Auctioning Substitutable Goods

Two or more authors:

Andrea Giovannucci [a], Juan A. Rodríguez-Aguilar [a], Jesús Cerquides [b]

[a] *Artificial Intelligence Research Institute, IIIA*
Spanish Council for Scientific Research, CSIC
08193 Bellaterra, Barcelona, Spain
{*andrea,jar,giuseppe*}*@iiia.csic.es*

[b] *Dept. de Matemàtica Aplicada i Anàlisi*
Universitat de Barcelona
Gran Via, 585
08007, Barcelona, Spain
cerquide@maia.ub.es

Abstract. In this paper we extend the notion of multi-unit combinatorial reverse auction by adding a new dimension to the goods at auction. In such a new type of combinatorial auction a buyer can express substitutability relationships among goods: some goods can be substituted by others at a substitution cost. Substitutability relationships allow a buyer to introduce his uncertainty as to whether it is more convenient to buy some goods or others. We introduce such uncertainty in the winner determination problem (WDP) so that not only does the auction help allocate the optimal set of offers —taking into account substitutability relationships—, but also assess the substitutability relationships that apply. In this way, the buyer finds out what goods to buy, to whom, and what *operations* (substitutions) to apply to the acquired goods in order to obtain the initially required ones. Finally, we empirically analyse how the introduction of substitutability relationships helps increase competitiveness among bidders, and thus obtain better deals.

Keywords.

1. Introduction

Since many auctions involve the selling or buying[1] of a variety of different assets, combinatorial auctions [**?**,**?**] (CA) have recently deserved much attention in the literature. In particular, a significant amount of work has been devoted to the problem of selecting the winning set of bids [**?**,**?**,**?**,**?**,**?**]. Nonetheless, to the best of our knowledge the impact that the eventual relationships among the different assets to sell/buy have not been conveniently addressed so far.

Consider that a company devoted to the assembly and repairing of personal computers (PCs) requires to assembly new PCs in order to fulfil his demand. Say that its warehouse contains most of the components composing a PC. However, there are no

[1]Depending on whether the auction is direct or reverse respectively.

components to assemble motherboards[2]. Therefore, the company would have to start a sourcing [**?**] process to acquire such components. For this purpose, it may opt for running a combinatorial reverse auction [**?**] with qualified providers. But before that, a professional buyer may realise that he faces a decision problem: shall he buy the required components to assemble them in house into motherboards, or buy already-assembled motherboards, or opt for a *mixed purchase* and buy some components to assemble them and some already-assembled motherboards? This concern is reasonable since the cost of components plus the transformation (assembly) costs may eventually be higher than the cost of the already-assembled motherboards. To tackle this issue, the buyer could think of running separate auctions for motherboards and their components, and after that decide whether to buy the whole or the parts. Notice though that besides impractical and costly (in general, the more transformation relationships among goods we consider, the larger number of auctions would be required) this method would be missing the opportunity represented by mixed purchases. Hence, the buyer requires a combinatorial reverse auction mechanism that provides: (a) a language to express required goods along with the relationships that hold among them; and (b) a winner determination solver that not only assesses what goods to buy and to whom, but also the transformations to apply to such goods in order to obtain the initially required ones.

In this paper we try to provide solutions to both issues. Firstly, notice that we can resort to a more general semantics when referring to relationships among goods: the semantics of *substitutability*. In the example above, if a buyer requires a motherboard, we can say that it can be *substituted* by 1 CPU, 4 RAM units, and 3 USB connectors at a certain *substitution* (transformation in our example) cost. Notice though that this notion of substitutability among goods is different from the classic notion of substitutability on the bidder side that we find in the CA literature [**?**]. Since commercial e-sourcing tools[**?**] only allow buyers to express fixed number of units per required good as part of the so-called *Request for Quotation* (RFQ), we have extended this notion to allow for the definition of substitutability relationships among goods. Thus, we introduce a formal definition of an RFQ with substitutable goods that largely borrows from Place/Transition Nets[**?**] where transitions stand for substitution relationships and places stand for required goods.

Secondly, we present the formalisation of multi-unit combinatorial reverse auctions with substitutability relationships among goods by applying the expressiveness power of multi-set theory. Complementarity, we provide a mapping of our formal model to integer programming that takes into account substitutability relationships to asses the winning set of bids along with the substitutions to apply in order to obtain the buyer's initial requirements. Notice that although our example above depicts a very simple scenario where only a substitution applies (from components to motherboard), much more complex scenarios where a larger number of substitutability relationships are defined (see for instance the example in section 2) do require that the winner determination solver does find the substitutions to apply as well as the winning bids. The introduction of relationships among goods has the effect of putting together to compete bidders that otherwise would not be competing (e.g. CPU, memory, and USB manufactures compete with motherboard manufacturers).

The paper is organised as follows. In section 2 we introduce an extended version of the above-described example that founds our definition of RFQ with substitutability re-

[2]In this particular case, we consider that a motherboard is composed of 1 CPU, 4 RAM units, and 3 USB connectors.

lationships. In section 4 we present the formal model of multi-unit combinatorial reverse auctions with substitutability relationships, along with its winner determination problem and its mapping to integer programming. Finally, section 5 draws some conclusions and outlines directions for future research.

2. Example

In this section we provide an extended version of the example introduced in section 1 to illustrate the type of substitutability relationships that we are interested in representing. Figure 1 graphically represents the way a PC is assembled. Our graphical description largely borrows from the representation of Place/Transition Nets (PTN) [**?**], a particular type of Petri Net[3]. Each circle (corresponding to a PTN *place*) represents a good to negotiate upon. Assembly/splitting operations are represented as horizontal bars connecting goods, likewise *transitions* in a PTN. The assembling and splitting operations are labelled with an indexed capital T, and shall be referred to as *good transformations*. In particular $T1$ and $T2$ represent splitting operations whereas $T3$ and $T4$ stand for assembling operations. The values in parentheses, labelling good transformations, stand for the cost of each transformation every time it is *fired* (carried out). The arcs connecting a set of goods $G1$ to a transformation $T1$ indicates that the goods in $G1$ are an *input* to transformation $T1$. The arcs connecting a transformation $T1$ to a set of goods $G2$ indicates that goods in $G2$ are an *output* from transformation $T1$. In the example in figure 1, the $T2$ transformation, representing the way a motherboard is taken into pieces, has a motherboard as *input good* and CPUs, RAM memories, USBs and empty motherboards as *output goods*. We call *input weights* the labels on the arcs connecting *input goods* to transitions, and *output weights* the labels on the arcs connecting *output goods* to transitions. They indicate the units required of each *input good* to perform a transformation and the units generated per *output good* respectively. For instance, the labels on the arcs connected to $T3$ in figure 1 indicate that 1 motherboard is composed of 1 CPU, 4 RAM units, 3 USBs and 1 empty motherboard at a cost of 8 EUR.

3. Background

3.1. Multisets

A *multi-set* is an extension to the notion of set, considering the possibility of *multiple appearances* of the same element. A *multi-set* $\mathcal{M}_X$ over a set X is a function $\mathcal{M}_X : X \to \mathbb{N}$ mapping X to the cardinal numbers. For any $x \in X$, $\mathcal{M}_X(x) \in \mathbb{N}$ is called the *multiplicity* of x. We formally represent a multi-set $\mathcal{M}_X$ by a sum as follows:

$$\sum_{x \in X} \mathcal{M}_X(x)`x$$

An element $x \in X$ *belongs* to the multi-set $\mathcal{M}_X$ if $\mathcal{M}_X(x) \neq 0$ and we write $x \in \mathcal{M}_X$. We denote the set of multi-sets over X by X_{MS}.

[3] In section 4 we further elaborate on the formal links with PTNs.

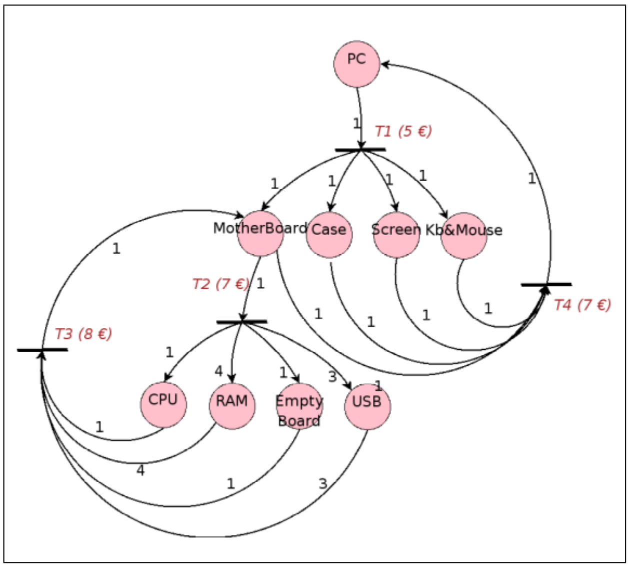

Figure 1. Graphical representation of an RFQ with substitutability relationships.

Given the multi-sets $\mathcal{M}_S, \mathcal{M}'_S \in S_{MS}$, their union is defined as:

$$\mathcal{M}_S \cup \mathcal{M}'_S = \sum_{s \in S} (\mathcal{M}_S(s) + \mathcal{M}'_S(s))\, `s$$

Operations over multi-sets (addition, multiplication, subtraction,...etc.) amount to the standard operations on their mapping functions.

Finally, notice that all in this paper we restrict to *finite* multi-sets.

3.2. Place Transition Nets

In what follows we recall the definition of a *Place/Transition Net* (PTN), a particular type of Petri Net [**?**].

Definition 3.1. A *Place/Transition Net* (PTN) is a tuple $PTN = (G, T, A, E, M_0)$ satisfying the requirements below:

1. G is a set of *places*.
2. T is a finite set of *transitions* such that $P \cap T = \emptyset$.
3. $A \subseteq (G \times T) \cup (T \times G)$ is a set of *arcs*.
4. $E : A \to \mathbb{N}_+$ is an *arc expression* function.
5. The initial marking $\mathcal{M}_0 \in G_{MS}$ represents the number of tokens initially present in each place.

A *Plca Transition Net Structure* $N = (G, T, A, E)$ does not specify any initial marking. A Place Transition Net with a given initial marking is denoted by $PTN = (N, M_0)$.

The graphical representation of a PTN structure is composed of the following graphical elements:

- places are represented as circles;
- transitions are represented as bars;
- Arcs connect places to transitions or transitions to places;
- E labels arcs with values; and

Definition 3.2. A *marking* is a multi-set over G. A *step* is a non-empty and finite multi-set over T. The *initial marking* $M_0 \in G_{MS}$ denotes the initial tokens distribution.

Definition 3.3. A step $\mathcal{S} \in T_{MS}$ is *enabled* in a marking $\mathcal{M} \in G_{MS}$ if the following property is satisfied: $\forall g \in G : \sum_{t \in \mathcal{S}} E(g,t)\mathcal{S}(t) \leq \mathcal{M}(g)$

Definition 3.4. Let the step $\mathcal{S}$ be enabled in a marking $\mathcal{M}_1$. Then, the step $\mathcal{S}$ may *occur*, changing the marking $\mathcal{M}_1$ to another marking $\mathcal{M}_2 \in G_{MS}$, defined as follows:

$$\forall g \in G : \mathcal{M}_2(g) = (\mathcal{M}_1(g) - \sum_{t \in \mathcal{S}} E(g,t)\mathcal{S}(t)) + \sum_{t \in \mathcal{S}} E(t,g)\mathcal{S}(t) \qquad (1)$$

Setting $Z(g,t) = E(g,t) - E(t,g)$ expression (1) becomes:

$$\forall g \in G : \mathcal{M}_2(g) = \mathcal{M}_1(g) + \sum_{t \in \mathcal{S}} Z(g,t)\mathcal{S}(t) \qquad (2)$$

Moreover, we say that the marking $\mathcal{M}_2$ is *directly reachable* from the marking $\mathcal{M}_1$ by the occurrence of the step $\mathcal{S}$, and we denote it by by $\mathcal{M}_1[\mathcal{S} > \mathcal{M}_2$.

Definition 3.5. A *finite occurrence sequence* $\mathcal{K}$ is a multi-set over T defined as follows:

$$\mathcal{K} = \{ \bigcup_{i \in \{1,2,..,n\}} \mathcal{S}^i \mid \mathcal{M}_1[\mathcal{S}_1 > \mathcal{M}_2....\mathcal{M}_n[\mathcal{S}_n > \mathcal{M}_{n+1}\} \qquad (3)$$

such that $n \in \mathbb{N}$ and $\mathcal{M}_i[\mathcal{S}_i > \mathcal{M}_{i+1}\ \forall i \in \{1..n\}$. $\mathcal{M}_1$ is called the *start marking*, while $\mathcal{M}_{n+1}$ is called the *end marking*.

Definition 3.6. A marking $\mathcal{M}''$ is *reachable* from a marking $\mathcal{M}'$ iff there exists a finite occurrence sequence having $\mathcal{M}'$ as start marking and $\mathcal{M}''$ as end marking, i.e. if there exists a finite occurrence sequence such that:

$$\mathcal{M}'[\mathcal{S}_1\mathcal{S}_2..\mathcal{S}_n > \mathcal{M}''$$

In this case we say that $\mathcal{M}''$ is *reachable* from $\mathcal{M}'$ in n steps and we denote it as:

$$\mathcal{M}'[\mathcal{K} > \mathcal{M}''$$

where $\mathcal{K} = \bigcup_{i=1..n} \mathcal{S}_i$. Furthermore start and end marking are related by the equation

$$\forall g \in G : \mathcal{M}''(g) = \mathcal{M}'(g) + \sum_{t \in \mathcal{K}} Z(g,t)\mathcal{K}(t) \qquad (4)$$

It has been showed that, if a place transition net has no directed circuits in it (*acyclic petri net*) then expression 4 completely describes the whole reachability set ([**?**]):

Definition 3.7. In an acyclic petri net a marking $\mathcal{M}''$ is *reachable* from a marking $\mathcal{M}'$ iff there exists a multi set $\mathcal{K} \in T_{MS}$ such that expression 4 holds.

As a consequence, the reachability set $[M_0 >$ is represented by:

$$[M_0 >= \{\mathcal{M} \mid \exists \mathcal{K} \in T_{MS}\ s.t. \forall g \in G : \mathcal{M}(g) = \mathcal{M}_0(g) + \sum_{t \in \mathcal{K}} Z(g,t)\mathcal{K}(t)\} \quad (5)$$

4. Multi-Unit Combinatorial Reverse Auctions with Substitutability Relationships among Goods (MUCRASG)

4.1. Request For Quotation with Substitutable Goods

In what follows we formally model a new type of RFQ in which it is possible to express substitutability relationships among goods with an associated *substitution cost*. We call such a new RFQ a Request For Quotation with Substitutable Goods (RFQSG):

Definition 4.1. A *Request For Quotations with Substitutable Goods* (RFQSG) is a triple $R = (N, \mathcal{U}, C)$, where:

- N is a Place-Transition Net Structure $N = (G, T, A, E)$ such that:
  1. The *places* G represent a set of negotiated goods.
  2. The *transitions* T represent a set of possible *substitutability relationships* among goods.
  3. The *directed arcs* in A connect goods to substitutability relationships.
  4. The weights assigned by the *arc expression* function E indicates the number of units of a given good either required or produced by a substitution. The values of E are the arc labels in figures 1, **??** and 2.

  T represents the set of possible substitutions among subsets of G. The arcs in A relate either goods to substitutions or substitutions to goods. A substitutability relationships states that the goods that are connected to it by incoming arcs (*input goods*) can substitute the goods connected to it by outgoing (*output goods*). The unit ratios according to which goods are substituted is expressed by E.
- $\mathcal{U} \in G_{MS}$ expresses a buyer's requirements (the number of required units per good).
- $C : T \to \mathbb{R}^+ \cup \{0\}$ is a cost function that associates a *substitution cost* to each *substitutability relationship*. The cost function C values are enclosed in parenthesis next to each transition in figures.

4.2. Example

The formal specification of the RFQSG graphically represented in figure 2 is:

- $G = \{g_1, g_2, g_3, g_4\}$

- $T = \{T_1\}$
- $A = \{(g_1, T_1), (g_2, T_1), (T_1, g_3), (T_1, g_4)\}$
- $E(g_1, t_1)$=3, $E(g_2, t_1)$=4, $E(t_1, g_3)$=2, $E(t_1, g_4)$=1
- $C(T_1) = 200$ EUR
- $\mathcal{U}(g_1) = 2, \mathcal{U}(g_2) = 2, \mathcal{U}(g_3) = 2, \mathcal{U}(g_4) = 1$

This RFQSG expresses that a buyer needs 2 units of g_1, g_2 and g_3, and 1 unit of g_4 ($\mathcal{U}$). Furthermore it describes a buyer's capacity of transforming the couple of goods $(g1, g2)$ into the couple $(g3, g4)$ by means of transformation $t1$. Multiplicities indicate that 3 units of good g_1 and 4 units of item g_2 can be transformed into 2 units of good g_3 and one unit of good g_4. C sets the substitution cost of T_1 to 200 EUR.

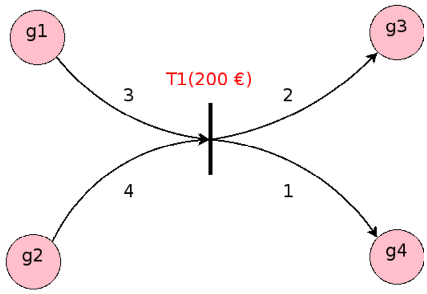

Figure 2. Graphical representation of a transformation

4.3. The Winner Determination Problem

In this scenario, what will it happen when a buyer receives a set of bids from providers?

A marking $\mathcal{M}_k$ in a RFQSG corresponds to a possible request configuration: $\mathcal{M}_k(g)$ is the number of required units of good g. Our aim is to define a set of request configurations that are equivalent for a buyer, although differing for a *substitution cost*. These configuration are equivalent since a buyer can obtain back its initial request configuration (I) by means of substitutions. Consider again the example in figure 2, in a classic multi-unit combinatorial reverse auction scenario, we would consider a single RFQ represented by I. Nonetheless, since substitutability relationships hold among goods, the buyer may have different alternatives depending on the bids he receives:

1. $\mathcal{M}_0 = [g_1\ g_1\ g_2\ g_2\ g_3\ g_3\ g_4]$. Buy all items as requested.
2. $\mathcal{M}_1 = [g_1\ g_1\ g_1\ g_1\ g_1\ g_2\ g_2\ g_2\ g_2\ g_2\ g_2]$. Buy 5 units of item g_1 and 6 units of item g_2 to transform respectively 2 units and 4 units of them into 2 units of g_3 and 1 unit of g_4 at a cost $c = 200$ EUR. The overall cost results from buyed units cost plus transformation cost c. Thus, there is an extra cost.

Notice that both possibilities allow the buyer to obtain his initial requirement, namely 2 unit of g_1 , 2 units of g_2, 2 units of g_3 an 1 unit of g_4, each one at a different cost. When running a MUCRA with the initial requirement in this example, the buyer faces a decision problem. According to the received bids he has to decide which of the two above explained alternatives minimizes its costs.

In a RFQSG *steps* map to substitutions involving only one step. In point (2) of the example above, a step results in substituting tokens in places g_1 and g_2 with tokens in places g_3 and g_4, which we will refer to as *substitution step*. We also say that the occurrence of the substitution step t_1 *transforms* the request configuration $\mathcal{M}_1$ into the initial configuration $\mathcal{M}_0$. In this way we define the concept of *transformation sequence* as the equivalent of *finite occurrence sequence* and the concept of *transformability* as the equivalent of *reachability*.

Our problem is to find all the possible markings $\mathcal{M}_k$ that are transformable by means of substitutions to the initial marking M_0. This is not equivalent to find the reachability set of the RFQSG! In fact it is right the dual of it. Thus, we define the *substitutability set* for a configuration $\mathcal{M}_0$ and an RFQSG π as the set:

$$S^{\Pi}_{M_0} = \{\mathcal{M} \in G_{MS} \mid \mathcal{M}_0 \in [\mathcal{M} >\}$$

This set contains all the markings that can be transformed into M_0. We mentioned that, for an acyclic PTN, the reachability set is completely defined by the 5 expression. With an algebraic manipulation of such equation we can obtain:

$$S^{\Pi}_{M_0} = \{\mathcal{M} \mid \exists \mathcal{K} \in T_{MS}\ s.t.\ \forall g \in G : \mathcal{M}(g) = \mathcal{M}_0(g) - \sum_{t \in \mathcal{K}} Z(g,t)\mathcal{K}(t)\} \quad (6)$$

that represent the substitutability set.

We also said that there is a cost associated to the transformation of a request configuration into another. How can we associate a substitution cost to each marking $\mathcal{M} \in S_{M_0}$? We know that if $M_0 \in [M >$ then it exists at least a tranformation sequence $\mathcal{K}$ such that $\mathcal{M}[\mathcal{K} > \mathcal{M}_0$. We also know that $\mathcal{K}(t)$ is the number of times a transition t is fired in a finite occurrence sequence. Thus, we compute the cost of transforming a request configuration $\mathcal{M} \in S_{M_0}$ into the initial configuration $\mathcal{M}_0$ by the formula:

$$C_{M_0}(\mathcal{M}) = \min_{\mathcal{K} \in K^M_{M_0}} \sum_{t \in \mathcal{K}} C(t)\mathcal{K}(t) \quad (7)$$

where $K^M_{M_0} = \{\mathcal{K} \mid \mathcal{M}[\mathcal{K} > \mathcal{M}_0\}$.

4.4. Winner Determination Problem for Multi-Unit Combinatorial Reverse Auctions with Substitutable Goods

Given a RFQSG and a set of bids B sent by a set of providers P, the winner determination problem is no solely focused on the determination of the set of winning bids as in the MUCRA case. Rather, the problem focuses on the determination of a requirement configuration $\mathcal{M}$ leading to the buyer's initial requirements via a substitution sequence, along with the optimal set of bids fulfilling the requirements expressed by $\mathcal{M}$. There are costs associated to both a substitution sequence (*substitution cost*) and to the selection of bids (the sum of the prices of selected bids). The WDP for a MUCRASG aims at determining both a substitution sequence and a set of winning bids minimising the overall cost, i.e. the *substitution cost* plus the selected bids' cost.

We define a multi-unit multi-item bid as a multiset over G $\mathcal{B}$, whose multiplicity represents the number of units offered per good.

We define also a pricing function $p : B \to \mathbb{R}_0^+$, that assigns a price to each bid.

An auction outcome is a pair $(W, \mathcal{M})$ where $\mathcal{M}$ stands for a requirement configuration leading to the initial requirements $\mathcal{M}_0$ via a substitution sequence $\mathcal{K}$, and $W \subseteq B$ stands for a set of bids fulfilling $\mathcal{M}$.

Definition 4.2. Given an initial requirement $\mathcal{M}_0$ and an RFQSG Π, the set of possible auction outcomes is:

$$\Omega = \{(W, \mathcal{M}), W \subseteq B, \mathcal{M} \in S^{\Pi}_{M_0} \mid \bigcup_{\mathcal{B} \in W} \mathcal{B} \supseteq \mathcal{M}_0\} \tag{8}$$

where $S^{\Pi}_{M_0}$ is the substitutability set for $\mathcal{M}_0$.

Definition 4.3. For each outcome $(W, \mathcal{M})$, we associate an *outcome cost* as follows:

$$c(W, \mathcal{M}) = \sum_{\mathcal{B} \in W} p(\mathcal{B}) + C_{M_0}(\mathcal{M}) \tag{9}$$

Given a set of auction outcomes, the aim of the WDP for a MUCRASG is to find the optimal outcome $(W^{opt}, \mathcal{M}^{opt}) \in \Omega_M$ that minimises the outcome cost $c(W, \mathcal{M})$. Formally,

$$(W^{opt}, \mathcal{M}^{opt}) = \arg \min_{(W, \mathcal{M}) \in \Omega_M} c(W, \mathcal{M}) \tag{10}$$

4.5. Mapping to Integer Programming

We model the problem of determining $(W^{opt}, \mathcal{M}^{opt})$ as an Integer Programming problem. In order to do this we need to express as integer variables:

- The set of selected bids W
- A substitution sequence $\mathcal{K}$.

In order to represent W we assign a binary variable $x_{\mathcal{B}}$ to each $\mathcal{B} \in B$, which represent whether the bid $\mathcal{B}$ is included ($x_{\mathcal{B}} = 1$) or not ($x_{\mathcal{B}} = 0$) in W. A multi-set is uniquely determined by his mapping function $\mathcal{K} : T \to \mathbb{N}$. Hence, we represent the multi-set $\mathcal{K}$ over the set $T = [t_1 \ t_2 \ ... \ t_m]$ via an ordered vector of bounded integer variables $\mathbf{q} = [q_{t_1} \ q_{t_2} \ ... \ q_{t_m}]$ Each q_{t_i} represents the multiplicity of element t_i in the $\mathcal{K}_T$ multi-set. This vector stands for a transformation sequence $\mathcal{K}$. Thus, the representation in integer programming of expression (10) becomes

$$min \sum_{\mathcal{B} \in B} x_B p(\mathcal{B}) + \sum_{tinT} q_t * c(t)$$

subject to $x_{\mathcal{B}} \in \{0, 1\}, q_t \in \{0, 1, ...max_t\}$

It is possible demonstrate that $q_t < k$, $k \in \mathbb{N}$, $k < \infty \ \forall tinT$ for acyclic petri nets. (Demonstration that q_t is always bounded!!!)

This expression generalises expression (**??**). Now we have to generalise expression (**??**). We know from equation (2) how to obtain the end requirement configuration given the *start requirement* and the substitution sequence $\mathcal{K}_T$. Rather, what we need is the substitutability set, i.e. all the requirement configurations $\mathcal{R}_G$ that are substitutable with $\mathcal{U}_G$. Thus, manipulating equation 2, and substituting $\mathcal{K}_T$ with his integer programming variable we obtain the substitutability set:

$$\{\mathcal{R}_G \in M \mid \forall g \in G : \mathcal{R}_G(g) = \mathcal{U}_G(g) - \sum_{t \in T} Z(g,t) q_t \tag{11}$$

Finally the constraint translating to integer programming expression (**??**) becomes:

$$\forall g \in G \quad \sum_{g \in \mathcal{C}_G} x_{\mathcal{C}_G} \mathcal{C}_G(g) \geq \mathcal{U}_G(g) - \sum_{t \in T} Z(g,t) q_t$$

that generalises expression (**??**)

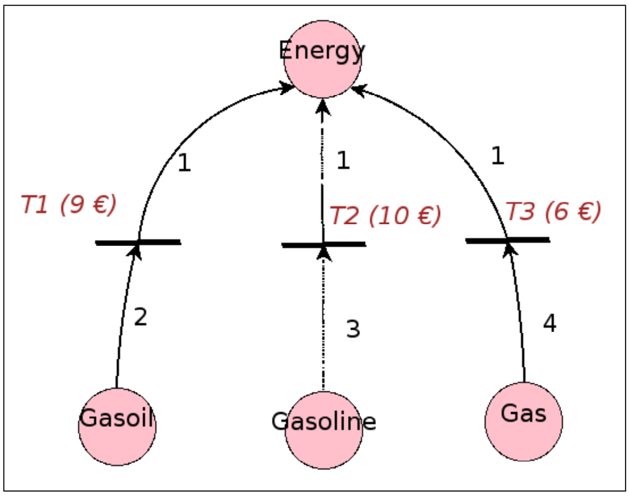

Figure 3. Energy sourcing example

5. Conclusions and Future Work

We defined a new powerful type of multi-unit combinatorial reverse auction in which the auctioneer can express substitutability relationships among goods at a certain added cost. Furthermore we provided a method for determining both the set of winning bids and the corresponding substitution to apply that minimizes the overall auctioneer's costs.

Dually to the case of substitutable goods for bidders, substitutability relationships among goods increase the competitivity among bidders, and though it allows better margins for auctioneerS. This is done via a market desegmentation: bidders that were not competing in a tradictional auction are engaged in a competition.

We also performed some preliminary and very simple experiments to measure increments in profits running a multi-unit combinatorial reverse auction with substitutable goods instead of a traditional MUCRA. The experiments showed different

- study the best instances
- study different price distribution
- Direct-reverse auctions
- *Alternative/OR semantic*:Consider that we can give different semantics to substitutability. One possible interpretation is related to transformations/assembling/dismantling operations. This is the case considered in figure 1. Alternatively it may indicate preferences between alternatives. Figure 3 shows an example of energy sourcing. An electrical company may decide to produce energy using three possible combustibles as well as to subcontract energy from other companies. The three fossil combustibles suppose different costs to the company due to their polluting emissions. The arc labels represent the fact that the three combustible have different efficiencies, so the energy quantities produced by the same quantity of each of the three fossils are different.(It would be better the example of USB2.0-FIREWIRE)
- are 0 valued arc weights allowed ?
- Going further it is possible that the buyer necessities are covered by only one between a set of substitutable goods. For example USB 1.1 and USB 2.0 may be equivalent for him.
- Possibility of running the combinatorial auction without transmitting information related to transformability in order to simplify bidding strategy.

NOTATION

References

[1]