation function g is a function $g : T \to 2^A$ such that:

1. every task $\tau \in T$ is allocated its requested number of agents so that $|g(\tau)| = s_\tau$; and
2. an agent can only be assigned to one team: for every pair of tasks $\tau, \tau' \in T$, such that $\tau \neq \tau'$, it holds that $g(\tau) \cap g(\tau') = \emptyset$.

The family of all feasible team allocation functions is denoted by G.

At this point, we want to highlight that the problem we address here is a non-trivial generalisation of the problem tackled in [Andrejczuk, 2018, Andrejczuk et al., 2019]. Unlike us, in these previous works, the authors consider a single task and cope with the problem of forming non-overlapping teams that all tackle (independently) the very same task. As we have mentioned earlier, in this thesis we illustrate the problem in the educational domain. For example, we consider students teams carrying out homework or semester project and students being assigned to internship programs as a team. Such team formation scenarios have in common that we seek for *balanced allocations*. [Andrejczuk, 2018] points out that in such scenarios, we cannot afford to have a single highly evaluated team while the remaining teams are poorly evaluated. Instead, all teams shall be equally good, i.e., all teams are more or less at the same level of collegiality. Beyond the educational domain, other team formation scenarios such as hackathons, volunteering missions, or crowdsoucing and social impact events seek for balanced allocations, as well.

Therefore, to achieve balanced allocations, the optimum team allocation function g^* should maximise the *a-la-Nash* product of collegiality of all teams since the product promotes fairness and homogenity [Nash, 1950]. Now we formally define the non-overlapping many teams to many tasks allocation problem as:

Definition 23 (*Non-Overlapping Many Teams to Many Tasks (NOMTMT) Allocation Problem).* Given a set of tasks T, and a set of agents A, the *Non-Overlapping Many Teams to Many Tasks Allocation Problem* is to find a team allocation function $g^* \in G$ that maximises the overall team collegiality:

$$g^* = \arg\max_{g \in G} \prod_{\tau \in T} \text{collegiality}\left(g(\tau), \tau\right) \tag{4.1}$$

We remind the reader that in order to compute the collegiality of every team in g, we need to compute the competence affinity, congeniality, motivation and social cohesion of each team, as discussed in section 3.4.5. Notably, competence affinity requires computing the optimum competence assignment as discussed in section 3.4.1. Thus, for each team allocation g, we need to solve $|T|$ optimisation problems (one per task) in order to determine $|T|$ competence assignment functions each of which is the best one for a pair $\langle \tau, g(\tau) \rangle$. The NOMTMT allocation problem is interrelated with the $|T|$ optimisation problems. However, for a fixed team allocation, the inner optimisation problems are independent of one another. In other words, although picking the optimum team allocation g^* depends on the collegial value of all teams and computing the collegial value of each team amounts to computing the optimum competence assignment, as soon as an allocation g is in place, computing the optimum competence assignment for two different teams $g(\tau)$ and $g(\tau')$ requires solving two independent optimisation problems.

4.2 Complexity Analysis

This section aims to discuss the complexity and hardness results of the non-overlapping many teams to many tasks allocation problem. Previously, we introduced the NOMTMT allocation problem as an optimisation one. In this problem, we need to find the best allocation function g that maps each task $\tau \in \mathcal{T}$ with a size-compliant set of agents. In contrast, no agent participates in more than one team, and each team is assigned a different task.

We begin by examining the vastness of the search space of the problem. This amounts to quantifying the number of feasible team allocation functions in G. For that, we start by splitting the tasks in $\mathcal{T}$ into $r \in \mathbb{N}_+$ buckets of tasks, where the tasks in the same bucket share a common requirement in team size. That is let $b_1, \cdots, b_r \subseteq \mathcal{T}$ denote the buckets, where $b_i \cap b_j = \emptyset,\ \forall i,j = 1, \cdots, r$ and $\bigcup_{i=1}^{r} b_i = \mathcal{T}$. Now, for each bucket b_i with $|b_i| = n_i$ (i.e., bucket b_i contains n_i different tasks), it holds that $s_{\tau_1} = s_{\tau_2} = \cdots = s_{\tau_{n_i}} = s_i$ for all $\tau_1, \tau_2, \cdots, \tau_{n_i} \in b_i$. Moreover, for any two different buckets b_i and b_j it holds that $s_i \neq s_j$, where s_i and s_j characterise b_i and b_j respectively. For example, assume that we have 10 tasks $\mathcal{T} = \{\tau_1, \tau_2 \cdot, \tau_{10}\}$, where τ_1, τ_2, τ_3 and τ_4 require a team of size 3 (i.e., $s_{\tau_1} = s_{\tau_2} = s_{\tau_3} = s_{\tau_4} = 3$), τ_5, τ_6 and τ_7 require a team of size 4 (i.e., $s_{\tau_5} = s_{\tau_6} = s_{\tau_7} = 4$), and τ_8, τ_9 and τ_{10} require a team of size 5(i.e., $s_{\tau_8} = s_{\tau_9} = s_{\tau_{10}} = 5$). In this case, we would have three buckets b_1, b_2 and b_3, where $b_1 = \{\tau_1, \tau_2, \tau_3, \tau_4\}$ with $n_1 = 4$ and $s_1 = 3$, $b_2 = \{\tau_5, \tau_6, \tau_7\}$ with $n_2 = 3$ and $s_2 = 3$, and $b_3 = \{\tau_8, \tau_9, \tau_{10}\}$ with $n_3 = 3$ and $s_3 = 5$. Next, we will distinguish three cases when counting the number of feasible teams in G. We assume that there are n agents in $\mathcal{A}$, i.e., $|\mathcal{A}| = n$.

Case I: $\sum_{\tau \in \mathcal{T}} s_\tau = \sum_{i=1}^{r} s_i \cdot |b_i| = n$, we have exactly as many agents as required by all tasks in $\mathcal{T}$. In this case, we seek for partition functions over $\mathcal{T}$. According to Theorem 3.4.19 in [Maddox, 2002], the space of G is

$$\frac{n!}{\prod_{i=1}^{r}(s_i!)^{b_i}}$$

Case II: $\sum_{\tau \in \mathcal{T}} s_\tau = \sum_{i=1}^{r} s_i \cdot |b_i| < n$, we have more agents than the required ones by all tasks in $\mathcal{T}$. Following the Example 3.4.20 in [Maddox, 2002], we assume one more bucket b_{r+1}

containing exactly one *auxiliary* program, which requires a team of size $s_{r+1} = \sum_{i=1}^{r} s_i \cdot |b_i| - n$. Now the number of different feasible team allocation functions is:

$$|G| = \frac{n!}{\prod_{i=1}^{r}(s_i!)^{b_i} \cdot \left(n - \sum_{i=1}^{r} |b_i| \cdot s_i\right)!}$$

Case III: $\sum_{\tau \in \mathcal{T}} s_\tau = \sum_{i=1}^{r} s_i \cdot |b_i| > n$, we have less agents than the required ones by all tasks in $\mathcal{T}$. In this case, first we need to introduce

$$cover(\mathcal{T}, \mathcal{A}) = \left\{ \mathcal{T}' \subset \mathcal{T} : \sum_{\tau \in \mathcal{T}'} s_\tau \leq n \wedge \nexists \tau' \in \mathcal{T} - \mathcal{T}' : s_{\tau'} \leq n - \sum_{\tau \in \mathcal{T}'} s_\tau \right\}$$

as the set that contains all the subsets of tasks $\mathcal{T}' \subset \mathcal{T}$ such that pair $\langle \mathcal{A}, \mathcal{T}' \rangle$ leads to Case I or Case II, and by adding any $\tau \notin \mathcal{T}'$ in τ' it will lead to Case III. The number of feasible team assignment functions is:

$$|G| = \sum_{\mathcal{T}' \in cover(\mathcal{T}, \mathcal{A})} \frac{n!}{\prod_{i=1}^{r}(s_i!)^{b_i} \cdot \left(n - \sum_{i=1}^{r} |b_i| \cdot s_i\right)!}$$

where variables $r, b_1, \cdots, b_r$ and $s_1, \cdots, s_r$ changes according to $\mathcal{T}'$. The number of subsets $\mathcal{T}'$ in $cover(\mathcal{T}, \mathcal{A})$ depends on the total number of agents, and the team sizes required by the tasks in T.

Note that the number of feasible team allocation functions quickly grows with the number of tasks and agents, hence leading to vast search spaces. Next, we show that the NOMTMT allocation problem is $\mathcal{NP}$-complete by reduction to a well-known problem.

Theorem 1. *The Non-Overlapping Many Teams to Many Tasks (NOMTMT) allocation problem for more than one task is $\mathcal{NP}$-complete.*

Proof. We can decide whether a given solution is feasible in polynomial time ($O(\sum_{\tau \in T} s_\tau)$). We now show that the problem is $\mathcal{NP}$-complete by using a reduction from *Single Unit Auctions with XOR Constraints and Free Disposals* (referred to as BCAWDP with XOR Constraints) which is shown to be $\mathcal{NP}$-complete [Sandholm et al., 2002]. In the BCAWDP with XOR Constraints, the auctioneer has N items to sell, the bidders place their bids $B_i = \langle \mathbf{b}_i, b_i \rangle$ with $\mathbf{b}_i$ a subset of items and b_i the price. Between two bids there can exist an XOR constraint–not necessarily to every pair of bids. The auctioneer allows free disposals, i.e., items can remain unsold.

Given an instance of BCAWDP with XOR Constraints, we construct an instance of the NOMTMT allocation problem as follows: "For each item i we create an agent a_i. For each

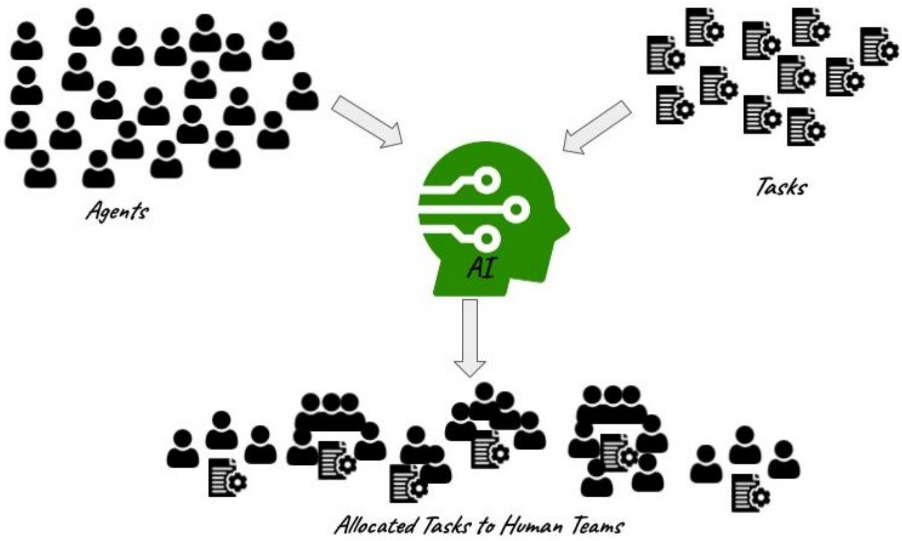


Figure 4.1: Task allocation for human teams.

task τ_j of size s_{τ_j} we create $\binom{n}{s_{\tau_j}}$ different bids $B_{jk} = \langle \mathbf{b}_{jk}, b_{jk} \rangle$, where n is the number of items, $|\mathbf{b}_{jk}| = s_{\tau_i}$, and $b_{jk} = \text{collegial}\mathbf{b}_{jk}, \tau_j$. All bids created for task τ_j are XOR-constrained bids. Moreover, each pair of bids $B_{j,k}, B_{q,l}$ such that $\mathbf{b}_{jk} \cap \mathbf{b}_{ql} \neq \emptyset$ are also XOR-constrained." Now, the non-overlapping many teams to many tasks allocation problem has a feasible solution if and only if BCAWDP with XOR constraints has a solution. □

4.3 Solving the NOMTMT Allocation Problem

The non-overlapping allocation problem is an $\mathcal{NP}$-complete problem with a vast search space, as discussed in the above section. As such, solving the problem by hand can be very hard and time-consuming, especially as the number of agents, tasks, and required team sizes increase. In this section, we put forward two solvers for the NOMTMT allocation problem:

1. an optimal solver employing integer linear programing (ILP), and
2. an anytime heuristic solver, especially devised for the problem.

4.3.1 Optimal solver

We start with the optimal solver. We can optimally solve the NOMTMT allocation problem by encoding the problem as an integer linear program. Given a set of agents $\mathcal{A}$ and a set of tasks

$\mathcal{T}$, for each task $\tau \in \mathcal{T}$ we denote with $\mathcal{K}_\tau = \{K \subseteq \mathcal{A} : |K| = s_\tau\}$ all the size-compliant teams for task τ, i.e., all the subset of agents that meet task's τ team size requirement. For each task τ and each size-compliant subset of agents $K \in \mathcal{K}_\tau$, we use a binary decision variable x_τ^K. We interpret a decision variable x_τ^K being equal to 1 as "agents in K form a team for task τ"; while a decision variable x_τ^K being equal to 0 corresponds to "agents in K *do not* form a team for task τ". Moreover, let function membership : $2^{\mathcal{A}} \times \mathcal{A} \to \{0, 1\}$ indicate whether an agent is a member in a subset of agents, i.e.,

$$\text{membership}(K, a) = \begin{cases} 1 & \text{if } a \in K, \\ 0 & \text{otherwise} \end{cases}.$$

Then, in order to solve a NOMTMT allocation problem, we need to solve the following non-linear program:

$$\max \prod_{\tau \in T} \prod_{K \in \mathcal{K}_\tau} \Big(\text{collegiality}(K, \tau)\Big)^{x_K^\tau} \tag{4.2}$$

subject to:

$$\sum_{K \subseteq \mathcal{K}_\tau} x_K^\tau \leq 1 \qquad \forall \tau \in T \tag{4.2a}$$

$$\sum_{\tau \in T} \sum_{K \subseteq \mathcal{K}_\tau} x_K^\tau \cdot \text{membership}(K, a) \leq 1 \qquad \forall a \in A \tag{4.2b}$$

$$x_K^\tau \in \{0, 1\} \qquad \forall\, \tau \in T, K \subseteq \mathcal{K}_\tau \tag{4.2c}$$

Constraint (4.2a) guarantees that at most one team is formed for each task. Constraint (4.2b) ensures that each agent will be assigned to at most one task. The combination of constraints (4.2a) and (4.2b) guarantees that each agent participates in at most one team, ensuring that we permit no overlaps.

Linear Transformation

Notice that the encoding above, precisely the objective function in Equation (4.2) is *non-linear*. Nevertheless, we can easily obtain an equivalent linear function. First, we change the function's domain. In order to use the logarithm of $\prod_{\tau \in T} \prod_{K \in \mathcal{K}_\tau} \Big(\text{collegiality}(K, \tau)\Big)^{x_K^\tau}$ we need to ensure that it is non-negative and non-zero. According to Equation (3.20), collegiality equals the summation of four non-negative values:

- the competence affinity of any team for its assigned task ranges in $[0, 1]$;
- the congeniality of any set of agents ranges in $[0, 1 + 3 \cdot c + d + e]$ (where $c, d, e \geq 0$, see Section 3.4.2);
- the motivation of any team for its assigned task ranges in $[0, 1]$;
- the social cohesion of any set of agents ranges in $[0, 1]$; and
- the regulating parameters are positive numbers, i.e., $\alpha, \beta, \gamma, \delta \in \mathbb{R}_+$.

Therefore, $\prod_{\tau \in T} \prod_{K \in \mathcal{K}_\tau} \left(\text{collegiality}(K, \tau)\right)^{x_K^\tau}$ is always non-negative since the collegiality of any team is always non-negative. However, we cannot guarantee that it is always non-zero. As such, without harming the monotonicity of the function, we change the objective function's domain by considering the following "surrogate" objective function:

$$\prod_{\tau \in T} \prod_{K \in \mathcal{K}_\tau} \left(1 + \text{collegiality}(K, \tau)\right)^{x_K^\tau}.$$

Now, we can use the logarithm to linearise the objective function, and thus, solving the non-linear program above is equivalent to solving the following binary linear program:

$$\max \sum_{\tau \in T} \sum_{K \in \mathcal{K}_\tau} x_K^\tau \cdot \log\left(1 + \text{collegiality}(K, \tau)\right) \tag{4.3}$$

subject to: equations (4.2a), (4.2b), and (4.2c), which are linear constraints.

We can solve this LP with the aid of an off-the-shelf solver (e.g. CPLEX [IBM, 2019], Gurobi [GUROBI, 2018], GLPK [GLPK, 2018], or SCIP [Gamrath et al., 2020]). Given sufficient time, an LP solver will return an optimal solution to the NOMTMT allocation problem.

Note that building such an LP requires to pre-compute the values of competence affinity, congeniality, motivation and social cohesion, with competence affinity being the computationally most challenging value to compute since it requires solving an optimisation problem for each candidate team for a task (see Section 3.4.1). This is bound to lead to large linear programs as the number of agents and tasks grows. Such large LPs require significant time to perform the necessary pre-computations before even solving the problem. Later in Chapter 5 we will empirically show that, indeed, this hinders the scalability of the optimal solver. With this in mind, in the following section, we introduce a novel anytime heuristic solver for the non-overlapping many teams to many tasks allocation problem.

4.3.2 Edu2Com: An anytime heuristic solver

Our proposed algorithm, called Edu2Com, consists of two stages in a similar manner as in [Andrejczuk et al., 2019]—as we already said, the problem we address in this work is a generalisation of the problem in the work of [Andrejczuk et al., 2019]. The first stage finds an initial feasible team allocation function. The second stage iteratively improves the allocation function by swapping agents between pairs of teams using different strategies. Edu2Com is presented in Algorithm 1.

Algorithm 1: Edu2Com

input : Agents $\mathcal{A}$, tasks $\mathcal{T}$

g_{init} = Initial Team Allocation($\mathcal{A}, \mathcal{T}$);

g = Improving Team Allocation($g_{init}, \mathcal{A}, \mathcal{T}$);

return g;

Building an initial team allocation

The algorithm finds an initial, feasible, and *promising* team allocation function; that is, the initial team allocation function involves non-overlapping teams, where tasks are matched with size-compliant teams that are highly evaluated (considering the evaluation metrics discussed in Section 3.4). It sequentially forms a team for each task, starting from the 'hardest' task to the 'lightest' one. We consider a task 'hard' if just a few agents can cover its competencies. Picking team members for the more challenging tasks first is a heuristic that allows us to avoid allocating the dew agents that can cover the most challenging tasks to 'lighter' tasks' teams.

Computing the allocation hardness of tasks. The allocation hardness (or simply 'hardness') of a task assesses the difficulty of finding agents who can adequately cover the task's required competencies. Intuitively, when, for some competence c, more agents adequately cover c (i.e., with high coverage on c), it is easier to find an agent for some task requiring c, and therefore the task is less hard. Inspired by the notion of the *moment of inertia* [Morrison and De Jong, 2002], we measure the difficulty of covering a competence, and therefore, the task's hardness requiring that competence, as the effort that the agents should make to reach the ideal competence coverage of c. We remind the reader that the coverage of a competence c ranges from 0 to 1. Thus, the ideal competence coverage for a competence occurs if every agent can fully cover the competence (i.e. competence coverage equals 1 for all agents). We compute the moment of inertia for some competence c as:

$$I(c) = \sum_{J \in \mathcal{I}} n_J^c \cdot \big(1 - mid(J)\big)^2 \tag{4.4}$$

where:

(i) $\mathcal{I} = \{[0, 0.1), [0.1, 0.2), [0.2, 0.3), [0.3, 0.4), [0.4, 0.5), [0.5, 0.6), [0.6, 0.7), [0.7, 0.8), [0.8, 0.9), [0.9, 1]\}$ is an interval partition of the domain of competence coverage $[0, 1]$;

(ii) $n_J^c = |\{a \in \mathcal{A} | \text{cvg}(c, a) \in J\}|$ is the number of agents in $\mathcal{A}$ whose coverage of competence c lies within interval J, and hence represents the *mass* of c in the interval; and

(iii) $mid(J)$ corresponds to the midpoint of interval J.

Now, we compute the hardness of each task from the hardness of each one of its required competencies as well as their relative importance weights. Specifically, the hardness of a task is proportional to the moment of inertia and the importance weight of each required competence. Thus, given task τ, we define its hardness as:

$$h(\tau) = \omega \cdot \sum_{c \in C_\tau} w_\tau(c) \cdot I(c) \tag{4.5}$$

where $\omega = \frac{1}{\sum_{c \in C_\tau} w_\tau(c)}$, is a normalising factor over the importance weights.

Building an initial team allocation. To build an initial team allocation function, our algorithm first sorts tasks according to their hardness and then proceeds by sequentially forming and allocating a team for each task, starting from the hardest one. Let $A_\tau \subseteq \mathcal{A}$ be the set of available agents to allocate to τ, i.e., the set of agents that have not been allocated to some task proceeding τ. First, the algorithm sorts the task's C_τ, based on their relative importance, into a sequence $\bar{C}_\tau$. We note as $\bar{C}_\tau^i$ the i-th competence in $\bar{C}_\tau$. The first agent to be allocated to τ's team is the agent in A_τ that can cover best competence $\bar{C}_\tau^1$—formally, we compute the first agent to pick for the team as:

$$\sigma_1 = \arg\max_{a \in A_\tau} \{\text{cvg}(\bar{C}_\tau^1, a)\}$$

After picking the first agent σ_1, the remaining available agents are $A_\tau - \{\sigma_1\}$. The following agent to join τ's team, σ_2, is the agent that covers best competence $\bar{C}_\tau^2$, and so on. As such, the i-th agent to be picked for the team of task τ is computed as:

$$\sigma_i = \arg\max_{a \in A_\tau - \Sigma_{i-1}} \{\text{cvg}(\bar{C}_\tau^j, a)\}$$

where $\Sigma_{i-1} = \bigcup_{k=1}^{i-1}\{\sigma(\bar{C}_\tau^k)\}$ stands for the agents allocated so far up to i-th agent, and $j = \left(i - 1 \pmod{|C_\tau|}\right) + 1$ indicates which competence agent σ_i shall cover best. Depending on the required team size and the number of required competencies, we discern two cases:

- $s_\tau < |C_\tau|$: the agents in the initial team are picked so that they cover best the s_τ most important required competencies; and
- $s_\tau \geq |C_\tau|$: the $|C_\tau|$ agents in the initial team are picked so that they cover best *all* required competencies. Then, agents are picked to reinforce the already covered competencies (prioritising the most important ones) until the required team size is reached.

After following the procedure above for some task τ, our algorithm allocates to τ the team consisting of:

$$K = \bigcup_{i=1}^{s_\tau} \sigma(\bar{C}_\tau^i)$$

The agents in K are no longer available for being chosen to participate in another team.

The initial feasible team allocation function can be built by following the algorithm described above and presented in Algorithm 2.

Improving team allocation

The second stage of our algorithm applies several heuristics implemented as *agent swaps*. This stage is similar to the approach proposed in [Andrejczuk et al., 2019], with the addition of an exploring step. The heuristics are applied until either:

1. no solution improvement occurs for a number of iterations; or
2. the algorithm is stopped by the user.

In all cases, the most recently found solution is returned. This stage performs two types of iterations:

1. **Single pairing.** We randomly select two tasks, and we apply over them the following swaps:
   a) **Exploiting swap.** Find the optimal team allocation just considering the agents in the teams currently allocated to both tasks. (Figure 4.2a)
   b) **Exploring swap.** Try a maximum of k times the following: (i) randomly select one of the two tasks, one agent within that task and an unassigned agent (if any);

Algorithm 2: Edu2Com: Initial Team Allocation

input : Agents $\mathcal{A}$, tasks $\mathcal{T}$

$C_{\text{all}} \leftarrow \bigcup_{\tau \in \mathcal{T}} C_\tau$;

for (*competence* $c \in C_{\text{all}}$ **):**

  `/* Compute the moment of inertia for every competence */`

  for (*interval* $J \in \mathcal{I}$ **):**

    $n_J^c \leftarrow$ count agents in $\mathcal{A}$ for which $\text{cvg}(a, c) \in J$;

  $I(c) \leftarrow \sum_{J \in \mathcal{I}} n_J^c \cdot \big(1 - mid(J)\big)^2$

for (*task* $\tau \in \mathcal{T}$ **):**

  `/* Compute the hardness for every tasks */`

  $\omega \leftarrow \frac{1}{\sum_{c \in C_\tau} w_\tau(c)}$;

  $h(c) \leftarrow \omega \cdot \sum_{c \in C_\tau} w_\tau(c) \cdot I(c)$;

$\mathcal{T}' \leftarrow$ Sort $\mathcal{T}$ in descending order according to h;

$\mathcal{A}_\tau \leftarrow$ Copy all agents in $\mathcal{A}$ to be considered as available agents;

for (*task* $\tau \in \mathcal{T}'$ **):**

  `/* Form a promising team for every task */`

  if($|\mathcal{A}'| < s_\tau$ **):** proceed to the next task ;

  $\bar{C}_\tau \leftarrow$ Sort C_τ in descending order according to w_τ;

  $K \leftarrow \emptyset$; `// Initialise an empty set of agents for task` τ

  agent_idx $\leftarrow 1$;

  while ($|K| < s_\tau$ **):**

    c_idx $\leftarrow \big((\text{agent\_idx} - 1) \,\%\, |C_\tau|\big) + 1$;

    $\sigma(\bar{C}_\tau^{\text{c\_idx}}) \leftarrow \arg\max_{a \in A_\tau} \{\text{cvg}(\bar{C}_\tau^{\text{c\_idx}}, a)\}$;

    Add $\sigma(\bar{C}_\tau^{\text{c\_idx}})$ to K; `// Add chosen agent in the set of agents.`

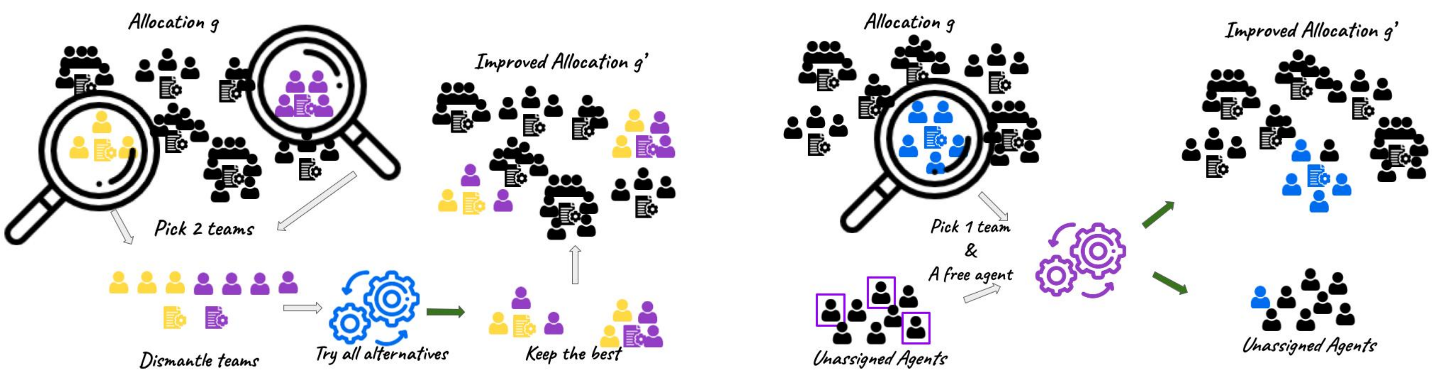
    Remove $\sigma(\bar{C}_\tau^{\text{c\_idx}})$ from A_τ; `// Remove chosen agent from the set with the available agents.`

    agent_idx $\leftarrow$ agent_idx $+ 1$;

  `/* Allocate set of agents` K `as a team to task` τ`. */`

  $g(\tau) \leftarrow K$;

return team allocation function g



(a) Exploiting swaps

(b) Exploiting swaps

Figure 4.2: Single pairing

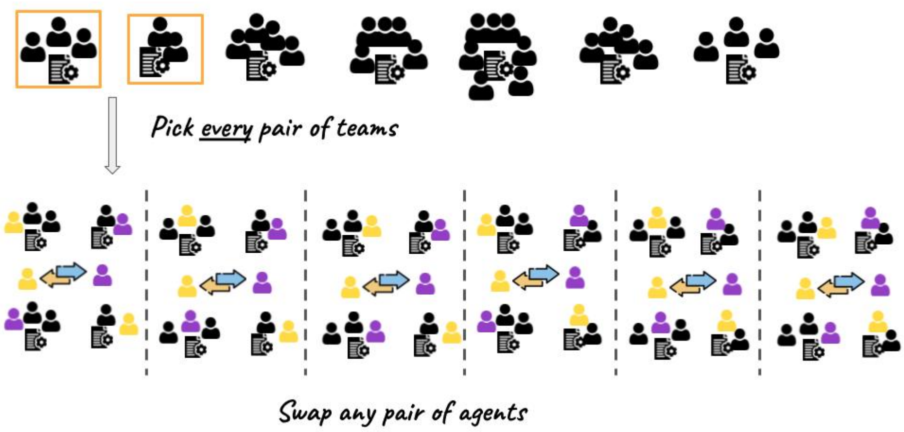


Figure 4.3: Exhaustive pairing

(ii) swap them; (iii) if the collegiality is improved, keep the change and stop the exploring swaps. (Figure 4.2b)

2. **Exhaustive pairing.** For *every* pair of tasks, swap every possible pair of agents within them. If the collegiality is improved, keep the change and stop the exhaustive pairing.(Figure 4.3)

An exhaustive pairing iteration is performed after a number of single pairing iterations. The second stage of Edu2Com is presented in Algorithm 3.

Algorithm 3: Edu2Com: Improving Team Allocation

input : Initial Allocation g, Agents $\mathcal{A}$, tasks $\mathcal{T}$

$A_{\text{available}} \leftarrow \mathcal{A} - \bigcup_{\tau \in \mathcal{T}} g(\tau)$; `// Available agents`

$iter_{\text{last\_impr}} \leftarrow 0$; `// Iteration of last improvement`

$iter_{\text{current}} \leftarrow 0$; `// Current iteration`

while (*$iter_{current} - iter_{last\_impr} <$ max iterations w/o improvement* **):**

    $iter_{\text{current}} \leftarrow iter_{\text{current}} + 1$;

    if(*interrupted by user* **):**

        return g;

    `/* Single Pairing Iteration: Exploiting swaps */`

    $\tau_1, \tau_2 \leftarrow$ Pick randomly two tasks from $\mathcal{T}$;

    $A_{\tau_1,\tau_2} \leftarrow g(\tau_1) \cup g(\tau_2)$;

    $K_1^*, K_2^* \leftarrow \arg\max_{K_1,K_2 \subseteq A_{\tau_1,\tau_2}} \big(\text{collegiality}(K_1, \tau_1) \cdot \text{collegiality}(K_2, \tau_2)\big)$;

    $g \leftarrow$ Update g such that $g(\tau_1) = K_1^*$ and $g(\tau_2) = K_2^*$;

    if(*overall collegiality is improved* **):**

        $iter_{\text{last\_impr}} \leftarrow iter_{\text{current}}$;

    `/* Single Pairing Iteration: Exploring swaps */`

    if(*there are available agents in $A_{available}$* **):**

        while (*$k \leq$ max number of exploring attempts* **):**

            $a_1 \leftarrow$ Randomly pick an agent from $A_{\text{available}}$;

            $a_2 \leftarrow$ Randomly pick an agent from A_{τ_1,τ_2};

            $K_1, K_2, A'_{\text{available}} \leftarrow$ Swap a_1 with a_2;

            if(*overall collegiality is improved* **):**

                $g \leftarrow$ Update g such that $g(\tau_1) = K_1$ and $g(\tau_2) = K_2$;

                $A_{\text{available}} \leftarrow$ Update available agents with $A'_{\text{available}}$;

                $iter_{\text{last\_impr}} \leftarrow iter_{\text{current}}$;

                Exit inner while loop;

            $k \leftarrow k + 1$;

    if(*interrupted by user* **):**

        return g;

    `/* Exhaustive Pairing Iteration */`

    if(*$iter_{current}$ % $iter_{exhaustive\ pairing}$ == 0* **):**

        for ($\tau_1 \in \mathcal{T}$ **):**

            for ($\tau_2 \in \mathcal{T}$ **):**

                for ($a_1 \in g(\tau_1)$ **):**

                    for ($a_2 \in g(\tau_2)$ **):**

                        $K_1, K_2 \leftarrow$ Swap a_1 with a_2;

                        if(*overall collegiality is improved* **):**

                            $g \leftarrow$ Update g such that $g(\tau_1) = K_1$ and $g(\tau_2) = K_2$;

                            $iter_{\text{last\_impr}} \leftarrow iter_{\text{current}}$;

                            Terminate exhaustive pairing iteration;

return g;

Complexity Analysis of the Edu2Com Algorithm

Note that our proposed heuristic algorithm is computationally efficient. On the one hand, Edu2Com's first stage is polynomial on the number of agents $|\mathcal{A}| = n$ and the number of tasks $|\mathcal{T}| = m$. Going through Algorithm 2, we see that most computationally hard process is to pick the agents to form a team for each task (lines 11-18). Specifically, the complexity for forming the initial teams for all tasks is $O(m \cdot n \cdot s^{\max})$, where $s^{\max} = \max_{\tau \in \mathcal{T}} s_\tau$ corresponds to the largest required team size. Notably, computing the hardness of all tasks (lines 6-8) costs $O(m)$, while computing the moment of inertia $I(c)$ for each competence is of complexity $O\big(|\mathcal{C}_{\text{all}}| \cdot m\big)$. At this point we assume that the total number of competencies involved in the problem instance is smaller than the total number of agents, i.e., $|\mathcal{C}_{\text{all}}| < n$. Thus, building an initial feasible solution with Edu2Com is polynomial on the number of agents and the number of tasks, and specifically $O(n \cdot m)$.

On the other hand, Edu2Com's second stage is an iterative process which terminates at any time at the user's request. The complexity of the improving stage depends on the complexity of the two types of iterations, namely the single pairing and exhaustive pairing. To be more precise, the complexity of improving the team allocation is polynomial on the number of agents $|\mathcal{A}| = n$ and factorial on the size of the two largest required team sizes.[1] That is, to perform an *exploiting swap* we consider all possible allocations involving just the agents currently allocated in the two tasks considered by the exploiting swap iteration. As such, in the worst case scenario, the complexity of an exploiting swap is $O\big((2 \cdot s^{\max})!\big)$. The computational complexity of an exploring swap depends on the number k we try to swap an already assigned agent with an unassigned one. Hence, given that $(2 \cdot s^{\max})! \gg k$, the complexity of a single pairing iteration is $O\big((2 \cdot s^{\max})!\big)$. Finally, an exhaustive pairing may result in swapping every possible pair of agents, therefore the complexity is $O(n^2)$. Notably, the complexity of an exploiting swap drives the computational complexity of the algorithm's improving stage. However, in practise the required team sizes are not large—[Andrejczuk, 2018] points out that teams within classrooms should have at most five members. As such, the factorial complexity of the exploiting swaps is not prohibitive in practice. Thus, our heuristic algorithm is computationally efficient, as we will empirically show in Chapter 5.

[1] Note that if there are more than one task requiring the largest team size $s^{\max}$, then the second largest required team size is $s^{\text{second max}} \equiv s^{\max}$.

4.4 Summary

This chapter introduces the problem we address in this work: the non-overlapping many teams to many tasks (NOMTMT) allocation problem. The NOMTMT allocation problem corresponds to the problem of, given a set of tasks and a set of agents, forming a single team of agents for each one of the tasks such that no agent can participate in more than one team and no agent can work on more than one task, i.e., the same set of agents compose a team for exactly one task. We formally defined the problem and cast it as an optimisation problem that intends to maximise the collegiality of all formed teams in a *balanced* way. Then, we discussed the complexity of the problem. Specifically, we first showed the vastness of the search space and provided the means to quantify the number of different solutions to the problem; then, we proved that the NOMTMT allocation problem is $\mathcal{NP}$-complete. Given the complexity of the problem, we put forward two solvers: an optimal solver and an anytime heuristic solver. More precisely, the optimal solver is an integer linear program encoding that can be solved with any off-the-shelf solver, given sufficient time. However, given the combinatorial nature of the problem, the optimal solver cannot handle large problem instances easily. Large ILPs may require an extreme amount of time to be solved and mainly to be built. That is, as we discussed in this chapter, our optimal solver considers a decision variable per each size-compliant team for each task. Therefore, the optimal solver needs to handle a large number of decision variables as the number of agents and tasks increases. In Chapter 5, we will empirically show that we cannot solve large real-world problems optimally. As such, we need an approximating anytime algorithm to solve the problem. Thus, finally, in this chapter, we devised and proposed Edu2Com, a novel anytime heuristic algorithm. Edu2Com consists of two stages: in the first stage, the algorithm builds a promising initial team allocation function, while in the second stage, the algorithm iterative improves the allocation until it is interrupted by the user or converges to a solution without improving for a large number of iterations. Moreover, we studied the computational complexity of Edu2Com algorithm, and we showed that it is computationally efficient. As we will see in the following chapter, where we will empirically evaluate the solving time of the two algorithms (optimal solver and Edu2Com), we can achieve significant time savings by using our anytime heuristic algorithm.

Chapter 5

Empirical Evaluation

In this chapter, we empirically evaluate the capability of our proposed team formation algorithm, Edu2Com, to cope with real-world problems. We begin by evaluating the behaviour of our heuristic algorithm when solving the NOMTMT allocation problem compared to optimal solvers. More precisely, first, we study the *quality* of the teams built with our algorithm, along with our algorithm's runtime and anytime behaviour. Given that the NOMTMT allocation problem is an optimisation problem, with the term quality, we refer to the objective achieved by the solution formed with Edu2Com compared to the objective of the optimal solution (i.e., the solution formed by an optimal solver). As such, Section 5.2 details the analysis of the solution quality, runtime performance and anytime capabilities of Edu2Com when pitched against CPLEX, a state-of-the-art linear programming solver, over synthetic data.

Next, we use real-world data to study the scalability of our algorithm and the limitations of optimal solvers. We investigate the behaviour of Edu2Com as the problem instances scale up (in terms of the number of agents, number of tasks, and required team sizes), and we show the frontier of problem instances that can be solved with an optimal solver. Sections 5.3.2 and 5.3.4 report on the scalability and the optimal solvers' limitations, respectively.

Afterwards, in Sections 5.3.3, 5.4.3, and 5.5.3, we employ Edu2Com to solve real-world instances of the NOMTMT allocation problem. In more detail, we solve several instances of the NOMTMT allocation problem found in the education domain, and we study the composition of the teams formed with Edu2Com and their performance. Specifically, experts in forming student teams (i.e., educational authorities with years of experience in forming teams of students for several activities) evaluate and compare the composition of teams formed with Edu2Com against teams formed by a human expert in forming student teams. Moreover, we study how the teams formed by Edu2Com perform in practice. We show that our algorithm builds student teams (who participate in school activities) that achieve higher marks compared to student teams formed following the teachers' current practices. Finally, we investigate the relationship between the team's collegiality and the team's performance.

5.1 Empirical Setup

The implementation of Edu2Com, along with all the necessary supporting code, was made in Python3.7. All the experiments ran on a PC with Intel Core i7 CPU, 8 cores, and 8Gb RAM. In our experiments below, we set our algorithm's parameters to: compute similarity with $\kappa = 0.35$, $\lambda = 0.75$; perform one exhaustive-pairing every 50 single-pairings; stop the algorithm after two blocks of single-pairings and exhaustive pairings have elapsed with no improvements.

5.2 Quality, Runtime and Anytime Analysis

In this section, we study the performance of Edu2Com, our proposed heuristic for the Non-Overlapping Many Team to Many Tasks allocation problem (NOMTMT-AP), across three dimensions: *(i)* solution quality, *(ii)* runtime behaviour, and *(iii)* anytime behaviour. Specifically, we pitch Edu2Com against CPLEX, a state-of-the-art optimal solver. Both algorithms were employed to solve the same instances of the NOMTMT-AP.

5.2.1 Synthetic Data Generation

To compare our heuristic algorithm against the optimal solver, we created three sets of problem instances with varying sizes (low, medium, and large). All synthetically generated problem instances can be solved with linear programming and CPLEX, and an optimal solution can be obtained within acceptable time limits.

Here we would like to point out that the application domains considered in this thesis do not require a real-time solution within tight limits. We mainly consider NOMTMT allocation problem instances within the educational domain, with non-restrictive time limits between gathering agents' profiles and reaching a team formation solution.

We generated three families of problem instances, considering settings involving 10, 15, and 20 tasks. Below we describe the process of generating the problem instances. Initially, we set the number of tasks m by selecting from the set $\{10, 15, 20\}$. Then, we generated the tasks. We remind the reader that a task τ is given by a tuple $\langle C_\tau, l_\tau, w_\tau, s_\tau \rangle$; thus for each task $\tau \in \mathcal{T} = \{\tau_1, \cdots, \tau_m\}$, we randomly selected its components. Specifically, we randomly selected:

1. the task's required team size from a uniform distribution in the range $[2, 3]$, i.e., $s_\tau \sim \mathcal{U}(2, 3)$;
2. the number of required competencies from a uniform distribution in the range $[2, 5]$, i.e. $|C_\tau| \sim \mathcal{U}(2, 5)$;

Data Family Size	Number of Tasks	Number of Problem Instances	Number of Agents		Number of Competencies	
			Average	St. Dev.	Average	St. Dev.
low	10	20	24.5	1.5035	3.41	1.0779
middle	15	20	37.7	2.0799	3.51	1.1090
large	20	20	50.55	2.7236	3.54	1.1461

Table 5.1: Synthetically generated data families.

3. the set of required competencies C_τ from the ESCO ontology, and in turn, for each competence in C_τ, we randomly selected:

    a) an importance weight from a normal distribution, i.e. $w_\tau(c) \sim \mathcal{N}(\mu, \sigma^2)$, with mean $\mu \sim \mathcal{U}(0, 1)$ and variance $\sigma^2 \sim \mathcal{U}(0.01, 0.1)$, and

    b) an expertise level $l_\tau(c) \sim \mathcal{N}(\mu, \sigma^2)$ with mean $\mu \sim \mathcal{U}(0, 1)$ and variance $\sigma^2 \sim \mathcal{U}(0.01, 0.1)$.[1]

Next, we generated the agents. This empirical evaluation considers agents' profiles containing only competence profiles. Moreover, we generated sufficient agents to tackle all the generated tasks here. To do so, for each task τ, we generated competent agents to tackle τ as follows. Given τ, we randomly generate s_τ competence profiles; we remind the reader that an agent's a competence profile is given with a tuple $\langle C_a, l_a \rangle$. For each agent generated for task τ, we randomly select $|C_\tau|$ competencies such that they are identical to some required competence in C_τ or a child node in the ESCO ontology of some required competence. That is, for an agent a generated for task τ and each required competence $c \in C_\tau$ we randomly select:

1. a competence c' from the set $\{c\} \cup \{c' \in$ESCO such that c' is a child node of $c\}$, and

2. an expertise level from a normal distribution $l_a(c) \sim \mathcal{N}(\mu, \sigma^2)$ with mean $\mu \sim \mathcal{U}(0, 1)$ and variance $\sigma^2 \sim \mathcal{U}(0.01, 0.1)$.

In total, we generated 60 problem instances of varying sizes. Specifically, we considered three data families containing 20 instances each. Each problem instance involves *(i)* 10 tasks and on average ~ 24.5 agents in the low-size family, *(ii)* 15 tasks and on average ~ 37.7 agents in the middle-size family, and *(iii)* 20 tasks and on average ~ 50.55 agents in the large-size family. Table 5.1 summarises the characteristics of each data family.

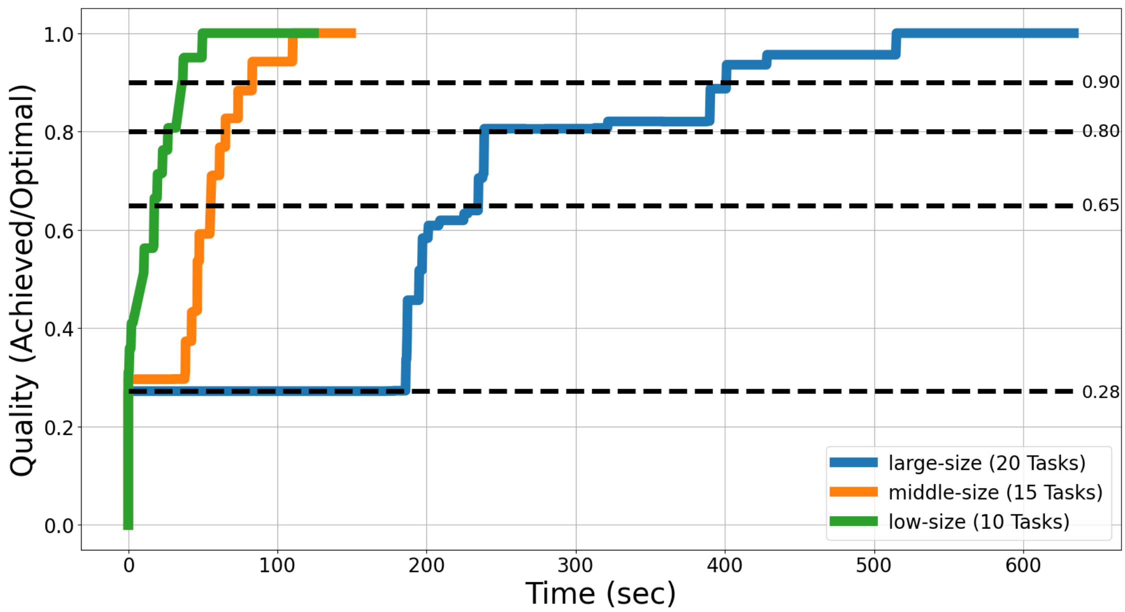


Figure 5.1: Competence quality of our algorithm along time.

5.2.2 Quality Analysis

Our first analysis regards the quality of the solutions computed with our proposed algorithm, Edu2Com. We define *quality of a solution* as the ratio between the competence affinity of the solution and the competence affinity of the optimal solution.

We tasked CPLEX to solve each problem instance optimally. The competence affinity of the optimal solution corresponds to the highest competence affinity that can be achieved. Then we task Edu2Com to compute a solution for each problem instance. The competence affinity achieved by Edu2Com is compared against the optimal competence affinity comprising the quality of the heuristic solution. Figure 5.1 illustrates the quality of the solutions built with Edu2Com across time. As we can see, our algorithm reaches the optimal solution for each data family (low-size involving 10 tasks, middle-size involving 15 tasks, large-size involving 20 tasks). That is, in all problem instances, Edu2Com computed a solution exhibiting competence affinity equal to the affinity of the corresponding optimal solution. Table 5.2 summarises the average solution time and the average converged solution quality. Notably, as shown in Table 5.2, all 60 problem instances (20 instances per data family) reached optimality, i.e., the average quality of the converged solution equals 1 with 0 standard deviation.

[1]Note that any importance weight or expertise level sampled below 0 is considered as 0, and any weight above 1 is considered as 1.

Data Family Size	Solving Time (sec)		Converged Solution Quality	
	Average	Standard Deviation	Average	Standard Deviation
low (10 tasks)	124.561	9.52	1	0
middle (15 tasks)	153.66	21.07	1	0
large (20 tasks)	633.962	13.51	1	0

Table 5.2: Average solving time and converged solution quality over 20 problem instances per data family.

Data Family Size	Time (sec)		Time-Savings (%)
	Edu2Com	CPLEX	
low (10 tasks)	124.561	407.214	55.70%
middle (15 tasks)	153.66	438.419	52.22%
large (20 tasks)	633.962	1899.769	64.69%

Table 5.3: Time savings to converge to an optimal solution *wrt.* CPLEX.

5.2.3 Runtime Analysis

In our second analysis, we focus on the runtime behaviour of our heuristic algorithm. The most significant advantage of Edu2Com is that it is much faster than CPLEX. Table 5.3 shows the time we can save with respect to CPLEX to reach optimality. Time-saving is quantified as the relative difference between the time needed by Edu2Com to converge to the optimal solution, denoted as $t_{edu2com}$, and the time needed by CPLEX to solve the problem instance optimally, denoted as t_{cplex}—we express time-savings as a percentage:

$$\text{time-saving} = \left(1 - \frac{t_{edu2com}}{t_{cplex}}\right) \cdot 100\%$$

In detail, considering the time we need to build the corresponding linear programs and solve the problem instances with CPLEX, we can save from ~52% to ~65% time by using Edu2Com instead. At this point, we should note that the primary time-consuming task for CPLEX is building the linear programs.

5.2.4 Anytime Analysis

In our third analysis, we study the anytime behaviour of our heuristic algorithm. Edu2Com reaches solutions of high-quality (i.e., with quality above 0.8) at least 7 times faster than CPLEX. In more detail, let us denote with t_{opt} the time in seconds that CPLEX required to yield a solution. Edu2Com finds a solution of quality at least 0.8 after:

Quality	Low-size (10 tasks)		Middle-size (15 tasks)		Large-size (20 tasks)	
	Time (sec)	Portion of t_{opt}	Time (sec)	Portion of t_{opt}	Time (sec)	Portion of t_{opt}
≥ 0.30	0.091	$2.23 \cdot 10^{-5}$	1.451	$3.31 \cdot 10^{-3}$	2.564	$1.135 \cdot 10^{-3}$
≥ 0.65	17.51	0.043	54.608	0.125	227.93	0.119
≥ 0.80	26.313	0.064	65.251	0.142	238.715	0.126
≥ 0.90	36.061	0.088	83.205	0.189	400.796	0.211
1.00	49.806	0.122	110.38	0.252	515.085	0.271

Table 5.4: Quality of the solution as time progresses, the time needed in seconds, and the proportion of time compared to the time required by CPLEX (t_{opt}).

- $t_{opt} \cdot 6.4\%$ in the low-size setting which corresponds to 26.313 seconds,
- $t_{opt} \cdot 14.2\%$ in the middle-size setting which corresponds to 65.251 seconds, and
- $t_{opt} \cdot 12.6\%$ in the large-size setting which corresponds to 238.715.

Moreover, our algorithm finds the first solution *(i)* of quality 0.32 after $2.25 \cdot 10^{-5}$ of the time t_{opt}, *(ii)* of quality 0.30 after $1.451 \cdot 10^{-3}$ of the time t_{opt}, and *(iii)* of quality 0.28 after $1.35 \cdot 10^{-3}$ of the time t_{opt} for the low-size settings (10 tasks), middle-size settings (15 tasks) and large-size settings (20 tasks), respectively. Table 5.4 describes the quality of the solutions as time progress for the three different families of datasets. Specifically, we provide the time needed by Edu2Com to reach a solution of a certain quality and the portion of time compared to t_{opt}.

5.3 Case Study: Fondazione Bruno Kessler (FBK) Institute

This section reports the findings resulting from our collaboration with the *Fondazione Bruno Kessler (FBK)*[2] institute in Italy in the framework of the European project "AI4EU: A European AI On Demand Platform and Ecosystems".[3] In this case study, we tested our algorithm, Edu2Com, in the real-world problem of forming student teams to be placed in internship programs. Our analysis concerns: *(i)* the capabilities of Edu2Com to solve large real-world problems, *(ii)* the limitations of solving optimally such problems, and *(iii)* the quality of the solutions computed by Edu2Com validated by experts when pitched against human expert's solutions.

[2]https://www.fbk.eu/en/
[3]https://www.ai4eu.eu/

This empirical evaluation was conducted with the aid of our collaborators Raman Kazhamiakin, Ornela Mich, and Alessio Palmero Aprossio from Fondazione Bruno Kessler and Jean Christophe Pazzaglia from SAP.

5.3.1 Real-world Data

In this empirical analysis section, we used real-world data. In more detail, we use 100 profiles of students provided by our FBK partner. The students' profiles correspond to past students' profiles who participated in the *School-Work Alternation (SWA) programme*. For this reason, these profiles contain only the students' competence profiles. The competencies were extracted from students' CVs and described in ESCO ontology—the *Competence and Skills Extraction Tool* developed by FBK can be found in [Georgara et al., 2023]. Students' profiles involved 118 distinct competencies, and each student acquired ~ 11.95 competencies.

Moreover, we used 50 real internship programs offered by FBK in the past. The Competence and Skills Extraction Tool was used for each internship program to extract the required competencies based on the internship description. The internship programs involved 34 distinct competencies, and each internship required 4 competencies (minimum 2, maximum 15).

5.3.2 Scalability Analysis

This analysis aims to highlight the scalability of the problem as tasks' required team sizes increase and assess Edu2Com on handling the problem. Notably, the actual data regarding the tasks (internship programs) did not specify the team size. Thus we synthetically populated the problem instances with specific team sizes. [Andrejczuk et al., 2018] points out that teams within classrooms shall have at most five members. As such, we considered problem instances where all tasks require the same team size, ranging in $[2, 5]$. Moreover, we considered problem instances with varying required team sizes. Specifically, the problem instances contained tasks requiring team sizes ranging in [2,3], [2,4], [2,5], [3,4], [3,5], and [4,5]. In each problem instance, the number of tasks requiring a specific team size is uniformly distributed across the team sizes in the corresponding range. As we discuss in Section 5.3.4, we cannot optimally solve instances with 100 agents, 50 tasks, and the aforementioned team sizes.

Analysis. In Figure 5.2, we show our findings with respect to the time needed by our algorithm to converge to a solution. Each bar in Figure 5.2 illustrates the average time (in minutes: seconds) over 20 problem instances per required team size. Here we highlight that Edu2Com converges to a solution in less than 50 minutes, especially in large settings that contain tasks requiring teams of size 5. Our experiments showed that Edu2Com needs less time to solve instances containing tasks requiring smaller team sizes—such a result is expected since the search

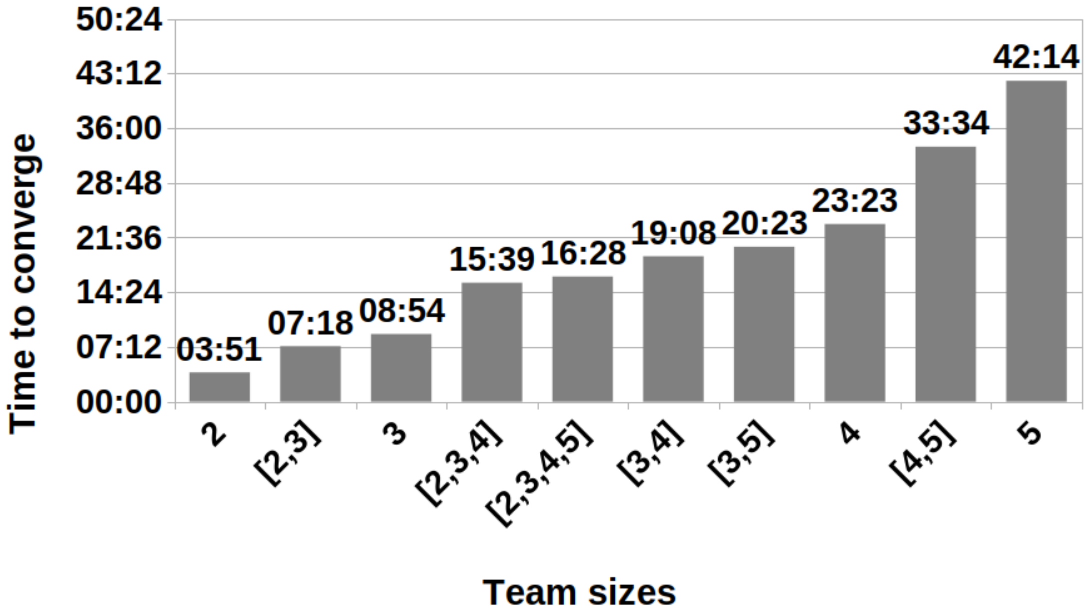


Figure 5.2: Time in Min:Sec required as team sizes grow.

space is much smaller in small team-size instances. Specifically, we observe that solving a problem instance with team size 2 requires less time than a problem instance with team size 3, which in its turn requires less time than a problem instance with team size 4, which in its turn requires less time than a problem instance with team size 5. Moreover, notice the time needed for finding a solution in problem instances with tasks requiring team sizes in a range $[a, b]$. In such problem instances, the time needed to find a solution falls between the time Edu2Com needs to solve *(a)* problem instances requiring teams only of size either a or b, and *(b)* instances requiring teams in team sizes in ranges $[a, b-1]$ and $[a+1, b]$. In the hardest scenario where all tasks' required team size is size 5, Edu2Com yields a solution in (approximately) less than 50 minutes. Given that this process does not need to be performed in real-time with highly demanding time constraints (i.e., we do not need to come up with a solution within a few seconds), it is acceptable. Notably, the educational authorities spend much more working time matching students to internships since the current practice is to do it manually; while solving the problem optimally with the means of an LP is infeasible since we cannot even generate the LP in time. Hence, our findings confirm our algorithm's feasibility in large problem instances and show that Edu2Com can handle the team allocation problem in this education scenario.

5.3.3 Expert Validation: Edu2Com vs Human Experts vs Random

In this part of our empirical analysis, we focus on validating our algorithm, Edu2Com, by educational authorities with experience in allocating students to internship programs. To do so, we pitched Edu2Com against teachers with such experience—we refer to them as *experts*. In more detail, we consider a synthetic problem instance—similar to the actual-world problem instances we employed in Section 5.3.2—with 100 agents (students) and 50 tasks (internships) requiring teams of sizes 1, 2 and 3. The problem instance used here is the largest regarding the number of tasks we can generate with the 100 student profiles at hand. Notably, solving this problem optimally (e.g., with CPLEX) would require more than 1.8 million decision variables.

Then, we followed the process below. Given the synthetically generated problem instance:

1. an expert matches each internship with a student team by hand,
2. Edu2Com computes an allocation of students to internships, and
3. we randomly allocate a student team to each internship

Henceforth, g_{expert}, g_{edu2com}, and g_{random} stand for the allocations built with each method, respectively. Next, we task eight (8) experts with experience in the allocating process to assess and compare the three allocations, ignoring by which method each allocated was built—we refer to these experts as *evaluators*. Here we would like to highlight that regarding the time needed to reach a solution Edu2Com required less than 1 hour and 45 minutes to build an allocation. In contrast, the expert needed approximately the time of a working week, including studying, analysing and matching each internship with a student team.

Evaluation Process. We asked each evaluator to study and assess the three allocations. Specifically, the evaluators should mark the team allocated to each internship as follows:

- 1 for high quality,
- 2 for medium quality, and
- 3 for low quality.

Moreover, the evaluators were allowed to mark with the same value teams assigned to the same internship that were produced by different methods in case they considered the teams were of the same quality

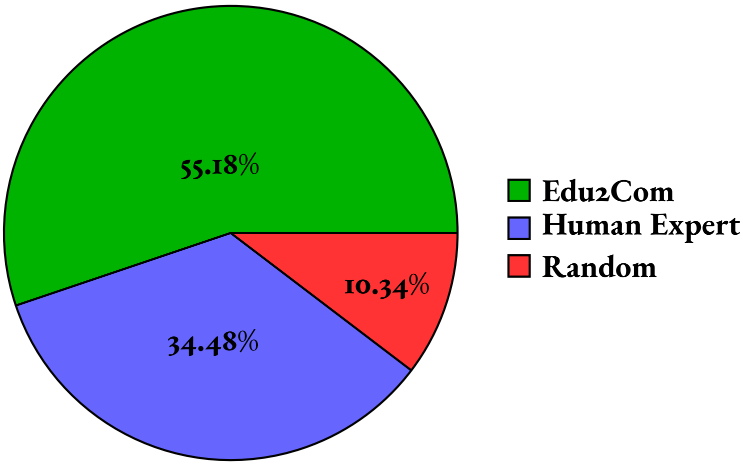


Figure 5.3: **Single winner**: Edu2com vs Human Expert vs Random. Percentage out of 29 tournaments.

Handling missing data. During the final analysis, we encountered a challenge of missing data where the expert could not form a team for every internship. Specifically, out of the 50 internships, the expert failed to provide a team for 13 of them, resulting in 23 students out of 100 being without internships. As a result, the evaluators had to work with incomplete data and provide incomplete evaluations. The absence of expert allocation for some tasks prevented the evaluators from rating all three allocation methods (g_{expert}, g_{edu2com}, and g_{random}). An auxiliary mark of 4 was used to signify the missing allocation, indicating absence, which was considered worse than a low-quality mark (mark 3). As a result, all evaluators marked any missing allocation with a four. Additionally, the evaluators missed marking the teams of some internships, resulting in low-quality marks (mark 3) for missing evaluations.

Analysis. Our analysis aims to determine the best method for allocating student teams to internships, founded on the evaluators' assessments. For that, we consider the evaluation of the teams assigned to each internship as a *tournament*, while a tournament involves three rounds of competing pairs:

1. Edu2Com vs Expert;
2. Edu2Com vs Random;
3. Expert vs Random

The evaluators' marks serve as the means for determining the winning allocation method for each round and, therefore, for each tournament. The allocation method that receives the higher aggregated mark in a round is declared the winner of that round and earns one point for the corresponding internship assignment. If in a round two internship assignments are of the same quality, i.e. there is a tie, both of their allocation methods receive half a point. By tallying the points earned from each round of the tournament, we utilise the Copeland$_\alpha$ voting rule [Conitzer and Sandholm, 2002] (with $\alpha = 0.5$) to determine the overall winner. This voting rule is shown in [Faliszewski et al., 2007] to be "resistant to all the standard types of (constructive) electoral control." In a nutshell, the allocation method that accumulates the highest number of points throughout the three rounds is declared the tournament winner. Again, if there is a tie between two allocation methods, each one earns half a point. For instance, suppose that for a given tournament, 8 evaluators considered the heuristic algorithm (Edu2Com) the best assignment, 5 evaluators preferred the expert's allocation to the random one, and 2 evaluators equally favoured both the expert and the random allocation. This would result in the heuristic algorithm earning 8 points, the expert's allocation receiving 6 points, and the random allocation getting 2 points. Therefore, the tournament winner, in this case, would be the heuristic algorithm.

In each tournament, there can be three outcomes:

1. a single winner,
2. a tie with no winners or
3. no winner

Our analysis shows that 58% out of 50 tournaments declared a single winner, 34% declared two winners in a tie, and 8% declared no winner. Figure 5.3 illustrates the tournament results that declared a *single winner*. As we can see, 55.17% of these tournaments were announced as the winning allocation method Edu2com, the heuristic algorithm, 34.48% of them were announced as the winning allocation method of the human exert, while 10.35% of the tournaments were won by the random allocation method. As such, our heuristic algorithm, Edu2Com, was the preferred allocation method to allocate student teams to internships.

Now, in Figure 5.4, we report the tournaments announcing *a tie with two winners*. As expected, the majority of the tournaments (52.92%) declared a tie between the methods of our heuristic algorithm and the human expert. Overall, Edu2Com was announced as a winner in a tie in 88.23% of the tournaments. To summarise, our analysis indicates that expert evaluators deem Edu2Com as the method of choice to assign student teams to internships.

5.3.4 Reasonable time limits for response and limitations of optimal solving

In this section, we study the limitations of forming student teams and assigning them to internship programs optimally. Notably, as mentioned in Section 5.2, the problem we address here

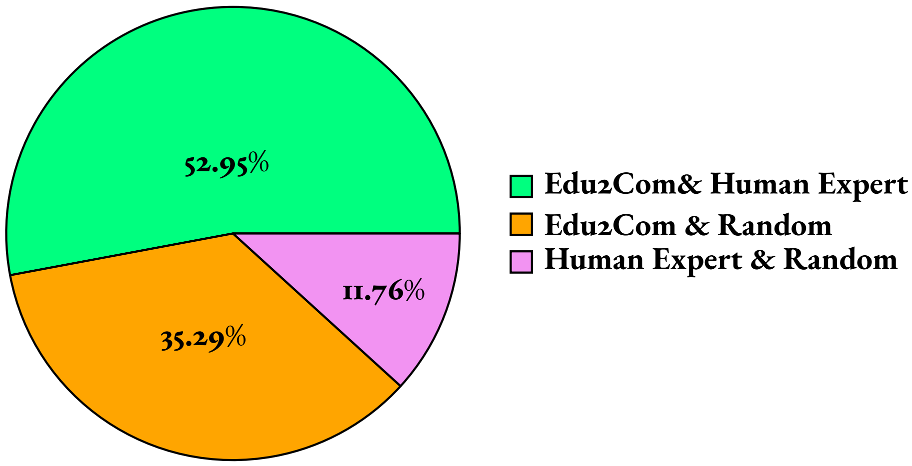


Figure 5.4: **Tie with two winners**: Edu2Com vs Human Expert vs Random. Percentage out of 17 tournaments.

does not require a real-time solution within tight time limits. As we stated in Section 5.3.3, a human expert may require a whole week in order to arrange a real-world school-work alternation programme. That is, a human expert needs around a week to manually solve a real-world problem instance with around 100 students and 50 internships. As such, we believe that 24 hours is a reasonable computing time for solving the problem.

As mentioned in Section 5.3.2, CPLEX cannot cope with real-world instances. This section aims to empirically characterise the instances that we can optimally solve and hence the limits of CPLEX. Figure 5.5 shows the configurations of problem instances (in terms of the number of agents, number of tasks, and team sizes) that CPLEX can optimally solve. Green squares show problem instance configurations that CPLEX managed to solve, whereas red squares show problem instance configurations that CPLEX could not solve (because it ran out of memory). The figure shows that CPLEX cannot handle real-world problem instances (with 100 agents and 50 internships) unless we limit team size to 2. Considering only pairs is not realistic: as pointed out in [Andrejczuk et al., 2018], educational scenarios require teams of sizes up to 5. As we increase the team size from 2 to 5, the range of configuration of problem instances that CPLEX can optimally solve dramatically decreases. We empirically observe that CPLEX did not manage to solve problem instances that lead to an LP containing more than around $8 \cdot 10^5$ decision variables. Figure 5.5 characterises the *frontier* of configurations of problem instances that CPLEX can optimally solve. In Appendix I, we include the *frontier* of configurations of problem instances along with the number of decision variables involved in the corresponding LPs and required solving time. We highlight that the solving time depends on

1. the number of agents,

2. the number of tasks, and
3. the required team sizes (by all tasks)

with the number of agents and team size being the most influential

5.4 Case Study: Technical University of Crete (TUC)

This section reports our observations from employing Edu2Com in undergraduate university courses. In collaboration with the *Technical University of Crete (TUC)*[4] in Greece—and specifically the School of Electrical and Computer Engineering—we formed student teams to work on their semester project for the "Design and Development of Information Systems"[5] course during the academic year 2021-2022. In this case study, we tested Edu2Com in yet another real-world problem from the educational domain.

This empirical evaluation was conducted with the aid of Georgios Chalkiadakis and Nikolaos Pappas from the School of Electrical and Computer Engineering of the Technical University of Crete.

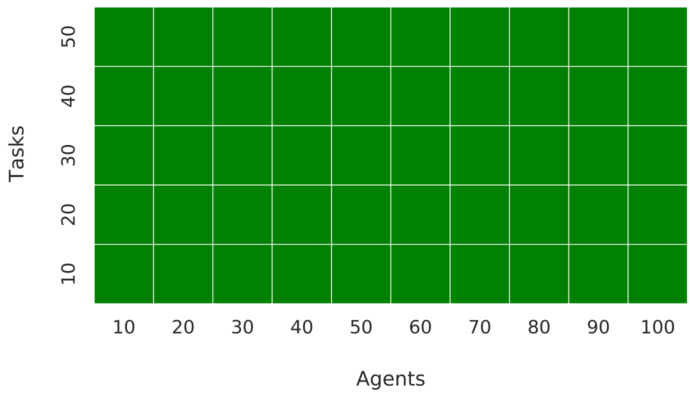
5.4.1 Real-world Data

In this empirical analysis, we used real-world data. In more detail, we collected data from 118 undergraduate students (ages ranging from 20 to 25 years old), including competence profiles, personality profiles, and preferences over projects and teammates. For their competence profiles, students were asked to self-assess their competency across 21 distinct competencies described in the ESCO ontology. Specifically, each student assessed their expertise level for each competence as *(i)* Novice, *(ii)* Advanced Beginner, *(iii)* Competent, *(iv)* Proficient or *(v)* Expert. We used the Post-Jungarian Personality Test developed by [Wilde, 2013] to collect the students' personality profiles.[6] Moreover, students declared their preferences over the different semester projects; each student was asked to specify their interest in working on each project (out of 10 project types) as *(i)* Not at all interested, *(ii)* Not so interested, *(iii)* Somewhat interested, *(iv)* Very interested or *(v)* Extremely interested. Finally, to extract students' preferences over teammates, we asked each student to name their top-five potential teammates and their top-five people to avoid. We opted for partial preferences over teammates—i.e., a preference relation consisting of the five most preferred and the five least preferred teammates, while everyone else is considered as neither preferred nor not-preferred—because (1) it is rather hard to engage people to respond in surveys requiring more than 10 minutes of their time [Revilla and Ochoa, 2017], and (2)

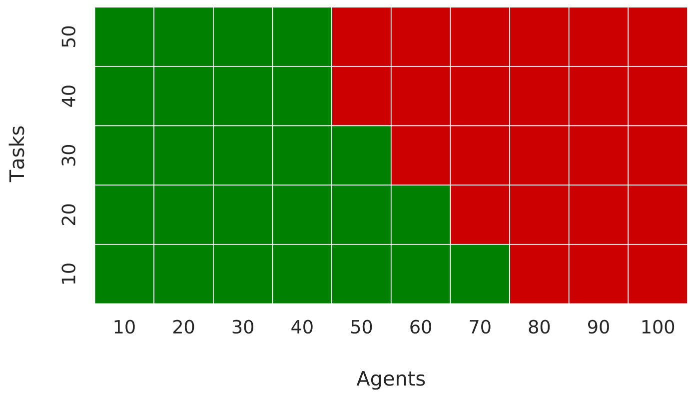
[4]https://www.tuc.gr

[5]Corresponding greek course title: "Σχεδίαση και Ανάπτυξη Πληροφοριακών Συστημάτων"

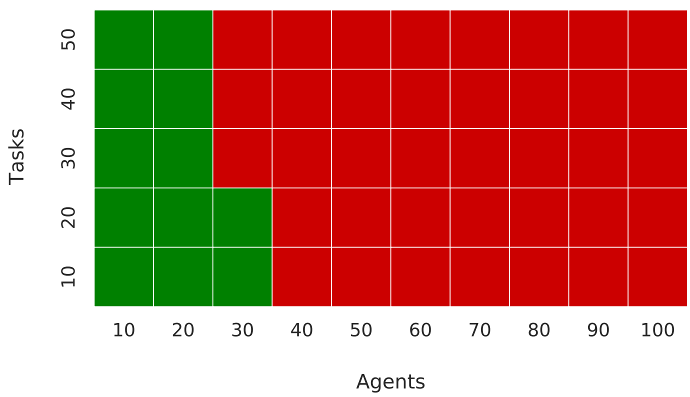
[6]We translated the personality test into Greek so the students could answer it in their primary language.



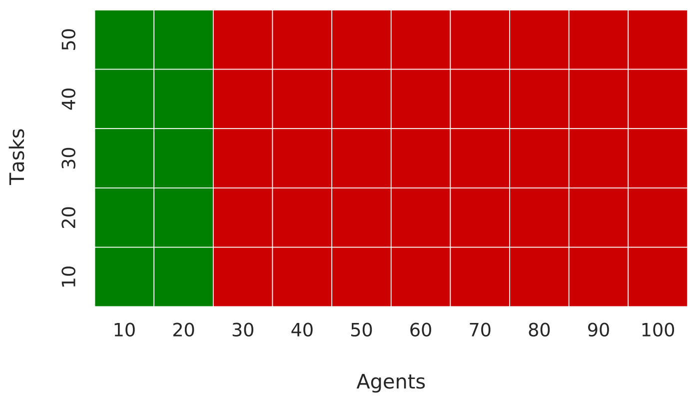
(a) Team Size 2



(b) Team Size 3



(c) Team Size 4



(d) Team Size 5

Figure 5.5: Problem instance configurations (number of agents, number of tasks, and team sizes) solvable by CPLEX. Red squares correspond to configurations that CPLEX cannot solve (because it runs out of memory), while green squares correspond to configurations that CPLEX can solve.

commonly, people are not familiar with every other individual to have an opinion regarding a potential partnership. Instead, they are aware of the few people they can collaborate with and the few they do not get along with. The questionnaires for extracting students' profiles can be found in Appendix II.

The semester project aims to engage the students to work on *requirement analysis* while designing an information system. To do so, each student team needs to find an *existing* business entity, analyse the needs of the business entity, and design an information system for them. Given the project's objective, the professor in charge of the course offered different types of businesses that could be the business target of the analysis. Therefore, the professor offered ten (10) variations of the project and described each project using eleven (11) competencies related to the project's main objective, along with capabilities in interacting with the client and presenting the team's work. Additionally, each project variation has one (1) competence related to the business target's functionality. All competencies were described in the ESCO ontology. The required team size for such a project was set to 4 or 5 students per team (regardless of the type of business target).

5.4.2 "Teachers' Methodology": A greedy approach

For comparison reasons, we developed a greedy approach following the "teachers' methodology", i.e., the practices in which a teacher usually forms student teams [Andrejczuk, 2018]. The greedy approach we followed aims to form heterogeneous teams by mixing strong and weak (with respect to their skills) students. This methodology allows students to help each other while working and promotes learning from each other. Moreover, such heterogeneous teams exhibit fewer conflicts among the team members [Sedaghat, 2018]. The teachers' methodology includes the steps below:

1. Characterise each student based on their skills as: *(i)* strong, *(ii)* weak, or *(iii)* average.
2. Build a team by:
    a) choosing **one strong** student;
    b) choosing **one weak** student; and
    c) **completing** the required team size **with average** students.

To characterise the students, we used their competence profiles. First, we compute for each student a their average competency as follows:

- we map the available qualitative expertise levels to a real number in $[0, 1]$:

Novice	0.2
Advanced Beginner	0.4
Competent	0.6
Proficent	0.8
Expert	1.0

- we compute the average competency:

$$\text{average\_competency}(a) = \frac{1}{21} \cdot \sum_{c} l_a(c)$$

- we normalise with

$$\text{normalising\_factor} = \frac{1}{\max_a \text{average\_competency}_a}$$

Given the (normalised) average competency, we characterise the top-m students as *strong*; the bottom-m students as *weak*, and anyone else as *average*, where m corresponds to the number of teams we need to form. Then, for each task τ we form a team by randomly selecting one strong student, one weak student, and $s_\tau - 2$ average students.

5.4.3 Validation in Practice: Edu2Com vs Teachers' Methodology

In this section, we study how teams perform in practice, i.e., the quality of the deliverables handed in by each team. For this reason, we compare teams formed by Edu2Com considering competence affinity, personality, and gender diversity, preferences over tasks, and preferences over teammates—our methodology—against teams formed by the teachers' methodology described in Section 5.4.2.

As mentioned earlier, we involved 118 undergraduate students to form teams and carry out a semester project, while 10 different variations of the semester project were offered. Each student shall be a member of exactly one team, and each team shall work on exactly one project variation. Given that there are 10 variations, two different teams may work on the same project variation: two different teams may work on the same type of business target (see Section 5.4.1), but they cannot work on the very same business target. Moreover, it was not necessary that all project variations would be assigned to at least one team—i.e., we could have multiple teams working on business target A and at the same time have no teams working on business target B.

The experimental process followed the next steps:

1. Divide the students into two distinct groups: the *test group* and the *control group*.

Business Target	Number of Teams		Business Target	Number of Teams	
	Test Group	**Control Group**		**Test Group**	**Control Group**
Accommodation Business	2	2	Lending Library	1	1
Construction Company	1	1	Pharmacy	3	3
Clinic	0	0	Rental Business	0	0
Event Center	0	0	Retail Business	2	2
Food Service Business	3	3	Training School	2	2

Table 5.5: Distribution of project's variations per group.

2. Use Edu2Com to form teams in the test group.

3. Use the Teachers' Methodology to form teams in the control group.

4. Each formed team (in either group) carries out their semester project and hand in the project's deliverables. Students do not know which way their team was formed.

5. The professor's assistant, without knowing how the teams were formed,[7] evaluates the deliverables by awarding a grade within $[0, 10]$ to each team.

In each group (test and control), we formed 11 teams consisting of 4 members and 3 teams consisting of 5 members. Moreover, the distribution of the projects' variation is shown in Table 5.5.

After forming the teams in each group, each team carried out their assigned project. Thereafter, the evaluator assessed each team across 4 criteria:

- Technical Report:
  - correctness of technical content,
  - report's structure
- Oral Presentation
- Teamwork & Members' Participation
- Client's Satisfaction

[7] To avoid any bias during the projects' evaluation, the evaluator was not informed that the teams were formed using AI tools and therefore was unaware of which team belonged to each group.

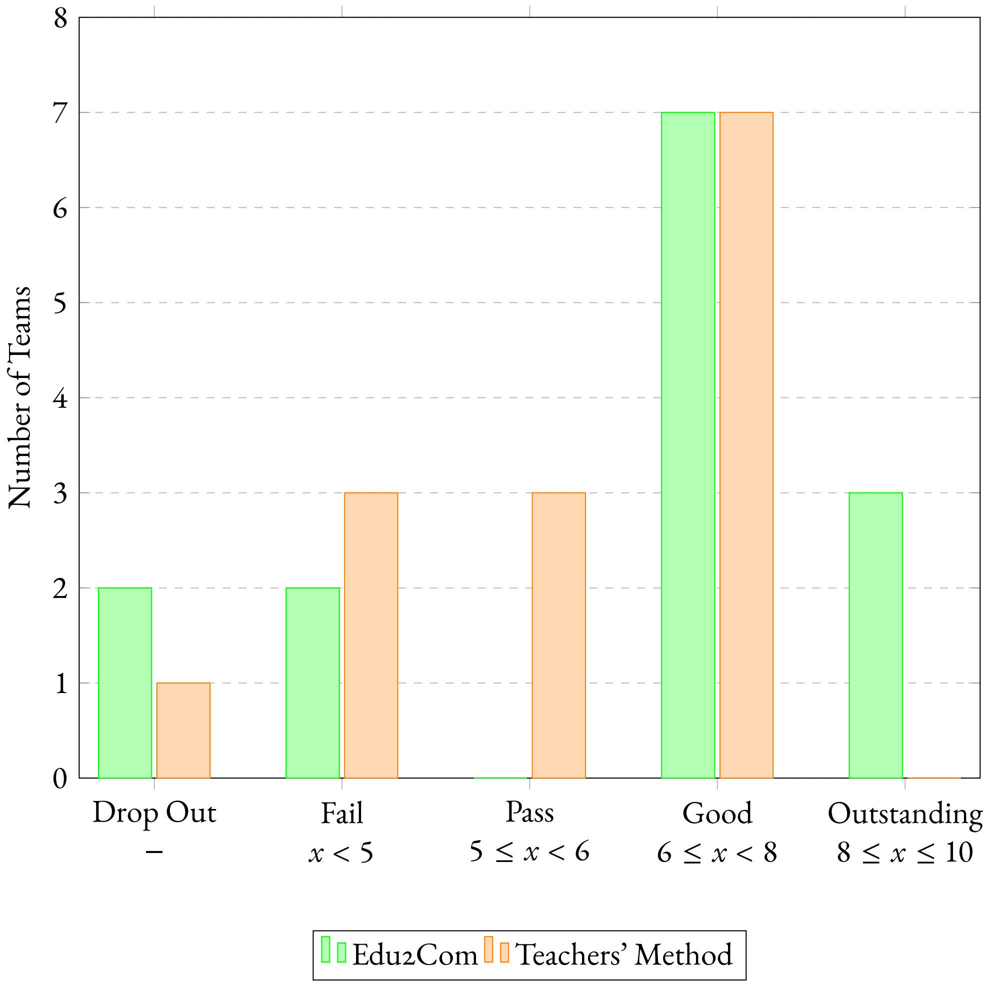


Figure 5.6: Teams Performance: Edu2Com vs Teachers' Method

grading each team with a mark ranging from 0 to 10, with 0 indicating the lowest quality, 10 indicating the highest quality, and 5 being the Pass / Fail bordeline. Given the teams' final marks, we distinguished five (5) categories of outcomes: *(i) Drop out*, for the teams that failed to deliver their project, *(ii) Fail*, for the teams that delivered their project but received a mark below the borderline ($= 5$), *(iii) Pass* for the teams that received a mark in the range $[5, 6)$, *(iv) Good* for the teams receiving a mark within $[6, 8)$, and *(v) Outstanding* for the teams receiving a mark within $(8, 10]$.

Analysis. This analysis aims to investigate whether teams built with Edu2Com *outperform* teams formed following the teachers' methodology. Figure 5.6 illustrates the results. As we can see, 10.71% of the teams (i.e., 3 teams out of 28) were awarded an 'Outstanding' grade, with all of these teams being formed with Edu2com. 50% of the teams (14 out of 28) were awarded a 'Good' grade, with Edu2Com and Teachers' Method forming 50% of these teams each. At the same time, 10.71% of the teams (3 out of 28) received a 'Pass' grade (just above the borderline = 5), all formed with the Teachers' Method. Regarding the teams that did not manage to reach the borderline and received a 'Fail' grade, we see that 17.86% of the teams (5 out of 28) failed, with 40% of these teams (2 out of 5) being formed with Edu2Com and 60% (3 out of 5) with the Teachers' Method. Finally, 10.71% of the teams (3 out of 28) dropped out, i.e., did not complete their semester project, with 66.67% of the teams being formed with Edu2Com (2 out of 3) and 33.33% of the teams (1 out of 3) being formed with the Teachers' Method.

Overall, both methodologies share some behavioural similarities, e.g., for each methodology, (1) 71.43% of the teams received a grade above the borderline, and (2) 50% of the teams received a 'Good' grade. However, looking closer at the grades awarded, the teams formed with Edu2Com outperformed the ones formed with the Teachers' Method since: (1) Edu2Com formed teams that received an 'Outstanding' grade while Teachers Method did not, and (2) Edu2Com formed fewer teams that failed compared to the Teachers' Method.

Figure 5.7 presents the team's performance (captured through the team's mark) with respect to the team's collegiality. Both Edu2Com and the Teacher's Method formed teams with collegiality ranging in [0.4, 0.65]. As shown in Figure 5.7, the team with the lowest collegiality (0.4) was graded with the lowest mark (4), and at the same time, the team with the highest collegiality (0.63) was graded with the highest mark (9). We used linear regression to model the relationship between teams' collegiality and teams' performance. The obtained line shows that the higher the team's collegiality, the higher the team's performance is as well.[8] This result supports our hypothesis that high collegiality boosts performance. As such, using collegiality as a guide can help form human teams that perform well.

At this point, we want to report that, according to the professor in charge of this course, forming teams with AI tools (referring to both Edu2Com and the Teachers' Method) resulted in lower dropout and failure rates compared to former years, when students self-organised in teams. At the same time, they exhibited higher-quality projects compared to former years. Notably, the professor expressed their willingness to reuse such AI tools, especially to form competitive teams with the students that fail to be self-organised in a team.

[8]The Mean Squared Error (MSE) is 0.7055.

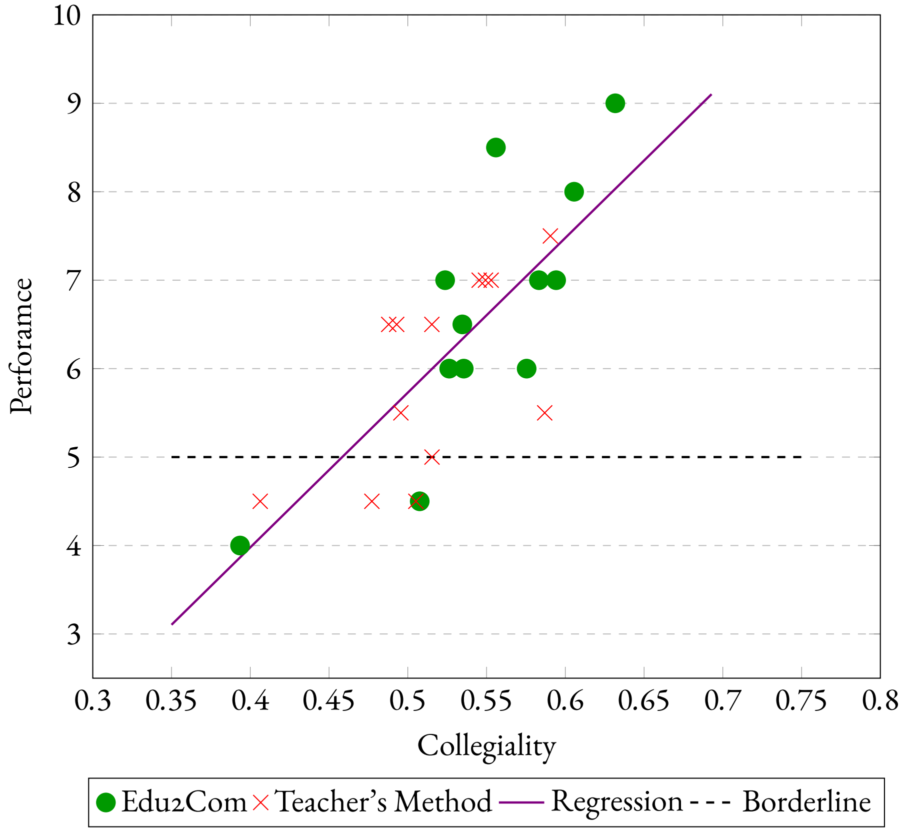


Figure 5.7: TUC Case: Teams' Performance vs Teams' Collegiality

5.5 Case Study: EADA Business School

In this section, we continue the study of the performance in practice of teams formed with our algorithm, Edu2Com. Here we collaborated with *EADA Business School*[9] in Spain. In this case study, we needed to form student teams to carry out a two-day activity as part of their Master's program. This empirical evaluation was conducted during the academic year 2022-2023, with the aid of Antony Lewis Poole and Mireia Montané Balagué from EADA Bussiness School.

5.5.1 Real-world Data

In this empirical evaluation, we used real-world data. Specifically, we collected data from 164 Master's students who attended Master's programs in several disciplines (e.g., management, finance, sustainable business & innovation, marketing, and hospitality). The collected data in-

[9]https://www.eada.edu/en

clude competence profiles, personality profiles, and task preferences. For their competence profiles, the students were asked to self-assess their competency across 9 different competencies described in the ESCO ontology. Similarly to the data collection in Section 5.4.1, each student assessed their expertise level as *(i)* Novice, *(ii)* Advanced Beginner, *(iii)* Competent, *(iv)* Proficient or *(v)* Expert. Again, we extracted students' personality profiles using the Post-Jugarian Personality Test [Wilde, 2013].[10] Finally, for each task, students specified their preferences as *(i)* I love to do it, *(ii)* I want to do it, *(iii)* I don't mind to do it, *(iv)* I slightly prefer not to do it, or *(v)* I prefer not to do it.

As part of the Master's programs, the students (from all disciplines) attended a two-day common module. In this module, the students were engaged in a creative and fun activity with interdisciplinary collaborations, aiming to wrap up the lessons learned. The professors in charge of the Master's program offered three different task types. Each task type was described using five (5) competencies in the ESCO ontology: two (2) competencies common to all three task types and three (3) distinct competencies per task type. The desired team size ranged from 3 to 5 members per team (regardless of the task type).

5.5.2 Experimental Process

The 164 students attended the two-day module in three groups:

Group	Masters' Program	Number of Participants
A	Finance, Hospitality, Marketing	63
B	Sustainable Business & Innovation	42
C	Management	59

In *Group A*, we split the students randomly into two sub-groups. For the first subgroup (group A_1), consisting of 32 students, we used Edu2Com to form 9 teams of sizes 3 and 4. For the second subgroup (group A_2), consisting of 31 students, we randomly distributed the students into 9 teams of sizes 3 and 4.

In *Group B*, students worked in their "stable teams". During the Master's program offered by EADA Bussiness School, students are put together in teams to work throughout their studies. People in EADA Bussiness School form these teams by hand, considering information regarding students' *academic/scientific background*, *behavioural assessments* obtained through DISC assessments[Marston, 2013], and their *nationality/cultural background* (in an attempt to form diverse teams). *Group B* involved 8 stable teams of sizes ranging from 4 to 6 members per team. To match the teams with tasks, we manually assigned each task to a team that was more motivated for the task.

[10] Students answered the personality test in English.

Group	Number of teams per task type		
	Task type 1	**Task type** 2	**Task type** 3
A_1	3	3	3
A_2	3	3	3
B	3	2	3
C_1	2	2	2
C_2	2	2	2

Table 5.6: Number of teams per task type per group.

In *Group C*, again, we split the students randomly into two sub-groups. For the first subgroup (group C_1), consisting of 30 students, we used Edu2Com to form 6 teams of size 5. For the second subgroup (group C_2), consisting of 29 students, we used Edu2Com considering *only* the students' competence and personality profiles to form 6 teams of sizes 4 and 5. We refer to the algorithm used in group C_2 as "SynTeam-base", as it uses the two components (competencies and personality) used by the Synteam Algorithm introduced in [Andrejczuk et al., 2019].[11]

According to the professors in charge of the Master's programs, task types should be assigned to a more or less equal number of teams in each group. Table 5.6 summarises the number of teams per task type per group. After each team carried out their assigned task, the evaluators (i.e., the professors offering the two-day module) graded wach team with a mark within the range of $[0, 10]$. In Group A and Group C, neither the students nor the evaluators were aware of the algorithm used to form each team—in Group B, students were in their stable teams.

5.5.3 Validation in Practice: Team Collegiality *wrt* Team Performance

In this section, we study the performance of the teams with respect to the teams' collegiality. Notably, all teams exhibited good performance, being graded with marks in the range $[7.3, 9.6]$. To make the differences easier to be observed between the teams, we "stretch" the teams' grades in the range $[5, 10]$, i.e., the team graded with the lowest mark 7.3 was mapped to the mark 5, and the team graded with the highest mark 9.6 was mapped to the mark 10. Let x be the mark of a team (with $7.3 \leq x \leq 9.6$); instead of x, we consider the stretched mark y defined as:

$$y = \frac{x - 7.3}{9.6 - 7.3} \cdot (10 - 5) + 5$$

Moreover, given the students' profiles, we observed that the majority of the students were indifferent among the three different tasks. Precisely, 24.44% of students did not specify any preference, while 27.44% declared the same preference for every task. In Group A, 50.73% of the

[11] Even though in Group C we formed 12 teams in total, only 8 teams carried out the activity due to a flue.

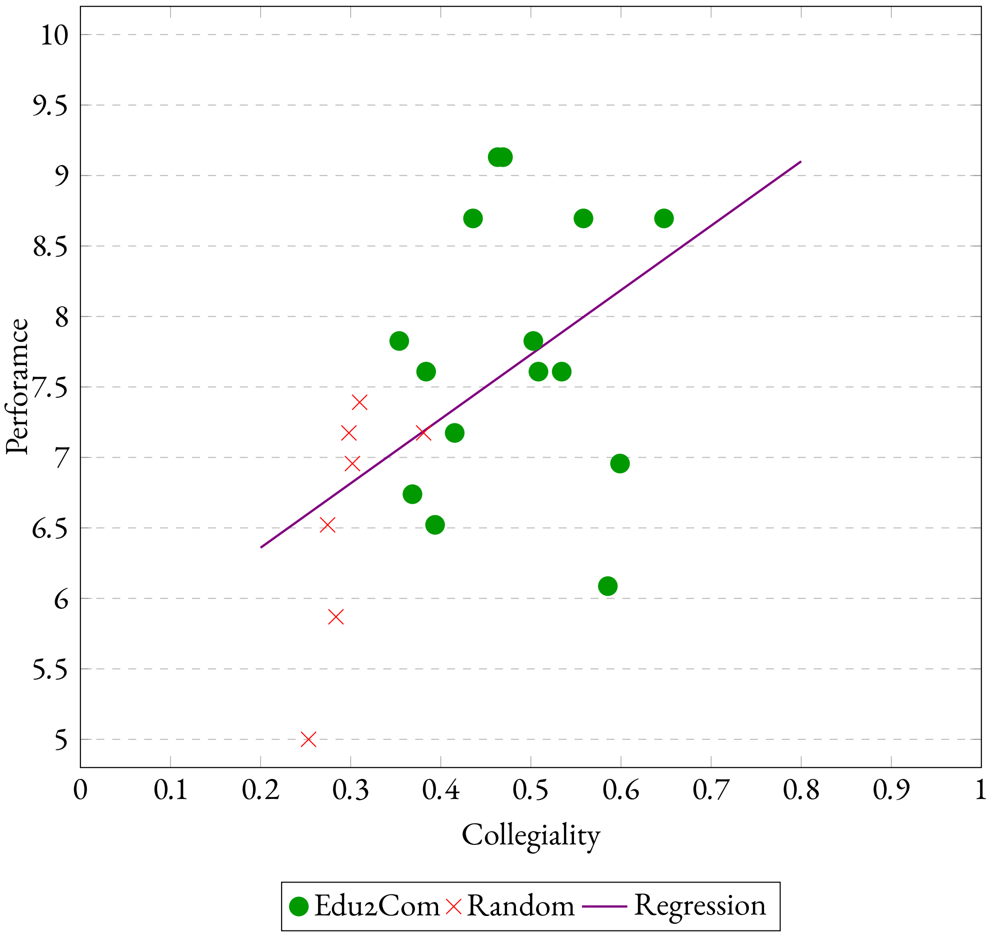


Figure 5.8: EADA Case: Teams' Performance vs Teams' Collegiality

students were indifferent; in Group B, the indifferent students were 28.57%; and in Group C, the indifferent students were 57.62%. The absence of preferences results in motivation having a negligible impact on the teams' collegiality. The negligible impact of motivation results in the Edu2Com algorithm and the SynTeam-base variation (in Group C) coinciding. As such, we will refer to the teams formed by either of these two algorithms as teams formed by Edu2Com.

Analysis. In this analysis, we study how collegiality affects teams' performance. Figure 5.8 illustrates the performance with respect to the team's collegiality for the "from-scratch-formed" teams, i.e., the teams formed randomly and the teams formed with Edu2Com. As we can see in the figure, teams formed randomly (red cross points) exhibit lower congeniality—which ranges in [0.2, 0.4]—compared to the collegiality of the teams formed with Edu2com (green circu-

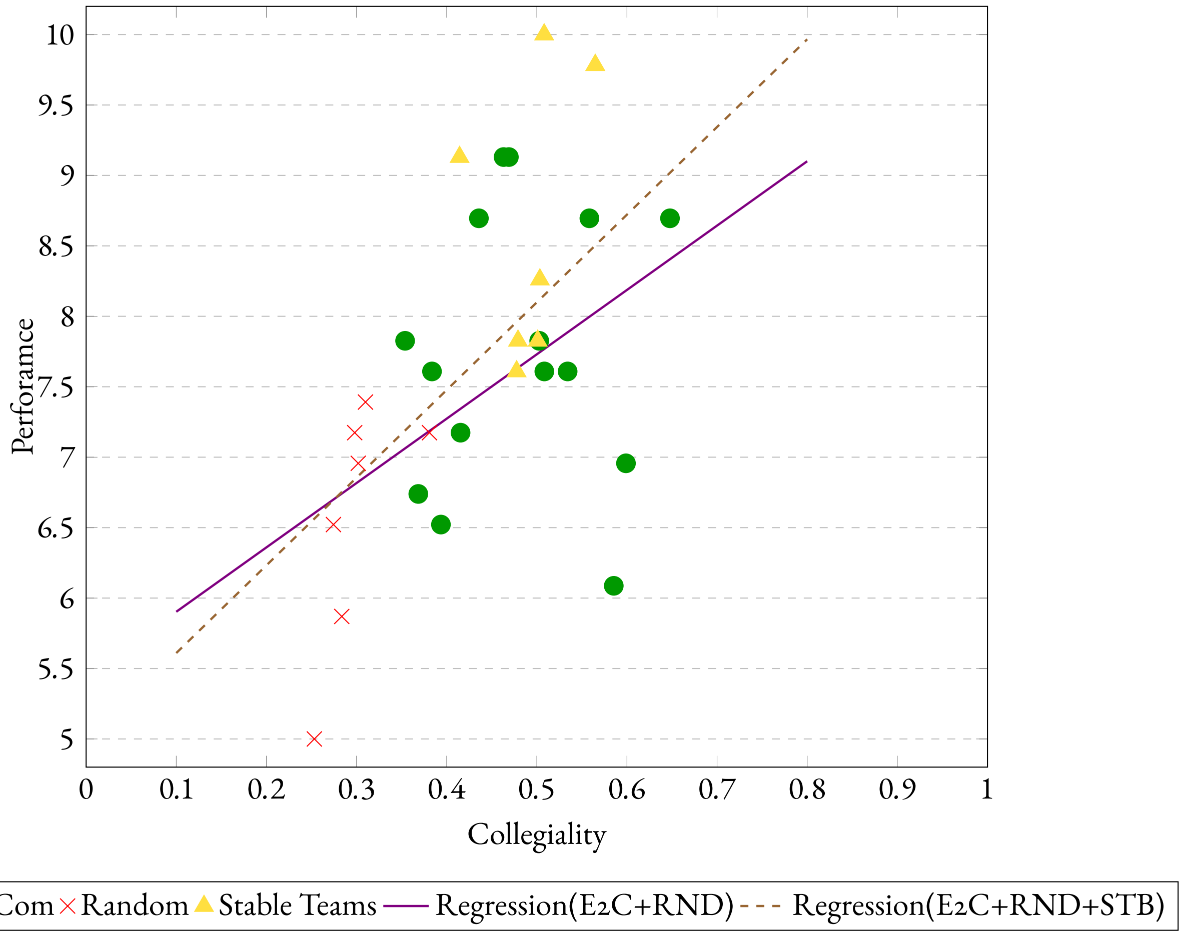


Figure 5.9: Team's Performance *wrt* Team's Collegiality: Edu2com vs Random vs Stable Teams

lar points)—which ranges in [0.3, 0.7]. At the same time, teams formed with Edu2Com exhibit better performance— which ranges in [6, 9.5]—than those formed randomly—the performance of which ranges in [5, 7.5]. We used linear regression over the data points to model the relation between collegiality and performance.[12] Even though the number of samples is low, the data show that increased performance is associated with increased collegiality, supporting our hypothesis. That is, the results indicate that the better the collegiality of a "from-scratch-formed" team is, the better the team's performance.

In Figure 5.9, we show the performance of all teams, including the stable teams, with respect to their collegiality. The stable teams (represented with yellow triangular points) exhibit better performance than the randomly formed teams (represented with red cross points) and

[12] The Mean Squared Error (MSE) is 0.8177.

slightly better performance than the teams formed with Edu2Com (represented with green circular points). Regarding the relationship between the team's collegiality and the team's performance, the stable teams are above the regression line obtained via the from-scratch-formed teams (purple straight line).[13] Such a result is expected due to the former interaction and collaboration among the members in the stable teams. In other words, stable teams are *experienced* teams, i.e., teams that have been working together repeatedly for an extended period of time; thus, such teams advance key teamwork factors, including trusting their teammates, knowing each member's strengths and weaknesses, and having established communication and coordination channels. Therefore, experienced teams are more likely to outperform teams formed from scratch, where the members have no prior interaction with each other.

Nonetheless, we observe that Edu2Com formed teams from scratch that can compete with the stable (i.e., experienced) teams. Teams formed with Edu2Com received a mark as high as the stable teams' marks. This observation suggests that in cases where we need to form teams from scratch, using collegiality can be a good guide for forming teams that perform well and are competitive with experienced teams. That is, when people have no prior interactions with each other or any collaborative experience as a team, we can use collegiality, and an algorithm such as Edu2Com, to form efficient teams.

Here, we would like to point out that all teams (formed-from-scratch and stable ones) received high marks—we remind the reader that every team received an actual mark in the range of $[7.3, 9.6]$. The generally high marks achieved by all teams may result from *(i)* very competent students and/or *(ii)* very determined students to successfully carry out their activity (regardless of the assigned task) and/or *(iii)* non-challenging tasks. Thus, in such cases—where we can find any of the *(i)*-*(iii)*—increased collegiality may not result in a significant added value in the teams' performance.

5.6 Summary

In this chapter, we reported a series of experiments to confirm our approach's effectiveness empirically. First, we tested the Edu2Com algorithm regarding solutions' quality, the algorithm's runtime performance and its anytime behaviour. Specifically, we pitched Edu2Com against the state-of-the-art optimal solver IBM CPLEX. The obtained results showed that *(i)* Edu2Com reaches optimality, *(ii)* it yields a solution much faster than CPLEX, and *(iii)* it reaches high-quality solutions after a few iterations.

Afterwards, we tested the capabilities of our algorithm to cope with large real-world problems. In particular, we employed Edu2Com to solve the real-world problem of forming student

[13] The regression line obtained via all the teams, including both from-scratch-formed teams and stable teams, is represented with the brown dashed line and exhibits Mean Squared Error (MSE) equal to 1.1733.

teams to be matched with internship programs using real-world data. We investigated the scalability of the problem as team sizes change, and we showed that Edu2Com reaches a solution within reasonable time limits, whereas solving the problem optimally is infeasible.

Moreover, in this empirical evaluation, we validated the quality of the solution provided by our algorithm. We tasked one human expert to form student teams and match them with internship programs, and after that, we asked different human experts to assess the teams provided by the human and Edu2Com. The results showed that *(i)* Edu2Com can be much faster than humans (less than 2 hours vs the working hours of a week) and that *(ii)* the experts preferred the teams formed by Edu2Com instead of the ones formed by the human.

Then we employed Edu2Com to solve the problem of forming teams of university students to work on semester projects. We pitched Edu2Com against the current teachers' practices used to form teams. The results showed that our approach formed better teams in terms of exhibiting fewer failures (teams receiving marks below the borderline) and higher marks. Moreover, we observed that teams with higher collegiality achieved higher performance (i.e., they received higher marks). Notably, the educational authorities expressed interest in using AI tools to form student teams.

Finally, we tasked Edu2Com to form teams of postgraduate students to work on a short-term activity related to their master's program. This empirical evaluation investigated the relationship between a team's performance and collegiality. We pitched teams formed by Edu2Com against randomly formed ones and stable teams, i.e., teams with students working together the whole academic year. The teams' performance was captured through the marks the teams' received. The results showed that higher collegiality results in better performance. Specifically, teams formed by Edu2Com exhibited higher collegiality than the randomly formed ones. At the same time, teams with higher collegiality exhibited better performance. Notably, we observed that experience gained through previous collaborations can significantly boost performance.

Chapter 6

Explainable Team Formation

Over the past decade, there has been a resurgence in explainable artificial intelligence. As more and more complex procedures are being automated with the aid of artificial intelligence, the need for humans to understand the rationale behind AI decisions becomes imperative. Adequate explanations for decisions made by an intelligent system help describe how the system works and earn users' trust. At the same time, recent legislation regarding data privacy, such as the General Data Protection Regulation (GDPR) within the EU or the California Consumer Privacy Act (CCPA) in the USA, highlighted the "right to explanation". That is, the users have the right to know how the several AI systems use their data.

In this thesis, we work towards developing artificial intelligence tools to aid the team formation problem. As such, in this chapter, we make headway towards explaining team formation algorithms. In particular, we propose a general methodology for explaining why certain teams are formed and others are not by a team formation algorithm (TFA). We introduce an algorithm that wraps up any existing team formation algorithm and builds explanations regarding the teams formed by such algorithms. Notably, this is done without modifying the team formation algorithm in any way. Moreover, we turn our attention to a vital challenge regarding explanations. Specifically, in this chapter, we also work towards privacy-aware explanations. As such, we put forward a privacy breach detector, i.e., we provide the means to analyse whether an explanation leads an explainability algorithm to incur privacy breaches when computing explanations for a user. Additionally, we propose a general framework that describes how our privacy breach detector interacts with a team formation algorithm (AI system) and an explanatory algorithm (XAI system) to approve or disapprove explanations.

In what follows, Section 6.1 discusses the need for providing explanations along with the challenges we need to tackle and presents our motivation towards explaining team formation scenarios. In Section 6.2, we introduce a general method for building contrastive explanations within team formation scenarios. In Section 6.3, we empirically evaluate the explanations built with our algorithms. The results showed that our contrastive explanations are easy to understand, requiring just the reading level of a high-school student. Finally, Section 6.4 addresses the challenge of preserving privacy when explaining team formation scenarios, where we introduce a general framework combining *(i)* team formation solutions, *(ii)* explanations for teams and *(iii)* a theory of mind that reasons over explanations to detect privacy breaches.

6.1 Motivation

In an era where artificial intelligence can be practically found in any system, it is increasingly common for people to make decisions guided by the suggestions and recommendations of some intelligent system. As these systems support everyday life decisions, they unavoidably make people curious about their functionality. Often, users question the rationale of their AI systems, bearing a feeling of 'distrust' with machines. Explainable artificial intelligence (XAI) [Miller, 2018, Arrieta et al., 2020] aims at alleviating this distrust by providing answers to questions like "How does this machine learning model operate?" or "Why did the AI system reach this decision?". Over the past years, XAI has attracted much attention, primarily focusing on explaining classification and machine learning (ML) models [Lipton, 2018, Adadi and Berrada, 2018, Došilović et al., 2018, Carvalho et al., 2019]. More precisely, in recent years, the AI community has welcomed several explainable algorithms and techniques with significant impacts, such as: the *LIME* [Ribeiro et al., 2016], *SHAP* [Lundberg and Lee, 2017] or *ANCHORS* [Ribeiro et al., 2018]. Briefly, regarding these widely used algorithms, LIME builds explanations for any classifier by building an interpretable linear model that approximates a prediction *locally*; SHAP exploits a celebrated game theoretic solution concept to identify the most critical features affecting a prediction; while ANCHORS provides model-agnostic explanations. Apart from general machine learning models, a particular family of ML algorithms that has gained much attention regarding explainability and interpretability is that of recommender systems [McAuley and Leskovec, 2013, Rossetti et al., 2013, Tintarev and Masthoff, 2015, Zhang and Chen, 2020, Kleinerman et al., 2018]. Providing explanations in such systems is especially critical as they need to earn the trust of users so that they accept the recommendations provided.

Recently, [Kraus et al., 2020] argued about the importance of explaining decisions in multiagent environments (xMASE), an area that has received little attention so far. In multiagent environments, agents must make decisions based on their goals and preferences and the actions and decisions of other agents and third parties. [Ramchurn et al., 2021] discusses the necessity of developing *trustworthy by design* systems to facilitate human-AI partnerships where both parties act autonomously, and the authors consider machine-generated explanations as an essential step towards trustworthiness. Explaining the agents' decisions in complex and dynamic environments is essential to ensure transparency, accountability, and trustworthiness. As described by the authors in [Kraus et al., 2020], providing explanations in a multiagent context is challenging. In particular, xMASE requires the following:

1. identifying the technical reasons for a decision,

2. adapting the answer to the different agents' preferences, and

3. deciding what kind of information can be revealed considering privacy and fairness issues.

Each of the above challenges is hard and complex to tackle individually, let alone all together. Moreover, XAI approaches usually ignore the explanations' *social nature* and tend to 'miss' the human factor while evaluating the explanations [Kraus et al., 2020]. This is an essential aspectsin explaining multiagent environments that should be addressed along with the aforementioned challenges.

While in fields like machine learning and recommender systems, there is already extensive (and still growing) research on providing explanations, when considering explainable multiagent environments, there is only a handful of recent works in a few application domains, as stated by [Kraus et al., 2020]. Among them, [Mosca and Such, 2021] introduces a system that recommends a sharing policy (i.e., a policy for sharing posts in online social networks) within multi-user environments that can justify its proposed sharing policies. [Pozanco et al., 2022] proposed a framework for explaining users' unsatisfied preferences in resource allocation scheduling problems where agents express their preferences over resources and time slots. Another work in the context of multiagent systems proposes an explanation scheme for the *multi-agent path finding* problem that justifies the agents' routes [Almagor and Lahijanian, 2020]. A further multiagent system contribution builds explanations to justify winners' selections in voting settings [Boixel and Endriss, 2020].

So far, in this thesis, we have studied the problem of forming human teams to work on some tasks. As we have already discussed, forming teams is relevant to many domain applications (e.g. education [Andrejczuk et al., 2018], industry [Ballesteros-Perez et al., 2012, Gutiérrez et al., 2016], search and rescue [Capezzuto et al., 2020], etc.). Despite the interest in team formation, to the best of our knowledge, the problem of explaining team formation decisions still needs to be addressed. In this chapter, we make headway in this matter. Thus, we propose a general method that allows building explanations for the decisions of team formation algorithms. Importantly, our proposed explanatory algorithm is not designed to explain the decisions of some specific team formation algorithm. Instead, it is a general method that *wraps* a team formation algorithm at hand and calls it to build *contrastive explanations*. Such explanations are motivated by the fact that people expect explanations justifying the decision taken by an AI system compared to some other alternative decision (that was not taken) [Miller, 2018].

We illustrate our method with the team formation problem we discussed in Chapter 4, as we believe is a general model for the problem of forming human teams to tackle several tasks. Nonetheless, the proposed method is not restricted to that specific problem. In other words, our method for building explanations can wrap existing algorithms for *any* team formation problem. To be more precise, as discussed in Section 2.1, in the literature, we can find several variations for the team formation problem, including problems (and respective algorithms) for

(i) forming a single team for tackling a single task [Lappas et al., 2009, Anagnostopoulos et al., 2010, Anagnostopoulos et al., 2012],

(ii) forming a single team for tackling multiple tasks [Crawford et al., 2016],

(iii) forming multiple teams for tackling a single (very same) task [Andrejczuk et al., 2019], and

(iv) forming multiple teams for tackling multiple tasks [Bachrach et al., 2010, Ballesteros-Perez et al., 2012, Capezzuto et al., 2020]).

In this chapter, we introduce a method for building explanations that can accompany team formation algorithms that solve the abovementioned problems *without modifying* the algorithm at hand.

One step further, the recent resurgence in explainable artificial intelligence brought up several challenging issues. [Kraus et al., 2020] identified the key challenges we should address when explaining multiagent environments. Among these key challenges is that of *preserving privacy*. AI systems, in general, use data often provided by the users in order to help them reach some decision. As the data provided may be sensitive or private, Kraus et al. argue that when one provides explanations for an AI system, one should consider that some of the information used shall not be disclosed. This issue is rather important within multiagent environments involving many individuals who provide their data. In this direction, [Goodman and Flaxman, 2017] points out the *right to explanation* as a consequence of recent legislations such as the GDPR in the EU or the CCPA in the USA. Such legislation focuses on protecting people's private data and ensuring that some AI system(s) do not misuse their information. All the above suggest that the right to explanation is limited by the right to privacy. In other words, while individuals have the right to know how their data is used by some AI system, at the same time they have the right to preserve their privacy and, therefore, explanations should avoid any privacy violation.

Hence, this chapter also addresses the challenge of preserving privacy when explaining team formation scenarios. One of the purposes for explaining an AI system is to earn the users' trust, and earning users' trust is rather difficult if the explanation reveals private information to third parties. We argue that it should be guaranteed that an explanation exposes no private data. Towards this, we propose using a *privacy breach detector* that can accompany an explanatory algorithm, assess whether the explanation built reveals some private information, and notify the explanatory algorithm accordingly. Finally, we put forward a general framework that describes the interactions among a team formation algorithm (AI system), an explanatory algorithm (XAI system) and the privacy breach detector (privacy-aware system).

6.2 Building Contrastive Explanations for Multi-Agent Team Formation Problems

In this section, we propose a novel explanatory algorithm that can wrap existing team formation algorithms without modifying them. In Section 6.2.1, we present the general team formation model for task allocations. Section 6.2.2 outlines our novel methodology for building explanations. In Section 6.2.2, we identify query templates that are relevant to team formation, and in Sections 6.2.3 and 6.2.4, we show how to handle the queries. Namely, how to translate and incorporate them into an extended version of the original team formation problem that must be solved to build explanations. Section 6.2.5 describes the building and tailoring of contrastive explanation. Finally, Section 6.2.6 systematically evaluates the quality of the explanations we generate.

6.2.1 A General Team Formation Problem For Task Allocation

Let us begin by introducing the general team formation problem we will refer to in this chapter. We denote with $\mathcal{A} = \{a_1, \cdots, a_n\}$ a set of n agents (with $|\mathcal{A}| = n$), and with $\mathcal{T} = \{\tau_1, \cdots, \tau_m\}$ a set of m tasks (with $|\mathcal{T}| = m$). A team of agents $K \subseteq \mathcal{A}$ corresponds to a subset of agents, who are put together to jointly tackle several tasks in $\mathcal{T}$ (similar to Definition 10). A team formation algorithm, TFA for short, forms such teams—e.g., the linear program and the heuristic algorithm Edu2Com presented Sections 4.3.1 and 4.3.2, respectively, are TFAs. More precisely, a TFA gets as input agents $\mathcal{A}$ and tasks $\mathcal{T}$ and returns a team allocation function that assigns teams to tasks denoted by g. Typically, a TFA forms teams based on several desired features (e.g., a team consists of agents with specific skills [Lappas et al., 2009, Anagnostopoulos et al., 2010, Andrejczuk et al., 2019], agents who are socially coherent [Ballesteros-Perez et al., 2012], agents whose location is close to the task's location [Capezzuto et al., 2020], etc.). The satisfaction of such features determines how good a team is for tackling a task. Let $f_i : 2^{\mathcal{A}} \times \mathcal{T} \rightarrow \mathbb{R}$ be an evaluating function for some feature i, which determines how good a subset of agents is for a task from feature i's perspective. We define the adequacy of matching subset of agents K with task τ with respect to a set of features as an *oracle* function:

$$u(K, \tau, F) \in \mathbb{R}^+ \cup \{0\} \tag{6.1}$$

where $F = \{f_1, f_2, \cdots, f_r\}$ is a set of feature evaluating functions. Using an oracle function is common in several team formation algorithms, e.g., [Bachrach et al., 2010, Ponda et al., 2015, Capezzuto et al., 2020]. As such, a team formation algorithm does not compute the quality of a team for its assigned task. Instead, it consults with the oracle u to obtain such information. For instance, our collegiality function (Definition 21) can be considered as an oracle, which we consult when solving the NOMTMT allocation problem presented in Chapter 4.

As a TFA builds an allocation function, any TFA is driven by the quality value of a whole team allocation as specified by g. Thus, the quality of allocation g is naturally an aggregation over the values $u(g(\tau), \tau, F)$ for all $\tau \in T$, i.e., the quality of g is given by

$$v(g) = \mathop{\mathcal{F}}_{\tau \in T} u\big(g(\tau), \tau, F\big)$$

where $\mathcal{F}$ is some aggregating function. For instance, the aggregation function can be the a-la-Nash product over the values $u\big(g(\tau), \tau, F\big)$ for every team formed according to g, as we did in Section 4.1 when we formalised the NOMTMT allocation problem (see Definition 23).

Moreover, a TFA may handle constraints imposed by the Team Formation Problem (TFP) at hand. Such constraints may refer to: whether an agent can participate in multiple teams, and if so, what is the maximum workload per agent; acceptable team sizes; whether every agent must be part of at least one team; etc. Notice, though, that not all TFAs deal with constraints [Czatnecki and Dutta, 2019, Lappas et al., 2009]. A TFA that does not handle constraints cannot be used if the solution must respect certain constraints; some TFAs can handle only specific types of constraints (e.g., each agent must be assigned to exactly one task) [Präntare and Heintz, 2018, Anagnostopoulos et al., 2010, Anagnostopoulos et al., 2012], others can solve a wide variety of constraints [Capezzuto et al., 2020, Andrejczuk et al., 2018]. With this in mind, we define our generalised model of team formation problem for task allocation as follows:

Definition 24 (Team Formation Problem for Task Allocation (TFP-TA))**.** *A Team Formation Problem for Task Allocation p is represented by a tuple $\langle \mathcal{A}, \mathcal{T}, F, u, C\rangle$, where $\mathcal{A} = \{a_1, \cdots, a_n\}$ is a set of agents; $\mathcal{T} = \{\tau_1, \cdots, \tau_m\}$ is a set of tasks; $F = \{f_1, f_2, \cdots, f_r\}$ is a set of feature evaluating functions; u is the oracle determining the suitability of a subset of agents $K \subseteq 2^A$ for a task $\tau \in T$ across all features in F; and $C = \{c_1, c_2, \cdots, c_s\}$ is a possibly empty set of linear constraints (with the understanding that a problem with $C = \{\emptyset\}$ is equivalent to an unconstrained problem).*

As mentioned above, there are many instances and variations of the team formation problem for task allocation, and therefore there are several corresponding algorithms (TFAs) [Bachrach et al., 2010, Ballesteros-Perez et al., 2012, Andrejczuk et al., 2019, Czatnecki and Dutta, 2019, Capezzuto et al., 2020, Präntare and Heintz, 2020].

Here, we propose building explanations for problem instances of the TFP-TA defined above without focusing on any particular TFA. In other words, given an instance of a TFP-TA, we can use any available TFA that can solve the problem, and we will be able to build contrastive explanations for a rich set of questions about the solution found. Moreover, this is without modifying the TFA. Before proceeding with the explanatory algorithm, let us present an example we will use throughout the chapter.

Example 1. *In an artificial intelligence course in a university, a professor must divide their 20 students into teams of size* 4 *for them to work on their semester projects. Each semester project shall cover a different topic of the course. The professor offers five projects: building a pathfinder agent-squad to explore unknown planets in a simulation* (pathfinder), *building competitive agents to play chess using reinforcement and Q learning* (chess), *building agents to solve SUDOKU problems using probabilistic inference and probabilistic graphical models* (sudoku), *building trading agents to play the game "Shelters of Catan (SoC)"* (trading), *and building competitive agents simulating electrical power producers and consumers* (energy). *While every student must be part of exactly one team, ideally, the professor would like to compose teams such that:* (i) *the members of each team have complementary skills to tackle their assigned semester project;* (ii) *the members of each team have diverse personalities; and* (iii) *students' preferences over projects are satisfied as much as possible. In this case, the TFP-TA would be* $p = \langle \mathcal{A}, \mathcal{T}, F, u, C\rangle$, *where:*

- $\mathcal{A} = \{a_1, a_2, a_3 \cdots, a_{20}\}$ *is a set of 20 students;*
- $T = \{Pathfinder, Chess, Sudoku, Trading, Energy\}$*;*
- $F = \{f_{skills}, f_{personality}, f_{preferences}\}$ *are the desired features since the value of a team working on a semester project depends on the students' complementary skills for the project* (f_{skills}), *their balanced personalities* ($f_{personality}$), *and their interests in the project* ($f_{prefernces}$)*;*
- *u is the oracle determining the quality of a team for a given task; and*
- $C = \{team\ size\ is\ 4, each\ student\ takes\ part\ in\ a\ single\ team\}$ *are the constraints to fulfil.*

6.2.2 The Explanatory Algorithm

In this section, we outline the algorithm wrapper to explain team membership and assignments of teams to tasks. Our algorithm builds *contrastive explanations*, i.e., explanations that contrast facts and foils. That is, our explanations justify why the team formation algorithm reached one solution (fact) instead of another alternative solution (foil). The alternative solution derives from the query questioning the solution originally reached by the TFA at hand. As we will detail later, a question q results in a set of additional constraints C_q, referred to as *query-constraints*, that indicate the alternative solution the explanatory algorithm shall contrast against the original solution. Therefore, in order to explain question q, given a TFP-TA $p = \langle A, T, F, u, C\rangle$ and a TFA that can solve p, we first need to *wrap* the TFA so that it can now deal with the extra constraints C_q. With TFA(u), let us denote the team formation algorithm that solves TFP-TA p by consulting the oracle u. We wrap the TFA so that it handles the query-constraints, by wrapping the oracle u. We will explain later on how this wrapper function is built. For now, let us note the wrapped TFA as TFA($\tilde{u}$) and see the definition of a Query-Compliant TFP-TA.

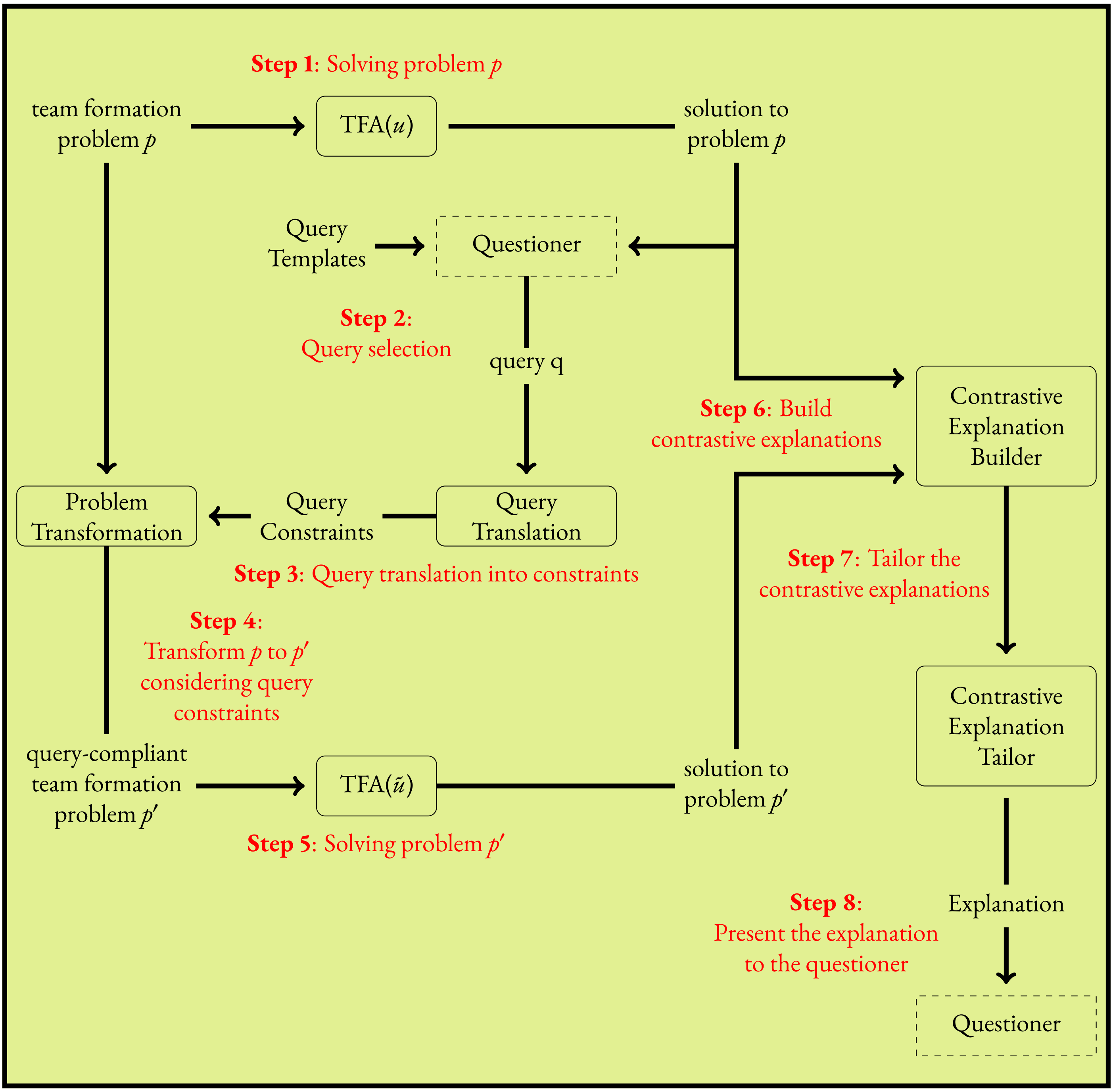


Figure 6.1: The Explanatory Algorithm.

Semester Project	Original (g)	Query-Compliant (g')
Pathfinder	{Alex, Ann, **Beth**, **Bob**}	{Alex, Ann, **Daniel**, **Fedra**}
Chess	{Cynthia, **Daniel**, Edward, **Fedra**}	{Cynthia, **Beth**, Edward, **Wiliam**}
Sudoku	{Helena, Ian, John, Jack}	{Helena, Ian, John, Jack}
Trading	{Kate, Martha, Roger, Stefani}	{Kate, Martha, Roger, Stefani}
Energy	{Suzan, Tania, Victor, **William**}	{Suzan, Tania, Victor, **Bob**}

Table 6.1: Original Allocation (g) vs Query-compliant (g') Allocation

Definition 25 (Query-Compliant TFP-TA (QTFP-TA))**.** *Given a Team Formation Problem for Task Allocation* $p = \langle A, T, F, u, C\rangle$*, and a question* q*, we define a Query-Compliant TFP-TA (QTFP-TA) as* $p' = \langle A, T, F, u, C, \tilde{u}, C_q\rangle$ *where* $\tilde{u}$ *is the wrapping function of* u *that satisfies the constraints* C_q *derived from question* q*.*

Our explanatory algorithm follows the steps illustrated in Figure 6.1 and described below:

1. The TFA(u) algorithm solves problem p and yields a team allocation g.

2. A questioner makes a query q regarding team allocation g.

3. Query q is translated into a set of query-constraints C_q.

4. A problem transformation process combines the original problem p with the query constraints in C_q to produce a *Query-Compliant* TFP-TA p'.

5. The TFA($\tilde{u}$) solves the QTFP-TA p' and outputs a query-compliant team allocation g'.

6. A builder of contrastive explanations compares the original team allocation (g) with the query-compliant allocation (g') to analyse their differences and generate explanations.

7. Finally, the contrastive explanations are tailored to highlight different perspectives and passed back to the questioner.

Next, we go through the steps above in more detail to explain the whole explanatory algorithm. However, further discussion follows in Sections 6.2.3-6.2.6. To begin with, given a problem TFP-TA $p = \langle \mathcal{A}, \mathcal{T}, F, u, C\rangle$, the TFA($u$) solves the problem and produces a team allocation g. Thereafter, a questioner questions the allocation g. That is, a questioner may ask, for instance, why a particular team was assigned to a task or why a questioner-made team was not assigned to the task. Following Example 1, consider that the TFA(u) outputs the team allocation depicted in the second column of Table 6.1, the one labelled as *Original*.

The professor studies the allocation and, acting as questioner, places a question regarding g: for instance, "Why is Bob assigned to the pathfinder project?". In Section 6.2.3, we introduce a number of query templates that focus on team formation problems. Our explanatory algorithm deals with such questions by providing contrastive explanations [Miller, 2018]. Contrastive explanations are based on findings in the philosophical and cognitive sciences literature indicating that people are not interested in the causes leading to a particular outcome (in our case, an allocation) *per se*, but, on the contrary, they are interested in the causes that explain a non-occurring outcome. In other words, people are interested (and tend to give) explanations regarding questions of the type "Why X instead of Y?". For instance, in our running example, the professor is interested in the causes that led TFA(u) to assign Bob to the Pathfinder project instead of assigning him to a different project.

To build such a contrastive explanation, we need to compute the *differences* between the original allocation produced by the TFA(u) and another alternative allocation, determined by the particular question of the questioner, produced by the TFA($\tilde{u}$). Thus, given a question q made by the questioner, the justification algorithm processes this question and proceeds to build an alternative, *query-compliant allocation*. This procedure is implemented by steps 3-5 in Figure 6.1. First, our explanatory algorithm translates the question posed into a set of constraints C_q. For instance, in our example, the question "Why is Bob assigned to the Pathfinder project?" is translated into one constraint that *forbids* Bob to be assigned to the Pathfinder project. Section 6.2.4 thoroughly discusses how to translate queries. This problem p' is a *query-compliant* TFP-TA, that is, p' is an extension of the original problem p with the query-constraints C_q and the wrapper function $\tilde{u}$.

Back to our example, problem p' would split the 20 students into teams of size 4, though constrained to assigning each team to exactly one semester project and ensuring that every student participates in exactly one team (likewise the original problem p). However, p' must consider an extra constraint: "Bob must not be assigned to the Pathfinder project". The QTFP-TA p' is then solved by TFA($\tilde{u}$). The output is a query-compliant allocation g' that respects the query constraints. For instance, in the allocation depicted along the third column in Table 6.1, Bob is assigned to the Energy project. Note that the query-compliant allocation is the best allocation that TFA($\tilde{u}$) could find while fulfilling the query constraints. Once our explanatory algorithm gets the original and the query-compliant allocations (g and g', respectively), it is ready to build a contrastive explanation by computing the differences between the two.

As soon as the contrastive explanation builder finds these differences (step 6 in Figure 6.1), the generated explanations go through a *tailoring process*. This process highlights different points of view. Specifically, as discussed in Section 6.2.6, we focus on different levels of abstraction in order to provide the questioner with a suitable explanation. In our example, our explanatory algorithm would generate three different types of explanations:

(a) one referring specifically to Bob;

(b) one referring to teams and their assigned tasks; and

(c) one referring to the allocation as a whole.

The following sections detail the main processes of the algorithm: Section 6.2.3 identifies several meaningful questions regarding team formation, Section 6.2.4 shows the translation process of these queries into query constraints, Section 6.2.5 transforms the initial TFP-TA to accommodate such constraints, and Section 6.2.6 elaborates on how to build contrastive explanations.

6.2.3 Identifying User Queries

Given a team formation scenario, a questioner may ask several questions regarding the allocation computed to solve a team formation problem. Here we identify a collection of intuitive and meaningful questions that cover the main points of interest regarding team formation scenarios. In Tables 6.2 and 6.3, we list the collected questions as query templates. There, we distinguish two types of queries:

Collaboration Queries question the established collaboration between agents. Thus, they question the teams formed while disregarding their assignments. Queries of this type consider the teams formed in a given allocation, focusing on the complete team (see queries Q8, Q9 in Table 6.3), or on individual agents (see queries Q10, Q11 in Table 6.3). This type of query also includes questions about the participation of specific agents in a team (see queries Q7, Q12 and Q13 in Table 6.3).
Assignment Queries challenge the assignment of tasks to teams and individuals. This type of query concerns the assignment of a complete team to a specific task (see queries Q3, Q4 in Table 6.2) or the assignment of certain agents to a specific task (see queries Q1, Q2, Q5 and Q6 in Table 6.2).

Following our running example 1, the question "Why is Bob assigned to the Pathfinder project?" is an assignment query questioning the assignment of Bob to a specific project, the Pathfinder project. The question "Why is team K = {Cynthia, Daniel,Edward,Fedra} not assigned to the Sudoku project?" is also an assignment query questioning the assignment of the complete team K to a specific project, and therefore the assignment of each student in K to that project. Alternatively, question "Why is student Cynthia in team K = {Cynthia, Daniel,Edward,Fedra}?" is a collaboration query questioning the participation of Cynthia in team K, and therefore, the collaboration of student Cynthia with each of the students in K.

After identifying these relevant team formation query templates, we show how to translate each of these queries into query constraints in the following section.

Code	Query Template	Query Type	Query Constraints	Constraints per *query-compliant* Allocations	Number of *query-compliant* Allocations
Q1	Why is agent a_i assigned to task τ?	Assignment	VETO assigning a_i to τ	1	1
Q2	Why is agent a_i not assigned to task τ?	Assignment	ENFORCE assigning a_i to τ	1	1
Q3	Why is team $K = \{a_1, \cdots a_{\lvert K \rvert}\}$ assigned to task τ?	Assignment	VETO assigning a_1 in τ **OR** VETO assigning a_2 in τ **OR** $\ldots$ VETO assigning $a_{\lvert K \rvert}$ in τ	1	|K|
Q4	Why is team $K = \{a_1, \cdots a_{\lvert K \rvert}\}$ not assigned to task τ?	Assignment	ENFORCE assigning a_1 in τ **AND** ENFORCE assigning a_2 in τ **AND** $\ldots$ ENFORCE assigning $a_{\lvert K \rvert}$ in τ	1	|K|
Q5	Why is agent a_i assigned to task τ, instead of a_j?	Assignment	VETO assigning a_i in τ **AND** ENFORCE assigning a_j in τ	2	1
Q6	Why is agent a_i assigned to task τ, while a_j is not?	Assignment	VETO assigning a_i in τ **OR** ENFORCE assigning a_j in τ	1	2

Table 6.2: Assignment query templates for a general team formation problem.

6.2.4 Query Translation

As mentioned in Section 6.2.2, we translate the queries posed by a questioner into constraints in order to compute an alternative allocation based on the query and thereafter build a contrastive explanation. Complying with a query constraint is necessary to compute an alternative allocation corresponding to a 'what-if' scenario, an alternative scenario. Query constraints are *hard constraints* and must be met when solving the new team formation problem that arises by imposing the query constraints. Looking at the query templates in Tables 6.2 and 6.3, we observe that a (collaboration or assignment) query poses:

(i) why a specific collaboration or assignment is established in a given allocation,*or*

(ii) why a specific collaboration or assignment is not established.

Such types of queries are then translated into one of two types of constraints:

(i) *veto constraints* that capture queries that question an established team or assignment, and

(ii) *enforcement constraints* that capture queries regarding why a team or assignment did not occur.

Consider the question asked by the professor in our running example, "Why is Bob assigned to the Pathfinder project?" which is an instance of query template Q1. To provide an explanation, we must find an allocation for the alternative what-if scenario: "What would happen if Bob were not assigned to the Pathfinder project?". By adding the appropriate constraint, we can compute alternative allocations suitable for answering the query posed by the professor. Thus, when a query template targets a team or an assignment established in the original allocation, it generally translates into veto constraints on the team or assignment. This translation is necessary to describe a meaningful what-if scenario and build meaningful contrastive explanations.

Alternatively, when a query template targets a team or assignment that did not occur in the original allocation, the query translates into enforcement constraints so that the query-compliant allocation contains the team or the assignment described in the query. For instance, consider the question "Why is team $K = \{a_1, a_3, a_5, a_7\}$ not formed", as an instance of query template Q9. The what-if scenario to consider would be: "What would happen if team K was formed?". Hence, replying to this query demands computing an allocation that enforces the formation of team K.

Depending on the query template, and more precisely on how many agents are involved in a query, the queries are translated into *(a)* a single constraint, or *(b)* a set of constraints. Queries translated into a single constraint are referred to as *simple queries* and involve a pair $\langle$agent, task$\rangle$ (Q1, Q2) or two agents (Q10, Q11). Regarding the queries translated into a set of constraints, we refer to them as *complex queries*, and we differentiate three translation patterns for them:

Code	Query Template	Query Type	Query Constraints	Constraints per *query-compliant* Allocations	Number of *query-compliant* Allocations
Q7	Why is agent a_i in team $K = \{a_1, \cdots, a_{\|K\|}\}$?	Collaboration	VETO collaborating a_i with a_1 **OR** VETO collaborating a_i with a_2 **OR** $\cdots$ VETO collaborating a_i with $a_{\|K\|}$	1	|K|-1
Q8	Why is team $K = \{a_1, \cdots, a_{\|K\|}\}$ formed?	Collaboration	VETO collaborating a_1 with a_2 **OR** VETO collaborating a_1 with a_3 **OR** $\cdots$ VETO collaborating a_1 with $a_{\|K\|}$ **OR** $\cdots$ VETO collaborating $a_{\|K\|-1}$ with $a_{\|K\|}$	1	$\binom{\|K\|}{2}$
Q9	Why is team $K = \{a_1, \cdots, a_{\|K\|}\}$ not formed?	Collaboration	ENFORCE collaborating a_1 with a_2 **AND** ENFORCE collaborating a_1 with a_3 **AND** $\cdots$ ENFORCE collaborating a_1 with $a_{\|K\|}$ **AND** $\cdots$ ENFORCE collaborating $a_{\|K\|-1}$ with $a_{\|K\|}$	1	$\binom{\|K\|}{2}$
Q10	Why are a_i and a_j in the same team?	Collaboration	VETO collaborating a_i with a_j	1	1
Q11	Why are a_i and a_j not in the same team?	Collaboration	ENFORCE collaborating a_i with a_j	1	1
Q12	Why is agent a_i in team $K = \{a_1, \cdots, a_{\|K\|}\}$, instead of agent a_x?	Collaboration	ENFORCE collaborating a_x with a_1 **AND** $\cdots$ ENFORCE collaborating a_x with a_{i-1} **AND** ENFORCE collaborating a_x with a_{i+1} **AND** $\cdots$ ENFORCE collaborating a_x with $a_{\|K\|}$ **AND** VETO collaborating a_x with a_i **AND** VETO collaborating a_x with a_y, $\forall a_y \notin K$	|A|	1
Q13	Why is agent a_i in team $K = \{a_1, \cdots, a_{\|K\|}\}$, instead of agent a_x?	Collaboration	ENFORCE collaborating a_x with a_1 **AND** $\cdots$ ENFORCE collaborating a_x with a_{i-1} **AND** ENFORCE collaborating a_x with a_{i+1} **AND** $\cdots$ ENFORCE collaborating a_x with $a_{\|K\|}$ **OR**	|K|-1	2
			VETO collaborating a_x with a_i **AND** VETO collaborating a_x with a_y, $\forall a_y \notin K$	|A|-|K|-1	

Table 6.3: Collaboration query templates for a team formation problem.

(i) conjunction of constraints (Q4, Q5, Q9, Q12), which deal with a team formation or assignment that did not occur;

(ii) disjunction of constraints (Q3, Q6, Q7, Q8), which focus on an established team formation or assignment; and

(iii) disjunction of conjunctions (Q13) considers both an established team and one that did not occur.

6.2.5 Problem Transformation

This section shows how to transform the original problem TFP-TA into a query-compliant one QTFP-TA by wrapping the oracle function u. The problem transformation process is based on defining the wrapper function $\tilde{u}$ and thus changing the evaluation function that determines the quality of the assignment of a team to a task, depending on the query constraints. In more detail, given a query q, we incorporate its query constraints C_q in the oracle function by imposing large penalties when a team-task pair violates a query constraint. The amount of the penalty depends on the function that aggregates the quality of each team-task allocation. For instance, if the aggregation is the product of the individual qualities, then the penalty would be zero when a query constraint is violated. The penalty would be a large negative value if the aggregation is based on adding qualities. In the equations below, we assume we have a product aggregation as an example. In other words, the new, query-compliant problem p' is given by the tuple $\langle A, T, F, u, C, \tilde{u}, C_q \rangle$ (see Definition 25) where:

$$\tilde{u}(K, \tau, F, C_q) = \psi(K, \tau, C_q) \cdot u(K, \tau, F) \tag{6.2}$$

and

$$\psi(K, \tau, C_q) = \begin{cases} 0, & \text{if } \langle K, \tau \rangle \text{ violates a constraint in } C_q \\ 1, & \text{otherwise} \end{cases} \tag{6.3}$$

Thus, when TFA($\tilde{u}$) is solving problem p', allocations that violate a query constraint are penalised and therefore avoided, since the allocation's quality depends on the quality of all teams for their assigned tasks—i.e., $v(g) = \mathcal{F}_{\tau \in T}\, \tilde{u}\Big(g(\tau), \tau, F\Big)$.

6.2.6 Building Contrastive Explanations

This section outlines the methodology for building and tailoring contrastive explanations to address queries related to team formation. As discussed in Section 6.2.2, a contrastive explanation involves comparing an initial allocation (g) with an alternative, query-compliant allocation (g') that satisfies a what-if scenario resulting from the query at hand. The explanation highlights the factors favouring the initial allocation over the alternative one. The initial and alternative allocations are compared at three levels of abstraction, referred to as the *explanation views*. Each explanation view provides a distinct perspective for explaining why the initial allocation is preferable to the alternative allocation. The three explanation views are:

- the *individual view (IV)*, which focuses on the agents identified in the query;
- the *local view (LV)*, which focuses on the individual tasks and the teams assigned to them; and
- the *global view (GV)*, which evaluates the overall quality of the allocations.

For each view, we compute the *relative differences* between the original allocation and the query-compliant one. Such differences quantify the gains or losses of one allocation with respect to the other. That is, we compute Δf_i^{IV}, Δf_i^{LV}, Δf_i^{GV} for each feature evaluation function $f_i \in F$. Specifically, we compute relative differences considering the oracle u in the individual and the local view, while in the global view, we compute relative differences considering the aggregation function $\mathcal{F}$. Notice that in this step, we need access to *(a)* the oracle function u, and *(b)* the aggregating function $\mathcal{F}$. However, the oracle function is accessible by our explanatory algorithm (through the wrapper $\tilde{u}$); while the aggregating function is known for a given TFA—remember that for our wrapper to impose the proper penalty during the problem transformation step, our explanatory algorithm must be aware of the aggregation function $\mathcal{F}$.

Moreover, we make the following assumption to compute the relative differences for the individual view. There is a way to compute the contribution of a single agent a to its team $g(\tau)$ (denoted as $|_{a \in g(\tau)}$) concerning *(i)* each feature evaluation function f_i, and *(ii)* the value yielded by the oracle u. Even though this may seem a strong assumption, it is common in the literature to employ oracles that allow such computation (e.g. [Czatnecki and Dutta, 2019, Andrejczuk et al., 2018]). In our empirical evaluation, we illustrate the use of one general oracle that allows us to compute relative differences. Given an original allocation g, a query-compliant allocation g', a task *tau* and an agent a, we compute relative differences as follows:

$$\textbf{Individual View:} \qquad \Delta f_i^{IV}(a, g') = \frac{f_i(g(\tau), \tau)|_{a \in g(\tau)} - f_i(g'(\tau'), \tau')|_{a \in g'(\tau')}}{f_i(g(\tau), \tau)|_{a \in g(\tau)}}$$

$$\textbf{Local View:} \qquad \Delta f_i^{LV}(\tau, g') = \frac{f_i(g(\tau), \tau) - f_i(g'(\tau), \tau)}{f_i(g'(\tau), \tau)}, \quad \forall \tau \in T$$

$$\textbf{Global View:} \qquad \Delta f_i^{GV}(g') = \frac{\mathcal{F}_{\tau \in T} f_i(g(\tau), \tau) - \mathcal{F}_{\tau \in T} f_i(g'(\tau), \tau)}{\mathcal{F}_{\tau \in T} f_i(g(\tau), \tau)}$$

where f_i is some feature evaluation function.

	Original Allocation g			**Query-Compliant Allocation** g'		
	f_{skills}	$f_{personality}$	$f_{prefernces}$	f_{skills}	$f_{personality}$	$f_{prefernces}$
Bob	0.67	-	0.9	0.44	-	0.75
Pathfinder	0.83	0.62	0.8	0.33	0.65	0.72
Chess	0.87	0.72	0.57	0.6	0.28	0.5
Sudoku	0.78	0.9	0.71	0.78	0.9	0.71
Trading	0.93	0.82	0.92	0.93	0.82	0.92
Energy	0.66	0.55	0.5	0.7	0.41	0.55
Overall	0.3457	0.1812	0.1489	0.1005	0.0551	0.1293

Table 6.4: Running Example: Contributions across each feature of *(i)* Bob, *(ii)* each team with respect to their corresponding task, and *(iii)* the overall allocation.

Explanation View		f_{skills}	$f_{personality}$	$f_{preferences}$
IV	Bob	-34.33%	-	-16.67%
LV	Pathfinder	-60.24%	4.84%	-10.00%
	Chess	-31.03%	-61.11%	-12.28%
	Sudoku	0.00%	0.00%	0.00%
	Trading	0.00%	0.00%	0.00%
	Energy	6.06%	-25.45%	10.00%
GV	*overall*	-70.92%	-69.61%	-13.16%

Table 6.5: Running Example: Relative differences per explanation view.

Back to our running example, assume that after compting the query-compliant allocation g', we measure the contribution of Bob to each assigned team and task across all features of interest ($F = \{f_{skills}, f_{personality}, f_{preferences}\}$) both the original allocation g and the alternative one g', as shown in Table 6.4. Similarly, we measure the evaluation across all features of each team formed either in g or g' and the evaluation of the overall allocation. Then we can compute the relative differences used in each explanation view, as illustrated in Table 6.5.

Explanation Template. Up to this point, we have discussed how to compute an alternative, query-compliant solution and the differences between this solution and the original one, i.e., how to compute the *reasons* why the original solution is more preferred than an alternative one. However, the explanations are built to address humans' doubts towards the AI algorithm at hand. As pointed out in [Kraus et al., 2020], explanations built within XAI (and XMASE) usually neglect the human factor. That is, even if by computing the differences between the two solutions, we conclude the reasons that explain why the AI algorithm decided on the original solution rather than the alternative one, we still need to *present* these reasons to the user. To do so, we use a *natural language template*. In other words, we pre-define a template per explanation view. We tailor the explanations by filling out a template according to *(i)* the computed relative differences and *(ii)* the explanation view.

Consider our running example again. The professor asked, "Why is Bob assigned to the Pathfinder project?" and our algorithm built an alternative, query-compliant allocation g' that assigns Bob to a project other than Pathfinder, as shown in Table 6.1. Given the relative differences shown in Table 6.5, our explanation builder generates one explanation per view as follows:

"If Bob was not assigned to the Pathfinder project, then...

IV: "Bob would have had to participate in the Energy project, for which they are less skilled."

LV: "40% of the tasks would have been assigned to less-skilled (up to 60.24%) teams, and 20% of the tasks would have been assigned to less-compatible (up to 25.45%) teams, while 40% of the tasks would have been assigned to equally skilled, compatible and satisfied teams."

GV: "The overall matching of teams to tasks would be 70.92% less-skilled, 69.61% less-diverse in terms of personality, and 13.16% less-satisfying. Thus, the alternative allocation would be 30.35% less-suitable considering all features".

The explanation example above is built by first computing the relative differences and then filling out an explanation template in natural language. The explanation template is generally common to all queries—changes only for the IV. The templates can be found in Appendix III .

6.3 Empirical Evaluation of Explanations

This section evaluates the quality of the explanations that our algorithm generates. First, Section 6.3.1 specifies the instance of the TFP-TA selected for our experiments, along with the TFA of choice. Then, Section 6.3.2 introduces our evaluation metrics for explanations, some adapted from metrics used in the ML literature. Finally, Section 6.3.3 reports the results of our evaluation.

6.3.1 Team Formation Problem, Algorithm and Datasets

Team Formation Problem For our empirical evaluation, we choose the team formation problem that we have been discussing throughout this dissertation. That is, we consider the problem of forming non-overlapping teams that tackle one task each, and in turn, each task is tackled by only one team, i.e., the NO-MTMT allocation problem introduced in Chapter 4.

Moreover, we choose the same features as described in the running example of Section 6.2, namely: skills, personality, and agents' preferences over tasks ($\{f_{\text{skills}}, f_{\text{personality}}, f_{\text{preferences}}\}$). Each task specifies the skills required by a team to perform it. We also adopt the typical team-size constraint per task (e.g. [Andrejczuk et al., 2018]), i.e., each task requires a team of a specified number of members. We use the metrics introduced in Chapter 3.3 as feature evaluation functions. Specifically, as f_{skills} we use the competence affinity metric (Section 3.4.1), as $f_{\text{personality}}$ we use the congeniality metric (Section 3.4.2), and as $f_{\text{preferences}}$ we use the motivation metric (Section 3.4.3). As oracle u that yields suitability, we define the linear combination of the three:

$$u(K, \tau, F) = \sum_{f_i \in F} w_i \cdot f_i(K, \tau).$$

Using a linear combination of multiple attributes to form a scalar function is a commonly used technique [Horn et al., 1994]. As aggregating function $\mathcal{F}$, we use the product ($\mathcal{F} \triangleq \prod$), since, as claimed by [Andrejczuk et al., 2018, Andrejczuk et al., 2019], a product promotes "balanced" team allocations —according to [Chevaleyre et al., 2006], it favours both increasing the overall team allocation utility and reducing the differences in individual team allocation quality.

Algorithm To solve the problem, we use the linear program encoding introduced in Section 4.3.1, and as a team formation algorithm (TFA), we use the state-of-the-art optimal solver CPLEX [IBM, 2019].

Datasets. For our empirical evaluation, we used synthetic data as instances for the problem of forming teams for task allocation. Specifically, we generated 20 datasets. Each dataset contains 10 tasks, and each task requires a team consisting of 2 to 4 team members—–we took special care that at least one task requires a team of 4 members. Each task requires 3 to 10 different skills, randomly selected out of 475 skills defined in the "European Skill/Competence, qualifications and Occupations" (version 1.0.8).[1] Given a dataset D (out of 20), we denote with Sk_D the set with the different skills required by all 10 tasks in D ($Sk_D \subset Sk_{ESCO}$).

An agent is defined through their competence profile (Def. 1), their personality profile (Def. 2), and their task preference profile (Def. 3). That is, agents count on *(i)* a skill set to tackle a task, *(ii)* a personality profile, and *(iii)* their set of graded preferences over tasks. Each agent in dataset D is equipped with 2 to 5 skills from Sk_D. An agent's personality is expressed by the four personality traits: Sensing–Intuition (SN), Thinking–Feeling (TF), Extroversion–Introversion (EI), and Perception–Judgement (PJ), and specifically by a real value in [1, 1] for each trait [Andrejczuk et al., 2018, Andrejczuk et al., 2019]. As such, we generate a personality profile for each agent by uniformly drawing 4 real numbers in the range [1, 1]. Finally, for each agent, we randomly

[1] `https://esco.ec.europa.eu/en`

select 1 to 10 tasks in the dataset, for which we express a preference degree in $[0, 1]$ (randomly selected).

The queries used in our experimental evaluation focus on tasks requiring a team of size 4 and its formed team. As mentioned above, we take special care that there exists at least one such task in every synthetically generated dataset. Let us denote with τ^{4m} a task requiring a 4-member team, and with K^{4m} the team formed for task τ^{4m} within the original allocation g, i.e., $g(\tau^{4m}) = K^{4m}$. Then, we generate a query per query template following the process below:

Q1: we randomly select an agent a from team K^{4m} (i.e., $a \in K^{4m}$), and generate a question why a is assigned to τ^{4m};

Q2: we randomly select an agent a not in team K^{4m} (i.e., $a \in A \setminus K^{4m}$, with A the set of all agents in the dataset), and generate a question why a is not assigned to τ^{4m};

Q3: we question why team K^{4m} is assigned to τ^{4m};

Q4: we randomly select 4 agents not in team K^{4m} forming team K' (i.e., $K' \subseteq A$ and $K' \cap K^{4m} = \emptyset$); and generate a question why team K' is not assigned to τ^{4m};

Q5: we randomly select an agent a from team K^{4m} (i.e., $a \in K^{4m}$), an agent a' not in team K^{4m} (i.e., $a' \in A \setminus K^{4m}$), and generate a question why a is assigned to τ^{4m} , instead of a';

Q6: we randomly select an agent a from team K^{4m} (i.e., $a' \in K^{4m}$), an agent a' not in team K^{4m} (i.e., $a \in A \setminus K^{4m}$), and generate a question why a is assigned to τ^{4m}, while a' is not;

Q7: we randomly select an agent a from team K^{4m} (i.e., $a \in K^{4m}$), and question why a participates in K^{4m} ;

Q8: we question why team K^{4m} is formed;

Q9: we randomly select 4 agents not in team K^{4m} forming team K' (i.e., $K' \subseteq A$ and $K' \cup K^{4m} = \emptyset$); and generate a question why team K' is not formed;

Q10: we randomly select 2 agents a and a' in team K^{4m} (i.e., $a, a' \in K^{4m}$, with $a \neq a'$), and generate a question why a and a' are in the same team;

Q11: we randomly select an agent a from team K^{4m} (i.e., $a \in K^{4m}$), an agent a' not in team K^{4m} (i.e., $a' \in A \setminus K^{4m}$) and generate a question why a and a' are not in the same team;

Q12: we randomly select an agent a from team K^{4m} (i.e., $a \in K^{4m}$), an agent a' not in team K^{4m} (i.e., $a' \in A \setminus K^{4m}$) and generate a question why a participates in K^{4m}, instead of a'; and finally

Q13: we randomly select an agent a from team K^{4m} (i.e., $a \in K^{4m}$), an agent a' not in team K^{4m} (i.e., $a' \in A \setminus K^{4m}$) and generate a question why a participates in K^{4m}, while a' does not.

6.3.2 Evaluation Metrics

We have singled out several *off-line* evaluation metrics aligned with the existing literature to evaluate the quality of explanations.

Number of features (NOF). Considering the number of features that appear within an explanation is a commonly used metric in many explainable models. Indeed, since most XAI models attempt to give insights on the functionality and the rationale of a black box (e.g., ML models), it is common to use the number of features used to explain a decision as a quality index for explanations [Zhang and Chen, 2020, Rosenfeld, 2021]. However, this metric is not restricted to explanatory systems related to ML models; instead, non-ML explanatory systems have considered the number of features as a quality index, as well. For instance, when justifying election winners in [Boixel and Endriss, 2020], the number of axioms that back up the elected winner is used to evaluate the quality of an explanation. The number of features can also be seen as the number of causes [Miller, 2018] displayed in an explanation, with the general guideline that good explanations are the simple ones containing a relatively small number of causes. Here, we consider the number of features that exhibit a utility decrease in the query-compliant allocation compared to the original allocation.

Mean explainability precision (MEP) [Abdollahi and Nasraoui, 2017, Mohseni et al., 2020]. Explainability precision—resembling the corresponding metric in information retrieval, classification, and ML in general— is the proportion of explainable items in a list relative to the total number of items. Regarding team formation, we measure MEP in terms of the percentage of agents, tasks, and attributes for which we can explain why the alternative query-compliant allocation is worse than the initial allocation.

Gunning Fog readability (GFR) index [Gunning, 1952]. Since we produce explanations in natural language, we also use the *Gunning Fog readability (GFR)* index [Gunning, 1952], a well-known readability metric in the literature. The GFR considers *(a)* the proportion of "complex words" relative to the total number of words; and *(b)* the number of words per sentence. The resulting score indicates the reading level needed to comprehend the text by grade.

6.3.3 Results

Our empirical evaluation employs 20 synthetic instances of the TFP-TA in Section 6.3.1 with 10 tasks and $\sim$ 26.5 agents each. Note that we can quickly compute relative differences for IV, since the oracle u in Section 6.3.1 allows us to compute the contribution to a team per agent at a low computational cost. We solved each one of the problem instances with an LP solver to obtain their *original* allocations.

After that, we generated one query for each query template and for each original allocation so that each allocation was artificially *challenged* by one query of each one of the thirteen query templates. We generated queries depending on the query template as follows. First, we selected an original allocation. Second, we randomly selected a team (K) in the allocation. Third, we built one query per query template by randomly selecting: agents from K (for templates Q1, Q7, Q10); agents from $A \setminus K$ (templates Q2, Q10); one agent from K and one from $A \setminus K$ (templates Q5, Q9, Q11, Q12, Q13). Finally, we handed the queries to our algorithm to compute explanations. We evaluated their quality using the metrics in Section 6.3.2. Tables 6.6-6.8 compile our results per metric grouped per query template, which we discuss below comparing explanation views.

Average number of features (NOF). Table 6.6 illustrates the average number of features ($\{f_{\text{skills}}, f_{\text{personalitt}}, f_{\text{preferences}}\}$) used to explain why the original allocation is preferable to the query-compliant one. That is, these are the average number of features per explanation that: *(a)* exhibit a relative gain, and *(b)* are part of the textual explanation. The results tell us that the individual view uses fewer features than the local view, which in turn uses fewer features than the global view when building an explanation for each query. Moreover, explanations of simple queries (Q1,Q2, Q7,Q8, Q10 or Q11) use fewer features compared to more complex ones.

Mean explainable precision (MEP). The results are shown in Table 6.7. As we can see, the GV reaches higher MEP (at least 67%) than the LV and the IV. That is, with the GV explanation, we can easily explain why the original allocation is preferred to the query-compliant one. This is because the TFA's goal is to optimise the overall allocation function; hence, the GV aligns with the algorithm's point of view. The LV reaches low MEP (below 50%) since many tasks are not affected by the query and, therefore, cannot explain why one allocation is preferred to the other. Finally, the IV exhibits a wide variability in MEP since the IV is highly query-dependent. Thus, the GV is more precise for explaining team formation.

Gunning Fog readability index. In Table 6.8, we observe that 92% (36 out of 39) of our explanations achieve scores between 8 (reading level of eighth grade) and 12 (reading level of a high school senior student) and 6% are very close to 12. Thus, we conclude that our explanations are easy to read and comprehend. In general, we see that simple queries achieve a low GFR index

($\sim$ 8.36), while complex queries achieve a higher GFR index ($\sim$ 10.33). Notably, the LV and GV explanations exhibit a 'stable' reading index (the LV index is always $\sim$ 8.8, and the GV index is $\sim$ 11). This is because, regardless of the query template, the LV always considers the same number of tasks (10 in all problem instances), and the GV considers a constant number of attributes. On the contrary, the readability score for the IV is query-dependent.

Query	Individual View		Local View		Global View	
	Average NOF	Standard Deviation	Average NOF	Standard Deviation	Average NOF	Standard Deviation
Q1	1	0	2.2	0.8	2.7	0.55
Q2	1	0	2.20	0.74	3.00	0.71
Q3	0.91	0.14	1.87	0.47	2.75	0.50
Q4	1.55	0.49	2.8	0.4	3.35	0.57
Q5	1.29	0.45	2.85	0.65	3.05	0.67
Q6	0.58	0.18	2.58	0.48	3.03	0.43
Q7	0.91	0.20	1.88	0.51	2.82	0.55
Q8	0.91	0.22	1.86	0.51	2.84	0.53
Q9	1.41	0.49	3.35	0.47	3.2	0.5
Q10	1.06	0.23	1.95	0.59	2.80	0.68
Q11	1.18	0.58	2.45	0.86	2.90	0.62
Q12	1.60	0.49	2.20	0.92	3.30	0.55
Q13	1.18	0.39	2.73	0.51	3.20	0.66

Table 6.6: Number of Features (NOF) per explanation view per query template.

Query	Individual View	Local View	Global View
Q1	65%	40%	67.5%
Q2	70%	42.5%	73.75%
Q3	91.25%	37.75%	68.75%
Q4	75%	80%	83.75%
Q5	47.5%	54.5%	76.25%
Q6	35%	41.25%	75.63%
Q7	55%	37.17%	70.42%
Q8	55.41%	36.58%	71.04%
Q9	61.67%	73.5%	80%
Q10	50%	38.5%	70%
Q11	35%	49.5%	72.5%
Q12	17.49%	39.5%	82.5%
Q13	48.87%	47.5%	80.62%

Table 6.7: Mean Explanable Precision per explanation view per query template

Query	Individual View	Local View	Global View
Q1	8.37	8.78	10.61
Q2	9.1	8.77	10.84
Q3	7.08	8.53	10.33
Q4	12.4	9.05	11.53
Q5	9.89	8.87	11.42
Q6	8.3	8.76	10.9
Q7	7.47	8.64	10.63
Q8	7.25	8.51	10.45
Q9	10.2	8.91	10.79
Q10	8.4	8.76	10.88
Q11	9.34	8.99	11.8
Q12	14.62	8.75	12.32
Q13	9.82	8.78	11.54

Table 6.8: Gunning Fog Readability index per explanation view per query template.

6.4 Privacy-Aware Explainable Team Formation

The method for building explanations we presented in Section 6.2 does not consider *privacy issues*, one of the key challenges in explainable multi-agent environments (XMASE) as identified by [Kraus et al., 2020]. The explanations built by our method use agents' individual data (profiles) along with information about teams' quality (in terms of, e.g., competence affinity, personality and gender diversity, motivation, etc.). This can result in revealing agents' private information to third parties. For example, an explanation under the individual view (see Section 6.2.6) may disclose information about an individual's competencies or preferences, while an explanation under the local view may disclose information about people's personality. This section aims to "enhance" our building of explanations to account for privacy issues.

As discussed earlier, explainable AI provides "inside information" regarding the inner functionality of an AI system in an attempt to be transparent and earn, in this way, the users' trust. More and more applications turn to AI to ease and automate complex procedures and demand an understanding of the solutions recommended by such systems. Besides the growing need for explanations, as pointed out by [Goodman and Flaxman, 2017], legislation such as the GDPR recently put forward by the EU leads to the *right to explanation*. That is, a user providing personal information as input data to some AI algorithm has the right to know why the algorithm makes a particular decision with their input data instead of another decision.

Note that in any AI system that assists people in making a decision or solving a problem, individuals need to feed the system with information (possibly private), which is therefore utilised by the system to reach a solution. For people to use an AI system and feed it with their data, the system must ensure that it works towards its users' best interest and does not misuse their data. In this direction, explainable AI sheds light on the practices used by the system at hand and shares internal information with the users. However, information and data shall not be shared lavishly, especially in environments involving many individuals. Instead, users' data must be treated with care when explanations are provided, and the AI system must guarantee that private information remains private and is not disclosed to third parties.

In this context, [Puiu et al., 2021] present recent developments on explainability and interpretability along with the limitation of data accessibility due to ethical constraints in cardiovascular diagnosis. [Sokol and Flach, 2019] points out the challenge to avoid revealing private information through counterfactual explanations, while [Goethals et al., 2022] proposes an algorithm for generating k-anonymous counterfactual explanations. [Sovrano et al., 2020] make a separation between explainable (X-) and explanatory (Y-) AI and propose a model for YAI under GPDR guidelines. According to Sorvano et al., making an AI system explainable (referred to as XAI) should be separated and independent of actually explaining the system (referred to as YAI).

In this section, we address the challenge of preserving privacy upon providing explanations within multi-agent environments and specifically in team formation scenarios. Specifically, we argue that an AI system should only offer explanations that guarantee no privacy breaches. To our knowledge, our proposal below is the first work tackling this challenge in team formation. As such, in this section, we propose a privacy breach detector capable of finding whether a given explanation is bound to lead to privacy breaches. As we detail later, we model the reasoning triggered by explanations using a theory of mind [Frith and Frith, 2005], which allows our detector to capture explanations bound to cause breaches. In addition, we propose a general framework that describes how our privacy breach detector interacts with a team formation algorithm (AI system) and an explanatory algorithm (XAI system) to approve or disapprove explanations. It is worth noting that our proposed framework is not restricted to the team formation algorithms presented in Chapter 4 and the explanatory algorithm presented in Section 6.2. Instead, the framework can involve team formation and explanatory algorithms other than the ones proposed in this dissertation.

The remaining of the section is structured as follows. In Section 6.4.1, we illustrate an example of the classroom team formation scenario which we use throughout this section. In Section 6.4.2, we describe a general framework that detects breaches of private information within explanations regarding team formation scenarios. To develop the privacy breach detector, Section 6.4.3 shows how to represent the concepts of knowledge and beliefs that we utilise in our framework. Then, in Section 6.4.4, we discuss the necessary inference rules used to reason over explanations. Section 6.4.5 puts forward a belief updater. The belief updater follows a theory of mind to simulate the line of reasoning of a questionnaire, and we illustrate this theory of mind in the classroom team formation scenario. Finally, in Section 6.4.6, we present the privacy checker that detects potential privacy breaches.

6.4.1 The Classroom Team Formation Scenario

Here we present a running example of the team formation scenario. We will be following this example throughout this section.

Let Renatta be a high-school teacher in charge of the history and literature courses. A group of six very enthusiastic students were interested in learning more about their local town. Thus, Renatta came up with three different "get-to-know-your-town" projects for the students to work on and investigate their town's local history and culture:

- **The PodCast Project (TPCP)**: In this project, the team in charge must prepare a podcast containing interviews with local people and narratives about the town's traditions.
- **The U-Video Project (TUVP)**: In this project, the team in charge must prepare a YouTube video series such that in each video, students visit monuments and historical places in town and briefly present the history of each place.

- **The WebSite Project (TWSP)**: In this project, the team in charge must prepare a website containing a "cultural map" of the town describing what kind of local festivities a visitor can enjoy (in which neighbourhood, at what time of the year, etc.).

Renatta wants her six students to work in pairs and each team to work on a different task so they can present it at the end of the school year. Ideally, Renatta would like each team to satisfy the following properties. Each team must be *(1) skilled* for its assigned project, *(2) diverse* in terms of personalities and gender, *(3) motivated* to work on its assigned project, and *(4) socially coherent*.

The students, namely Alex, Beth, Fedra, Jack, John and Suzan, provide their profiles containing their competencies, personalities, and preferences over projects and teammates. Then Renatta uses an AI team formation algorithm (TFA)—e.g., one of the algorithms presented in Chapter 4—to form teams and allocate them to projects. When the TFA yields a teams-to-tasks allocation, Beth challenges the explanatory algorithm (EA)—e.g., the algorithm described in Section 6.2—with the following query:[2]

> "Why is Jack in my team instead of Alex?"

Given the query, the explanatory algorithm computes an alternative allocation in which Beth and Alex are forced to work together. Consider that according to the current allocation, Beth works with Jack on The PodCast Project, while according to the alternative allocation, Beth works with Alex on The Website Project. Then the EA computes the differences between the teams in both allocations for Beth. That is, the EA compares ⟨Beth and Jack, TPCP⟩ against ⟨Beth and Alex, TWSP⟩ with respect to the requirements placed by the teacher:

- each team being skilled for their assigned project;
- each team being diverse in terms of personality;
- each team being satisfied with their assigned project in terms of individuals' preferences over projects; and
- each team being socially coherent in terms of individuals' preferences over teammates.

Finally, the EA builds the following explanation for Beth:

> "If Alex were on your team instead of Jack, then you would be in a less diverse team in terms of personality than the team you are currently in."

[2] We remind the reader that the framework we propose in this section is general; the team formation algorithms in Sections 4.3.1 and 4.3.2, and the explanatory algorithm in Section 6.2 are particular algorithms that we can use.

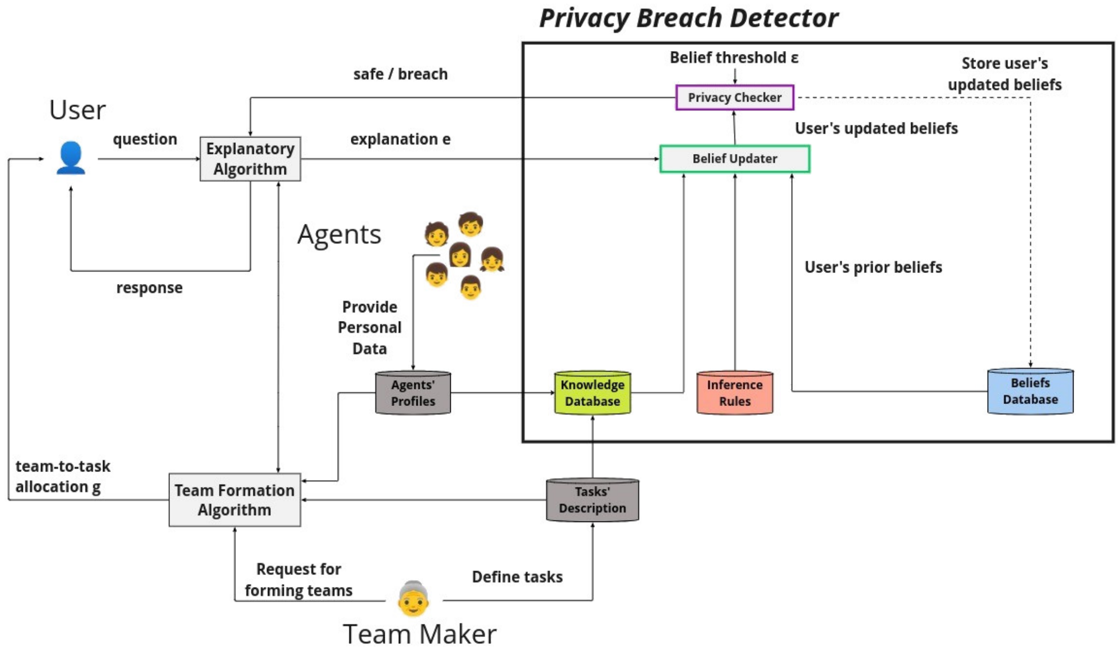


Figure 6.2: General Framework for Privacy-Aware Explanations for Team Formation

According to the EA, the desired property that explains best why Beth should be working with Jack and not with Alex is the property of diverse personalities. As we will see later, this is bound to lead to a privacy violation.

6.4.2 A General Framework for Privacy-Aware Explainable Team Formation

In this section, we describe a general framework that combines team formation solutions, explanations of these solutions, and a mechanism for checking whether some explanations may cause a privacy breach. Consider a team formation scenario (e.g., the one presented in the previous section) involving a set of agents $\mathcal{A}$ and a set of tasks $\mathcal{T}$. Moreover, let o be the 'orchestrator' or team-maker, i.e., the person who requests forming a team-to-tasks allocation using some team formation algorithm. A *user* corresponds to a person who challenges the teams-to-tasks allocation and can be either a team-maker or an agent. In this work, we assume that each agent holds a view of the world which consists of *(i) known facts* such as their private information, the description of the tasks, and the teams-to-tasks allocation, and *(ii) beliefs* over other agents' private information. Similarly, the team-maker also holds their view of the world, consisting of some known facts and their beliefs over the agents' private information. Figure 6.2 illustrates our proposed framework, which, in a nutshell, consists of the following components:

1. A team formation algorithm (TFA) that forms a teams-to-tasks allocation.
2. An explanatory algorithm (EA)—interacting with the TFA—that generates explanations regarding a teams-to-tasks allocation.
3. A privacy breach detector (PBD) that assesses whether an explanation may incur in privacy breaches. The PBD is composed of:
   a) a belief updater (BU) that computes posterior beliefs that the user is expected to form upon receiving an explanation; and
   b) a privacy checker (PC) that assesses whether the user's expected posterior beliefs exceed a belief threshold.

In more detail, the team-maker uses the TFA to solve a team formation problem and form an allocation, while the TFA notifies the team-maker and the agents of the allocation formed. As mentioned earlier, there is a plethora of TFAs that solve team formation problem. As such, depending on the problem at hand, one shall use the corresponding TFA. Considering the example in Section 6.4.1, Renatta, the teacher, acts as the team-maker and uses Edu2Com (Section 4.3.2) as the TFA to group her students (who correspond to agents) into teams to work on their "get-to-know-your-town" projects (which correspond to tasks). The TFA computes the teams along with their allocation to tasks. After that, the TFA communicates the resulting teams and allocations to the teacher and the students.

Then, say that some user challenges the TFA's result. That is, a user may argue that there is a better allocation than the one yielded by the TFA. Hence, the user poses a question to the explanatory algorithm. For example, student Beth asks why Jack is in her team instead of her friend Alex. The EA processes the user's question and builds an appropriate explanation by interacting with the TFA—following, for example, the process of building contrastive explanations we described earlier in this chapter. For example, the EA builds the following explanation to answer Beth's question:

> "If Alex were on your team instead of Jack, then you would be in a less diverse team in terms of personality than the team you are currently in."

Next, the EA passes the generated explanation to the privacy breach detector, particularly to the belief updater. Each agent holds knowledge regarding the world and beliefs over other agents' private information. The BU is responsible for exploiting the information conveyed by an explanation, combining it with the user's knowledge and current beliefs to extract valuable conclusions. Specifically, the BU follows a theory of mind [Frith and Frith, 2005] on the user to simulate the reasoning the user is expected to follow (based on the user's knowledge and beliefs). As a result, the BU forms an updated version of beliefs, which the user is expected to reach after receiving the explanation. In our running example, Beth holds some prior beliefs about the personalities of Alex and Jack. Beth is expected to update her prior beliefs based on the explanation provided by the EA. After that, the BU passes the expected posterior beliefs to the privacy checker. The PC assesses whether the user's expected posterior beliefs exceed the belief threshold ε. The belief threshold corresponds to a maximum probability that a user may believe that some agent's information is true without violating this agent's privacy. For example, with a belief threshold $\varepsilon = 0.5$, if Beth is expected to update her beliefs that Alex is of personality role 'leader' to 0.3, then this *causes no violation* of Alex's private information. On the other hand, if Beth is expected to update her belief that Jack is of personality 'implementer' to 0.7, then this *causes a violation* of Jack's private information. Notably, each user may set their belief threshold regarding their private information. For example, Alex may set his belief threshold regarding his personality to 0.2. On the other hand, Jack may set his corresponding threshold to 0.75. In this case, the explanation that answers Beth's question would cause a violation of Alex's privacy, but it would not violate Jack's privacy.

Finally, the privacy checker outputs an answer for the explanatory algorithm. Specifically, the privacy checker's responds with an appropriate message indicating whether the explanation is *safe* in case our Privacy Breach Detector detected no privacy breaches on private information or *violating* otherwise. Depending on the PC's response, the explanatory algorithm either *yields* with the explanation or *handles* the situation. That is, in case the explanation is safe according to the PC, then the EA can share the explanation with the user. Otherwise, the EA needs to resolve the privacy issue, e.g., by computing a different explanation or denying an answer due to a privacy breach.

6.4.3 Representing knowledge and beliefs

In this section, we discuss how to represent *knowledge* and *beliefs* within our framework (see Figure 6.2). Recall that both the agents and the team-maker hold a view of the team formation problem. This view consists of known facts and beliefs. *Knowledge* corresponds to known facts that an agent has over the team formation scenario. Such known facts include the tasks' description and the team-to-tasks allocation published by the TFA. Moreover, for an agent a, known facts also include their own personal profile—in our example, Alex's knowledge includes his competence profile, personality profile, and preference profiles over tasks and teammates. Besides knowledge, individuals can form *beliefs* over others. Specifically, individuals form beliefs regarding knowledge they do not own, i.e., beliefs over another agent's profile.

Agent's Knowledge. An agent holds *knowledge* that can be either *private* or *public*. Given an agent $a \in \mathcal{A}$, their *private knowledge* refers to characteristics that comprise agent a's own profile. Each agent holds their own private knowledge, withheld from anyone else. For instance, "John acquires the competence of Video Editing" is part of John's private knowledge.

Public knowledge refers to the tasks made public by the team-maker. That is, "The U-Video Project requires the competence of Video Editing" is public knowledge. Moreover, public knowledge includes the team-to-tasks allocation announced by the team formation algorithm. For example, "John has been assigned to work on The U-Video Project" is public knowledge. All agents at the outset share the same public knowledge, i.e. public knowledge is common to all agents.

We represent knowledge using *first-order predicates* with *ground terms*. For an agent $a \in \mathcal{A}$, we denote the private knowledge of a as Γ_a corresponding to a set of first-order predicates with ground terms referring to a. In our running example, we use the following predicates for agent a's private knowledge:

- acquires(a, c), where $a \in \mathcal{A}$ is an agent, and $c \in \mathcal{C}_a$ is one of a's acquired competencies;
- personality(a, p), where $a \in \mathcal{A}$ is an agent, and p is a's personality role;[3]
- wants_to_work_on(a, τ), where $a \in \mathcal{A}$ is an agent, and $\tau \in \mathcal{T}$ is a task; and
- wants_to_work_with(a, b), where $a \in \mathcal{A}$ is an agent, and $b \in \mathcal{A} \setminus \{a\}$ is an agent different to a.

So, predicate acquires(John, Video Editing) $\in \Gamma_a$ corresponds to some of John's private knowledge. We denote with Γ_τ the public knowledge that each agent holds regarding task τ. In our example, we use the following predicates for public knowledge related to a task's description:

- requires(τ, c), where $\tau \in \mathcal{T}$ is a task and $c \in \mathcal{C}_\tau$ is one of τ's required competencies;
- size(τ, s_τ), where $\tau \in \mathcal{T}$ is a task and $s_\tau \in \mathbb{N}$ is the required team size (with $s_\tau \geq 2$).

Finally, with Γ_g, we denote the public knowledge that each agent holds regarding the team-to-task allocation announced by the TFA. In our example, we use the predicate works_on(g, a, τ) which is read as "According to team-to-task-allocation g, agent a is member of the team allocated to work on task τ". Therefore, an agent a's knowledge consists of: $\Gamma_a \cup \Gamma_\tau \cup \Gamma_g$.

Team-Maker's Knowledge The team maker only holds public knowledge regarding the tasks' description and team-to-tasks allocations. Thus, the team-maker's knowledge is $\Gamma_o \equiv \Gamma_\tau \cup \Gamma_g$.

[3]Assuming a finite set of Personality Roles, e.g., the Belbin's nine roles or the 16 MBTI.

Agents' Beliefs. Each agent holds beliefs over other agents' private knowledge. That is, an agent sets a probability with which they believe that some private knowledge of another agent is true. For example, let Beth believe that John is knowledgeable in Video Editing with a probability of 0.7, which comprises Beth's belief over some of John's private knowledge. This belief is, in fact, a probability over a predicate in Γ_{John}, i.e., $P[\text{acquires(John, Video Editing)}] = 0.7$. Thus, an agent's a beliefs correspond to a probability function over predicates in $\bigcup_{b \in \mathcal{A}} \Gamma_b$.

Team-maker's Beliefs. The team-maker holds beliefs over the agents' private information as well. Similarly, the team-maker's beliefs correspond to a probability function over predicates in $\bigcup_{a \in \mathcal{A}} \Gamma_a$.

Beliefs Initialisation. When the different parties (participants, team-maker, observers, etc.) have no prior interaction, then we have a team formation environment under full uncertainty. In such cases, some generally known guidelines help individuals form their initial beliefs about other agents' private information. In our running example, we can consider four such guidelines, one per feature comprising an agent's profile:

init_1 : An agent $a \in \mathcal{A}$ may either acquire a competence $c \in \mathcal{C}$ **OR** a may not acquire c.

init_2 : An agent $a \in \mathcal{A}$ may be of **exactly one** out of sixteen (16) personality roles

init_3 : An agent $a \in \mathcal{A}$ may either like working on a task $\tau \in \mathcal{T}$ **OR** a may not like working on τ.

init_4 : An agent $a \in \mathcal{A}$ may either like working with another agent $b \in \mathcal{A} \setminus \{a\}$ **OR** a may not like working with b.

Interrelation Properties of Beliefs. In addition, we also have some properties that govern beliefs, indicating the interrelations among the features and profiles of different agents. In our running example, we have four such properties:

property_1: Agent a acquiring competence c is independent to agent b acquiring competence c'.

property_2: Agent a being of perosnality p is independent to agent b being of personality p'.

property_3: Agent a liking task τ is independent to agent b liking task τ'.

property_3: Agent a liking agent b is independent to agent c liking agent d.

6.4.4 Inference Rules

Here, we discuss the *inference rules* used by our model within the privacy breach detector (see Figure 6.2). In particular, we use 'IF-THEN' rules that guide the *belief updater* (BU) component to reason over new information deriving from an explanation. We consider two types of rules, namely:

(i) rules that determine when a team satisfies a desired requirement placed by the teamaker, and

(ii) rules that interpret a contrastive explanation.

Considering our classroom example in Section 6.4.1, we have one rule per desired requirement to determine when a team satisfies this requirement. For example, such a rule is:

> "IF the members of team K are of different personality roles THEN the team K is diverse".

If we use first-order predicates, we can write the rule as follows:

$$\forall x, y, \forall p \quad \text{inTeam}(x, K) \wedge \text{inTeam}(y, K) \wedge \text{personality}(x, p) \wedge \neg \text{personality}(y, p) \Rightarrow \text{isDiverse}(K)$$

We also have rules for interpreting contrastive explanations, i.e., interpreting the comparisons described in the explanations. In our example, a contrastive explanation involves a comparison of the form:

> "team A assigned to task τ satisfies requirement f, while team B assigned to task σ does not"

or a comparison of the form:

> "both team A assigned to task τ and team B assigned to task σ (do not) satisfy property f".

Using first-order predicates, the comparisons above can be written as follows:

$$\text{assignedTo}(A, \tau) \wedge \text{assignedTo}(B, \sigma) \wedge \text{isBetter}(A, B, f) \Rightarrow \text{satisfies}(A, \tau, f) \wedge \neg \text{satisfies}(B, \sigma, f)$$

and

$$\text{assignedTo}(A, \tau) \wedge \text{assignedTo}(B, \sigma) \wedge \text{isEqual}(A, B, f) \Rightarrow \\ \big(\text{satisfies}(A, \tau, f) \wedge \text{satisfies}(B, \sigma, f)\big) \vee \big(\neg \text{satisfies}(A, \tau, f) \wedge \neg \text{satisfies}(B, \sigma, f)\big)$$

Next, we provide the inference rules used in our running example. We express the inference rules in natural language.

r_1 : IF the team members count on all the skills required by the task THEN the team is skilled for the task.

r_2 : IF the team members are of different personality roles THEN the team is diverse in terms of personality.

r_3 : IF each team member likes the task, THEN the team is satisfied with the task.

r_4 : IF each team member likes the task, THEN the team is satisfied with the task.

r_5 : IF team A assigned to τ is a better match than team B assigned to σ *wrt* property f THEN team A satisfies f AND team B does not satisfy f.

r_6 : IF team A assigned to τ is a worse match than team B assigned to σ *wrt* property f THEN team A does not satisfy f AND team B satisfies f.

r_7 : IF team A assigned to τ is an equally good match to team B assigned to σ *wrt* property f THEN *both* teams A and B satisfy property f OR *both* teams A and B do not satisfy property f.

We can formally express the rules above with the aid of first-order predicates as follows:

$$r_1: \quad \forall c \quad \Big((\exists\, x \quad \mathrm{inTeam}(x, K) \wedge \mathrm{assignedTo}(K, \tau)) \vee \neg \mathrm{requires}(\tau, c) \Big) \Rightarrow \mathrm{isSkilled}(K, \tau)$$

$$r_2: \quad \forall\, x, y,\ \forall p \quad \mathrm{inTeam}(x, K) \wedge \mathrm{inTeam}(y, K) \wedge \mathrm{personality}(x, p) \wedge \neg \mathrm{personality}(y, p) \Rightarrow \mathrm{isDiverse}(K)$$

$$r_3: \quad \forall\, x \quad \mathrm{inTeam}(x, K) \wedge \mathrm{assignedTo}(K, \tau) \wedge \mathrm{wants\_to\_work\_on}(x, \tau) \Rightarrow \mathrm{isMotivated}(K)$$

$$r_4: \quad \forall\, x, y \quad \mathrm{inTeam}(x, K) \wedge \mathrm{inTeam}(y, K) \wedge \mathrm{wants\_to\_work\_with}(x, y) \Rightarrow \mathrm{isCoherent}(K)$$

$$r_5: \quad \mathrm{assignedTo}(A, \tau) \wedge \mathrm{assignedTo}(B, \sigma) \wedge \mathrm{isBetter}(A, B, f) \Rightarrow \mathrm{satisfies}(A, \tau, f) \wedge \neg \mathrm{satisfies}(B, \sigma, f)$$

$$r_6: \quad \mathrm{assignedTo}(A, \tau) \wedge \mathrm{assignedTo}(B, \sigma) \wedge \mathrm{isWorse}(A, B, f) \Rightarrow \neg \mathrm{satisfies}(A, \tau, f) \wedge \mathrm{satisfies}(B, \sigma, f)$$

$$r_7: \quad \mathrm{assignedTo}(A, \tau) \wedge \mathrm{assignedTo}(B, \sigma) \wedge \mathrm{isEqual}(A, B, f) \Rightarrow \big(\mathrm{satisfies}(A, \tau, f) \wedge \mathrm{satisfies}(B, \sigma, f)\big) \vee \big(\neg \mathrm{satisfies}(A, \tau, f) \wedge \neg \mathrm{satisfies}(B, \sigma, f)\big)$$

Depending on the desired requirement f the term satisfies(K, τ, f) corresponds to:

desired requirement f	satisfies(K, τ, f)
competencies	isSkilled(K, τ)
personality	isDiverse(K)
preferences over tasks	isMotivated(K, τ)
preferences over teammates	isCoherent(K)

Given these rules, we can handle the process of inference with a rule-based forward reasoner [Rattanasawad et al.,]. Next, the inference is used to update the beliefs that the explainee holds over private information of the agents appearing in the explanation, following a theory of mind approach [Frith and Frith, 2005].

6.4.5 Belief updater

In this section, we discuss how we model a user's reasoning upon receiving an explanation about a team formation allocation. A user is expected to perform inferences by using the new knowledge in the explanation and their prior knowledge and beliefs. As such, here we model the user's reasoning following a *theory of mind* [Frith and Frith, 2005] approach. That is, we present a belief updater (BU) that simulates the user's expected reasoning upon receiving an explanation, exploiting the user's knowledge and beliefs.

First, we remind the reader of our running example. Our user is student Beth. Beth challenges the explanatory algorithm to explain why she has not been teamed up with Alex instead of Jack, her current teammate. In reply, the EA builds the following explanation:

> e ="if you teamed up with Alex, then you would be part of a less diverse team in terms of personality."

Now, if Beth receives this explanation, we consider that she reasons as follows. Considering the inference rule r_6, Beth infers that the team consisting of Beth and Alex does not satisfy the property of diversity. In contrast, the team consisting of Beth and Jack satisfies the property of diversity. This, in turn, triggers inference rule r_2 regarding personality diversity. Now, Beth can switch the focus of her inference from the team's diversity to the individual's personality types. Hence, she can make two new inferences directly related to agents' profiles, expressed as statements:

s1 Beth and Alex are of the same personality type; and

s2 Beth and Jack are of different personality types.

Statement **s1** can be expressed in terms of probabilities using property$_2$ (see Section 6.4.3) of the user's beliefs' properties:

$$\sum_{p=p_1,\cdots p_9} P\big[\text{personality}(\text{Beth}, p)\big] \cdot P\big[\text{personality}(\text{Alex}, p)\big] = 1 \tag{6.8}$$

Similarly, statement **s2** can be expressed in terms of probabilities using property$_2$:

$$\sum_{p=p_1,\cdots p_9} \sum_{p' \neq p} P\big[\text{personality}(\text{Beth}, p)\big] \cdot P\big[\text{personality}(\text{Jack}, p')\big] = 1 \tag{6.9}$$

Let us assume that Beth has no knowledge regarding Jack's and Alex's personality roles. As such, initially, Beth is in an environment with full uncertainty, and therefore, she initialises her prior beliefs accordingly, i.e., using the rules for beliefs initialisation described in 6.4.3. According to guideline init$_2$, each one is of precisely one out of possible personality roles with equal probability. Formally, Beth's initial beliefs are:

$$P\big[\text{personality}(agent, p)\big] = \frac{1}{16} \qquad \text{for } agent = \text{Jack, Alex} \quad \text{and } p = p_1, \cdots, p_{16}$$

On the other hand, Beth knows that her own personality role is p_3, i.e.,

$$P\big[\text{personality}(\text{Beth}, p_3)\big] = 1$$

and

$$P\big[\text{personality}(\text{Beth}, p)\big] = 0 \quad \text{for } p = p_1, p_2, p_4 \cdots, p_{16}$$

Considering the simple statement **s1** as expressed in Eq (6.8), Beth updates her beliefs regarding Alex being of personality role p_3

$$\begin{aligned}
\sum_{p=p_1,\cdots p_{16}} P\big[\text{personality}(\text{Beth}, p)\big] \cdot P\big[\text{personality}(\text{Alex}, p)\big] &= 1 \Rightarrow \\
P\big[\text{personality}(\text{Beth}, p_3)\big] \cdot P\big[\text{personality}(\text{Alex}, p_3)\big] &= 1 \Rightarrow \\
P\big[\text{personality}(\text{Alex}, p_3)\big] &= 1
\end{aligned}$$

This, in its turn, triggers Beth's beliefs update regarding Alex being of any other personality role, i.e.:

$$\begin{aligned}
&\sum_{p=p_1\cdots,p_{16}} P\big[\text{personality}(\text{Alex}, p)\big] && = 1 \Rightarrow \\
&P\big[\text{personality}(\text{Alex}, p_3)\big] + \sum_{p\neq p_3} P\big[\text{personality}(\text{Alex}, p)\big] && = 1 \Rightarrow \\
&1 + \sum_{p\neq p_3} P\big[\text{personality}(\text{Alex}, p)\big] && = 1 \Rightarrow \\
&\sum_{p\neq p_3} P\big[\text{personality}(\text{Alex}, p)\big] && = 0 \Rightarrow \\
&P\big[\text{personality}(\text{Alex}, p)\big] = 0,\ \forall\, p = p_1, p_2, p_4, \cdots, p_{16}
\end{aligned}$$

Similarly, considering the simple statement **s2** as expressed in Eq (6.9), Beth updates her beliefs regarding Jack being of personality role p_3:

$$\begin{aligned}
&\sum_{p=p_1,\cdots p_{16}} \sum_{p'\neq p} P\big[\text{personality}(\text{Beth}, p)\big] \cdot P\big[\text{personality}(\text{Jack}, p')\big] && =1 \Rightarrow \\
&\sum_{p'\neq p_3} P\big[\text{personality}(\text{Beth}, p_3)\big] \cdot P\big[\text{personality}(\text{Jack}, p')\big] && =1 \Rightarrow \\
&\sum_{p'\neq p_3} P\big[\text{personality}(\text{Jack}, p')\big] && =1
\end{aligned} \tag{6.12}$$

Given guideline init_2 combined with Eq (6.12), it holds that:

$$\begin{aligned}
&\sum_{p=p_1\cdots,p_{16}} P\big[\text{personality}(\text{Jack}, p)\big] && =1 \overset{Eq\ (6.12)}{\Rightarrow} \\
&\sum_{p=p_1\cdots,p_{16}} P\big[\text{personality}(\text{Jack}, p)\big] && = \sum_{p'\neq p_3} P\big[\text{personality}(\text{Jack}, p')\big] \Rightarrow \\
&P\big[\text{personality}(\text{Jack}, p_3)\big] + \sum_{p\neq p_3} P\big[\text{personality}(\text{Jack}, p)\big] && = \sum_{p'\neq p_3} P\big[\text{personality}(\text{Jack}, p')\big] \Rightarrow \\
&P\big[\text{personality}(\text{Jack}, p_3)\big] = 0
\end{aligned}$$

This, in its turn, triggers Beth's beliefs update regarding Jack being of any other personality role, i.e.:

$$\begin{aligned}
&\sum_{p=p_1\cdots,p_{16}} P\big[\text{personality}(\text{Jack}, p)\big] && = 1 \Rightarrow \\
&P\big[\text{personality}(\text{Jack}, p_3)\big] + \sum_{p=p_1,p_2,p_4\cdots,p_{16}} P\big[\text{personality}(\text{Jack}, p)\big] && = 1 \Rightarrow \\
&0 + \sum_{p=p_1,p_2,p_4\cdots,p_{16}} P\big[\text{personality}(\text{Jack}, p)\big] && = 1 \Rightarrow \\
&\sum_{p=p_1,p_2,p_4\cdots,p_{16}} P\big[\text{personality}(\text{Jack}, p)\big] && = 1
\end{aligned}$$

Beth rules out the possibility that Jack is of personality role p_3. At the same time, she believes that Jack's personality role is one out of the remaining eight personality roles with equal probability i.e.,

$$P\big[\text{personality}(\text{Jack}, p)\big] = \frac{1}{15} \qquad \text{for } p = p_1, p_2, p_4 \cdots, p_{16}$$

Here we highlight that when Beth finishes her beliefs updating, she is expected to reach certain conclusions regarding Alex's and Jack's personalities, which she did not know before. That is, she has learnt that

1. Alex is of personality role p_3, and
2. Jack is not of personality role p_3.

In the example above, we illustrated how a user is expected to think upon an explanation and ultimately reach some conclusions that allow the user to update their beliefs regarding some agent's private information. As such, the above reasoning can be systematically described as a Belief Updater algorithm (see Alg. 4), which graphically is depicted in Figure 6.3.

In words, according to the proposed belief updater (BU), given an explanation e, the user first encodes the explanation in the form of new facts. Then, the user needs to make some inferences based on the inference rules at hand and reach some simple statements. This process includes selecting the appropriate inference rules and triggering one rule after another towards the simple statements. Notably, the inference can be handled by a rule-based forward reasoner [Rattanasawad et al.,]. Then, each simple statement s is expressed in terms of probabilities s_{prob} considering the beliefs' properties of the user. Next, if there is a known fact in the user's knowledge that *applies* in the probabilistic form s_{prob}, the user applies the knowledge in s_{prob}. That is, if there exists a predicate in Γ_u that appears in s_{prob}, the user exploits this predicate's known probability (that is, either 0 or 1) to obtain a conclusion. Then, the user exploits

Algorithm 4: Belief Updater

Input : explanation e
output : posterior beliefs

new_facts $\leftarrow$ encodeExplanation(e);
while (*new_facts*):
  fact $\leftarrow$ pop(new_facts);
  $R_{\text{fact}} \leftarrow$ select inference rules related to the fact;
  $S \leftarrow$ use inference rules in R_{fact} to inference simple statements;
  for ($s \in S$):
    $s_{prob} \leftarrow$ Express s as a statement related to agent's a profiles;
    $P_u \leftarrow$ Get user's u prior beliefs from database;
    $\Gamma_u \leftarrow$ Get user's u knowledge from database;
    if(*there is knowledge* $\gamma \in \Gamma_u$ *that applies in* s_{prob}):
      $s'_{prob} \leftarrow$ apply(γ, s_{prob});
      $P' \leftarrow$ Backpropagate(s'_{prob}, P);
      if(*there is new fact in* P'): add new fact in new_facts ;
    else :
      stop reasoning over s_{prop};
return P'

this conclusion using probabilistic rules and beliefs' properties to form their posterior beliefs. Finally, if the posterior beliefs reach some new facts (e.g., Alex is of personality role p_3 with probability 1), the user must also reason about these new facts.

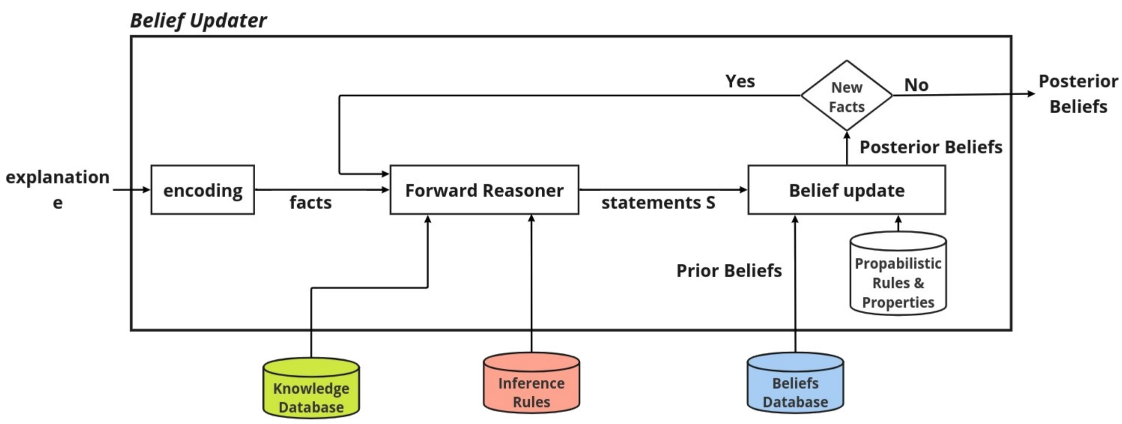


Figure 6.3: Belief updater

Now, when our privacy breach detector uses the above line of reasoning, the user's posterior beliefs correspond to the beliefs we *expect* that the user will form when they receive the explanation. Next, our detector uses these expected posterior beliefs to assess whether some private information is disclosed, as we detail in Section 6.4.6.

6.4.6 Checking privacy breaches

In this section, we describe the *privacy checker* (PC) of our architecture in Figure 6.2. As discussed in the previous section, the belief updater (BU) computes the user's expected posterior beliefs, with the user's beliefs being (re-)formed upon receiving an explanation built by the explanatory algorithm. The PC assesses whether these expected posterior beliefs cause a *privacy breach*. Our model assumes a *belief threshold* ε, representing an upper limit on agents' beliefs. That is, if some belief reaches a probability ε or higher, the privacy checker considers that there is a privacy breach. In our running example, if the belief threshold is $\varepsilon = 0.5$, Beth's expected posterior beliefs regarding Jack's personality do not lead to a privacy breach. Remember that Beth's beliefs regarding Jack are:

$$P\big[\text{personality}(\text{Jack}, p)\big] = 1/15 \leq \varepsilon \quad \text{for all } p = p_1, p_2, p_4, \cdots, p_9$$

and

$$P\big[\text{personality}(\text{Jack}, p_3)\big] = 0 \leq \varepsilon$$

On the other hand, Beth's expected posterior beliefs regarding Alex's personality lead to a privacy breach since her beliefs regarding Alex are:

$$P\big[\text{personality}(\text{Alex}, p_3)\big] = 1 \geq \varepsilon$$

Now, let us formally define the notion of *privacy breach*. Let u be the user and $a \in \mathcal{A}$ be an agent participating in the team formation scenario with $u \neq a$. The EA builds an explanation e and forwards it to the privacy breach detector (PBD) to assess whether it is safe for the EA to respond to the user with explanation e. First, the detector computes the user's expected posterior beliefs using the BU. Assume that the BU computes u's posterior belief regarding some of a's private information $\gamma \in \Gamma_a$ by reasoning over the new knowledge contained in e. Moreover, let $P[\gamma] = p$ be u's prior belief before receiving e, and $P'[\gamma] = p'$ be u's expected posterior belief. Then a privacy breach is defined as:

Definition 26 (Privacy Breach)**.** *An explanation e given to user u causes a privacy breach if there exists an agent $a \in \mathcal{A}$ (with $u \neq a$) and some of a's private information $\gamma \in \Gamma_a$ such that: user u's prior belief over γ is $p \leq \varepsilon$; and the explanation e leads to an updated belief over γ to p' such that $p' > \varepsilon$, where $\varepsilon \in [0, 1]$ is a belief threshold.*

The PC detects privacy breaches through the user's prior beliefs and the expected posterior beliefs computed by the BU component. Specifically, the PC compares both the prior P and the (expected) posterior P' beliefs against the belief threshold ε. A privacy breach is detected if the information conveyed by the explanation leads from a non-violating situation to a violating situation regarding some agent's privacy, i.e., if the posterior belief exceeds the threshold while the prior belief does not. In more detail, the PC proceeds as follows:

1. first, it iterates over all of the private knowledge for which the user holds posterior beliefs[4],

2. finds the owner of the private knowledge of the belief, and

3. compares both the user's prior and expected posterior belief with the belief threshold.

Finally, if there exists a privacy breach on any of this information, then the PC notifies the explanatory algorithm (EA) with an appropriate message indicating the breach. Otherwise, if no privacy breach exists, then the PC notifies the EA that the explanation is safe.

So far, we have presented a mechanism for detecting privacy breaches on explanations regarding team formation scenarios. We assume that the explanations are given to the user *in private* and that the user keeps the explanations for themselves. However, we may assume that individuals can exchange information, particularly that they share explanations with each other. In such a case, the information conveyed in an explanation becomes available to everyone. As such, a privacy breach may occur by any individual, not just the user. Therefore, the privacy breach detector needs to compute the expected posterior beliefs of any individual. Here, we note that each individual holds their own knowledge, interacts with the EA and forms their own beliefs, even if they share the explanations.

Definition 27 (Collective Privacy Breach). *An explanation e given to user u causes a* collective privacy breach *if there exists an individual $i \in \mathcal{A} \cup \{o\}$ (with $u \neq a$), an agent $a \in \mathcal{A}$ (with $i \neq a$) and some of a's private information $\gamma \in \Gamma_a$ such that: the individual's i prior belief over γ is $p \leq \varepsilon$; and the explanation e leads to an updated belief for individual i over γ to p' such that $p' > \varepsilon$, where $\varepsilon \in [0, 1]$ is a belief threshold.*

Thus, in order to detect a collective breach, the PBD acts as follows. First, the PBD runs the BU and the PC for the user. If no privacy breach is detected, then the PBD *sequentially* runs the BU and the PC for every individual until either *(i)* a privacy breach is detected or *(ii)* the process (a run of the BU and the PC) is repeated for all the individuals. Notably, each time the PBD runs the BU and the PC components for some individual i, the algorithms use

[4] We restrict this iteration over the private knowledge for which the BU computed a posterior belief different to the corresponding prior.

the knowledge and the beliefs of this particular individual i. Finally, the PBD notifies the EA with an appropriate message. Specifically, it outputs the message 'breach' if the PBD detects a privacy breach incurred by the user, a 'collective breach' message if the PBD detects a privacy breach incurred by any other individual, or 'safe' if the PBD detects no privacy breaches.

6.5 Summary

In this chapter, we turned our attention to eXplainable AI (XAI) within multiagent environments, and specifically, we made advances towards explaining team formation. First, in Section 6.2, we introduced a general methodology for building contrastive explanations in team formation scenarios. Importantly, our proposed explanatory algorithm is not designed for a specific team formation algorithm. Instead, our explanatory algorithm can *wrap* around existing team formation algorithms with no modifications needed. In a nutshell, our explanatory algorithm builds on the notion of facts and foils [Miller, 2018], and explores 'what-if' scenarios in order to explain teams formed by the team formation algorithm at hand. Specifically, our explanatory algorithm compares the original teams formed by the team formation algorithm (facts) against teams compliant with a what-if scenario (foils). Finally, the explanatory algorithm generates explanations highlighting different perspectives. Alongside with the explanatory algorithm, we identified a collection of thirteen intuitive query templates that widely cover questions regarding team formation. To evaluate the quality of our explanations we conducted a preliminary empirical evaluation over synthetic data. Specifically, we assessed the explanations generated in terms of

(i) the number of causes appearing in the explanations,

(ii) the mean explainability precision of the explanations, and

(iii) the reading level required to understand the explanation.

Our results showed that our explanations are easy to read, they are simple by using a small number of causes (attributes), and they exhibit an acceptable mean explainable precision.

Moreover, in this chapter we tackled the reserving privacy challenge, one of the key challenge identified by [Kraus et al., 2020]. We argue that providing explanations should guarantee that agents' private information is not disclosed. Towards this, we proposed a general framework that combines team formation solutions and explanations over these solutions, while it detects whether an explanation would lead to a privacy breach. In particular, we put forward a privacy breach detector that complements an explanatory algorithm, and assesses the explanations built with respect to privacy breaches. Our proposed framework describes how our privacy breach detector interacts with a team formation algorithm and an explanatory algorithm

(XAI system) to detect potential privacy violations. Notably, our proposed framework for privacy detection is not restricted to the team formation algorithms presented in Sections 4.3.1 and 4.3.2 or the explanatory algorithm presented in Section 6.2.

Chapter 7

Conclusions and Future Work

In this dissertation, we study *human teams* and how to form efficient teams to be *allocated to tasks*. Literature suggests that teams and teamwork have a crucial role in a wide range of application areas in the public and private sectors, including the educational domain, industry, open innovation challenges, and crowdsourcing events. Here, we address several open problems regarding task allocation for human teams, and we contribute to the literature by *(i)* incorporating valuable findings regarding team composition from psychology and social sciences into formal algorithmic processes, *(ii)* proposing algorithms for solving the problem of forming many teams for many tasks with no overlaps, and *(iii)* explaining team formation algorithms. In what follows, we summarise this dissertation, discuss the lessons learned during this study, and explore the future directions that arise from this thesis.

7.1 Summary and Lessons learned

First, we reviewed the literature to identify the current practices on forming teams and the key components that drive people's behaviour during teamwork. Specifically, we went through research in Computer Science, Psychology, and Social Sciences. Research in Computer Science focuses on solving the *team formation problem*, i.e., the problems of selecting one or more groups of agents (not necessarily humans) who shall tackle a job or task. Existing research explores the several variations of the team formation problem, namely forming a single team to work on a single task and forming multiple teams to work on multiple tasks that either permit overlaps or not. There, we find several algorithms that tackle the aforementioned team formation problems. We note that this thesis focuses on the team formation problem involving multiple teams and multiple tasks permitting no overlaps. The scant research on our team formation problem ([Präntare and Heintz, 2018, Czatnecki and Dutta, 2019]) exhibits severe limitations regarding scalability and applicability. Among the existing works, we distinguish the work of [Andrejczuk et al., 2019], who address the problem of partitioning a set of agents into similar-sized teams so that each team tackles the same task. In this thesis, we extend the approach of Andrejczuk et al. and address the problem of partitioning a set of agents into teams (of various team sizes) so that each team tackles a different task. By doing so, we addressed a non-trivial generalisation of the problem previously tackled by [Andrejczuk et al., 2019], and we introduced an anytime heuristic solver for solving large real-world instances of the problem.

Research in Psychology and Social Sciences mainly studies the human characteristics that play a key role in teamwork and team performance. In particular, we find research in Organisational Psychology investigating the impact of a team's composition in terms of team members' personalities on teamwork and team performance. The evidence presented supports that personality diversity within a team boosts team performance. In Motivational Psychology, we find research regarding the relationship between people's motivation for tackling a task and their performance. Many studies across different application areas (including sports, the educational domain, hospitals, and construction companies) investigate how motivation affects team performance. According to the findings in the literature, motivated teams exhibit better performance. Therefore, aiming at motivated teams during the formation process is bound to improve team performance. Notably, to the best of our knowledge, there has yet to be a team formation algorithm considering team motivation. As such, this dissertation contributes to this matter by taking into account individuals' motivation while forming teams. Finally, research in Social Sciences focuses on the relation between people's social bonds and team performance. Researchers studying the social interactions among the members within a working team claim that the more socially coherent a team is, the more likely the team will bloom and prosper [Friedkin, 2004]. The relevant studies support this claim. Therefore, in this thesis, we studied the empirical evidence from Psychology and Social Sciences regarding the components that boost team performance. Hence, we developed the means to exploit the components acknowledged to boost team performance while we form teams.

Then, in Chapter 3, we focused on how to formally model the several components of the team formation problem, namely the agents, the tasks, and the teams. In accordance with the relevant literature, we modelled human agents to be comprised of their competencies, personalities, and preferences. In more detail, we showed how to formally define *(i)* an agent's acquired competencies, *(ii)* their personality following the Mayers-Briggs Type Indicator [Myers et al., 1998], and their preferences over *(iii)* different tasks and *(iv)* potential teammates. Also, we provided the formal modelling of tasks to be defined through a set of requirements in terms of competencies and team size. Regarding the concept of teams, we consider a team to be more than just a group of individuals. Instead, we argue that a team corresponds to a group of people who share a common goal. In other words, a team consists of a group of individuals who collectively tackle the same task. Thus, we formally defined a team in this light. After that, given a team, we thoroughly discussed how to integrate the team members and task requirements.

Towards a team-level aggregation, we proposed metrics that allow us to incorporate the team members' individual profiles. We proposed metrics that allow us to integrate individual profiles at a team level. Specifically, our metrics include:

- The means to align the competencies offered by a team with the ones required by a task, contributing in this way to bridging the gap between offered and required competencies.
- An enhanced personality and gender diversity metric. Our metric builds on the one proposed in [Andrejczuk, 2018], and propose a less restrictive model of gender diversity. That is, our enhanced diversity metric can handle more than just the binary gender paradigm of male vs. female.
- The means to capture a team's motivation through the team's collective willingness to work on the assigned task.
- A metric for computing a team's social cohesion in terms of each individual accepting the team and vice versa.

Thus, our thesis advanced the literature by introducing new and enhancing existing ways to evaluate a team concerning four key aspects that boost teamwork and team performance.

In Chapter 4, we introduced the team formation problem that we address in this thesis, i.e., the problem of forming many teams to tackle many tasks without overlaps, which we refer to as the *Non-Overlapping Many Teams to Many Tasks (NOMTMT) Allocation Problem*. Specifically, we formally described the NOMTMT allocation problem and cast it as an optimisation problem. Given a set of agents and a set of tasks, the NOMTMT allocation problem regards the formation of multiple teams so that each team works on exactly one task. Additionally, each agent can participate in at most one team, while each task can be tackled by at most one team. Notably, as we discussed earlier, the NOMTMT allocation problem is a non-trivial generalisation of the *Synergistic Team Composition Problem* presented in [Andrejczuk, 2018]. Our optimisation problem aims to form balanced teams considering the agents' profiles and the tasks' requirements while it respects the constraints mentioned above. After that, we studied the complexity of the problem. We characterised the problem's search space and provided the means to quantify the number of feasible solutions. Moreover, we investigated the problem with regard to its complexity class.

Next, we put forward two solvers to tackle the NOMTMT allocation problem. First, we proposed an *optimal solver*. That is, we proposed solving the problem optimally with the means of *Integer Linear Programming (ILP)*. We provided an ILP encoding for the NOMTMT allocation problem that, given sufficient time, can be solved with any out-of-the-shelf ILP solver. However, the combinatorial nature of the problem, along with our complexity analysis, indicates that solving the problem optimally is inefficient in practice. That is, building the input for problem instances involving many individuals, many tasks and large teams is bound to lead to large ILPs and, therefore, impractical. Hence, we proceeded to develop a heuristic solver. More precisely, we devised *Edu2Com*, an anytime heuristic algorithm to solve the NOMTMT allocation problem. Our algorithm first builds a promising initial allocation of teams to tasks, and thereafter, it iteratively improves the initial allocation. In order to improve the allocation, we consider different types of *swaps of agents* between teams. Edu2Com yields the most recent solution at any time if interrupted by the user or with the solution to which it has converged after many iterations with no improvement. Notably, the polynomial complexity of Edu2Com's first stage along with the anytime nature of its second stage allow us effectively to solve large (in terms of number of agents and number of tasks) problem instances.

In Chapter 5, we conducted a manifold empirical evaluation to study the behaviour and capabilities of the proposed algorithms. At first, we focused on the effectiveness of our heuristic algorithm, Edu2Com, compared to the optimal solver. In some detail, we studied Edu2Com with respect to solution quality, runtime and anytime behaviour when pitched against the state-of-the-art optimal solver IBM CPLEX. Then, we tested our algorithm Edu2Com in solving *real-world* instances of the NOMTMT allocation problem. Specifically, we tasked Edu2Com to solve problems from the educational domain, including *(i)* forming student teams to work on internship programs in collaboration with the Fondazione Bruno Kessler Institute, *(ii)* forming undergraduate student teams to carry out semester projects in collaboration with the Technical University of Crete, and *(iii)* forming master student teams to realise short-term activities in collaboration with the EADA Business School. We explored the scalability of our algorithm in large real-world problems, and we investigated the performance of teams formed with our algorithm compared to teams formed with other current practices in the educational domain.

Through our empirical evaluation, we learned some valuable lessons. First of all, we confirmed that our heuristic algorithm is much more efficient in terms of solving time compared to building an ILP and solving the NOMTMT allocation problem optimally. Specifically, we showed that considering problem instances involving $\sim$ 50 agents and 20 tasks, we can reach the optimal solution of the problem and, at the same time, save up to $\sim$ 65% of time (compared to the time required using an optimal ILP solver). Moreover, we learned that even after the initial stage of the algorithm, we obtained high-quality solutions, making the iteratively improving stage yield high-quality solutions at any time.

By tasking Edu2Com to solve real-world scenarios, we observed that our algorithm was able to handle large problem instances. Specifically, we had the opportunity to test our algorithm in problem instances considering different numbers of agents, numbers of tasks, and required team sizes. Moreover, we studied the limitations of optimally solving the NOMTMT allocation problem, specifically with the state-of-the-art solver CPLEX. The results revealed that while Edu2Com can solve instances involving 100 agents, 50 tasks—considering required team sizes of 2, 3, 4 and 5—CPLEX could not solve the corresponding problems unless the size of teams needed is restricted to 2. Notably, all of the three factors mentioned above (number of agents, number of tasks, required team sizes) affect the time needed to solve the problem. However, the most influential factors are the number of agents and the required team sizes.

Additionally, our experiments allowed us to evaluate the effectiveness of our approach that exploits people's competencies, personalities, and preferences. First, we witnessed that the teams formed with Edu2Com were better (according to experts in team formation) than teams formed by a human expert. Regarding team performance, the results showed that teams with higher collegial values tend to exhibit better performance. Specifically, we observed that Edu2Com formed teams considering collegiality that exhibit better performance than those formed using the teachers' current practices considering solely competencies. Moreover, we learned that teams formed-from-scratch were able to compete with stable teams that have been collaborating for a period of time. Such an observation suggests that collegiality is indeed a good team performance indicator while forming teams. Nonetheless, we also learned that collegiality might not have significant added value to team performance when individuals are very competent or very determined to successfully carry out their activity (regardless of the assigned task), or tasks are not challenging. Notably, from our experience with real-world problems, we see the need to study further how prior experience of a team affects performance.

Finally, in the last part of this dissertation, we delved into Explainable Artificial Intelligence. Following the observations of [Miller, 2018] and [Kraus et al., 2020], we focused on the multi-agent team formation problem and explored how to explain decisions made within team formation scenarios. Thus, in Chapter 6, we introduced a novel generic algorithm for providing *contrastive explanations*. Importantly, our proposed explanatory algorithm wraps existing team formation algorithms with no modifications needed. We build on the notion of facts and foils [Miller, 2018] and explain the teams formed by exploring 'what-if' scenarios. To do so, first, we identified an intuitive and meaningful collection of queries that, in our opinion, cover the main points of interest regarding team formation scenarios. Thereafter, we showed how to handle these queries in order to build an explanation, and we proposed a tailoring mechanism to highlight different perspectives of the explanation. To test our explanatory algorithm we first introduced three evaluation metrics and then conducted an emperical evalutiuon, considering instances of NOMTMT allocation problem, and an optimal solver. We observed that depending on the level of abstraction, our explanations include different number of causes and exhibit different explainable precision. Nonetheless, our results showed that we can build easy-to-read explanations that require the reading level of a high school student.

Last but not least, acknowledging one of the key challenges identified by [Kraus et al., 2020] regarding explainable multiagent environments, we made headway towards preserving privacy upon explaining team formation decisions. As such, we proposed a general framework that combines team formation solutions and explanations over these solutions, while it detects potential privacy breaches upon offering explanations. In some detail, we model the reasoning triggered by explanations using a theory of mind [Frith and Frith, 2005], which allows our proposed breach detector to capture explanations bound to cause breaches.

7.2 Contributions

In this section, we discuss the contributions made through this dissertation by tackling each one of the research questions posed in Section 1.2.

7.2.1 Modeling Human Agents

Reviewing the relevant literature in Computer Science, Organisational Psychology, Motivational Psychology and Social Sciences, we identified some highly-valued components that play vital roles in teamwork and team performance. Thus, we were able to answer our first research question:

> Q.A1: Which human aspects identified in organisational psychology, motivational psychology and social sciences should be considered when building teams?

In accordance with the existing research, we discern four key human characteristics that, when considered properly, boost team performance. Namely, these four characteristics are:

- skills and competencies;
- personality and gender;
- preferences over tasks; and
- preferences over potential teammates.

When a team is competent and sufficiently skilled, it can successfully carry out a task and achieve high-quality outcomes. Similarly, when the team members are *compatible* in terms of their personalities, the team is more likely to prosper and reach better performance. Specifically, [Wilde, 2011] introduced a set of empirical rules for composing teams that, according to empirical findings, boost teams' performance. Moreover, much research suggests that people perform better when motivated by their assigned tasks. That is, when people find a purpose or interest in their assigned task—i.e., they are motivated by their task— they tend to be more productive [Deci et al., 2017]. Existing surveys support that motivated teams, i.e., teams consisting of members interested in their assigned task—are more likely to achieve greater performance. Finally, as reported in existing literature, when team members share strong social bonds, they tend to stick together, and the team is more likely to prosper. Our findings supported the literature observations above during the empirical evaluation that we conducted. Specifically, we evaluated the performance of teams in several real-world case studies (in the educational domain), and our findings confirmed that the aforementioned characteristics have an impact on teamwork and team performance.

Given the above findings, we proceeded to address our second research question:

> QA.2: How can we model the identified beneficial-to-teamwork hu- man characteristics and, therefore, human agents?

We formally modelled human agents with competencies, personality, gender, and preferences over tasks and potential teammates. Considering competencies, we adopted a *graded competence model* over a boolean one since it is richer and closer to reality. That is, in real life, an individual acquires some competence at some *expertise level*. As such, we model an individual's competency with a set of acquired competencies, while each competence is accompanied by either a quantitative or qualitative indicator of the expertise level (e.g., using the level of expertise scale in the Dreyfus model [Dreyfus and Dreyfus, 1980]).

Regarding personality, we followed the quantitative personality theory developed by [Wilde, 2011], according to which we represent an individual's personality with a four real number tuple. An individual's gender is modelled as a categorical value from a finite set of such values. Notably, in contrast to [Andrejczuk, 2018], here we expanded the possible options for gender (e.g., male vs. female) to a broader, more inclusive set of possible genders, as we acknowledge that people may identify themselves differently to only female or male.

To model people's preferences, we considered *preoders*. Due to their desired properties *completeness*, *reflexivity* and *transitivity*, preorders are a common tool used to represent preferences over alternatives in social choice theory [Brandt et al., 2016] and alternative coalitions in hedonic game theory [Chalkiadakis et al., 2011]. Specifically, for each pair of items—which in our thesis corresponds to *(i)* tasks and *(ii)* potential partners—there is a relation indicating that one item is *strictly more preferred* or *at least as much preferred as* or *strictly less preferred* than the other. By using preorders, we can capture an individual's willingness to either work on some task or collaborate with another individual with respect to the rest of the available alternatives.

Moreover, moving from an individual level to a team level, we faced the challenge of integrating individuals' profiles to evaluate teams and, therefore, compare one team with another. Towards this purpose, we proposed metrics that allow us this integration. These metrics include:

- *competence affinity* to measure the collective competency of a team to tackle its assigned task;
- *congeniality* to measure team diversity in terms of personality as proposed by [Andrejczuk et al., 2019], enhanced with a diversity metric in terms of gender that considers a broader set of genders;
- *motivation* to measure team members' collective preference to work on a task;
- *social cohesion* to measure the collective acceptance of the team by its team members; and
- *collegiality* to balance the influence of the four aforementioned metrics.

While devising the metric of competence affinity, we realised that existing competence models are rather limited. According to the existing competence models, an individual is adequate to handle a task if and only if the agent acquires the same competencies as the ones required by the task. However, we acknowledge that an individual may be suitable for tackling a task even if they hold not the same (but similar enough) competencies as the required ones, especially in multidisciplinary environments. As such, in response to our research question

> QA.3: How can we define semantic similarities between competencies to characterise a team's competency for tackling a task?

we adopted the concept of *competence ontology*. More and more countries and organisations construct competence ontologies in an attempt to create a map of competencies across the different domains to bridge, for example, competencies provided by education and those required in industry. Thus, here, we seized the opportunity to use such existing ontologies, and we proposed a method to define semantic similarity between competencies given a competence ontology. By doing so, we enriched the existing competence models towards more realistic ones.

7.2.2 Forming multiple Teams for Multiple Tasks

Given the above, we studied the *Non-Overlapping Many Teams to Many Tasks Allocation Problem*. First, we focused on the complexity of the problem, and we worked towards answering our next research question:

> QB.1: What is the computational cost of forming multiple teams for multiple tasks with no overlaps?

We showed that the search space grows rapidly as the number of agents, tasks, and required team sizes increase, and we provided the means to quantify the search space. Large numbers of agents and tasks, along with large required teams, result in a vast number of alternative solutions since one needs to consider all possible size-compliant combinations of agents for each task. Acknowledging the combinatorial nature of the problem, we showed that solving the NOMTMT allocation problem is $\mathcal{NP}$-complete.

Then, we put forward two solvers for the NOMTMT allocation problem. First, we intended to solve the problem optimally with the means of linear programming. However, this is bound to lead to large linear programs as the number of agents and tasks grows; therefore, solving the problem optimally becomes impractical. Notably, in our empirical evaluation, we presented the limitations of the proposed optimal solver with respect to the problem instance configuration that we can solve within reasonable time limits. Thus, in response to research question

QB.2: Can we efficiently form multiple promising teams for multiple tasks with no overlaps?

we devised Edu2Com, a novel anytime heuristic solver. As we showed through our empirical evaluation, our proposed heuristic approach reaches high-quality solutions. At the same time, it is much faster than a state-of-the-art optimal solver and human experts solving the problem manually.

7.2.3 Explaining Teams and Task Allocations

Finally, we turned our attention to explainable artificial intelligence and specifically to explaining why a team formation algorithm formed some teams and not others. Towards this direction, we identified a collection of 13 intuitive types of queries that, in our view, cover the main points of interest within a team formation environment. First, we discerned queries regarding task allocation, i.e., questioning why certain agents were (or were not) involved in a team working on a specific task. Then, we discerned queries regarding collaborations, i.e., questioning why two or more agents did (or did not) participate in the same team. With this query collection, we answered the following research question:

QC.1: What are the typical queries that team members and team makers will pose?

After that, we concentrated on answering the identified typical queries and building the corresponding explanations. As such, we introduced a novel algorithm for building contrastive explanations as a response to the following research question:

QC.2: How can we build contrastive and comprehensive explanations?

Specifically, we proposed addressing a query by investigating alternative scenarios. Each identified query can be translated into some 'what-if' scenarios that lead to a set of alternative solutions. That is, a solution to a what-if scenario is such that the questioner would feel no need to pose the query at hand. To find a solution to a what-if scenario, our explanatory algorithm reruns the team formation algorithm so that the solution respects some additional query constraints. Then, we can contrast the (best) alternative solution with the solution originally yielded by the team formation algorithm. Therefore, we propose building an explanation utilising the differences spotted between the original and the alternative solution and filling out properly a natural language template. Our empirical evaluation showed that our generated explanations are easy to understand, only requiring the reading level of a high-school student.

Moreover, to address the following research question

QC.3: What is the computational cost of explanations?

we studied the computational cost of building an explanation. As mentioned above, each query corresponds to some what-if scenarios, and for finding a solution to a what-if, we need to rerun the team formation algorithm. As such, we consider the computational cost of explanations in terms of reruns of the team formation algorithm. In particular, we discern queries corresponding to a single what-if scenario, queries corresponding to two what-if scenarios, and queries corresponding to multiple what-if scenarios (linear or quadratic to the number of agents). We presented a detailed table illustrating the number of reruns needed for each one of the typical queries. Notably, regardless of the complexity, all the explanations built with our explanatory algorithms are comprehensive, as shown in our empirical evaluation.

Our proposed algorithm is not ad-hoc for a specific team formation algorithm. Instead, it is a general-purpose explanatory algorithm that can *wrap* around existing team formation algorithms without modifying them, which answers our following research question:

> QC.4: Is there a general-purpose framework for building explanations for team formation algorithms?

Specifically, we acknowledge that it is common for team formation algorithms to use *oracle functions* to determine the adequacy of a team. Our explanatory algorithm considers specially constructed query constraints and, importantly, handles these constraints by wrapping the oracle function and employing large penalties whenever a query constraint is violated. As such, the proposed algorithm for building explanations within team formation scenarios comprises a general framework.

Last but not least, we focused on one of the challenges pointed out by [Kraus et al., 2020] that one should pay attention to when explaining multiagent environments. Specifically, we addressed the challenge of preserving privacy upon explaining team formation scenarios and worked towards answering our last research question:

> QC.5: Can an explanatory algorithm preserve team members' private information?

When we explain teams, providing, for example, contrastive explanations as we proposed in this thesis, we need to provide information regarding the *quality characteristics* of the team at hand. That is, we need to explain which are those characteristics (or reasons) that make one team more preferable than others. Especially considering human teams, we acknowledge that to reason why we form a specific team, we need to delve into people's personal profiles and build an explanation using people's information, which in many cases is private. We argue that providing explanations should guarantee that agents' private information is not disclosed. As such, we proposed a general framework that combines team formation solutions and explanations over these solutions while it detects potential privacy breaches upon offering explanations. In particular, we put forward a privacy breach detector to complement an explanatory algorithm.

Our privacy breach detector emulates the reasoning expected by the explainee upon receiving an explanation using a theory of mind [Frith and Frith, 2005]; assesses an explanation as prone to violate privacy; and notifies the explanatory algorithm accordingly. Therefore, we can preserve privacy by detecting explanations that are about to cause privacy breaches and avoiding such explanations.

7.3 Future Work

In this thesis, we studied human teams, explored how to allocate teams to tasks, and explained team formation scenarios. Despite the contributions made in this work regarding human team formation, we find many research paths that need attention, and we believe that this dissertation can be a springboard for future research. In what follows, we discuss the open challenges organised around two main points of interest: *(i)* forming teams and *(ii)* explaining formed teams.

7.3.1 Forming Teams

Explore additional human factors. In this thesis, we reviewed existing research regarding teamwork from psychology and social sciences, and we singled out four main human factors that are observed to boost teamwork and team performance. However, there exist other factors that are believed to have an impact on teamwork. It is increasingly common when teams are involved to require that the teams are diverse (e.g., in terms of age, nationality, religion, etc.). Crowdsourcing, open innovation events, and social impact tasks often seek teams that include members of different nationalities, ages, or cultural origins. Similarly, in workplaces working groups consist of members coming from different backgrounds (in terms of nationality, race, religion, etc.). At the same time, studies—mainly in the domain of Organisational Psychology—investigate how such factors influence their performance while collaborating [Pesch et al., 2015, Ayub and Jehn, 2018, Minehart and Foldy, 2020]. As such, future work could explore how to integrate more of human nature in automated team formation procedures.

Consider vetos. Even though in this work we considered social relations among people and intended to form socially coherent teams to avoid rivalries within a team, it is quite common that the team maker detects some 'problematic' collaborations that are not captured via social relations. For example, within the educational domain, it is very common that teachers (i.e., the team-makers) specifically request certain students to be placed in different teams, vetoing certain collaborations in this way, regardless of the social relations between these students—for instance, if two close friends are placed in the same team, they may spend time playing around instead of working, undermining the team's performance. This is a rather challenging problem to deal with, especially when we have contradicting vetos. That is, considering a large number of arbitrary collaboration vetos might lead to contradictions and, therefore, to infeasibility. Thus, tackling the problem of forming teams while considering vetos efficiently is an open problem that needs attention.

Partial team formation. In this thesis, we focused on forming human teams (to be allocated to tasks) from scratch. That is, given a set of individuals and a set of tasks, we proposed algorithms that form for each task at most one *complete* team. However, in real-life settings, we often encounter the challenge of completing or reinforcing existing teams. There is a plethora of real-world examples that require partial team formation. In many cases, we see people abandoning their team, making their team *incomplete*. For example, in industry, we often find people resigning or being fired; in schools and universities, we find students dropping out from their classes and projects; in crowdsourcing events, it is common for people to leave, turning (in all these examples) the already formed teams shorthanded and therefore incomplete. Moreover, it is not rare that the needs of an assigned task change (or were initially underestimated), so the already formed team needs to be reinforced with new members for the team to be adequate for the task. Beyond these, considering multidisciplinary teams, we often find the challenge of “synthesising” partial teams, i.e., partially formed teams from different domains need to be matched to form a complete team. All the above are real-world problems that team makers across several domains face. As such, we believe that future work should address the problem of partial team formation.

Permitting overlaps. This work addressed the problem of forming many teams to tackle many tasks without permitting overlaps. Specifically, we disallowed all types of overlaps, i.e., individuals participating in many teams, one team working on multiple tasks, or many teams contributing to the same task. Even though overlaps are undesired in areas such as the educational domain (which is the application area of this thesis) or social impact and crowdsourcing events, in application areas like the industry, overlaps are permitted and, in many cases, welcomed. Therefore, we find work towards overlapping team formation a promising research path for future work.

Boost personal growth through teamwork. An interesting future direction concerns individuals’ personal growth through teamwork. Humans are not static, unchanged entities. Instead, through every single experience, they absorb new knowledge, adapt to new environments, and alter their beliefs, desires, and values. As such, teamwork can help people grow; for instance, participate in teams that allow them to acquire new skills and advance existing ones, collaborate with new people and expand their social network, or engage with people and tasks that open new ways of thinking. In this light, in our opinion, future work should include investigating how an individual personally benefits from teamwork and, one step further, forming teams to boost people's *expected* personal growth.

7.3.2 Explaining Team Formation Scenarios

In this thesis, we made headway towards explaining multiagent team formation environments. Nonetheless, it is not a complete study on the subject; instead, our research scratches the surface and opens up several research paths.

Design ad-hoc explanatory algorithms. In this work, we proposed a general explanatory algorithm that wraps around existing team formation algorithms. Our decision was made upon the observation that many team formation algorithms adopt the notion of *oracle function*, which they consult regarding a team's adequacy. Therefore, our proposed algorithm can wrap around the oracle function and employ penalties when necessary. However, there is a need for further study on explaining team formation algorithms when the oracle function is not accessible or when the algorithm does not use one. Moreover, our proposed algorithm requires rerunning, perhaps several times, the team formation algorithm at hand. This, in large team formation settings, may become impractical and inefficient. Having spotted this obstacle, we strongly believe that future work should include designing explanatory algorithms, perhaps ad-hoc for a given team formation algorithm, that avoids resolving the team formation problem—e.g., by keeping track of the necessary information during the team formation process.

Increase user satisfaction. Kraus et al. highlighted the need for explaining decisions within multiagent environments and identified the key challenges in this direction. Increasing users' satisfaction with received explanations stands among these challenges. As such, in our view, aiming to improve users' satisfaction is important for explainable team formation and provides a fruitful field of research. For instance, target personalised explanations based on user's preferences and needs. Although our proposed approach considers different points of view—that match with different user roles such as participants, the team-maker, and external observers—there is room for personalising explanations. Additionally, we could improve user satisfaction by exploiting the latest advances made in natural language tools, such as ChatGTP. Moreover, one could explore options beyond textual explanations and turn, for example, to visual explanations.

Tackle privacy breaches. As we have already pointed out, another key challenge in explaining multiagent environments is preserving privacy. This dissertation addresses this challenge by proposing a general framework for detecting privacy breaches upon explanations. Nonetheless, how to tackle the detected privacy breaches is yet to be addressed. As such, it is crucial to work in this direction, and there is a great need to explore how to handle privacy breaches, e.g., by generating entirely new explanations or obfuscating current ones.

Repeated interactions. Users might challenge the explanatory algorithm multiple times. That means that a user can have multiple interactions with the explanatory algorithm either because they pose different queries regarding the initial team formation solution or because the previously given explanations do not entirely convince them. In such cases, the explanatory algorithm must ensure that they produce consistent explanations, do not repeat themselves, and, of course, do not disclose private information. Repeated interactions with the explanatory mechanism is a broad and challenging research path that needs attention.

Appendix I

Limitations of optimal solving: Frontier of problem instance configurations that CPLEX can solve

The required time is the average over five problem instances per combination ⟨number of tasks, number of agents, team size⟩.

Number of Tasks	Number of Agents	Team Size	Number of Decision Variables	Required Time(sec)	Standard Deviation(sec)
10	100	2	49500	376.3446	17.45
20	100	2	99000	836.6631	17.29
30	100	2	148500	1153.8280	28.03
40	100	2	198000	1958.0188	74.93
50	100	2	247500	2305.5095	90.82
10	70	3	547400	4588.9023	74.45
20	60	3	684400	5505.4640	94.48
30	50	3	588000	5107.8094	181.27
40	40	3	395200	4045.5223	73.42
50	40	3	494000	4188.8293	179.38

Number of Tasks	Number of Agents	Team Size	Number of Decision Variables	Required Time(sec)	Standard Deviation(sec)
10	30	4	274050	2386.0373	280.69
20	30	4	548100	5377.4400	408.94
30	20	4	145350	1328.7434	152.06
40	20	4	193800	1776.9613	193.64
50	20	4	242250	2228.5940	20.16
10	20	5	155040	1539.4960	243.25
20	20	5	310080	2906.0331	746.35
30	20	5	465120	6243.2947	61.33
40	20	5	620160	8438.8540	230.19
50	20	5	775200	8468.0339	130.67

Table I .1: Frontier of problem instance configurations (number of tasks, number of agents, team size) along with the number of decision variables and required solving time.

Appendix *II*

Case study: Technical University of Crete (TUC) Questionaires

Here, we provide the questionnaire used in our experiment in collaboration with the Technical University of Crete. The students answered questionnaires regarding their competencies, their personality, and their preferences over different semester projects and potential partners. All questionnaires were answered in the Greek language. In what follows, we provide all four questionnaires in both English and Greek.

Competence Self Assessment. We asked students to self-assess themselves concerning a collection of 21 competencies described in the ESCO ontology. We used a five-level Likert scale [Likert, 1932] to model the expertise level:

English	Greek	Likert scale item
Novice	Αρχάριος/Αρχάρια	0.2
Advanced Beginner	Ερασιτέχνης	0.4
Competent	Ικανός/Ικανή	0.6
Proficient	Έμπειρος/Έμπειρη	0.8
Expert	Ειδικός/Ειδική	1.0

Then, the students characterised the competencies below with respect to their expertise level:

1. **English:** Software and Applications Development and Analysis
   Greek: *Ανάπτυξη και Ανάλυση Λογισμικού και Εφαρμογών*
   ESCO url: `http://data.europa.eu/esco/isced-f/0613`

2. **English:** Database and Network Design and Administration
   Greek: Σχεδιασμός και Διαχείριση Βάσεων Δεδομένων και Δικτύων
   ESCO url: `http://data.europa.eu/esco/isced-f/0612`

3. **English:** Communication, Collaboration and Creativity
   Greek: Επικοινωνία, Συνεργασία και Δημιουργικότητα
   ESCO url: `http://data.europa.eu/esco/skill/S1.0.0`

4. **English:** Liaising and Networking
   Greek: Επικοινωνία και Δικτύωση
   ESCO url: `http://data.europa.eu/esco/skill/S1.2.0`

5. **English:** Solving Problems
   Greek: Επίλυση Προβλημάτων
   ESCO url: `http://data.europa.eu/esco/skill/S1.9.0`

6. **English:** Evaluating Systems, Programmes, Equipment and Products
   Greek: Αξιολόγηση Συστημάτων, Προγραμμάτων, Εξοπλισμού και Προϊόντων
   ESCO url: `http://data.europa.eu/esco/skill/f2cf57fe-d4cb-4b4a-831d-73171cc73909`

7. **English:** Organising, Planning and Scheduling Work and Activities
   Greek: Οργάνωση, Σχεδιασμός και Προγραμματισμός Εργασιών και Δραστηριοτήτων
   ESCO url: `http://data.europa.eu/esco/skill/S4.2.0`

8. **English:** Presenting Information
   Greek: Παρουσίαση Πληροφοριών
   ESCO url: `http://data.europa.eu/esco/skill/S1.4`

9. **English:** Technical or Academic Writing
   Greek: Τεχνική ή Επιστημονική Συγγραφή
   ESCO url: `http://data.europa.eu/esco/skill/S1.13.3`

10. **English:** Greek Language (Ability to comprehend spoken and written Greek and to speak and write in Greek)
    Greek: Ελληνική Γλώσσα (Δυνατότητα κατανόησης προφορικού και γραπτού λόγου στα Ελληνικά και ανάπτυξης προφορικού και γραπτού λόγου στα Ελληνικά)
    ESCO url: `http://data.europa.eu/esco/skill/ea4ebfa1-e17a-4416-ac54-955f33e6ade7`

11. **English:** English Language (Ability to comprehend spoken and written English and to speak and write in English)
    Greek: Αγγλική Γλώσσα (Δυνατότητα κατανόησης προφορικού και γραπτού λόγου στα Αγγλικά και ανάπτυξης προφορικού και γραπτού λόγου στα Αγγλικά)
    ESCO url: `http://data.europa.eu/esco/skill/6d3edede-8951-4621-a835-e04323300fa0`

12. **English:** Familiar with Accommodating Bussiness
    Greek: Εξοικείωση με Επιχειρήσεις Καταλυμάτων
    ESCO url: `http://data.europa.eu/esco/occupation/53cdd34e-22f2-41bd-b3c7-de22bf9bbcae`
13. **English:** Familiar with Construction Companies
    Greek: Εξοικείωση με Κατασκευαστικές Εταιρίες
    ESCO url: `http://data.europa.eu/esco/occupation/faed05c0-c1d1-4e34-b575-0dea96459e56`
14. **English:** Familiar with Clinics and Healthcare Institutions
    Greek: Εξοικείωση με Ιατρεία, Κλινικές και Οργανισμούς Υγείας
    ESCO url: `http://data.europa.eu/esco/occupation/4a29eab1-5f02-4723-a863-e2d5a0614dfa`
15. **English:** Familiar with Event Centers
    Greek: Εξοικείωση με Κέντρα Εκδηλώσεων
    ESCO url: `http://data.europa.eu/esco/occupation/1b38a27d-ef98-4d9f-b1b2-8c109bf47e79`
16. **English:** Familiar with Food Service Bissiness
    Greek: Εξοικείωση με Επιχειρήσεις Εστίασης
    ESCO url: `http://data.europa.eu/esco/occupation/d5eb6150-bbff-4a9c-9d0c-21eab4dbe2b6`
17. **English:** Familiar with Lending Libraries
    Greek: Εξοικείωση με Δανειστικές Βιβλιοθήκες
    ESCO url: `http://data.europa.eu/esco/occupation/24d39e12-e104-49d3-8224-8b7a5f9b99d1`
18. **English:** Familiar with Pharmacy Stores
    Greek: Εξοικείωση με Φαρμακεία
    ESCO url: `http://data.europa.eu/esco/occupation/1b3e150f-8ec1-47e2-a2ef-d02632efe0d5`
19. **English:** Familiar with Rental Bussiness
    Greek: Εξοικείωση με Επιχειρήσεις Ενοικίασης
    ESCO url: `http://data.europa.eu/esco/occupation/de31a27f-c6ba-4d4f-87cd-b405e1852121`
20. **English:** Familiar with Retail Bussiness
    Greek: Εξοικείωση με Επιχειρήσεις Λιανικής
    ESCO url: `http://data.europa.eu/esco/occupation/9e81adde-9983-44fa-b74b-c548d0dbfbdd`
21. **English:** Familiar with Training and Tutoring Schools
    Greek: Εξοικείωση με Εκπαιδευτήρια
    ESCO url: `http://data.europa.eu/esco/isco/C1345`

Personality Test. We asked students to answer the Post-Jungian Personality test developed by [Wilde, 2013]. Below, we provide first the personality test in English and then the translated test in Greek.

English.

1.	(TF)	Judges should be:	impartial (T)	merciful (F)	neither (-)
2.	(PJ)	You prefer things:	open-ended (P)	planned (J)	neither (-)
3.	(SN)	You prefer the:	traditional (S)	novel (N)	neither (-)
4.	(EI)	You prefer:	groups (E)	individuals (I)	neither (-)
5.	(TF)	You are more:	skeptical (T)	tolerant (F)	neither (-)
6.	(PJ)	You work better:	pressured (P)	unpressured (J)	neither (-)
7.	(PJ)	You are more:	improviser (P)	methodical (J)	neither (-)
8.	(SN)	You are more:	practical (S)	theoretical (N)	neither (-)
9.	(EI)	You are more:	sociable (E)	reserved (I)	neither (-)
10.	(TF)	You are more:	curious (T)	accommodating (F)	neither (-)
11.	(EI)	You are more:	expressive (E)	contained (I)	neither (-)
12.	(TF)	You are more:	honest (T)	diplomatic (F)	neither (-)
13.	(SN)	You prefer the:	specific (S)	abstract (N)	neither (-)
14.	(EI)	You are more:	talkative (E)	quiet (I)	neither (-)
15.	(EI)	You learn better by:	listening (E)	reading (I)	neither (-)
16.	(SN)	You are more:	practical (S)	conceptual (N)	neither (-)
17.	(TF)	You prefer:	logic (T)	empathy (F)	neither (-)
18.	(SN)	You prefer to:	investigate (S)	speculate (N)	neither (-)
19.	(PJ)	You are more:	informal (P)	systematic (J)	neither (-)
20.	(PJ)	You prefer:	variety (P)	routine (J)	neither (-)

Greek.

1.	(TF)	Ένας δικαστής πρέπει να είναι:	αμερόληπτος (T)	συμπονετικός (F)	Τίποτα από τα δύο (-)
2.	(PJ)	Προτιμάτε τον:	αυτοσχεδιασμό (P)	προσχεδιασμός (J)	Τίποτα από τα δύο (-)
3.	(SN)	Προτιμάτε το:	παραδοσιακό (S)	μοντέρνο (N)	Τίποτα από τα δύο (-)
4.	(EI)	Προτιμάτε να βρίσκεστε με:	παρέες (E)	μεμονωμένα άτομα (I)	Τίποτα από τα δύο (-)
5.	(TF)	Είστε περισσότερο:	δύσπιστος/η (T)	δεκτικός/ή (F)	Τίποτα από τα δύο (-)
6.	(PJ)	Αποδίδετε καλύτερα:	υπό πίεση (P)	χωρίς πίεση (J)	Τίποτα από τα δύο (-)
7.	(PJ)	Είστε περισσότερο:	αυθόρμητος/η (P)	μεθοδικός/η (J)	Τίποτα από τα δύο (-)
8.	(SN)	Είστε περισσότερο της:	πράξης (S)	θεωρίας (N)	Τίποτα από τα δύο (-)
9.	(EI)	Είστε περισσότερο:	κοινωνικός/ή (E)	επιφυλακτικός/ή (I)	Τίποτα από τα δύο (-)
10.	(TF)	Είστε περισσότερο:	περίεργος/η (T)	συμβιβαστικός/ή (F)	Τίποτα από τα δύο (-)
11.	(EI)	Είστε περισσότερο:	εκφραστικός (E)	εγκρατής (I)	Τίποτα από τα δύο (-)
12.	(TF)	Είστε περισσότερο:	ειλικρινής (T)	διπλωματικός/ή (F)	Τίποτα από τα δύο (-)
13.	(SN)	Προτιμάτε το:	συγκεκριμένο (S)	αφηρημένο (N)	Τίποτα από τα δύο (-)
14.	(EI)	Είστε περισσότερο:	ομιλητικός (E)	σιωπηλός (I)	Τίποτα από τα δύο (-)
15.	(EI)	Μαθαίνετε ευκολότερα:	ακούγοντας (E)	διαβάζοντας (I)	Τίποτα από τα δύο (-)
16.	(SN)	Είστε περισσότερο της:	πράξης (S)	σκέψης (N)	Τίποτα από τα δύο (-)
17.	(TF)	Προτιμάτε την:	λογική (T)	εν-συναίσθηση (F)	Τίποτα από τα δύο (-)
18.	(SN)	Προτιμάτε να:	ερευνάτε (S)	εικάζεται (N)	Τίποτα από τα δύο (-)
19.	(PJ)	Είστε περισσότερο:	άτυπος (P)	συστηματικός (J)	Τίποτα από τα δύο (-)
20.	(PJ)	Προτιμάτε την:	ποικιλία (P)	ρουτίνα (J)	Τίποτα από τα δύο (-)

Preferences over Projects. We asked students to declare their preferences over the different types of business they would like to work on during their semester project. We used a five-level Likert scale to evaluate how much an individual would like to work on a specific type of business:

English	Greek	Likert scale item
Not at all interested	Δεν ενδιαφέρομαι καθόλου	0.0
Not so interested	Δεν ενδιαφέρομαι ιδιαίτερα	0.25
Somewhat interested	Ενδιαφέρομαι λίγο	0.5
Very interested	Ενδιαφέρομαι πολύ	0.75
Extremely interested	Ενδιαφέρομαι απόλυτα	1.0

Then we asked them to rate how much they would like to work on each of the types of business below:

1. **English:** Accommodating Bussiness (hotels, hostels, camps)
   Greek: Επιχειρήσεις Καταλυμάτων (ξενοδοχεία, ξενώνες, κατασκηνώσεις)

2. **English:** Construction Company
   Greek: Κατασκευαστικές Εταιρίες

3. **English:** Private Doctors' Offices / Clinics
   Greek: Εξοικείωση με Ιατρεία / Κλινικές

4. **English:** Event Centers
   Greek: Κέντρα Εκδηλώσεων

5. **English:** Food Service Business (restaurants, cafeterias, bars)
   Greek: Επιχειρήσεις Εστίασης (εστιατόρια, καφετέριες, μπαρ)

6. **English:** Lending Libraries
   Greek: Δανειστικές Βιβλιοθήκες

7. **English:** Pharmacy Stores
   Greek: Φαρμακεία

8. **English:** Rental Bussiness (car, boat, bikes, etc.)
   Greek: Εξοικείωση με Επιχειρήσεις Ενοικίασης (αυτοκινήτων, σκαφών, ποδηλάτων, κλπ.)

9. **English:** Retail Bussiness (clothing, shoe, toy, etc.)
   Greek: Επιχειρήσεις Λιανικής (ρούχων, υποδημάτων, παιχνιδιών, κλπ.)

10. **English:** Familiar with Training and Tutoring Schools (foreign language schools, dance schools, conservatories, etc.)
    Greek: Εξοικείωση με Εκπαιδευτήρια (φροντιστήρια ξένων γλωσσών, σχολές χορού, ωδεία, κλπ.)

Preferences over Potential Partners. Finally, we asked the students to indicate the top five most preferred individuals they would like to work with, along with the top five least preferred individuals. Notably, we asked students to rank the potential partners so that the 1^{st} one is more/less preferred than the 2^{nd}, who in turn is more/less preferred than the 3^{rd}, and so on.

Potential partners I would **like to** work with *Πιθανοί συνεργάτες με τους οποίους* ***θα ήθελα*** *να συνεργαστώ*		
1^{st}.	Name/ Όνομα	Surname / Επώνυμο
2^{nd}.	Name/ Όνομα	Surname / Επώνυμο
3^{rd}.	Name/ Όνομα	Surname / Επώνυμο
4^{th}.	Name/ Όνομα	Surname / Επώνυμο
5^{th}.	Name/ Όνομα	Surname / Επώνυμο

Potential partners I would **like not to** work with *Πιθανοί συνεργάτες με τους οποίους* ***δεν θα ήθελα*** *να συνεργαστώ*		
1^{st}.	Name/ Όνομα	Surname / Επώνυμο
2^{nd}.	Name/ Όνομα	Surname / Επώνυμο
3^{rd}.	Name/ Όνομα	Surname / Επώνυμο
4^{th}.	Name/ Όνομα	Surname / Επώνυμο
5^{th}.	Name/ Όνομα	Surname / Επώνυμο

Explanation Natural Language Templates

To build explanations, we first express the "what-if" scenario depending on the query template, as shown in Table III .1. Then we follow the procedure below:

Individual View. The individual view focuses on the agents identified in the query. For each identified agent a we:

1. select the feature r that exhibits the largest relative difference *in absolute value*, i.e., $|\Delta f_r^{IV}(a, g')| = \max_{f_i \in F} |\Delta f_i^{IV}(a, g')|$;
2. we characterise the relative distance of feature r as "dramatic" in case $|\Delta f_r^{IV}(a, g')| > 75\%$; and as "slight" in case $|\Delta f_r^{IV}(a, g')| < 25\%$; and
3. we discern if the relative difference of feature r corresponds to a *gain* (when $\Delta f_r^{IV}(a, g') > 0$), a *loss* (when $\Delta f_r^{IV}(a, g') < 0$), or a *neutrality* (when $\Delta f_r^{IV}(a, g') = 0$).

Then we fill out the template:
"… agent {agent id} would have had to take part in task {alternative task id}, for which they are $\left\{ \begin{array}{ll} \text{\{dramatically/$\cdot$ /slightly\} \{more / less\} \{skilled/ satisfied/ etc.\}} & \text{, if } \Delta f_r^{IV} \neq 0 \text{, or} \\ \text{\{equally skilled/satisfied, etc\}} & \text{, otherwise} \end{array} \right\}$".

Local View. The local view focuses on each task and the teams assigned to it according to the original and the query-compliant allocation. For each task τ we:

1. select the feature r that exhibits the largest relative difference *in absolute value*, i.e., $|\Delta f_r^{LV}(\tau, g')| \equiv \max_{f_i \in F} |\Delta f_i^{LV}(\tau, g')|$;

Query Code	What-if scenario
Q1	If agent a_i was not assigned to task τ then ...
Q2	If agent a_i was assigned to task τ then ...
Q3	If team $K = \{a_1, \cdots, a_{\vert K\vert}\}$ was not assigned to task τ then ...
Q4	If team $K = \{a_1, \cdots, a_{\vert K\vert}\}$ was assigned to task τ then ...
Q5	If agent a_j was assigned to task τ, instead of a_i then ...
Q6	If agent a_j was assigned to task τ, while a_i was not then ...
Q7	If agent a_i was not in team $K = \{a_1, \cdots, a_{\vert K\vert}\}$ then ...
Q8	If team $K = \{a_1, \cdots, a_{\vert K\vert}\}$ was not formed then ...
Q9	If team $K = \{a_1, \cdots, a_{\vert K\vert}\}$ was formed then ...
Q10	If agents a_i and a_j were not in the same team then ...
Q11	If agents a_i and a_j were in the same team then ...
Q12	If agent a_x was in team $K = \{a_1, \cdots, a_{\vert K\vert}\}$ instead of a_i then ...
Q13	If agent a_x was in team $K = \{a_1, \cdots, a_{\vert K\vert}\}$ while a_i was not then ...

Table III .1: Natural language explanation templates for each query template.

2. we characterise the relative distance of feature r as "dramatic" in case $|\Delta f_r^{LV}(\tau, g')| > 75\%$; and as "slight" in case $|\Delta f_r^{LV}(\tau, g')| < 25\%$—in case that $25\% \leq |\Delta f_r^{LV}(\tau, g')| \leq 75\%$; and

3. we discern if the relative difference of feature r corresponds to a *gain* (when $\Delta f_r^{LV}(\tau, g') > 0$), a *loss* (when $\Delta f_r^{LV}(\tau, g') < 0$), or a *neutrality* (when $\Delta f_r^{LV}(\tau, g') = 0$).

Then, we compute the percentage (%) of teams that exhibit "dramatic gain" ($prc_{\text{dramatic gain}}$), "gain" ($prc_{\text{gain}}$), "slight gain" ($prc_{\text{slight gain}}$), "dramatic loss" ($prc_{\text{dramatic loss}}$), "gain" ($prc_{\text{loss}}$), "slight loss" ($prc_{\text{slight loss}}$), and "neutrality" ($prc_{\text{neutrality}}$).

The explanation template for the local view is:
"... $\Big\{$\{$prc_{\text{dramatic loss}}$\} of the tasks would have been assigned to a dramatically less {skilled/ satisfying/ compatible/ etc}[1] team$\Big\}$, $\Big\{$\{prc_{loss}\} of the tasks would have been assigned to a less {skilled/ satisfying/ compatible/ etc} team$\Big\}$, and $\Big\{$\{$prc_{\text{slight loss}}$\} of the tasks would have been assigned to a slightly less {skilled/ satisfying/ compatible/ etc} team$\Big\}$.

[1]Include all the features that exhibit dramatic loss.

On the other hand $\Big\{\{prc_{\text{dramatic gain}}\}$ of the tasks would have been assigned to a dramatically more {skilled/ satisfying/ compatible/ etc} team$\Big\}$, $\Big\{\{prc_{\text{gain}}\}$ of the tasks would have been assigned to a more {skilled/ satisfying/ compatible/ etc} team$\Big\}$, and $\Big\{\{prc_{\text{slight gain}}\}$ of the tasks would have been assigned to a slightly more {skilled/ satisfying/ compatible/ etc} team$\Big\}$.

$\Big\{$While $prc_{\text{neutrality}}$ of the tasks would have been assigned to equally skilled, compatible and satisfied teams.$\Big\}$ ".

When a percentage is at 0%, we disregard the corresponding part.

Global View. The global view focuses on the quality of the overall allocation. For each feature $f_i \in F$ we compute the relative differences $\Delta f_i^{GV}(g')$. Moreover, we compute the relative difference of the matching adequacy of the teams in the original allocation g and the teams in the query-compliant allocation g':

$$\Delta u^{GV}(g') = \frac{\mathcal{F}_{\tau \in \mathcal{T}}\, u\big(g(\tau), \tau, F\big) - \mathcal{F}_{\tau \in \mathcal{T}}\, u\big(g'(\tau), \tau, F\big)}{\mathcal{F}_{\tau \in \mathcal{T}}\, u\big(g(\tau), \tau, F\big)} = \frac{v(g) - v(g')}{v(g)}$$

Then, the explanation template for the global view is:

"… the overall matching of teams to task would be
$\left\{ \begin{array}{ll} |\Delta f_i^{GV}(g')| \text{ \{less / more\} \{skilled/ satisfying/ compatible/ etc\}} & \text{, if } \Delta f_i^{GV} \neq 0 \text{, or} \\ \text{\{equally skilled/satisfied/compatible, etc\}} & \text{, otherwise} \end{array} \right., \forall f_i \in F \Big\}$. Thus, the alternative allocation would be $|\Delta u^{GV}(g')|$ {less/more}-suitable considering all features."

Bibliography

[APA, 2023] (2023). American Psychological Association: Industrial and Organisational Psychology. `https://www.apa.org/ed/graduate/specialize/industrial`. Accessed: 2023.

[Abdollahi and Nasraoui, 2017] Abdollahi, B. and Nasraoui, O. (2017). Using explainability for constrained matrix factorization. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*, RecSys '17, page 79–83, New York, NY, USA. Association for Computing Machinery.

[Abdulsalam and Mawoli, 2012] Abdulsalam, D. and Mawoli, M. A. (2012). Motivation and job performance of academic staff of state universities in nigeria: the case of ibrahim badamasi babangida university, lapai, niger state. *International Journal of Business and management*, 7(14):142.

[Adadi and Berrada, 2018] Adadi, A. and Berrada, M. (2018). Peeking inside the black-box: A survey on explainable artificial intelligence (xai). *IEEE Access*, 6:52138–52160.

[Almagor and Lahijanian, 2020] Almagor, S. and Lahijanian, M. (2020). Explainable multi agent path finding. AAMAS '20, page 34–42, Richland, SC. International Foundation for Autonomous Agents and Multiagent Systems.

[Anagnostopoulos et al., 2010] Anagnostopoulos, A., Becchetti, L., Castillo, C., Gionis, A., and Leonardi, S. (2010). Power in unity: Forming teams in large-scale community systems. pages 599–608.

[Anagnostopoulos et al., 2012] Anagnostopoulos, A., Becchetti, L., Castillo, C., Gionis, A., and Leonardi, S. (2012). Online team formation in social networks. In *Proceedings of the 21st International Conference on World Wide Web*, WWW '12, page 839–848, New York, NY, USA. Association for Computing Machinery.

[Andrejczuk, 2018] Andrejczuk, E. (2018). Artificial intelligence methods to support people management in organisations.

[Andrejczuk et al., 2018] Andrejczuk, E., Berger, R., Rodríguez-Aguilar, J. A., Sierra, C., and Marín-Puchades, V. (2018). The composition and formation of effective teams: computer science meets organizational psychology. *Knowledge Eng. Review*, 33:e17.

[Andrejczuk et al., 2019] Andrejczuk, E., Bistaffa, F., Blum, C., Rodríguez-Aguilar, J. A., and Sierra, C. (2019). Synergistic team composition: A computational approach to foster diversity in teams. *Knowledge-Based Systems*, 182(104799).

[Arrieta et al., 2020] Arrieta, A. B., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., García, S., Gil-López, S., Molina, D., Benjamins, R., et al. (2020). Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information fusion*, 58:82–115.

[Ayub and Jehn, 2018] Ayub, N. and Jehn, K. A. (2018). Exploring diversity effects: nationality composition and nationality context in workgroups. *European Journal of Work and Organizational Psychology*, 27(5):616–628.

[Baard et al., 2004] Baard, P. P., Deci, E. L., and Ryan, R. M. (2004). Intrinsic need satisfaction: a motivational basis of performance and weil-being in two work settings 1. *Journal of applied social psychology*, 34(10):2045–2068.

[Bachrach et al., 2010] Bachrach, Y., Meir, R., Jung, K., and Kohli, P. (2010). Coalitional structure generation in skill games. volume 2.

[Back, 1951] Back, K. W. (1951). Influence through social communication. *The Journal of Abnormal and Social Psychology*, 46(1):9.

[Ballesteros-Perez et al., 2012] Ballesteros-Perez, P., González-Cruz, M. C., and Fernández-Diego, M. (2012). Human resource allocation management in multiple projects using sociometric techniques. *International Journal of Project Management*, 30(8):901–913.

[Barrick et al., 2001] Barrick, M. R., Mount, M. K., and Judge, T. A. (2001). Personality and performance at the beginning of the new millennium: What do we know and where do we go next? *International Journal of Selection and assessment*, 9(1-2):9–30.

[Beal and Cohen, 2003] Beal, D. J. and Cohen, R. R. (2003). Cohesion and performance in groups: a meta-analytic clarification of construct relations. *Journal of applied psychology*, 88(6):989.

[Belbin, 1993] Belbin, R. (1993). *Team Roles at Work*. Butterworth-Heinemann.

[Bergner, 2020] Bergner, R. M. (2020). What is personality? two myths and a definition. *New Ideas in Psychology*, 57:100759.

[Boixel and Endriss, 2020] Boixel, A. and Endriss, U. (2020). Automated justification of collective decisions via constraint solving. In *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '20, page 168–176, Richland, SC. International Foundation for Autonomous Agents and Multiagent Systems.

[Brandt et al., 2016] Brandt, F., Conitzer, V., Endriss, U., Lang, J., and Procaccia, A. D. (2016). *Handbook of computational social choice*. Cambridge University Press.

[Briggs et al., 1995] Briggs, I. M., Myers, I. B., and Myers, P. B. (1995). *Gifts differing: Understanding personality type*. Davies-Black.

[Bruffee, 1993] Bruffee, K. A. (1993). *Collaborative learning*.

[Capezzuto et al., 2020] Capezzuto, L., Tarapore, D., and Ramchurn, S. D. (2020). Anytime and efficient coalition formation with spatial and temporal constraints.

[Carron et al., 2002] Carron, A. V., Colman, M. M., Wheeler, J., and Stevens, D. (2002). Cohesion and performance in sport: A meta analysis. *Journal of sport and exercise psychology*, 24(2):168–188.

[Carvalho et al., 2019] Carvalho, D. V., Pereira, E. M., and Cardoso, J. S. (2019). Machine learning interpretability: A survey on methods and metrics. *Electronics*, 8(8).

[Chalkiadakis and Boutilier, 2012] Chalkiadakis, G. and Boutilier, C. (2012). Sequentially optimal repeated coalition formation under uncertainty. *Autonomous Agents and Multi-Agent Systems*, 24(3):441–484.

[Chalkiadakis et al., 2011] Chalkiadakis, G., Elkind, E., and Wooldridge, M. (2011). *Computational Aspects of Cooperative Game Theory (Synthesis Lectures on Artificial Inetlligence and Machine Learning)*. Morgan & Claypool Publishers, 1st edition.

[Chevaleyre et al., 2006] Chevaleyre, Y., Dunne, P., Endriss, U., Lang, J., Lemaitre, M., Maudet, N., Padget, J., Phelps, S., Rodrguez-Aguilar, J., and Sousa, P. (2006). Issues in multiagent resource allocation. *Informatica*, 30:3–31.

[Cohen and Bailey, 1997] Cohen, S. G. and Bailey, D. E. (1997). What makes teams work: Group effectiveness research from the shop floor to the executive suite. *Journal of management*, 23(3):239–290.

[Conitzer and Sandholm, 2002] Conitzer, V. and Sandholm, T. (2002). Complexity of manipulating elections with few candidates. In *Eighteenth National Conference on Artificial Intelligence*, page 314–319, USA. American Association for Artificial Intelligence.

[Crawford et al., 2016] Crawford, C., Rahaman, Z., and Sen, S. (2016). Evaluating the efficiency of robust team formation algorithms. In Osman, N. and Sierra, C., editors, *Autonomous Agents and Multiagent Systems*, pages 14–29, Cham. Springer International Publishing.

[Czarnecki and Dutta, 2021] Czarnecki, E. and Dutta, A. (2021). Scalable hedonic coalition formation for task allocation with heterogeneous robots. *Intelligent Service Robotics*, 14(3):501–517.

[Czatnecki and Dutta, 2019] Czatnecki, E. and Dutta, A. (2019). Hedonic coalition formation for task allocation with heterogeneous robots. In *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, pages 1024–1029.

[da Silva and Krohling, 2018] da Silva, I. E. and Krohling, R. A. (2018). A fuzzy sociometric approach to human resource allocation. In *2018 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pages 1–8. IEEE.

[Deci et al., 2017] Deci, E. L., Olafsen, A. H., and Ryan, R. M. (2017). Self-determination theory in work organizations: The state of a science. *Annual Review of Organizational Psychology and Organizational Behavior*, 4(1):19–43.

[Deci and Ryan, 1985] Deci, E. L. and Ryan, R. M. (1985). *Intrinsic Motivation and Self-Determination in Human Behavior.* Springer New York, NY.

[Deist and Winterton, 2005] Deist, F. D. L. and Winterton, J. (2005). What is competence? *Human Resource Development International*, 8(1):27–46.

[Došilović et al., 2018] Došilović, F. K., Brčić, M., and Hlupić, N. (2018). Explainable artificial intelligence: A survey. In *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pages 0210–0215.

[Dreyfus and Dreyfus, 1980] Dreyfus, S. E. and Dreyfus, H. L. (1980). A five-stage model of the mental activities involved in directed skill acquisition.

[Driskell et al., 2006] Driskell, J. E., Goodwin, G. F., Salas, E., and O'Shea, P. G. (2006). What makes a good team player? personality and team effectiveness. *Group Dynamics: Theory, Research, and Practice*, 10(4):249.

[Ellemers et al., 2004] Ellemers, N., De Gilder, D., and Haslam, S. A. (2004). Motivating individuals and groups at work: A social identity perspective on leadership and group performance. *Academy of Management review*, 29(3):459–478.

[Faliszewski et al., 2007] Faliszewski, P., Hemaspaandra, E., Hemaspaandra, L. A., and Rothe, J. (2007). Llull and copeland voting broadly resist bribery and control. AAAI'07, page 724–730. AAAI Press.

[Festinger et al., 1950] Festinger, L., Schachter, S., and Back, K. (1950). Social pressures in informal groups; a study of human factors in housing.

[Foss et al., 2008] Foss, N. J., Klein, P. G., Kor, Y. Y., and Mahoney, J. T. (2008). Entrepreneurship, subjectivism, and the resource-based view: toward a new synthesis. *Strategic Entrepreneurship Journal*, 2(1):73–94.

[Friedkin, 2004] Friedkin, N. E. (2004). Social cohesion. *Annual Review of Sociology*, 30(1):409–425.

[Frith and Frith, 2005] Frith, C. and Frith, U. (2005). Theory of mind. *Current biology*, 15(17):R644–R645.

[Furnham, 1996] Furnham, A. (1996). The big five versus the big four: the relationship between the myers-briggs type indicator (mbti) and neo-pi five factor model of personality. *Personality and individual differences*, 21(2):303–307.

[Gallie et al., 2009] Gallie, D., Zhou, Y., Felstead, A., and Green, F. (2009). Teamwork, productive potential and employee welfare.

[Gamrath et al., 2020] Gamrath, G., Anderson, D., Bestuzheva, K., Chen, W.-K., Eifler, L., Gasse, M., Gemander, P., Gleixner, A., Gottwald, L., Halbig, K., Hendel, G., Hojny, C., Koch, T., Le Bodic, P., Maher, S. J., Matter, F., Miltenberger, M., Mühmer, E., Müller, B., Pfetsch, M. E., Schlösser, F., Serrano, F., Shinano, Y., Tawfik, C., Vigerske, S., Wegscheider, F., Weninger, D., and Witzig, J. (2020). The SCIP Optimization Suite 7.0. Technical report, Optimization Online.

[Georgara et al., 2023] Georgara, A., Kazhamiakin, R., Mich, O., Palmero Approsio, A., Pazzaglia, J.-C., Rodríguez Aguilar, J. A., and Sierra, C. (2023). The ai4citizen pilot: Pipelining ai-based technologies to support school-work alternation programmes. *Applied Intelligence*.

[GLPK, 2018] GLPK (2018). Glpk: Gnu linear programming kit). `https://www.gnu.org/software/glpk/`.

[Goethals et al., 2022] Goethals, S., Sörensen, K., and Martens, D. (2022). The privacy issue of counterfactual explanations: explanation linkage attacks. *arXiv preprint arXiv:2210.12051*.

[Goodman and Flaxman, 2017] Goodman, B. and Flaxman, S. (2017). European union regulations on algorithmic decision-making and a "right to explanation". *AI Magazine*, 38(3):50–57.

[Gunning, 1952] Gunning, R. (1952). *The Technique of Clear Writing*. McGraw-Hill.

[GUROBI, 2018] GUROBI (2018). Gurobi optimizer 8.0. `https://www.gurobi.com/`.

[Gutiérrez et al., 2016] Gutiérrez, J. H., Astudillo, C. A., Ballesteros-Pérez, P., Mora-Melià, D., and Candia-Véjar, A. (2016). The multiple team formation problem using sociometry. *Comput. Oper. Res.*, 75(C):150–162.

[Hager and Gonczi, 1996] Hager, P. and Gonczi, A. (1996). What is competence? *Medical teacher*, 18(1):15–18.

[Harwood and Thrower, 2020] Harwood, C. G. and Thrower, S. N. (2020). Chapter 9 - motivational climate in youth sport groups. In Bruner, M. W., Eys, M. A., and Martin, L. J., editors, *The Power of Groups in Youth Sport*, pages 145–163. Academic Press.

[Higgs, 1996] Higgs, M. (1996). A comparison of myers briggs type indicator profiles and belbin team roles. *WORKING PAPER SERIES-HENLEY MANAGEMENT COLLEGE HWP*.

[Higgs et al., 2005] Higgs, M., Plewnia, U., and Ploch, J. (2005). Influence of team composition and task complexity on team performance. *Team Performance Management: An International Journal*, 11(7/8):227–250.

[Hoegl and Gemuenden, 2001] Hoegl, M. and Gemuenden, H. G. (2001). Teamwork quality and the success of innovative projects: A theoretical concept and empirical evidence. *Organization science*, 12(4):435–449.

[Hoegl and Parboteeah, 2007] Hoegl, M. and Parboteeah, K. P. (2007). Creativity in innovative projects: How teamwork matters. *Journal of engineering and technology management*, 24(1-2):148–166.

[Horn et al., 1994] Horn, J., Nafpliotis, N., and Goldberg, D. (1994). A niched pareto genetic algorithm for multiobjective optimization. In *Proceedings of the First IEEE Conference on Evolutionary Computation. IEEE World Congress on Computational Intelligence*, pages 82–87 vol.1.

[Howard et al., 2016] Howard, J., Gagné, M., Morin, A. J., and Van den Broeck, A. (2016). Motivation profiles at work: A self-determination theory approach. *Journal of vocational behavior*, 95:74–89.

[Høigaard et al., 2006] Høigaard, R., Säfvenbom, R., and Tønnessen, F. E. (2006). The relationship between group cohesion, group norms, and perceived social loafing in soccer teams. *Small Group Research*, 37(3):217–232.

[IBM, 2019] IBM (2019). Ibm ilog cplex optimization studio 12.10. `https://www.ibm.com/us-en/marketplace/ibm-ilog-cplex`.

[Juárez et al., 2021] Juárez, J., Santos, C., and Brizuela, C. A. (2021). A comprehensive review and a taxonomy proposal of team formation problems. *ACM Computing Surveys (CSUR)*, 54(7):1–33.

[Kargar and An, 2011] Kargar, M. and An, A. (2011). Discovering top-k teams of experts with/without a leader in social networks. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 985–994.

[Kargar et al., 2013] Kargar, M., Zihayat, M., and An, A. (2013). Finding affordable and collaborative teams from a network of experts. In *Proceedings of the 2013 SIAM international conference on data mining*, pages 587–595. SIAM.

[Khawam et al., 2017] Khawam, A. M., DiDona, T., and Hernández, B. S. (2017). Effectiveness of teamwork in the workplace. *International Journal of Sciences: Basic and Applied Research (IJSBAR)*, 32(3):267–286.

[Kirschbaum, 2005] Kirschbaum, R. (2005). Open innovation in practice. *Research-Technology Management*, 48(4):24–28.

[Kleinerman et al., 2018] Kleinerman, A., Rosenfeld, A., and Kraus, S. (2018). Providing explanations for recommendations in reciprocal environments. In *Proceedings of the 12th ACM Conference on Recommender Systems*, RecSys '18, page 22–30, New York, NY, USA. Association for Computing Machinery.

[Kozlowski and Ilgen, 2006] Kozlowski, S. W. and Ilgen, D. R. (2006). Enhancing the effectiveness of work groups and teams. *Psychological science in the public interest*, 7(3):77–124.

[Kraus et al., 2020] Kraus, S., Azaria, A., Fiosina, J., Greve, M., Hazon, N., Kolbe, L., Lembcke, T.-B., Muller, J., Schleibaum, S., and Vollrath, M. (2020). Ai for explaining decisions in multi-agent environments. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34:13534–13538.

[Kurtan et al., 2020] Kurtan, C., Yolum, P., Dastani, M., De Giacomo, G., Catala, A., Dilkina, B., Milano, M., Barro, S., Bugarín, A., Lang, J., et al. (2020). An ideal team is more than a team of ideal agents. *ECAI 2020*, 325:43–50.

[Laal and Ghodsi, 2012] Laal, M. and Ghodsi, S. M. (2012). Benefits of collaborative learning. *Procedia-social and behavioral sciences*, 31:486–490.

[Lappas et al., 2009] Lappas, T., Liu, K., and Terzi, E. (2009). Finding a team of experts in social networks. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, page 467–476, New York, NY, USA. Association for Computing Machinery.

[Leadership Centre,] Leadership Centre. Leadership Centre: MBTI Myers Briggs type indicator. `https://www.leadershipcentre.org.uk/artofchangemaking/theory/mbti-myers-briggs-type-indicator/`. Accessed: 2023.

[Levi, 2001] Levi, D. (2001). *Group dynamics for teams*. Sage Publications.

[Li and Shan, 2010] Li, C.-T. and Shan, M.-K. (2010). Team formation for generalized tasks in expertise social networks. In *2010 IEEE Second International Conference on Social Computing*, pages 9–16.

[Li et al., 2003] Li, Y., Bandar, Z. A., and McLean, D. (2003). An approach for measuring semantic similarity between words using multiple information sources. *IEEE Transactions on knowledge and data engineering*, 15(4):871–882.

[Likert, 1932] Likert, R. (1932). A technique for the measurement of attitudes. *Archives of psychology*.

[Lipton, 2018] Lipton, Z. C. (2018). The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue*, 16(3):31–57.

[Lucius and Kuhnert, 1997] Lucius, R. H. and Kuhnert, K. W. (1997). Using sociometry to predict team performance in the work place. *The journal of Psychology*, 131(1):21–32.

[Lundberg and Lee, 2017] Lundberg, S. M. and Lee, S.-I. (2017). A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30.

[Maddox, 2002] Maddox, R. B. (2002). *Mathematical Thinking and Writing: A transition to abstract mathematics*. Academic Pr, San Diego, 1st edition.

[Mallett, 2005] Mallett, C. J. (2005). Self-determination theory: A case study of evidence-based coaching. *The sport psychologist*, 19(4):417–429.

[Manganelli et al., 2018] Manganelli, L., Thibault-Landry, A., Forest, J., and Carpentier, J. (2018). Self-determination theory can help you generate performance and well-being in the workplace: A review of the literature. *Advances in Developing Human Resources*, 20(2):227–240.

[Manzoor et al., 2011] Manzoor, S. R., Ullah, H., Hussain, M., Ahmad, Z. M., et al. (2011). Effect of teamwork on employee performance. *International Journal of Learning and Development*, 1(1):110–126.

[Marston, 2013] Marston, W. M. (2013). *Emotions of normal people*, volume 158. Routledge.

[Mathieu et al., 2015] Mathieu, J. E., Kukenberger, M. R., D'Innocenzo, L., and Reilly, G. (2015). Modeling reciprocal team cohesion–performance relationships, as impacted by shared leadership and members' competence. *Journal of Applied Psychology*, 100(3):713.

[Mayer, 2007] Mayer, J. D. (2007). Asserting the definition of personality. *The online newsletter for personality science*, 1(1):1–4.

[McAuley and Leskovec, 2013] McAuley, J. and Leskovec, J. (2013). Hidden factors and hidden topics: understanding rating dimensions with review text. In *Proceedings of the 7th ACM conference on Recommender systems*, pages 165–172.

[McLean and Mallett, 2012] McLean, K. N. and Mallett, C. J. (2012). What motivates the motivators? an examination of sports coaches. *Physical Education and Sport Pedagogy*, 17(1):21–35.

[Miller, 2018] Miller, T. (2018). Explanation in artificial intelligence: Insights from the social sciences.

[Minehart and Foldy, 2020] Minehart, R. D. and Foldy, E. G. (2020). Effects of gender and race/ethnicity on perioperative team performance. *Anesthesiology Clinics*, 38(2):433–447.

[Miranda et al., 2017] Miranda, S., Orciuoli, F., Loia, V., and Sampson, D. (2017). An ontology-based model for competence management. *Data & Knowledge Engineering*, 107:51–66.

[Mohseni et al., 2020] Mohseni, S., Zarei, N., and Ragan, E. D. (2020). A multidisciplinary survey and framework for design and evaluation of explainable ai systems.

[Moreno and Jennings, 1938] Moreno, J. L. and Jennings, H. H. (1938). Statistics of social configurations. *Sociometry*, 1(3/4):342–374.

[Morrison and De Jong, 2002] Morrison, R. W. and De Jong, K. A. (2002). Measurement of population diversity. In Collet, P., Fonlupt, C., Hao, J.-K., Lutton, E., and Schoenauer, M., editors, *Artificial Evolution*, pages 31–41, Berlin, Heidelberg. Springer Berlin Heidelberg.

[Mosca and Such, 2021] Mosca, F. and Such, J. M. (2021). ELVIRA: An explainable agent for value and utility-driven multiuser privacy. In *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems*, page 916–924. International Foundation for Autonomous Agents and Multiagent Systems.

[Mount et al., 1998] Mount, M. K., Barrick, M. R., and Stewart, G. L. (1998). Five-factor model of personality and performance in jobs involving interpersonal interactions. *Human performance*, 11(2-3):145–165.

[Myers et al., 1998] Myers, I. B., McCaulley, M. H., Quenk, N. L., and Hammer, A. L. (1998). *MBTI manual: A guide to the development and use of the Myers-Briggs Type Indicator*. Consulting Psychologists Press.

[Nash, 1950] Nash, J. F. (1950). The bargaining problem. *Econometrica*, 18(2):155–162.

[Nizri et al., 2022] Nizri, M., Hazon, N., and Azaria, A. (2022). Explainable shapley-based allocation (student abstract). In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 13023–13024.

[Norman, 1963] Norman, W. T. (1963). Toward an adequate taxonomy of personality attributes: Replicated factor structure in peer nomination personality ratings. *J. Abnorm. Soc. Psychol.*, 66(6):574–583.

[Ogbonnaya et al., 2018] Ogbonnaya, C., Tillman, C. J., and Gonzalez, K. (2018). Perceived organizational support in health care: The importance of teamwork and training for employee well-being and patient satisfaction. *Group & Organization Management*, 43(3):475–503.

[Osman et al., 2014] Osman, N., Sierra, C., Mcneill, F., Pane, J., and Debenham, J. (2014). Trust and matching algorithms for selecting suitable agents. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 5(1):1–39.

[Pesch et al., 2015] Pesch, R., Bouncken, R. B., and Kraus, S. (2015). Effects of communication style and age diversity in innovation teams. *International Journal of Innovation and Technology Management*, 12(06):1550029.

[Ponda et al., 2015] Ponda, S. S., Johnson, L. B., Geramifard, A., and How, J. P. (2015). *Cooperative Mission Planning for Multi-UAV Teams*, pages 1447–1490. Springer Netherlands, Dordrecht.

[Popov et al., 2022] Popov, D., Snelson, S., and Baily, T. (2022). Review of skills taxonomies: Report prepared for the skills and productivity board: May 2022. *Frontier Economics*.

[Pozanco et al., 2022] Pozanco, A., Mosca, F., Zehtabi, P., Magazzeni, D., and Kraus, S. (2022). Explaining preference-driven schedules: the expres framework. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 32, pages 710–718.

[Präntare and Heintz, 2018] Präntare, F. and Heintz, F. (2018). An anytime algorithm for simultaneous coalition structure generation and assignment. In Miller, T., Oren, N., Sakurai, Y., Noda, I., Savarimuthu, B. T. R., and Cao Son, T., editors, *PRIMA 2018: Principles and Practice of Multi-Agent Systems*, pages 158–174, Cham. Springer International Publishing.

[Präntare and Heintz, 2020] Präntare, F. and Heintz, F. (2020). An anytime algorithm for optimal simultaneous coalition structure generation and assignment. *Autonomous Agents and Multi-Agent Systems*, 34(1):1–31.

[Puiu et al., 2021] Puiu, A., Vizitiu, A., Nita, C., Itu, L., Sharma, P., and Comaniciu, D. (2021). Privacy-preserving and explainable AI for cardiovascular imaging. *Studies in Informatics and Control, ISSN 1220-1766*, 30(2):21–32.

[Ramchurn et al., 2021] Ramchurn, S. D., Stein, S., and Jennings, N. R. (2021). Trustworthy human-ai partnerships. *iScience*, 24(8):102891.

[Rattanasawad et al.,] Rattanasawad, T., Saikaew, K. R., Buranarach, M., and Supnithi, T. A review and comparison of rule languages and rule-based inference engines for the semantic web. In *2013 International Computer Science and Engineering Conference (ICSEC)*.

[Revilla and Ochoa, 2017] Revilla, M. and Ochoa, C. (2017). Ideal and maximum length for a web survey. *International Journal of Market Research*, 59(5).

[Ribeiro et al., 2016] Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). "Why should I trust you?" Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144.

[Ribeiro et al., 2018] Ribeiro, M. T., Singh, S., and Guestrin, C. (2018). Anchors: High-precision model-agnostic explanations. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

[Richter et al., 2011] Richter, A., Dawson, J., and West, M. (2011). The effectiveness of teams in organizations: A meta-analysis. *The International Journal of Human Resource Management*, 22(13):2749–2769.

[Rosenfeld, 2021] Rosenfeld, A. (2021). *Better Metrics for Evaluating Explainable Artificial Intelligence*, page 45–50. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC.

[Rossetti et al., 2013] Rossetti, M., Stella, F., and Zanker, M. (2013). Towards explaining latent factors with topic models in collaborative recommender systems. In *2013 24th International workshop on database and expert systems applications*, pages 162–167. IEEE.

[Ryan and Deci, 2000] Ryan, R. and Deci, E. (2000). Self-determination theory and the facilitation of intrinsic motivation, social development, and well-being. *The American psychologist*, 55:68–78.

[Sandholm et al., 2002] Sandholm, T., Suri, S., Gilpin, A., and Levine, D. (2002). Winner determination in combinatorial auction generalizations. In *Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems: Part 1*, AAMAS '02, page 69–76, New York, NY, USA. Association for Computing Machinery.

[Sedaghat, 2018] Sedaghat, A. (2018). Factors affecting the team formation and work in project based learning (pbl) for multidisciplinary engineering subjects. *Journal of Problem Based Learning in Higher Education*, 6(2).

[Sokol and Flach, 2019] Sokol, K. and Flach, P. A. (2019). Counterfactual explanations of machine learning predictions: Opportunities and challenges for ai safety. *SafeAI@ AAAI*.

[Sovrano et al., 2020] Sovrano, F., Vitali, F., and Palmirani, M. (2020). Making things explainable vs explaining: Requirements and challenges under the GDPR. In *AI Approaches to the Complexity of Legal Systems XI-XII: AICOL International Workshops 2018 and 2020: AICOL-XI@JURIX 2018, AICOL-XII@JURIX 2020, XAILA@JURIX 2020, Revised Selected Papers*, page 169–182, Berlin, Heidelberg. Springer-Verlag.

[Strom and Strom, 2011] Strom, P. S. and Strom, R. D. (2011). Teamwork skills assessment for cooperative learning. *Educational Research and Evaluation*, 17(4):233–251.

[Suliman and Al-Sabri, 2009] Suliman, A. M. and Al-Sabri, N. (2009). Surviving through the global downturn: employee motivation and performance in healthcare industry. *The Open Business Journal*, 2(1).

[Tabassi et al., 2012] Tabassi, A. A., Ramli, M., and Bakar, A. H. A. (2012). Effects of training and motivation practices on teamwork improvement and task efficiency: The case of construction firms. *International Journal of Project Management*, 30(2):213–224.

[Thirasak, 2020] Thirasak, V. (2020). Building an effective startup team. In *International Conference on Advances in Business and Law (ICABL)*, volume 3, pages 18–27.

[Tietjen and Myers, 1998] Tietjen, M. A. and Myers, R. M. (1998). Motivation and job satisfaction. *Management decision*, 36(4):226–231.

[Tintarev and Masthoff, 2015] Tintarev, N. and Masthoff, J. (2015). Explaining recommendations: Design and evaluation. *Recommender systems handbook*, pages 353–382.

[Tremblay et al., 2009] Tremblay, M. A., Blanchard, C. M., Taylor, S., Pelletier, L. G., and Villeneuve, M. (2009). Work extrinsic and intrinsic motivation scale: Its value for organizational psychology research. *Canadian Journal of Behavioural Science/Revue canadienne des sciences du comportement*, 41(4):213.

[Vazirani, 2010] Vazirani, N. (2010). Review paper competencies and competency model-a brief overview of its development and application. *SIES Journal of management*, 7(1):121–131.

[West, 2012] West, M. A. (2012). *Effective teamwork: Practical lessons from organizational research*. John Wiley & Sons.

[Wi et al., 2009] Wi, H., Oh, S., Mun, J., and Jung, M. (2009). A team formation model based on knowledge and collaboration. *Expert Systems with Applications*, 36(5):9121–9134.

[Wideman, 2023] Wideman, M. (2023). Project teamwork, personality profiles and the population at large: Do we have enough of the right kind of people?

[Wilde, 2009] Wilde, D. J. (2009). *Teamology: the construction and organization of effective teams*, volume 3. Springer.

[Wilde, 2011] Wilde, D. J. (2011). *Jung's personality theory quantified*. Springer Science & Business Media.

[Wilde, 2013] Wilde, D. J. (2013). Post-jungian personality theory for individuals and teams. *SYDROSE LP*.

[Zhang and Chen, 2020] Zhang, Y. and Chen, X. (2020). Explainable recommendation: A survey and new perspectives. *Foundations and Trends® in Information Retrieval*, 14(1):1–101.