

Resource bartering in data grids

Can Özturan

Department of Computer Engineering, Bogazici University, Istanbul, Turkey

Tel.: +90 212 359 7225; Fax: +90 212 287 2461; E-mail: ozturaca@boun.edu.tr

Abstract. We present models for bartering of resources on grids. Bartering models can be useful for making resource allocation decisions in grids and perhaps even for building a so called barter grid whereby distributed resources such as electronic media can be bartered. Grids allow various resources to be shared among many users. This sharing however definitely does not mean that everyone will have unrestricted use of the resources. Some mechanism such as pricing or quotas can be employed in order to enforce controlled sharing of resources. A barter model for resource sharing can enable people or computer centers to directly get something in return for letting their resources to be used by others. We utilize directed hypergraphs to develop a barter model in which multiple resources can be traded. We prove that the decision version of the multi-resource bartering problem is NP-complete. We present an integer programming formulation for the bartering problem. We also present a linear time algorithm to compute components that may contain feasible bartering solutions. We generalize our multi-resource bartering formulation to the case where multiple instances of resources are present. Finally, we present various computational results from our software that makes use of LP-SOLVE and CPLEX mixed integer programming libraries to solve example bartering problems.

Keywords: Grid, resource management, barter, auction

1. Introduction

Grids allow various resources such as cpu cycles, storage, licensed software executions and data to be shared among many users. For small academic and research communities that know and trust each other, free unrestricted sharing of resources may not pose a big problem since people in these communities are courteous and avoid abuse of resources. On the other, when we move to larger scale grids and in particular to commercial platforms, sharing, however, will definitely not mean that everyone will have unrestricted use of the resources. Some mechanism such as pricing or quotas would have to be employed in order to enforce controlled sharing of resources. The best example of what may happen if resources are provided free of charge is the spam mail problem on the Internet. If grids are to be successfully deployed on a large scale, then mechanisms must be devised to manage resources. Buyya et al. [3] and Wolski et al. [28] discuss several economic models that can be used for resource management in grids. These are commodity market, posted price, bargaining, spot market, tendering, auction, coalition, bartering, monopoly and oligopoly models.

Prices of grid resources such as cpu cycles, storage and data can be too subjective and dynamic. Therefore, we believe that pricing-based schemes such as auctions can be disadvantageous especially for parties that are going to both sell and buy resources. To illustrate this, we draw an example from the Internet domain name registration scene. Domain names are resources. Suppose party A have registered domain name X and party B has registered domain name Y. Each party knows that his domain name is valuable. But let's say that party A needs the domain name Y owned by B. Likewise, party B needs the domain name X owned by A. Knowing that their names are valuable, each party may put his domain name for sale for a huge sum of money (say \$ 50,000). If these parties do not have any money, they can swap their names provided that they realise that their bids form a cycle (of length two). This realisation however, maybe difficult especially in cases where we have bid structures that form cycles of length n where n is greater than 2. In such cases, since parties may not have enough money, they are not able to bid for the requested amount and hence trading of the n names may not be possible. Yet, if we were able to detect bids forming a cycle of length n , then the n names could

all be traded. This example illustrates the disadvantage of markets based on pricing with money. By pricing items with money, we lose rich trading structures that would enable us to come up with feasible trading patterns. The well known word for direct trading of items without involvement money is bartering. In this paper, we propose the use of bartering for grid resource management and provide a formal mathematical model that enables us to develop scheduling algorithms for bartering problems.

Grids first appeared for enabling sharing of computational resources among researchers in academia and research laboratories. If grids are to be deployed successfully on a massive scale, they have to address the needs of ordinary people that make up the mass market. What do the ordinary people need? Ordinary people do not run sophisticated applications that require massive computational resources like the researchers do. They are more interested in data-centric resources such as mp3 music files, ebooks, financial reports, historical stock quotes, news and movies. We, therefore, believe that data grids rather than computational grids have much more potential to get the the attention of mass market. The popularity of file sharing networks such as Napster [16] and Kazaa [13] is a clear evidence that demonstrates real interests of the mass market. On the other hand, even though these file sharing networks are quite popular, they have one drawback and that is that files can be shared illegally. We believe that if we get rid of the sharing mechanism (that may result in illegal free copying of items) and replace it with a bartering mechanism and hence come up with a so called "bartergrid", this barter grid will get the attention of mass market. Consider someone who has purchased an ebook. After reading the ebook he may not need the the ebook any longer, so he may use the so called bartergrid to barter his ebook for something that he needs, for example, a movie. Such bartering transactions will not pose any legal problems since people will be trading their legally owned items for other legally owned items. Such a barter grid will require mechanisms for floating ownerships so that ownership of an item is transferred from one user to another. Another mechanism that will be required is a model for bartering, that is, problem definitions and algorithms that will produce feasible bartering solutions. The subject matter of this paper is only the latter, that is development of a model and algorithms which will take barter bids from users and produce optimal bartering solutions according to some objective function.

Recently complementary auction and double auction markets have received a lot of attention among

researchers [9,12,15,27]. These markets have purchase/sell prices as a fundamental component in their problem formulation. We believe that in its most general form involving multiple resources and multiple resource instances, bartering offers a much more general and stronger combinatorial market and that it probably contains complementary auction and double auction markets as special cases. On the Internet, several sites that offer bartering services can be found [25]. These sites, however, utilize barter units in place of money. In this paper, we focus on a market that allows *direct bartering*. In direct bartering, multiple resources can be traded *directly* without involvement of money by traders. However, unlike the auctions, the bids in the barter market are made in terms of resource(s) offered and resource(s) requested. In our barter market, we want to choose a feasible set of bids that can be satisfied and that optimizes some objective. We will call this problem multi-resource bartering problem.

We review previous work on auctions, bartering and graph theoretic results that will be useful for bartering in Section 2. In Section 3, we present a graph theoretic model for multi-resource bartering problem. We also look into the problems of eliminating unnecessary resources and decomposing the problem into smaller subproblems. We present a linear time algorithm to do this in Section 4. In Section 5, we prove that multi-resource bartering problem is NP-complete. In Sections 6 and 7, we give integer programming formulation of our problem respectively for single and multiple instance versions. Finally, the last two sections present timings from various tests and a discussion of results.

2. Previous work

The difficulty of finding "double coincidence of wants" in direct bartering was formally stated by Jevons [10]. Since then a lot of studies have been carried on the analysis of bartering and monetary issues from economists' perspectives using various models [6, 11,23].

Depending on the number of distinct resources and the number of instances (units) of these resources that can appear in a bid, we can classify the bartering problems into four categories: (i) Single instance single resource, (ii) Multiple instance single resource, (iii) Single instance multiple resource, (iv) Multiple instance multiple resource problems. The subject matter of this paper is bartering problems of type (iii) and type (iv). We have addressed in particular problems of type (i)

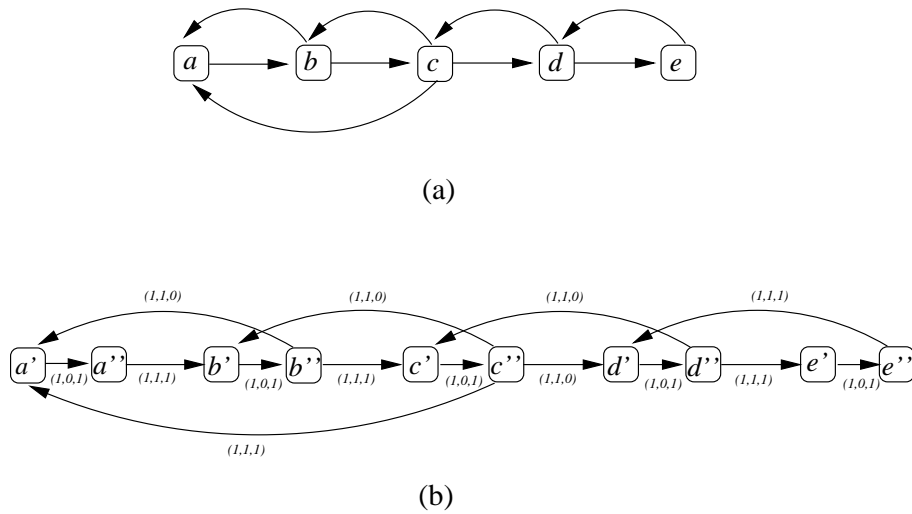


Fig. 1. Network showing bartering bids (a) and the transformed network with split nodes (b). The arcs in (b) are labelled (capacity, cost, flow) values.

and (ii) in our earlier work [17,18]. These two problems have polynomial time solution. Problems of type (i) in which we maximize the number of resources bartered is in fact the graph theoretic maximum vertex disjoint cycles problem that is addressed in Gutin et al. [8] (or see [2, p. 146]). Gutin et al. solves this problem with $O(n^3)$ complexity by transforming it to an assignment problem. The solution of problem (i) which is formulated as a minimum cost circulation problem is illustrated through an example in Fig. 1. We represent each barter bid as an arc where the tail of the arc represents the item to be given and the head of the arc represents the item that is requested. Figure 1(a) shows an example bid graph. Given a bid graph, a capacitated network (with all arc capacities equal to 1), is constructed by employing *node splitting procedure* [1, p. 41]. As part of the node splitting procedure, a node v is replaced by two nodes v' and v'' with an arc $\langle v', v'' \rangle$ in between them. The aim of node splitting procedure is to implement node capacities with 1 unit and hence enable at most one bid to win an item. A cost of -1 is assigned to arcs corresponding to bid arcs and 0 to the arcs between the split nodes. It is easy to verify that the solution of the minimum cost circulation problem [1, p. 7] on this network will give us the maximum vertex disjoint cycles. Hence, the maximum number of satisfied bids can be found this way. Figure 1(b) shows the transformed network with split nodes. The arc labels give the (capacity, cost, flow) values. The arcs (representing bids) with flow value 1 give the maximum solution. We also note that by introducing nodes that correspond to barterers and assigning capacities that

reflect the number of instances of each item, solution of problems of type (ii) can also be found in a similar manner [17].

It is also important to note the relationship of bartering to that of matching problems. In the most popular and the simplest matching problem on an undirected graph, if we replace each undirected edge (u, v) with two directed arcs, $\langle u, v \rangle$ and $\langle v, u \rangle$, then maximum cardinality matching amounts to finding maximum vertex disjoint cycles of length two. In the special case of bartering in which a single item is to be bartered with another single item, we are solving a maximum vertex disjoint cycles problem with no restriction on cycle length. A similar relationship can possibly be established for the multi-item bartering problem we study in this paper and the gangmatching problem that is addressed in Condor by Raman et al. [21,22].

Double auction mechanisms [9,12,27] in which simultaneous buy and sell bids may be placed may look similar to bartering mechanisms. Pricing, however, is a major component in the formulation of the double auction problems. Double auction mechanisms can be used in barter exchanges in which barter units are employed to value (price) resources. However, in this paper, we address direct barter exchange, in which resources are not priced by barter units.

3. Multi-item bartering model

Our multi-resource barter exchange is a synchronous market which consists of two main components:

- Resources: Each barterer puts forward a set of distinct (i.e., single instance) resources that he owns and combinations of which he may want to trade away for resource(s). The set of all resources of all the barterers is denoted by R .
- Bids: Each barterer makes a number of bids. A bid consists of a set of owned resources of the barterer that is offered for another set of resources owned by other barterers. The following example illustrates the form of a bid:

$$\begin{aligned} &\{\text{resource1}, \text{resource2}\} \Rightarrow \\ &\{\text{resource3}, \text{resource4}, \text{resource5}\} \end{aligned}$$

We will call the offered (requested) set of resources on the the left (right) hand side as the tail (head) set of a bid. The bid simply declares that the barterer can give away resources in the tail set provided he can get the resources in the head set. We will let B denote the set of all bids by all barterers.

Having collected these bids over a period of time, we can then close the market (hence the name synchronous) and then solve the bartering problem which in the simplest case may have the objective of maximizing the number of bids that can be satisfied. In this way, we may be able to trade away grid resources directly for other grid resources without involvement of money. We put forward a few rules that must be satisfied by a bartering solution:

- Rule 1: If a bid is selected (satisfied) in a bartering solution, then *all* the resources in the head set must be given to the barterer owning the satisfied bid.
- Rule 2: In a bartering solution produced, it is possible that a subset of (i.e., none, some, or all) of resources in the tail set may be given in order to get all the resources in the head set in a satisfied bid. What this means is that it is possible that we can give fewer resources than what appears in the declared tail set and yet be able to get all the resources requested. This also implies that it is possible to get something simply as free without giving away anything at all.
- Rule 3: the third rule simply says that in a bartering solution, a resource can appear in the head set of only one satisfied bid. Violating this rule may mean giving the same resource to more than one barterer which is unacceptable. We also enforce this rule even if the bids satisfied belong to the same barterer.

Directed hypergraphs are generalizations of directed graphs. Whereas single vertices appear at the tail and

head of arcs in directed graphs, arbitrary sized sets of vertices appear at the tail and head of *hyperarcs* in directed hypergraphs. Hence in our bartering problem, bids can be represented by hyperarcs and resources as vertices in directed hypergraphs. In the rest of the paper, we use the notation R for the set of vertices and B for the set of hyperarcs since these coincide respectively with resources and bids in our problem. We now give formal directed hypergraph definitions that we will utilize while formulating a solution technique for the bartering problem. A *directed hypergraph* $D(R, B)$ consists of two sets, R and B where R is a set of vertices and B is a set of hyperarcs. Each hyperarc $b = \langle R_t, R_h \rangle$ is an ordered pair of non-empty disjoint subsets R_t and R_h of R . Here, R_t and R_h are the sets of vertices that appear respectively in the *tail* and *head* of the hyperarc b . For the example shown in Fig. 2(a), $R = \{a, b, c, d, e\}$ is the set of vertices (resources), $B = \{b_1, b_2, b_3, b_4, b_5, b_6, b_7\}$ is the set of hyperarcs (bids) with $b_1 = \langle \{e\}, \{a\} \rangle$, $b_2 = \langle \{a\}, \{e\} \rangle$, $b_3 = \langle \{a, b\}, \{c\} \rangle$, $b_4 = \langle \{d\}, \{b\} \rangle$, $b_5 = \langle \{c\}, \{d\} \rangle$, $b_6 = \langle \{e\}, \{c\} \rangle$ and $b_7 = \langle \{d\}, \{c\} \rangle$. The *indegree*(r) of vertex r is defined to be the number of times vertex r appears in the heads of hyperarcs. Similarly, the *outdegree*(r) of vertex r is the number of times vertex r appears in the tails of hyperarcs. The set of vertices that appear in the tail or head of a hyperarc is called a *hypernode*. A $B' \subseteq B$ *induced directed subhypergraph* $D'(R', B')$ of $D(R, B)$ is defined as a directed hypergraph with $R' = (\bigcup_{b \in B'} \text{head}(b)) \cup (\bigcup_{b \in B'} \text{tail}(b))$.

Directed hypergraphs are also known as AND/OR graphs [20, p. 21]. In the *AND/OR graph representation*, a directed graph is constructed with two types of nodes: AND nodes which represent hyperarcs (bids) and OR nodes which represent vertices (resources). Figure 2(b) shows the AND/OR graph representation of the example in Fig. 2(a). In the figure, the white nodes represent the AND nodes and the black nodes represent the OR nodes.

For the example in Fig. 2, the possible maximal (maximal in the sense that the set cannot be grown bigger) bartering solutions are $S_1 = \{b_1, b_2, b_3, b_4, b_5\}$ and $S_2 = \{b_1, b_2, b_5, b_7\}$. If our objective is to maximize the number of bids satisfied, we choose S_1 as our maximum solution. The bold edges in the Fig. 2(a) show this optimal solution.

In summary, the resources and the bids in our multi-resource bartering problem can be represented by a directed hypergraph. The feasible solutions of the bartering problem are bids, B' , that have their head

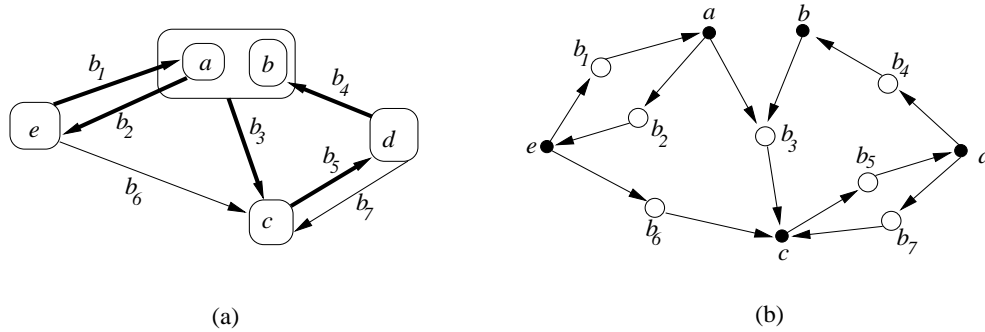


Fig. 2. An example showing bartering bids (a) and its AND/OR graph representation (b).

sets disjoint (because of Rule #3) and whose induced subhypergraphs, $D'(R', B')$, have $outdegree(r) \geq indegree(r)$ for each vertex $r \in R'$ (because of Rules #1 and #2).

3.1. Another example illustrating problem reduction

We give another more complicated example in Fig. 3 to illustrate how some of the unsatisfiable bids and unnecessary resource nodes can be removed. The bold edges in this figure again give the solution that maximizes the number of bids satisfied. We remark the following about this solution:

- The owner of resource a is able to get the resources b and c for free without giving away his resource.
- Some of the bids and resources cannot appear in a feasible solution: These are resources β, x, w and bids b_{18}, b_{19}, b_{20} . It is clear that the resource β cannot appear in a solution, since it does not appear in the tail set of any bid. If we remove β , then the head set of bid b_{20} cannot be satisfied. Hence bid b_{20} cannot be satisfied and must be removed. This removal may propagate. Bids b_{18} and b_{19} and resources w and x are removed because of propagation. Having removed all these resources and the bids that can never be part of a solution, what we are left with is a reduced directed graph which possibly contains disconnected components. In Fig. 3, there are two such components shown in the shaded regions. We will call a connected maximal subset of hyperarcs such that its induced directed subhypergraph has all its vertices, r , with $outdegree(r) > 0$ as a *barter-candidate (bc) component*. Here by connected, we mean the connected underlying undirected AND/OR graph with the directions on the arcs removed. We present the details of a linear time algorithm to compute the bc-components in Section 4.

- If we look at resource s , we see that it was used to satisfy tail sets of two bids of the same barterer. Just like getting something for free is possible, it is also possible to get multiple bids satisfied with the same resource(s). In the next Subsection 3.2, we cover the details of this objective in more detail and also introduce another model in which a resource can only appear exclusively in the tail set of one satisfied bid.

3.2. Other objectives and tail inclusive and exclusive problem cases

So far, we assumed we had the objective of maximizing the number of satisfied bids. Alternatively, in order to give the barterers a chance of dictating the priority (importance) of their bids, we can perhaps let the barterers state what fee they are willing to pay if their bid is satisfied. This can be done by associating a fee f_b with each bid b . Our multi-resource problem then becomes that of coming up with a set of bids that maximize total fees. If we do this, then it is possible that a barterer may not want his resources to appear in the tail set of more than one satisfied bid even if it means that he is able to get more of his bids satisfied. Figure 4 exemplifies these two different objectives. The bold hyperarcs indicate the bids that are satisfied. Note that in this particular problem, all the bids can be satisfied as shown in Fig. 4(a) and we can collect a total of \$235 in fees. But let's say the owner of resource c is willing to pay \$100 to only one satisfied bid involving c . Then the solution that can be returned is shown in Fig. 4(b). In this case, the total fees that will be collected is \$135.

Motivated by the terms inclusive-OR and exclusive-OR, we call the solution objective exemplified by Fig. 4(a) as the *inclusive case* and and that of Fig. 4(b) as the *exclusive case*. Because of bartering Rule #3, it is clear that in a feasible solution the *head* sets of satisfied

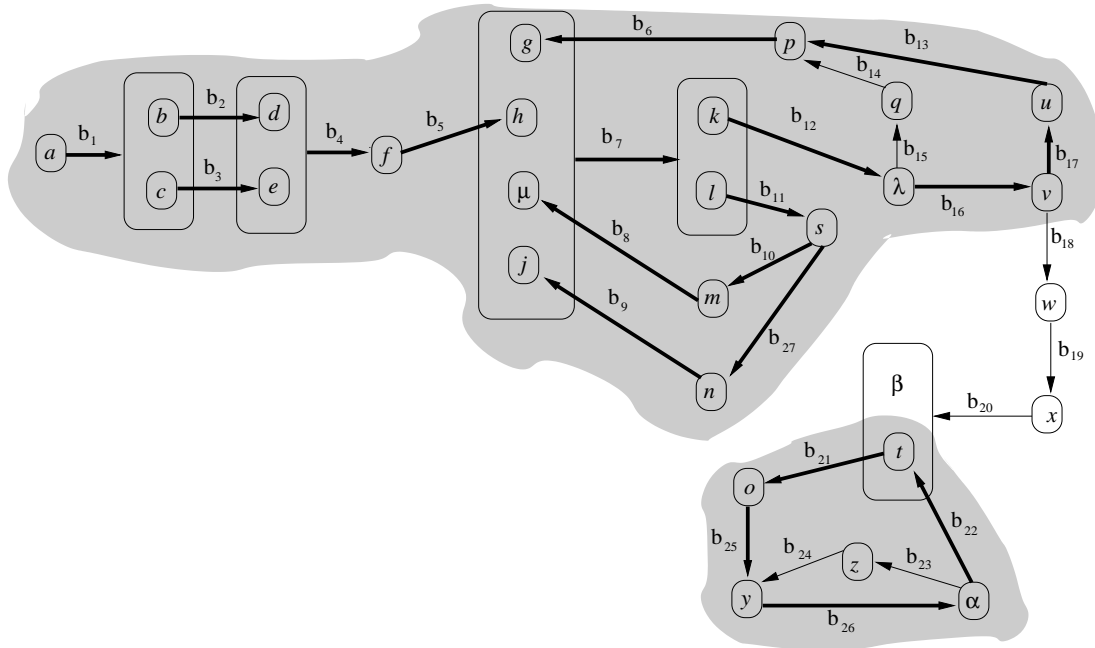


Fig. 3. An example showing bartering bids and the two bc-components present in the directed hypergraph.

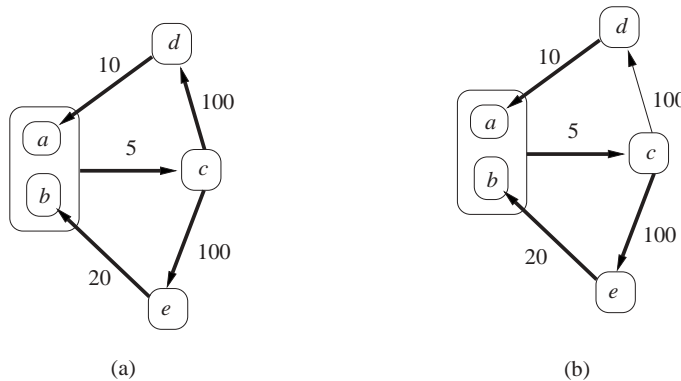


Fig. 4. Example illustrating inclusive solution (a) and exclusive solution (b) for the same problem instance.

bids are disjoint. The exclusive (inclusive) case puts (does not put) similar restrictions on the *tail* sets. In other words, in the exclusive case, tail sets of satisfied bids are disjoint in a feasible solution.

4. Finding BC-components in directed hypergraphs

Let $G(V, E)$ be the AND/OR graph representation of the hypergraph $D(R, B)$. A linear, $(|V| + |E|)$, algorithm for finding the bc-components of the directed hypergraph can be implemented by a two-phase procedure as follows:

- (i) Remove all the resource (OR) nodes that have outdegree equal to 0 either initially or as a result of propagation. Remove also all the bid (AND) nodes that have one or more removed resource nodes in their head sets.
- (ii) Find the connected components on the underlying undirected graph (i.e., graph obtained by removing directions on the arcs) by depth first traversal. The connected components obtained are the bc-components.

It is clear that step (ii) takes $O(|V| + |E|)$. Step (i) functions very similar to the topological sorting algorithm [26, p. 286]. Algorithm for phase (i) is given

```

remove_nodes(V,Nr)
{
  for(r=0 ; r < Nr ; r++) {
    V[r].outdegree = V[r].forward.size() ;
    if (V[r].outdegree == 0) Q.enqueue(r) ;
  }

  while ( Q.size() > 0 ) {
    r = Q.dequeue() ;
    V[r].label = REMOVED ;
    for(j=0 ; j < V[r].back.size() ; j++) {
      b = V[r].back[j] ;
      if (V[b].label == REMOVED) continue ; /* to next j iteration */
      V[b].label = REMOVED ;
      for(k=0 ; k < V[b].back.size() ; k++) {
        V[V[b].back[k]].outdegree-- ;
        if (V[V[b].back[k]].outdegree == 0) Q.enqueue(V[b].back[k]);
      }
    }
  }
}

```

Fig. 5. Procedure for the removal of unsatisfiable bids and resources.

in Fig. 5 using C++/STL (Standard Template Library) syntax. Here, V is the adjacency linked list data structure containing both the resource and the bid nodes as shown in Fig. 6. Both forward arc list (i.e., all u such that $\langle v, u \rangle \in E$) as well as back arc list (i.e., all u such that $\langle u, v \rangle \in E$) are stored for each node v . The size method (function) returns the cardinality of the lists or arrays. The variable Nr is the number of resources (i.e., it is equal to $|R|$). The `remove_nodes` procedure starts by first setting the outdegree field for each resource node and putting those that have outdegree equal to 0 into the queue Q . In the body of the while loop, a resource node with outdegree 0 is removed from the queue and marked as `REMOVED`. All the bid nodes that point to it are also removed (since the resource in the head set of these bid nodes cannot be obtained). For each removed bid node, we also update (decrease by one) the outdegree of resource nodes that appear in its tail set. If outdegree of a resource node becomes 0, it is entered into the queue. The while loop repeats until the queue is empty. It should be clear that this removal process takes $O(|V| + |E|)$.

5. Complexity of the multi-resource bartering

The following theorem establishes the complexity of the multi-resource bartering problem.

Theorem 1. The decision version of the multi-resource bartering problem is NP-complete.

Proof. Given a directed hyper-strongly connected hypergraph $D(R, B)$ which represents multi-resource bartering bids and a positive integer $K_B \leq |B|$, we ask the decision question: Does B contain a multi-resource bartering solution with at least K_B hyperarcs (bids)? Clearly multi-resource bartering problem is in NP, since we can guess a set of K_B directed hyperarcs and check in polynomial time whether the vertices (in the subhypergraph induced by guessed hyperarcs) have (i) indegree either 0 or 1 and (ii) outdegree greater than or equal to the indegree.

We transform the maximum independent set problem [7, p. 53] to the multi-resource bartering problem. An independent set in an undirected graph $G(V, E)$ is a subset $V' \subseteq V$ such that for all $u, v \in V'$, the edge $\{u, v\}$ is *not* in E . The decision version of the independent set problem asks whether there exists an independent set of size at least K_G in the graph G . We perform a polynomial transformation of the independent set problem to the multi-resource bartering problem as follows: Let $adjedges(v)$ denote the set of edges that are incident on vertex v in graph G . A directed hypergraph $D(R, B)$ is constructed from the graph $G(V, E)$ by letting $R = V \cup E$ and $B = \{ \langle \{v\}, adjedges(v) \rangle : v \in V \} \cup \{ \langle adjedges(v), \{v\} \rangle : v \in V \}$. Clearly this construction takes polynomial time. We illustrate this construction by the example graph given in Fig. 7.

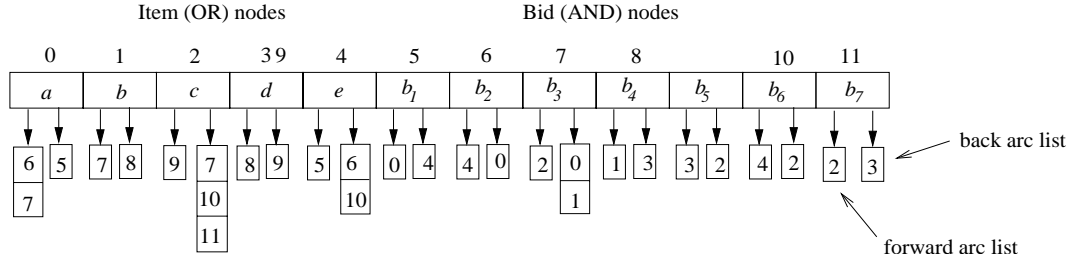
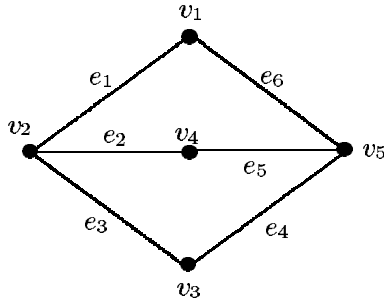


Fig. 6. Adjacency linked data structures for the directed hypergraph in Fig. 1.



$$D(R, B) \text{ where } R = \{v_1, v_2, v_3, v_4, v_5, e_1, e_2, e_3, e_4, e_5, e_6\}$$

$$\text{and } B = \{ \langle \{v_1\}, \{e_1, e_6\} \rangle, \langle \{e_1, e_6\}, \{v_1\} \rangle, \langle \{v_2\}, \{e_1, e_2, e_3\} \rangle, \langle \{e_1, e_2, e_3\}, \{v_2\} \rangle, \langle \{v_3\}, \{e_3, e_4\} \rangle, \langle \{e_3, e_4\}, \{v_3\} \rangle, \langle \{v_4\}, \{e_2, e_5\} \rangle, \langle \{e_2, e_5\}, \{v_4\} \rangle, \langle \{v_5\}, \{e_4, e_5, e_6\} \rangle, \langle \{e_4, e_5, e_6\}, \{v_5\} \rangle \}$$

Fig. 7. Transformation of independent set problem on graph $G(V, E)$ to the bartering problem $D(R, B)$.

Here, an undirected graph with 5 vertices and 6 edges are given. The corresponding directed hypergraph with vertices and hyperarcs that will be constructed are also given on the right.

The independent set problem on $G(V, E)$ can then be stated as a bartering problem on $D(R, B)$: Does B contain a multi-resource bartering solution with at least $2K_G$ hyperarcs? To show that solution of the multi-resource bartering problem will give an independent set of size K_G , we note that K_G of these bids will be the $\langle \{v\}, adjedges(v) \rangle$ hyperarcs and the other K_G will be the corresponding $\langle adjedges(v), \{v\} \rangle$ hyperarcs. The bartering solution guarantees that the head sets of satisfied bids are disjoint (because of Rule #3). Hence, the head sets of $\langle \{v\}, adjedges(v) \rangle$ hyperarcs are disjoint, meaning the set of all incident edges, i.e., $adjedges(v)$ of each of the K_G vertices will be disjoint. Since, each vertex v in an independent set of a graph also has $adjedges(v)$ disjoint, then it is clear that the K_G vertices in the tail of $\langle \{v\}, adjedges(v) \rangle$ hyperarcs form an independent set for the graph $G(V, E)$. Conversely, a solution to the independent set problem induces a solution to the bartering problem: Corresponding to each vertex v that appears in the independent set, bids $\langle \{v\}, adjedges(v) \rangle$ and $\langle adjedges(v), \{v\} \rangle$ appear in the solution of the bartering problem.

6. Integer programming formulation

We solve the inclusive and exclusive case multi-resource bartering problems by formulating them as integer programming problems. Let us define an $|R| \times |B|$ hypergraph matrix T as follows:

$$T_{r,b} = \begin{cases} 1 & \text{if resource } r \in tail(b) \\ 0 & \text{otherwise} \end{cases}$$

and similarly the H matrix as:

$$H_{r,b} = \begin{cases} 1 & \text{if resource } r \in head(b) \\ 0 & \text{otherwise} \end{cases}$$

For the example in Fig. 1(a), nonzero entries of matrices T and H are as follows:

T	b_1	b_2	b_3	b_4	b_5	b_6	b_7
a		1	1				
b			1				
c					1		
d				1			1
e	1					1	

H	b_1	b_2	b_3	b_4	b_5	b_6	b_7
a	1						
b				1			
c			1			1	1
d					1		
e		1					

Let us denote $\bar{0}$ as vector of 0's and $\bar{1}$ as a vector of 1's (both of which are of size $|R|$). We also define a fee (or weight) vector f of size $|B|$ which has each component set to the fee (or weight) amount f_b that we would like to maximize.

6.1. Inclusive case

We can now express the inclusive multi-resource bartering problem as the following integer programming problem:

$$\text{Maximize } f^T x \quad (1)$$

subject to constraints :

$$Hx \leq \bar{1} \quad (2)$$

$$(T - H)x \geq \bar{0} \quad (3)$$

$$x_b \in \{0, 1\} \quad \text{with } b \in B. \quad (4)$$

Note that if the variable x_b is 1 in the solution, then this means bid b is satisfied. Constraints of type (2) basically enforce our rule which says that a resource can be in the head set of only one satisfied bid. Constraints of type (3) simply says that the outdegree of a resource must be greater than or equal to the indegree the feasible solution. The objective function maximizes the total fee (weight). If we want to maximize the number of bids satisfied, then we can set $f = \bar{1}$.

6.2. Exclusive case

In order to get a formulation for the exclusive multi-resource bartering problem, we simply add the constraint set $Tx \leq \bar{1}$ to the inclusive multi-resource bartering problem in order to enforce the restriction that a resource can only appear in the tail set of one satisfied bid. However, by substituting this new constraint set into (3), we now see that the $Hx \leq \bar{1}$ constraints become redundant and hence can be removed. As a result, we get the following formulation of the exclusive multi-resource bartering problem:

$$\text{Maximize } f^T x \quad (5)$$

subject to constraints :

$$Tx \leq \bar{1} \quad (6)$$

$$(T - H)x \geq \bar{0} \quad (7)$$

$$x_b \in \{0, 1\} \quad \text{with } b \in B. \quad (8)$$

6.3. Redundant constraints

We also remark that redundant constraints may be generated in (2), (3), (6), and (7). To identify these, let $t_b = \text{tail}(b)$ and $h_b = \text{head}(b)$. Let also $S_t = \{t_1, \dots, t_{|B|}\}$ and $S_h = \{h_1, \dots, h_{|B|}\}$. We will define $\text{conset}(S, r)$ as the set of members of S that contain resource r , i.e., $\text{conset}(S, r) = \{s \in S \mid r \in s\}$. In constraints of type (2), resources that have $\text{conset}(S_h, r)$ identical will have the same row values. Hence, we can delete the redundant constraints of type (2) by finding two or more resources that have identical consets . We can repeat the same process for constraints of type (6) this time by considering $\text{conset}(S_t, r)$. Similarly, for constraints of type (3) and (7), resources which have both $\text{conset}(S_t, r)$ and $\text{conset}(S_h, r)$ identical will have the same row values. The redundant constraints that result in this way can also be removed.

7. Multiple instance multi-resource bartering

We now formulate the most general form of the bartering problem in which we have multiple resources with multiple instances.

To illustrate the multi-instance multiple resource bartering, we present the following example scenario with three barterers:

- Barterer u_1 owns 2 units of resource a and places the following bids:

$$a(2) \Rightarrow b(1) \text{ with fee} = 1$$

$$a(1) \Rightarrow d(1) \text{ with fee} = 1$$

- Barterer u_2 owns 1 unit of resource b places the following bid:

$$b(1) \Rightarrow c(1) \text{ with fee} = 1$$

- Barterer u_3 owns 2 units of resource c and 1 unit of resource d and places the following bid:

$$d(1) + c(2) \Rightarrow a(5) \text{ with fee} = 1$$

- Barterer u_4 owns 3 units of resource a places the following bids:

$$a(3) \Rightarrow c(1) \text{ with fee} = 10$$

$$a(2) \Rightarrow b(1) \text{ with fee} = 5$$

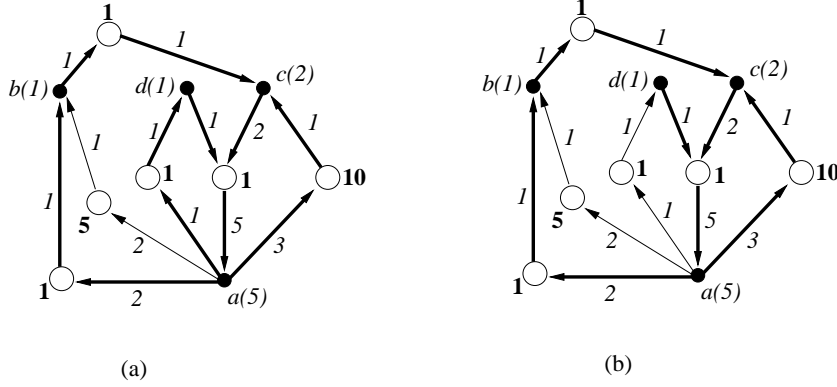


Fig. 8. Multiple instance bartering example with inclusive-MIMR solution (a) and exclusive-MIMR solution (b).

Note that numbers in parantheses above give the number of instances of resources given or requested. The directed hypergraph corresponding to this example is given in Fig. 8.

We now introduce additional notation to facilitate the formulation. Let:

- U stand for the set of barterers and $u \in U$ denote a single barterer.
- $B^{(u)}$ stand for the set of bids made by barterer u .
- $S_r^{(u)}$ stand for the number of instances of resource r owned (supplied) by barterer u .
- $f_b^{(u)}$ stand for the fee associated with bid b of the barterer u .
- $T^{(u)}$ and $H^{(u)}$ stand respectively for the $|R| \times |B^{(u)}|$ tail and head matrices for the user u . Note that $T_{r,b}^{(u)}$ ($H_{r,b}^{(u)}$) denote the number of instances of resource r that is given away (requested) in bid b of barterer u .
- $x_b^{(u)}$ stand for the binary variable associated with bid b of the user u .

For the aforementioned example, the supplies variable has the following values: $S_a^{(u_1)} = 2$, $S_b^{(u_2)} = 1$, $S_c^{(u_3)} = 2$, $S_d^{(u_4)} = 1$, $S_a^{(u_4)} = 1$. Assuming rows 1, 2, 3 and 4 represent the resources a, b, c and d respectively, the head matrices are given as follows:

$$H^{(u_1)} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}, H^{(u_2)} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix},$$

$$H^{(u_3)} = \begin{bmatrix} 5 \\ 0 \\ 0 \\ 0 \end{bmatrix}, H^{(u_4)} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 0 \end{bmatrix}.$$

Similarly, the tail matrices are given as follows:

$$T^{(u_1)} = \begin{bmatrix} 2 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, T^{(u_2)} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix},$$

$$T^{(u_3)} = \begin{bmatrix} 0 \\ 0 \\ 2 \\ 1 \end{bmatrix}, T^{(u_4)} = \begin{bmatrix} 3 & 2 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}.$$

7.1. Exclusive multiple instance multiple resource (Exclusive-MIMR) bartering

Our earlier formulation given by constraints (6–8) for the single instance version generalizes in a straightforward way to the multiple instance of this problem. We formulate Exclusive-MIMR problem as follows:

$$\text{Maximize } \sum_{u \in U} \sum_{b \in B^{(u)}} f_b^{(u)} \cdot x_b^{(u)} \quad (9)$$

subject to constraints :

$$\sum_{b \in B^{(u)}} T_{r,b}^{(u)} \cdot x_b^{(u)} \leq S_r^{(u)} \text{ for each } (u, r), \quad (10)$$

$$u \in U, r \in R \text{ s.t. } S_r^{(u)} \neq 0,$$

$$\sum_{u \in U} \sum_{b \in B^{(u)}} (T_{r,b}^{(u)} - H_{r,b}^{(u)}) \cdot x_b^{(u)} \geq 0 \quad (11)$$

$$\text{for each } r \in R,$$

$$x_b^{(u)} \in \{0, 1\} \text{ with } u \in U \text{ and } b \in B^{(u)}. \quad (12)$$

In constraint (10) the 2-tuple (u, r) stands for barterer u who owns and offers for bartering resource r . In this constraint, we simply sum the instances of the resource

r that are to be given away by barterer u in his selected bids and constrain this quantity to be no more than the barterer's resource r supply. Constraint (11) is the multi-instance version of the constraint (7) previously given.

7.2. Inclusive multiple instance multiple resource (Inclusive-MIMR) bartering

Unlike the exclusive-MIMR case, it is not easy to start with the single instance inclusive constraints (2–4) and extend them in a straightforward way to the multiple instance case. It is better to rethink the inclusive-MIMR constraints carefully. If we concentrate on a resource r , then we surely should not give away more than the total number of instances that is owned by the barterers. However, is this constraint enough? As in the single instance case, the answer is no. We have to give only if we are able to satisfy our bids that offer them. In the single instance case, since a resource has only one unit, that means there is single barterer owning it and hence there is no possibility of a resource being used to satisfy (i.e., appear in the tail of) bid(s) of another barterer. But in the case of inclusive-MIMR, since we keep the total number of resources, using the total number of resources only to constrain the number of resources given away exposes us to the danger of someone's resource being used to satisfy others' bids. Therefore, we have to limit the supply provided by a barterer to the minimum of the number of units he owns and the quantities that are used in his satisfied bids where this particular resource appear. We express this as follows:

$$\sum_{u \in U} \sum_{b \in B^{(u)}} H_{r,b}^{(u)} \cdot x_b^{(u)} \leq \sum_{u \in U} \min \left(S_r^{(u)}, \sum_{b \in B^{(u)}} T_{r,b}^{(u)} \cdot x_b^{(u)} \right) \text{ for each } r \in R. \quad (13)$$

Let us now go back to the *single* instance case and see whether the above constraint indeed generates the single instance inclusive constraints (2–3). In the single instance, we have $S_r^{(u)} = 1$ and one barterer u' that owns the particular resource r . So, the above inequality simplifies to:

$$\sum_{u \in U} \sum_{b \in B^{(u)}} H_{r,b}^{(u)} \cdot x_b^{(u)} \leq \min \left(1, \sum_{b \in B^{(u')}} T_{r,b}^{(u')} \cdot x_b^{(u')} \right) \text{ for each } r \in R.$$

We can get rid of the min term by writing the above as two constraints:

$$\sum_{u \in U} \sum_{b \in B^{(u)}} H_{r,b}^{(u)} \cdot x_b^{(u)} \leq 1 \text{ for each } r \in R,$$

$$\sum_{u \in U} \sum_{b \in B^{(u)}} H_{r,b}^{(u)} \cdot x_b^{(u)} \leq \sum_{b \in B^{(u')}} T_{r,b}^{(u')} \cdot x_b^{(u')} \text{ for each } r \in R.$$

We can conclude that the above two constraints are indeed the single instance inclusive constraints 2 and 3 respectively. If we return back to the inclusive-MIMR case, instances of a resource can be owned by more than one barterer. If there are n_r barterers that own resource r , then we can get rid of the min term by writing 2^{n_r} constraints each of which takes either the $S_r^{(u)}$ term or the $\sum_{b \in B^{(u')}} T_{r,b}^{(u')} \cdot x_b^{(u')}$ term from each barterer. For example, if we have two barterers u_1 and u_2 that own resource r , then we would write the following four constraints:

$$\sum_{u \in U} \sum_{b \in B^{(u)}} H_{r,b}^{(u)} \cdot x_b^{(u)} \leq S_r^{(u_1)} + S_r^{(u_2)},$$

$$\sum_{u \in U} \sum_{b \in B^{(u)}} H_{r,b}^{(u)} \cdot x_b^{(u)} \leq S_r^{(u_1)} + \sum_{b \in B^{(u_2)}} T_{r,b}^{(u_2)} \cdot x_b^{(u_2)},$$

$$\sum_{u \in U} \sum_{b \in B^{(u)}} H_{r,b}^{(u)} \cdot x_b^{(u)} \leq \left(\sum_{b \in B^{(u_1)}} T_{r,b}^{(u_1)} \cdot x_b^{(u_1)} \right) + S_r^{(u_2)},$$

$$\sum_{u \in U} \sum_{b \in B^{(u)}} H_{r,b}^{(u)} \cdot x_b^{(u)} \leq \sum_{b \in B^{(u_1)}} T_{r,b}^{(u_1)} \cdot x_b^{(u_1)} + \sum_{b \in B^{(u_2)}} T_{r,b}^{(u_2)} \cdot x_b^{(u_2)}.$$

The above enumeration scheme may be attractive if n_r is a small number. For larger n_r , an alternative procedure is to introduce additional variables that replace the min terms. We introduce a nonnegative integer variable, $y_r^{(u)}$ in place of the corresponding min term. Hence, for a particular resource r , we can now rewrite the constraint (13) as the following constraints:

$$\sum_{u \in U} \sum_{b \in B^{(u)}} H_{r,b}^{(u)} \cdot x_b^{(u)} \leq \sum_{u \in U} y_r^{(u)},$$

$$y_r^{(u)} \leq \sum_{b \in B^{(u)}} T_{r,b}^{(u)} \cdot x_b^{(u)},$$

$$y_r^{(u)} \leq S_r^{(u)},$$

$$0 \leq y_r^{(u)}.$$

We can now give the complete inclusive-MIMR problem formulation as follows:

$$\text{Maximize } \sum_{u \in U} \sum_{b \in B^{(u)}} f_b^{(u)} \cdot x_b^{(u)} \quad (14)$$

$$\left(\sum_{b \in B^{(u)}} T_{r,b}^{(u)} \cdot x_b^{(u)} \right) - y_r^{(u)} \geq 0 \text{ for each } (u, r), \quad u \in U, r \in R \text{ s.t. } S_r^{(u)} \neq 0, \quad (15)$$

$$\sum_{u \in U} y_r^{(u)} - \sum_{u \in U} \sum_{b \in B^{(u)}} H_{r,b}^{(u)} \cdot x_b^{(u)} \geq 0 \quad (16)$$

for each $r \in R$,

$$x_b^{(u)} \in \{0, 1\} \text{ with } u \in U \text{ and } b \in B^{(u)}, \quad (17)$$

$$y_r^{(u)} \in \{0, \dots, S_r^{(u)}\} \text{ for each } (u, r), \quad (18)$$

$u \in U, r \in R \text{ s.t. } S_r^{(u)} \neq 0.$

Note that we only create the $y_r^{(u)}$ variables if $S_r^{(u)} \neq 0$. In the summation over u in constraint (16), we implicitly assume that $y_r^{(u)} = 0$ for the $S_r^{(u)} = 0$ cases. The bold arcs in Fig. 8(a) and (b) give the solutions for inclusive and exclusive MIMR problems respectively. Notice that in the exclusive solution in Figure 8(b), barterer u_3 has to give only $c(2)$ and that he can keep $d(1)$.

8. Implementation and tests

We have developed code which solves the multi-resource bartering problem by making use of existing mixed integer programming (MIP) packages. We have used two MIP solvers. The first is the LP_SOLVE [5] which is a linear programming solver that is freely distributed under GNU Lesser license. LP_SOLVE also contains a MIP solver that employs branch and bound technique. The second is the CPLEX MIP Solver [4] which is a commercial package and has a sophisticated MIP solver. We have generated various test cases for single instance multiple resource bartering problem by randomly generating bids. We have used 4 parameters while generating the test cases. These are:

- β : denotes the number of barterers.
- ω : denotes the bound on the number of resources owned by each barterer. The actual number of resources for each barterer is generated randomly in the range $1, \dots, \omega$.
- ρ : denotes the bound on the number of bids made by each barterer. The actual number of bids for each barterer is generated randomly in the range $1, \dots, \rho$.
- ϕ : denotes the bound on the number of resources in the head set of a bid. The actual number of resources for each bid is generated randomly in the range $1, \dots, \phi$.

Note that the number of resources in the tail set of each bid is also generated randomly in the range $1, \dots, \omega$. Finally, the resources that will appear on the head and tail sets are also generated. The rand() function was used on the Solaris operating system as random number generator. The characteristics of a number of tests generated are shown in Table 1.

Solutions of the generated single instance multiple resource bartering problems were obtained by solving (1–8). The weight of each bid was taken to be one so that we could maximize the number of bids satisfied. The tests were carried out on a SUN HPC 4500 server with 400 Mhz processors and 3 GBytes of memory. The execution times for the tests described in Table 1 are given in Table 2. The test runs that have >24 hrs. as their solution times were terminated because they took very long and did not produce any answer after 24 hours. These tests are not meant to be comprehensive performance evaluation of the LP_SOLVE and CPLEX packages. Rather, they were used to test the developed models and routines. The tests, however, do give us some feeling about execution times. We also ran other tests and our observations are the following:

- LP_SOLVE execution times exploded when the number of bids reached above 200.
- The exclusive case always took longer time than the inclusive case. If we look at the constraints for both problems, we see that in the inclusive case, only the head sets of bids in the solution should be disjoint. But in the exclusive case, not only the head sets but also the tail sets should be disjoint. This is probably making the problem computationally harder.
- CPLEX MIP solver worked fast until around 1000 bids. After that it also started to take longer. We had test cases with around 1800 bids which did not terminate within 24 hours even for the inclusive case.

Table 1
Characteristics of tests generated

Test	β	ω	ρ	ϕ	No. of Resources($ R $)		No. of bids($ B $)		No. of bc-components
					original	after bc	original	after bc	
(a)	25	10	10	4	153	141	110	89	1
(b)	25	10	10	5	139	133	146	126	1
(c)	35	5	10	3	103	95	212	172	1
(d)	40	10	10	4	217	206	248	216	1
(e)	100	10	10	4	541	513	623	543	1
(f)	400	10	10	4	2727	2360	2632	1861	1

Table 2
Solution times (in seconds) for the various tests given in Table 1

Test	Inclusive Case		Exclusive Case	
	LP_SOLVE	CPLEX	LP_SOLVE	CPLEX
(a)	0.7	0.2	0.9	0.8
(b)	24.7	0.1	98.6	2.6
(c)	6.6	0.4	387.9	2.8
(d)	124	2.06	>24hrs.	18.9
(e)	>24hrs.	1.8	>24hrs.	1334
(f)	>24hrs.	45318	>24hrs.	>24hrs.

9. Discussion and conclusion

In this paper, we have proposed the use of direct bartering techniques for managing resources in grids. We believe that bartering techniques offer advantages since they make it possible to have more complex resource trading patterns. This, for example, may help us to get rid of liquidation problems and may make trading possible in cases where due to subjective pricing a solution may not exist. We have presented a new mathematical model for the multi-resource bartering problem by representing it as a directed hypergraph problem. We have also presented the notion of bc-components in directed hypergraphs which help us to remove unnecessary resources and bids from the original problem. We have implemented programs that make use of existing integer programming solvers. We have performed a number of tests. However, in order to determine suitability of the MIP packages for real life use, more computational experiments need to be done.

We have also proposed a so-called bartergrid where items all over the world could be bartered. Such a bartergrid would be an excellent medium for trading electronic media. Looking at recent news, we believe that a bartergrid can even be established for the matching of other items such as human organs. The clever three-way transplant done by doctors in Johns Hopkins University [14,24] is in fact an excellent example that demonstrates the power of bartering. We also note that existing computational grids can also make use bartering: For example, different computer centers can barter jobs with each other.

The software we have developed for testing our models is called BarterMachine. An alpha version of this software is available on the Internet [19]. Users can enter bids and the barter engine will return the multi-resource bartering solution that maximizes the number of bids (or total weights assigned to bids). Note that our publicly available software uses LP_SOLVE and can only solve small problems.

As the results of our tests show, faster solvers are needed to solve the multi-bartering problem. However these results should not make us pessimistic about adapting bartering in real life. Heuristics can be built for example by filtering bids so that polynomial time algorithms (single/multiple instance and single resource bartering) can be applied at first to find some solutions if there exists any. Heuristics can also be devised that partition the large number of bids into smaller sets that can be solved by the optimal solver in parallel in a reasonable amount of time. Another approach that can be followed is that we can enforce small number of bids by (i) keeping the bidding time short and/or (ii) limiting the number of bids per user. Hence when we reach a threshold number that can be solved in a reasonable time by the optimal solver, the algorithm is run. Winning bids can be removed from the system and the remaining stay for the next run of bartering solver and so on.

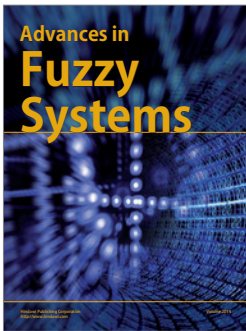
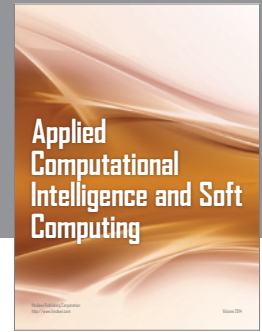
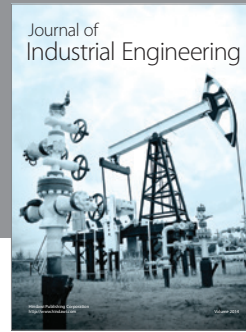
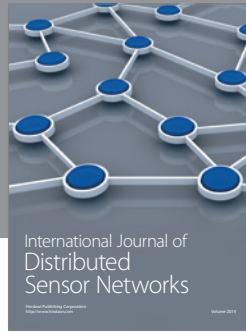
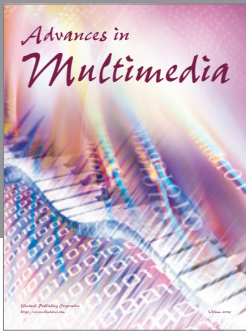
Another issue that needs to be clarified is that when developing our model, we assumed the barterers always bid non-empty set of resources. But what happens if a barterer wants to give away resource(s) for free or bid for resource(s) by offering nothing for them. This problem can also be taken care of in our model. We introduce a fictitious (dummy) resource called NOTHING. Then if someone wants to give away resource(s) for free, he places the bid: resource(s) => NOTHING. If on the other hand, someone wants to acquire resources(s) for free, then he places the bid: NOTHING => resource(s). We can also adjust bid priorities (fees) in order to have the NOTHING bids to be picked up in a solution. Let n be the number of bids that contain

NOTHING. We can let each NOTHING bid to have a fee of 1 and the other bids to have a fee of $f_b \cdot (n + 1)$ where f_b is the original fee associated with each bid. In this way, we give more priority to the bids that trade non-empty set of resources.

Finally, we would like to remark that some economists may object to bartering saying that it is not an efficient market mechanism. However, if bartering is studied from a computer science perspective and efficient algorithms and software are designed for it, then we believe that a serious barter economy will develop and compete with monetary economy.

References

- [1] R.K. Ahuja, T.L. Magnanti and J.B. Orlin, *Network Flows, Theory, Algorithms and Applications*, Prentice Hall, 1993.
- [2] J. Bang-Jensen and G. Gutin, *Digraphs: Theory, Algorithms and Applications*, Springer-Verlag, 2000.
- [3] R. Buyya, D. Abramson, J. Giddy and H. Stockinger, Economic Models for Resource Management and Scheduling in Grid Computing, *J. of Concurrency and Computation: Practice and Experience* **14** (2002), 13–15.
- [4] CPLEX, <http://www.ilog.com>.
- [5] M. Berkelaar, LP_SOLVE, Linear programming code, ftp://ftp.es.ele.tue.nl/pub/lp_solve.
- [6] M. Engineer, Bargains, barter and money, *Review of Economic Dynamics* **4** (2001), 188–209.
- [7] M. Garey and D.S. Johnson, *Computers and Intractability, A Guide to the Theory of NP-Completeness*, W.H. Freeman and Company, 1979.
- [8] G. Gutin, Finding a longest path in a complete multipartite digraph, *SIAM J. Discrete Math.* **6** (1993), 270–273.
- [9] J. Hu, *Learning in Dynamic Non-Cooperative Multi-agent Systems*, PhD thesis, 1999.
- [10] W.S. Jevons, *Theory of Political Economy*, MacMillan and Co., New York, 1888.
- [11] R. Jones, The origin and development of Media of Exchange, *J. of Political Economy* **84** (1976), 757–776.
- [12] J.R. Kalagnanam, A. Davenport and H.S. Lee, Computational aspects of clearing continuous call double auctions with assignment constraints and indivisible demand. *Electronic Commerce Research Journal* **1** (2001), 221–238.
- [13] Kazaa Network, (<http://www.kazaa.com>).
- [14] D. Kohn, *Three-way kidney transplant makes history, happy patients*, Baltimore Sun, Aug. 2, 2003. (<http://www.baltimoresun.com>).
- [15] V. Krishna, *Auction Theory*, Academic Press, 2002.
- [16] Napster, (<http://www.napster.com>).
- [17] C. Ozturan, Network flow models for electronic barter exchanges, Technical Report FBE/CMPE01/2001-9, Dept. of Computer Engineering, Bogazici University, Istanbul, July 2002. To appear in *J. of Organizational Computing and Electronic Commerce*, (<http://www.cmpe.boun.edu.tr/ozturan/netflowbarex.ps>).
- [18] C. Ozturan, Used car salesman problem: A differential auction-barter market, To appear in *Annals of Math. and Artificial Intelligence*, (<http://www.cmpe.boun.edu.tr/ozturan/aucbar.ps>).
- [19] C. Ozturan, BarterMachine Software, (<http://www.cmpe.boun.edu.tr/ozturan/bm.tar> and <http://www.bartergrid.org>).
- [20] J. Pearl, *Heuristics*, Addison Wesley, 1984.
- [21] R. Raman, M. Livny and M. Solomon, *Matchmaking: Distributed Resource Management for High Throughput Computing*, Proceedings of the Seventh IEEE International Symposium on High Performance Distributed Computing, Chicago, IL., July 28–31, 1998.
- [22] R. Raman, M. Livny and M. Solomon, *Policy Driven Heterogeneous Resource Co-Allocation with Gangmatching*, Proceedings of the Twelfth IEEE International Symposium on High-Performance Distributed Computing, Seattle, WA., June, 2003.
- [23] S. Shinohara and Y.P. Gunji, Emergence and collapse of money through reciprocity, *Applied Math. and Computation* **117** (2001), 131–150.
- [24] A.B. Swingle, Triple-Swap transplant, *DOME* **54**(7) Sept 2003, (http://www.hopkinsmedicine.org/dome/0903/top_story.cfm).
- [25] Top Ten Business Barter Sites, (<http://www.top10links.com/cat.php/Business:Barter/>).
- [26] M.A. Weiss, *Data Structures and Algorithm Analysis in C*, Addison Wesley, 1997.
- [27] P.R. Wurman, W.E. Walsh and M.P. Wellman, Flexible double auctions for electronic commerce, *Decision Support Systems* **24** (1998), 17–27.
- [28] R. Wolski, J. Brevik, J. Plank and T. Bryan, Grid resource allocation and control using computational economies, in: *Grid computing: Making the global infrastructure a reality*, F. Berman, G.C. Fox and A.J.G. Hey, eds, 2003, pp. 747–787.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

