Mechanism Design on Trust Networks

Arpita Ghosh
Yahoo! Research

Mohammad Mahdian
Yahoo! Research

Daniel M. Reeves
Yahoo! Research

David M. Pennock
Yahoo! Research

Ryan Fugger
RipplePay.com

Ver. 105: 2007-07-14 02:25

Abstract

We introduce the concept of a trust network—a decentralized payment infrastructure in which payments are routed as IOUs between trusted entities. The trust network has directed links between pairs of agents, with capacities that are related to the credit an agent is willing to extend another; payments may be routed between any two agents that are connected by a path in the network. The network structure introduces *group budget constraints* on the payments from a subset of agents to another on the trust network: this generalizes the notion of individually budget constrained bidders.

We consider a multi-unit auction of identical items among bidders with unit demand, when the auctioneer and bidders are all nodes on a trust network. We define a generalized notion of social welfare for such budget-constrained bidders, and show that the winner determination problem under this notion of social welfare is NP-hard; however the flow structure in a trust network can be exploited to approximate the solution with a factor of $1 - 1/e$. We then present a pricing scheme that leads to an incentive compatible, individually rational mechanism with feasible payments that respect the trust network's payment constraints and that maximizes the modified social welfare to within a factor $1 - 1/e$.

1 Introduction

Consider an auction where payments take the form of IOUs. That is, the winning buyer(s) do not immediately pay with dollars or other standardized currency, but instead commit to compensate the seller appropriately at some later date. In this setting, the seller must consider not only the magnitude of a buyer's bid, but also the risk of the buyers defaulting on their commitments. Naturally the seller may not wish to accept a large IOU from an unknown or untrustworthy bidder.

Now suppose the seller will not accept an IOU from buyer Alice. Alice might still be able to compete in the auction if someone that the seller does trust, say Bob, in turn trusts Alice. Then Alice can pass an IOU to Bob who can pass an IOU to the seller. The seller is paid with a commitment from Bob, someone the seller trusts, and Bob receives a commitment from Alice, someone he trusts.

In Section 3, we formalize this notion of a payment infrastructure based on a trust network. The trust network induces pairwise (directed) limits on how much compensation can flow from any one agent to another. In this way, trust networks generalize the notion of budget constraints where, instead of a single budget per agent denominated in a common currency, there may exist different budgets for every subset of agents. Note that it is not enough to consider budget limits on how much individual agents can pay the seller since, for example, Bob may be a bottleneck. That is, there may be multiple buyers for whom the seller requires Bob to vouch for their payments. Thus, the amount that one buyer can pay the seller depends on the degree to which other buyers have exhausted the Bob link.

We examine the problem of mechanism design in trust networks. In particular, we consider the generalized winner determination problem when the mechanism must explicitly account for the limits that the trust network imposes on possible payments.

We consider a multi-item auction of identical items. The auctioneer is a node in the network, and payments to the auctioneer are constrained by link capacities: The payments from a subset of bidders cannot exceed the maximum flow from these nodes to the auctioneer on the trust network. It is not possible to design incentive compatible mechanisms to maximize social welfare in this setting so we define a modified notion of social welfare based on budget-capped values.

In Section 4, we show that the winner determination problem—choosing the set of winners that maximizes this modified notion of welfare—is NP-hard. We also describe in Section 4.3 how the flow structure in the trust network can be exploited to approximate the solution within a factor $1-1/e$. In Section 5, we present a pricing scheme that leads to an incentive compatible, individually rational mechanism with feasible payments that respect the group budgets and that approximately maximizes the modified social welfare to within a factor $1-1/e$.

The next section compares trust networks with more traditional payment infrastructures and describes the existing trust network implementations that motivate our research.

2 Payment as a routing problem

Currencies can in fact operate as abstract IOUs, or obligations. Modern currencies are issued in the form of abstract obligations to provide value of some form, be it banks' obligations to redeem account balances for government notes, governments' obligations to redeem those notes as credit toward taxes due, or e-gold's obligations to store gold in trust for account holders. A decision to accept a certain currency[1] is a decision to trust the issuer to fulfill its obligations. From this perspective, a loan repayment agreement is currency issued by the borrower and accepted by the lender.

Payment is the transfer of obligations from one entity, the payer, to another, the recipient, in a form the recipient will accept. In other words, to make payment, the payer must present obligations from a currency issuer that is trusted by the recipient. The payer is faced with the problem of how to route the payment: how to convert obligations that it holds or can readily obtain (for example, via a line of credit) into obligations from an issuer that the recipient considers trustworthy. This routing takes place in a trust network.

The most ubiquitous routable financial trust network is the banking system. At the national level this is essentially a tree, with the central bank at the root, regular banks as children of the central bank, and bank customers as the leaves. This arrangement makes it feasible to route payments manually, since there is only one path between any two nodes in a tree.

[1] Currency here is defined as obligations from a certain issuer, as considered separately from the units of value in which those obligations are accounted.

Analogy to computer networks. Computer networks are built to route information from one computer to another. The evolution of computer networks follows a similar course to that of currency networks. For a small network, computers can be directly connected to each other as needed using wires. As the number of computers grows, this soon becomes unwieldy, and it is easier to connect all the computers to a special intermediary computer (a router), which relays information between computers in the network. Routers accept and transmit data like any other computer, but act as hubs for transferring messages between computers because they are highly connected in the computer network, just as banks act as hubs for transferring obligations between people because they are highly connected in the financial trust network.

Eventually, it is desirable to send information between networks, and to accomplish this, several routers can be connected to a super-router, and these in turn can be connected to an even higher router, and so on in a hierarchical fashion as needed. Since there is only a single route between any two points, routing messages in strictly hierarchical networks is simple.

The designers of the internet did not build it as a strict hierarchical network—primarily because to withstand a nuclear attack, it couldn't have any single points of failure. As a side effect, the Internet can operate as the most democratic forum for communication ever known, because it does not require, and is in fact resistant to, control by special groups. A non-hierarchical financial network can have similar advantages.

Two systems recently implemented by the authors—first Ripple [Fugger, 2004] and then Yootles [Reeves et al., 2006]—demonstrate this powerful generalization of the usual financial trust tree. These systems route payments through arbitrary financial trust networks much like the internet routes data through arbitrary computer networks, demonstrating how advances in routing enable the formation of decentralized routable payments. This paper formally defines the concept of a trust network.

The Yootles system allows its users to conduct auctions of a variety of types [Reeves et al., 2006]. This paper describes how auction design is impacted by the payment constraints implied by the network.

3 The model

We first define our proposed decentralized payment infrastructure. We denominate the hypothetical currency in utils, representing an abstract measure of utility [Reeves et al., 2006], but this choice is orthogonal to our results.

3.1 Trust networks

A trust network, or decentralized ledger, consists of two directed graphs defined on a set of vertices $V = \{0, \ldots, m\}$ representing entities, or agents. (Vertex 0 will be treated specially in the next section.) A set of edges E_O gives the pairwise account balances between nodes. The weight o_{ij} on an edge $(i, j) \in E_O$ quantifies the obligations that i has to j, that is, i is committed to increasing j's utility by o_{ij} utils or, if $o_{ij} < 0$ then j owes $-o_{ij}$ utils to i. By definition, $o_{ij} = -o_{ji}$ for all i, j and $o_{ii} \equiv 0$.

A set of edges E_T gives pairwise credit limits between agents. The weights on these edges quantify the trust in the trust network. An edge $(i, j) \in E_T$ with weight t_{ij} specifies that i has extended j a credit line of t_{ij} utils. In practice these edges may have concomitant interest rates

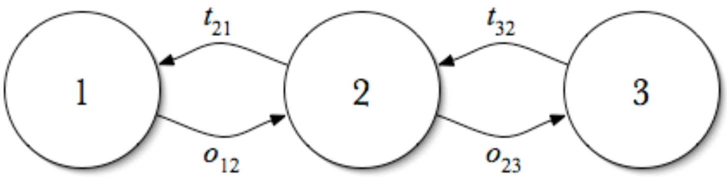

Figure 1: A trust network with three agents. Credit extended from 1 to 2 and from 2 to 3 are not shown.

and there may be multiple lines of credit issued between agents at different interest rates. In this paper we ignore interest and assume that every directed pair of agents has exactly one credit limit, possibly zero.

The power of a trust network defined above is that, if sufficiently well-connected, arbitrary payments can be made by passing obligations between agents that explicitly trust each other. For example, in Figure 1 agent 1 can make a payment of x utils to agent 3 (which 1 has no connection to) by issuing an obligation of x utils to agent 2 and agent 2 issuing an obligation for the same amount to agent 3, increasing both o_{12} and o_{23} by x utils. (Note that agent 2's net balance remains unchanged.) The payment is feasible as long as both 1's remaining credit with 2 (i.e., $t_{21} - o_{12}$) and 2's remaining credit with 3 (i.e., $t_{32} - o_{23}$) are greater than or equal to the payment amount of x utils. This generalizes to arbitrarily long payment chains in the obvious way. The maximum x satisfying the credit constraints along a path from i to j is the *payment capacity* of that path. The overall payment capacity for (i, j) is the amount that could be paid from i to j if each path from i to j was maxed out in sequence—that is, the *maximum flow* [Ahuja et al., 1993] from i to j.[2]

3.2 Auctions on trust networks

We consider multi-unit auctions of k identical items among n bidders on the trust network. We label the nodes $V = \{0, \ldots, m\}$ so that auctioneer is node 0, and the bidders are $\mathcal{A} = \{1, \ldots, n\}$. (The remaining nodes $\{n + 1, \ldots, m\}$ may be used for routing payments, but do not participate in the auction.)

To study this problem, we do not need to consider the account balances and credit limits on the edges of the trust network separately—all that matters is the remaining credit on a link. The trust network can therefore be defined by a single graph G comprising the same set of vertices V and a set of edges E representing the payment capacities of edges, where an edge $(i, j) \in E$ has capacity $c_{ij} = t_{ji} - o_{ij}$.

Every bidder i has a private value v_i for the item. Bidders have unit demand—that is, they want no more than one unit of the item. We assume that the network structure and link capacities c_{ij} are publicly known: bidders cannot strategically report link capacities to the auctioneer.

The link capacities limit the maximum payment that can be made by any subset of bidders $S \subseteq \mathcal{A}$ to the auctioneer. We denote by $\mathrm{c}(S)$ the maximum flow that can be routed from S to 0 on the graph with link capacities c_{ij}; $\mathrm{c}(S)$ is a *group budget*, or combined budget, for the nodes

[2] In practice, the system may limit routing, for example by only considering paths up to a certain length or only considering a subset of all paths. If so, the system will be computing lower bounds on the true payment capacities. Payment feasibility thus degrades gracefully with computational restrictions on payment routing.

in S. This generalizes the notion of individually budget-constrained bidders [Borgs et al., 2005]. However, note that our setting is a special case of combined budgets: not all values for combined budgets can be derived from maximum flow constraints on a network with link capacities.

We will refer to payments that can be routed along the trust network without violating any link capacity constraints as *feasible payments*. Feasible payments correspond exactly to those where the total payment from every subset of nodes is less than or equal to the budget constraint for that subset.

Due to the budget constraints, it is not possible to design an incentive compatible mechanism to maximize social welfare, or the sum of private values of winning bidders.[3] Instead we define a modified notion of welfare. For any subset of bidders $S \subseteq \mathcal{A}$, let $\mathrm{v}(S)$ denote the budget-capped value of this set, defined as the optimal value of the linear program with variables x_i:

$$\begin{aligned} \text{maximize} \quad & \sum_{i \in S} x_i \\ \text{s.t.} \quad & \sum_{i \in T} x_i \leq \mathrm{c}(T) \quad \forall\, T \subseteq S \\ & 0 \leq x_i \leq v_i. \end{aligned} \tag{1}$$

In a regular auction, a bidder's private value can be thought of as the maximum individually rational payment the bidder is willing to make to the auctioneer; here the budget-capped value of a set of bidders is the maximum individually rational payment from this set of bidders that are feasible on the trust network. When the group budgets, $\mathrm{c}(T)$, are sufficiently large, that is, $\mathrm{c}(T) \geq \sum_{i \in T} v_i$ for all T, then the modified welfare is exactly the sum of the valuations of the bidders: $\mathrm{v}(S) = \sum_{i \in S} v_i$.

We define $\mathrm{v}(S, b)$ as the value of (1) with v_i replaced by b_i (*i.e.*, with the constraint $0 \leq x_i \leq b_i$). As usual, we will use b_{-i} to denote the vector b with the ith component removed.

We will be interested in mechanisms that maximize modified welfare—that is, choose as winners a set of bidders with highest budget-capped value.

4 Complexity of winner determination

In this section we study the winner determination problem in a k-unit auction on a trust network. The problem is to select a set of at most k bidders on a trust network to maximize the budget-capped social welfare defined in the previous section. This is essentially equivalent to the problem of selecting k sources (among a pre-specified set of possible sources) in a graph that can send the maximum amount of flow to a given destination (the auctioneer). The problem can be studied in two models: one where ex-post individual rationality is required (the ex-post IR model), and another where it is enough to satisfy ex-ante individual rationality (the ex-ante IR model).

4.1 Problem formulation

We start by formulating the problem in the ex-post IR model as a mathematical program.

[3] To see why, consider a mechanism that tries to maximize welfare in the face of budget-constrained bidders. The mechanism would have to base allocation decisions on agents' reported values—values that can be greater than what the agents could pay. But then low valuation agents with the same ability to pay would report valuations that made them appear identical to the high-valuation agents.

$$
\begin{aligned}
\text{maximize} \quad & \sum_{i\in\mathcal{A}} x_i && \text{(2a)}\\
\text{s.t.} \quad & \forall i \in \mathcal{A}:\ x_i \le v_i y_i && \text{(2b)}\\
& \forall u \in \{n+1,\ldots,m\}:\ \sum_{(u,w)\in E} z_{u,w} = \sum_{(w,u)\in E} z_{w,u} && \text{(2c)}\\
& \forall u \in \mathcal{A}:\ \sum_{(u,w)\in E} z_{u,w} - \sum_{(w,u)\in E} z_{w,u} \ge x_i && \text{(2d)}\\
& \forall (u,w) \in E:\ 0 \le z_{u,w} \le c_{u,w} && \text{(2e)}\\
& \sum_{i\in\mathcal{A}} y_i \le k && \text{(2f)}\\
& \forall i \in \mathcal{A}:\ y_i \in \{0,1\} && \text{(2g)}
\end{aligned}
$$

The binary variable y_i in the above program indicates whether the bidder i is selected as a winner. The variable x_i is the amount of "value" extracted from bidder i. The constraints (2c)–(2e) guarantee that these amounts are routable through the trust network. The variable $z_{u,w}$ in these constraints corresponds to the amount of flow routed through the directed edge (u, w) in the graph.

The ex-ante IR problem can be formulated similarly, except the constraint (2g) is relaxed to $0 \le y_i \le 1$. The value of y_i means that the bidder i receives a unit of the good with probability y_i. Note that the ex-ante IR property allows us to charge a bidder, who (due to the outcome of the coin flip) does not receive any item, as long as the expected value the bidder receives is not more than the expected amount she pays. Using this formulation, the winner determination problem in the ex-ante IR case can be solved exactly by solving a linear program. For the rest of this section, we will focus on the winner determination problem in the ex-post IR model, and show tight hardness and approximability results.

4.2 Hardness of the ex-post problem

The following theorem shows that the winner determination problem is hard to approximate within any factor better than $1 - 1/e$, even if all edges of the trust network have capacity 1.

Theorem 1. *If the winner determination problem for ex-post IR multi-unit auctions on trust networks can be approximated within a factor of* $1-1/e+\varepsilon$ *for any* $\varepsilon > 0$*, then* $NP \subseteq TIME(n^{O(\log\log n)})$.

Proof. We reduce the problem of maximum k-coverage to this problem. An instance of the max k-coverage problem consists of a number k and a collection of subsets $S_1, S_2, \ldots, S_p$ of a universe $\mathcal{U}$. The goal is to find a subcollection $S_{i_1}, \ldots, S_{i_k}$ of size k whose union has the maximum size. Given such an instance, we construct an instance of the winner determination problem as follows: the parameter k corresponds to the number of items available for sale, each set S_i corresponds to a bidder i, and each element of $\mathcal{U}$ corresponds to a non-bidder node in the trust network. The only other node in the trust network corresponds to the auctioneer, which is denoted by 0. For every element $j \in \mathcal{U}$, there is an edge from j to 0, and for every $i \in \{1, \ldots, p\}$ and $j \in S_i$, there is an edge from the vertex i to the vertex j. The capacity of all edges are 1. The value of each bidder $i \in \{1, \ldots, p\}$ is $|S_i|$. It is easy to observe that the budget-capped value of a collection of bidders is equal to the size of the union of the corresponding sets. Therefore the solution of the winner determination problem is precisely equal to the solution of the max k-coverage problem. The hardness result follows from a theorem of Feige [1998], who show that the max k-coverage problem is hard to approximate within any factor better than $1-1/e$, unless $NP \subseteq TIME(n^{O(\log\log n)})$. □

4.3 Approximation algorithm

The above theorem shows that the ex-post IR winner determination problem is at least as hard as the max k-coverage problem. For the max k-coverage problem, there is a well-known greedy algorithm that achieves an approximation factor of $1 - 1/e$. Using this algorithm, and a lemma proved by Chandra et al. [2004] for a different problem, we can show that the ex-post IR winner determination problem can be approximated within a factor of $1 - 1/e$.

Our algorithm, which is a natural generalization of the greedy algorithm for max k-coverage, proceeds as follows. Start with $S = \emptyset$. In every iteration, select a bidder that maximizes the *marginal value* $\mathrm{v}(S \cup \{i\}) - \mathrm{v}(S)$, and add this bidder to S. Continue this for k iterations, until $|S| = k$.

To prove the approximation factor of this algorithm, we need the following lemma, which is an adaptation of Lemma 3 in Chandra et al. [2004].

Lemma 1. *Let G be a directed graph with capacities on the edges, and S_1 and S_2 be two subsets of vertices of G. Consider a maximum flow f from the vertices in S_1 to a special vertex $0 \notin S_1 \cup S_2$, and let f_i denote the amount of flow originating from the vertex $i \in S_1$ in this solution. Then there is a solution to the maximum flow problem from vertices in $S_1 \cup S_2$ to the vertex 0, in which the amount of flow originating from every vertex $i \in S_1$ is precisely f_i.*

Proof sketch. We use the Ford-Fulkerson [Ahuja et al., 1993] algorithm for solving the maximum flow problem from $S_1 \cup S_2$ to 0. This algorithm proceeds in iterations. In each iteration, the algorithm takes the current feasible flow, and finds an augmenting path to increase the total flow sent from $S_1 \cup S_2$ to 0. The Ford-Fulkerson theorem guarantees that such an augmenting path can be found in any non-optimal flow. To prove the lemma, we apply this algorithm starting from the flow f. If in each iteration we find the shortest augmenting path from $S_1 \cup S_2$ to 0, the path cannot contain any vertex of $S_1 \cup S_2$ as an interior vertex, and therefore it will never change the amount of flow originating from a vertex in S_1. Hence, in the final maximum flow computed by this algorithm, the amount of flow originating from every $i \in S_1$ is precisely f_i. □

Theorem 2. *The greedy algorithm achieves an approximation ratio of $1 - 1/e$ for the winner determination problem in ex-post IR multi-unit auctions on trust networks.*

Proof. Consider an instance of the problem, and let OPT denote the value of the optimal solution on this instance, and S^* denote the set of winners in this solution. Let T_i denote the value of the solution found at the end of the i'th iteration of the greedy algorithm, and set $T_0 = 0$. The main ingredient of the proof is the following inequality, which bounds the amount of marginal value in iteration r:

$$T_r - T_{r-1} \geq \frac{\mathrm{OPT} - T_{r-1}}{k} \tag{3}$$

To prove this, we construct the graph G' constructed from the trust network by adding a *shadow* vertex i' for every bidder i, and connecting i' to i with an edge of capacity v_i. Clearly, the budget-capped value of any set S of bidders is equal to the maximum amount of flow that can be sent from the set of shadow vertices of bidders in S to the vertex 0. Let S_1 denote the set of shadow vertices corresponding to the bidders selected in the first $i-1$ iterations of the greedy algorithm, and S_2 denote the shadow vertices for bidders in S^*. Consider a solution to the maximum flow problem from the vertices in S_1 to 0, and denote by f_i the amount of flow originating from $i \in S_1$ in this solution. By Lemma 1, there is a maximum flow $\tilde{f}$ from the vertices of $S_1 \cup S_2$ to 0 in which the flow originating from any vertex $i \in S_1$ is precisely f_1. On the other hand, since the amount of flow that can be sent from S_2 to 0 is OPT, the value of the flow $\tilde{f}$ is also at least OPT. Therefore in $\tilde{f}$, vertices in $S_2 \setminus S_1$ send at least $\mathrm{OPT} - \sum_{i \in S_1} f_i = \mathrm{OPT} - T_{r-1}$. Since there are at most k vertices in $S_2 \setminus S_1$, there must be a vertex i' in this set (corresponding to the bidder i), which sends at least $(\mathrm{OPT} - T_{r-1})/k$ units of flow to 0. This implies that the marginal value resulting from adding the vertex i in the r'th iteration of the algorithm is at least $(\mathrm{OPT} - T_{r-1})/k$. Since the algorithm always adds a vertex with the highest marginal value, the inequality (3) follows.

Inequality (3) can be re-arranged as follows:

$$\left(1-\frac{1}{k}\right)^{-r} T_r \geq \frac{\mathrm{OPT}}{k}\left(1-\frac{1}{k}\right)^{-r} + \left(1-\frac{1}{k}\right)^{-(r-1)} T_{r-1}.$$

By adding these inequalities for $r = 1, \ldots, k$ and simplifying, we obtain

$$T_k \geq \mathrm{OPT}\left(1-\left(1-\frac{1}{k}\right)^{k}\right) \geq \left(1-\frac{1}{e}\right)\mathrm{OPT}.$$

This completes the proof of the theorem, as T_k is the value of the greedy solution. □

The above proof heavily uses the combinatorial structure of the budgets imposed by the trust network, and therefore does not generalize to the more abstract model of collective budgets. In fact, it is not hard to prove that the winner determination problem in the abstract model cannot be approximated to within any factor better than $n^{1-\varepsilon}$, even if all subsets that have a budget are of size two. This is proved by the following reduction from the maximum independent set problem: each node of the given graph G corresponds to a bidder of value 1, and the collective budget of any pair of bidders connected by an edge in G is 1. No other subset of bidders has a collective budget. It is straightforward to show that the solution of the winner determination problem in this instance corresponds to a maximum independent set in G. By the hardness of the maximum independent set problem Håstad [1999], the winner determination problem in this case is hard to approximate.

4.4 Algorithms for special cases

Despite the hardness result shown in Theorem 1, the winner determination problem can be solved exactly in some special cases, most notably in the case that the trust network is hierarchical (has a tree structure), as in the case of a national banking system.

Theorem 3. *If the underlying undirected graph of the trust network G is a tree, the winner determination problem for ex-post IR multi-unit auctions on G can be solved in polynomial time.*

Proof. First we transform the tree T rooted at the auctioneer into an (incomplete) binary tree T' with bidders as leaf nodes, as follows. For every bidder i, add a shadow node i', and add a link from i' to i with capacity v_i. All leaf nodes in the tree which are not bidders in the auction can be removed with no change to the solution. To convert the tree into a binary tree, an internal node with k children can be replaced by a binary subtree of depth $k-1$ (for example, if an internal node v has children u, w, y, create dummy node v' with children u and w (and link capacities $c_{u,v}$ and $c_{w,v}$); v now has children v' and y (with link capacities ∞ and $c_{y,v}$)). The solution to the maximum flow problem in T' also solves the maximum flow problem in T.

Define $V[v, l]$ to be the maximum flow that can be routed to an internal node v from at most l leaves in the subtree rooted at v. Then

$$V[v, l] = \max_{l_1 + l_2 = l} (\min(V[v_1, l_1], c_{v_1,v}) + \min(V[v_2, l_2], c_{v_2,v})),$$

where v_1 and v_2 are the children of v. For all leaf nodes, $V[v, l] = \infty$ for all l. The solution to the winner determination problem is $V[0, k]$ (recall that 0 is the auctioneer node, and k is the number of items to be sold).

Construct a table that stores the values of $V[v, l]$ for all nodes in T'. Since we added at most one additional node for every internal node in T, and one shadow node for every bidder, the number of nodes in T' is no more than $3|T|$, so the table has size $O(k|T|)$. Every entry in the table can be computed in time $O(k)$. So the optimal solution $V[0, k]$ can be computed in time $O(k^2|T|)$, where $|T|$ is the size of the trust network. □

5 Mechanism design

In this section, we discuss the question of designing an incentive compatible mechanism that maximizes the modified welfare. The solution to the winner determination problem specifies the allocation of items amongst bidders that maximizes, or approximately maximizes, modified welfare; the pricing scheme must be chosen to ensure incentive compatibility, as well as feasible payments.

We show that mechanisms $\mathcal{M}$ and $\mathcal{M}'$, stated below, are incentive compatible, individually rational mechanisms with feasible payments, that respectively maximize and approximately maximize modified welfare: $\mathcal{M}$ assumes that the winner determination problem can be solved exactly, and allocates items according to this solution, while $\mathcal{M}'$ allocates items according to the greedy algorithm in the previous section. (Although the mechanisms look very similar, the proofs for feasibility of payments are different, so we present them separately.)

Mechanism $\mathcal{M}$: Every bidder submits a bid b_i to the auctioneer.

- *Allocation*: The *winning* set is the lexicographically first subset S^* of bidders that maximizes $\mathrm{v}(S, b)$ over all subsets with $|S| \leq k$. Assign the k items to bidders in the winning set S^*.
- *Pricing*: Charge bidder $i \in S$ the smallest value $p_i \leq b_i$ such that i would still belong to the winning set with bids (b_{-i}, p_i).[4]

Note that the winning set need not be S^* with input b_j for $j \neq i$ and p_i—we only require that the winning set contains i.

Theorem 4. *The mechanism $\mathcal{M}$ is incentive compatible, ex-post individually rational, maximizes modified social welfare, and leads to payments that are feasible on the trust network.*

[4] Note that this price p_i need not be the same as the smallest report p'_i at which i still belongs to some set with the highest value (*i.e.*, not necessarily the lexicographically first set): clearly $p_i \geq p'_i$; p_i can in fact can be strictly larger.

Proof. Incentive compability follows from the results in Archer and Tardos [2001]; we include a direct proof here for completeness. Suppose a bidder bids $b_i > v_i$: if i belongs to the winning set with $b_i = v_i$, the winning set with the increased bid must still contain i, since $\mathrm{v}(S, b)$ (Equation 1) does not decrease for sets S containing i, and does not change for sets not containing i. So i still wins an item and the price p_i it pays does not change. If i did not belong to the winning set with $b_i = v_i$, then the utility of i is negative for reporting $b_i > v_i$ and winning an item, by definition of its payment: at $b_i = v_i$, the winning set does not contain i, so $p_i > v_i$. Similarly, i has no incentive to bid $b_i < v_i$: if i did not belong to the winning set earlier, it cannot belong to a winning set by decreasing its value since $\mathrm{v}(S, b)$ is a non-increasing function of b_i; if it belonged to a winning set, its payment is independent of its bid so long as it wins an item, and it gets utility 0 if it does not win an item.

Next we show that these payments are feasible, *i.e.*, they can be routed to the auctioneer along the network. For this, we need to show that the payments p_i satisfy the first set of constraints in (1).

Let $w^* = \mathrm{v}(S^*)$ denote the value of the winning set, when $b_j = v_j$ for all bidders j. Abusing notation slightly, let $\mathrm{v}(S, b_i)$ denote the value of set S when bidder i bids b_i and all other bidders continue to bid v_j. For $i \in S^*$, let v'_i be the smallest value such that $v(S^*, v'_i)$ is still w^*. In fact, $v'_i = x^*_i$, where x^*_i is the smallest value of x_i amongst all optimal solution vectors x for the linear program (1) (with the true values v_i as input).

With $b_i = v'_i$, S^* is still the winning set: $\mathrm{v}(S)$ is unchanged for sets not containing i, and $\mathrm{v}(S, v'_i) \leq \mathrm{v}(S, v_i)$ for sets containing i, since $v'_i \leq v_i$ (the feasible set in (1) with $b_i = v'_i$ is a subset of the feasible set with $b_i = v_i$, so the optimal value cannot increase). Since S^* was the lexicographically first set with $b_i = v_i$, and the value of no set increases when b_i decreases to v'_i, S^* is still the lexicographically first set with the highest value when $b_i = v'_i$. Thus the bid at which i still belongs to the winning set is at least as small as v'_i, *i.e.*, $p_i \leq v'_i = x^*_i$. Since x (the optimal solution to 1 with entry x^*_i for bidder i) is feasible, we have

$$x^*_i + \sum_{j \neq i, j \in T} x_j \leq \mathrm{c}(T) \quad \forall\ T,$$

Using identical arguments for all other winners j, $p_j \leq x^*_j \leq x_j$, and substituting above, we get

$$\sum_{i \in T} p_i \leq \sum_{i \in T} x^*_i \leq \mathrm{c}(T) \quad \forall\ T,$$

i.e., the payments are feasible.

Individual rationality follows from the fact that $p_i \leq v'_i \leq v_i$ if the bidder wins an item, and is 0 otherwise. □

The greedy algorithm in Section 4.3 can be used to design a mechanism that approximates modified social welfare to a factor $1 - 1/e$, when the winner determination problem cannot be solved exactly.

Mechanism $\mathcal{M}'$: All bidders submit bids b_i.

- *Allocation*: Choose the set of winning bidders according to the greedy algorithm in Theorem 2, breaking all ties in favor of the bidder with the lower index.

- *Pricing*: Charge bidder $i \in S$ the smallest value $p_i \leq b_i$, such that i would still be chosen by the greedy algorithm when all bidders $j \neq i$ report b_j, and bidder i reports p_i.

Theorem 5. *The mechanism $\mathcal{M}$ approximates modified social welfare by a factor $1 - 1/e$, is incentive compatible, ex-post individually rational, and results in feasible payments.*

Proof. Incentive compatibility and individual rationality follow from the same arguments as before. The mechanism approximates welfare to a factor $1 - 1/e$ from Theorem 2.

To see that the payments are feasible, we use Lemma 1. Suppose bidder i was chosen at step j_i in the greedy algorithm ($1 \leq j_i \leq k$). Let v'_i be the smallest bid at which i is still chosen at the same step j_i by the greedy algorithm. (Again, v'_i is the smallest value of x_i in the optimal solution to (1) for the set S_{j_i}—the set of bidders chosen by the greedy algorithm in steps $1, \ldots, j_i$). By definition of the payments, we must have $p_i \leq v'_i$. From Lemma 1, there is a maximum flow from $S_{j'}$, $j' \geq j$, in which v'_i is the flow routed from node i. Since this is true for bidders chosen at every step in the greedy algorithm, the set of flows $(v'_1, \ldots, v'_k)$ satisfy budget constraints in S_k (of course, this set of flows need not be a maximum flow). That is,

$$\sum_{i \in T} p_i \leq \sum_{i \in T} v'_i \leq \mathrm{c}(T),$$

for all $T \subseteq S_k$, or the prices are feasible. □

6 Discussion

We have defined the concept of a general trust network as infrastructure for arbitrary payment routing. Analyzing such networks is a rich area for research and this paper only scratches the surface. We are not aware of previous work addressing the graph structure on credit extensions and the concomitant spending constraints. Here we considered the special case of a single auctioneer selling homogeneous goods to bidders with unit demand on such a network. We found that the winner determination problem in this domain (using a generalized notion of social welfare for budget-constrained bidders) is NP-hard but that the trust network structure can be exploited to approximate the solution within a factor of $1 - 1/e$. We then derived an incentive compatible and individually rational mechanism for this domain that respects the payment constraints of the trust network.

There are myriad additional mechanism design problems that can be studied in the context of trust networks. Multiple sellers may be considered, other preference functions, heterogeneous goods, etc. The designer may prefer to maximize revenue instead of welfare or may have any number of other design goals and constraints [Dash et al., 2003]. However, note that the impossibility result of Borgs et al. [2005] on mechanism design with individually budget constrained bidders shows that in many of these cases achieving incentive compatibility in dominant strategies is impossible.

A natural extension to consider, particularly for the case of repeated auctions, is interest rates on the credit links. Interest causes positive balances to become more positive over time and negative balances to become more negative. Not only does this complicate the payment routing problem (unless a single universal interest rate is used) but it means link capacities decrease over time, impacting the mechanism design problem.

Other mechanisms besides auctions are also impacted in interesting ways by the constraints inherent in a trust network. For example, betting games like poker become complicated when not every bet is honored by every player and if the degree to which a bet is honored depends on how other players—intermediate nodes in the trust network—fare in the game. Other mechanisms of interest to study in this setting include decision auctions, prediction markets [Wolfers and Zitzewitz, 2004] (with and without automated market makers [Hanson, 2003, 2007, Pennock, 2004]), and various incentive schemes for participation in the trust network itself.

The continued growth and development of online services and protocols for building decentralized financial trust networks will also pose questions and challenges in areas such as routing, distributed transactions, online identity verification, reputation systems, and spam prevention.

Both in terms of design and analysis of trust networks and in terms of mechanism design problems on trust networks, we hope that this paper opens a number of interesting research avenues.

Acknowledgments

We thank Varsha Dani and Lance Fortnow helping to clarify the mechanism design problem in this paper. We thank Bethany Soule for valuable suggestions and for (yootling us for) creating the figure.

References

Ravindra K. Ahuja, Thomas L. Magnanti, and James B. Orlin. *Network Flows: Theory, Algorithms, and Applications.* Prentice Hall, 1993.

Aaron Archer and Eva Tardos. Truthful mechanisms for one-parameter agents. In *IEEE Symposium on Foundations of Computer Science*, pages 482–491, 2001.

Christian Borgs, Jennifer Chayes, Nicole Immorlica, Mohammad Mahdian, and Amin Saberi. Multi-unit auctions with budget-constrained bidders. In *Proceedings of the 6th ACM Conference on Electronic Commerce (EC)*, pages 44–51, 2005.

Ranveer Chandra, Lili Qiu, Kamal Jain, and Mohammad Mahdian. Optimizing the placement of integration points in multi-hop wireless networks. In *Proceedings of the 12th IEEE International Conference on Network Protocols (ICNP)*, 2004.

Rajdeep K. Dash, Nicholas R. Jennings, and David C. Parkes. Computational mechanism design: A call to arms. *IEEE Intelligent Systems*, 18:40–47, 2003.

Uriel Feige. A threshold of $\ln n$ for approximating set cover. *J. ACM*, 45:634–652, 1998.

Ryan Fugger. The ripple project. `http://ripple.sourceforge.net`, 2004.

J. Håstad. Clique is hard to approximate within $n^{1-\epsilon}$. *Acta Mathematica*, 182:105–142, 1999.

Robin D. Hanson. Combinatorial information market design. *Information Systems Frontiers*, 5(1): 107–119, 2003.

Robin D. Hanson. Logarithmic market scoring rules for modular combinatorial information aggregation. *Journal of Prediction Markets*, 1(1):1=15, 2007.

David M. Pennock. A dynamic pari-mutuel market for hedging, wagering, and information aggregation. In *Proceedings of the Fifth ACM Conference on Electronic Commerce (EC'04)*, May 2004.

Daniel M. Reeves, Bethany M. Soule, and Tejaswi Kasturi. Yootopia! *SIGecom Exchanges*, 6:1–26, 2006.

Justin Wolfers and Eric Zitzewitz. Prediction markets. *Journal of Economic Perspective*, 18(2): 107–126, 2004.