From Single Static to Multiple Dynamic Combinatorial Auctions

Don Perugini [1,2], Dale Lambert [1], Leon Sterling [2], Adrian Pearce [2]
[1] Defence Science and Technology Organisation, Edinburgh, S.A., 5111, Australia
[2] NICTA Victorian Lab., Dept. Comp. Sci. and Soft. Eng. The University of Melbourne, Vic, Aus.
[1] {firstname.lastname}@dsto.defence.gov.au, [2] {leon, pearce}@cs.mu.oz.au

Abstract

We apply the Provisional Agreement Protocol (PAP) as a new approach to single static, single dynamic and multiple combinatorial auction problems, and empirically evaluate PAP. PAP benefits over one-shot auctions include: bidders not required to submit all bids and their dependencies; interaction with a changing environment during the auction can improve the solution; less communication when each bidder possesses many bids. PAP's backtracking may allow a better solution to be found than the first (greedy) solution, but can be detrimental with multiple auctions when bids (resources) are limited. With multiple auctions, dynamics and competition increases as resources becomes scarce. Therefore, PAP is likely to perform better when many resources are available, which is when auctions are useful anyway. PAP scales well, and applying PAP to a second domain shows its generality.

1. Introduction

Provisional Agreement Protocol (PAP) was introduced in [1, 2] to enable agents to plan and allocate tasks in decentralised, dynamic and open environments. In this paper, we apply PAP to combinatorial auctions [3, 4]. An auctioneer must allocate a set of non-identical goods $\boldsymbol{G}_i = \{g_1, ..., g_g\}$ to bidders, and bidders may submit bids b_j for a portion of the goods ($b_j \subseteq \boldsymbol{G}_i$) for price p_j. Typically there is free disposal, so not all goods need to be allocated, and each good can only be allocated once. The aim is for the auctioneer(s) to find an allocation of bids that maximises their individual (*local*) price. In the multiple auction case, we are not aiming to maximise the global price (sum of individual prices), as in mechanism design. Our aim is to provide a protocol to facilitate interaction that is present in many real world situations – auctioneers finding themselves a suitable plan and allocate tasks in a complex and dynamic environment, in the presence of other auctioneers that it must compete with for bids.

PAP is applied to the well-known case of the single auction with static bids, and the *less studied* single auction with dynamic bids and multiple simultaneous combinatorial auctions. Bids may be dynamic, for example, as bidders enter or leave the system during an auction, resulting in new bids surfacing and old bids retracted. Multiple auctions are inherently dynamic – during one auction, bids may be accepted by other auctioneers.

In auctions, a bidder may have many bids to communicate, or may not want to send *all* its bids (private information). Bids may have complex dependencies that may not be easily described with the OR-of-XOR language [4]. With multiple auctions, it is not clear how to deal with dependencies between bids in different auctions. Therefore, it may not be practicable for bidders to send all their bids and dependencies to auctions for processing, which is required in the one-shot (centralised) combinatorial auction, e.g. [3, 4]. With multiple auctions, auctioneers may be reluctant for a mediator to determine an allocation of bids for them. The PAP addresses these issues. Applying the PAP to combinatorial auctions (second domain) demonstrates the generality of the protocol.

2. PAP

The protocol is shown in figure 2. The five boxes along each vertical line represents the five steps of the protocol. Only one message or event may occur at any step (except bidding at step 2 may occur at anytime). The dashed arrows outside the vertical lines that start at a message or event indicates the next step of the protocol if that message or event occurs, and a diamond indicates the protocol exits. For more information on PAP, see [1, 2].

2.1. Protocol Policies

- *Commitment policy:* Bidders are not committed to their bid unless it's *provisionally granted*, and the grant is accepted. Auctioneers are committed to a bid after a *confirm grant*, and can *reject a provisionally granted bid*.
- *Persistence policy:* Agents store bids and auction announcements (goals) for future use, e.g. during backtracking. Bids and goals may not be available when agents decide to use them. Goals and bids are considered persistent until agents are informed otherwise.
- *Bidding policy:* Bidders may bid for a goal *anytime* they believe the goal is available – even after the deadline as the goal may be revisited during backtracking. *One* bid may be sent, the bid that the auctioneer prefers (see later) and *fully* or *partially* achieves the goal. Bidders *may* send an *updated bid* if the worst submitted bid is rejected or attempted to be provisionally granted but is (provisionally) withdrawn, to replace the bid. An *updated better bid* is sent when a bid becomes available that is better than the worst submitted bid for a goal.

Proceedings of the 2005 IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT'05)
0-7695-2416-8/05 $20.00 © 2005 IEEE

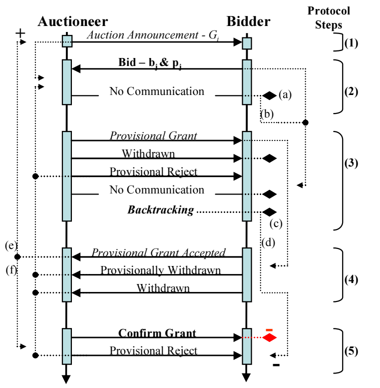

Figure 2. Protocol Flow Diagram.

2.2. Protocol Steps and Example

PAP begins at **step 1** with an **auction announcement** to all bidders. The message contains the goal $\boldsymbol{G_i}$, a deadline d_i, and bid evaluation function f. f informs bidders how the auctioneer will evaluate bids, so bidders can submit their best bid for $\boldsymbol{G_i}$. Example is $\boldsymbol{G_1} = \{1, 2, 3, 4, 5\}$, $d_1 = 5$ seconds, f = *number goods/price*.

In **step 2**, bidders may submit one **bid** b_j with price p_j, or submit nothing – **no communication** – in which case the protocol either exits for the bidder for that goal (path (a)), or an updated better bid can be submitted later (take path (b)). Example of a bid is $b_1 = \{1, 4\}$, $p_1 = \$10$.

In **step 3**, after d_i, the best bid, based on f, is given a **provisional grant**, proceed to step 4. If a submitted bid is unsuitable, a **provisional reject** is sent, allowing the bidder to send an updated bid at step 2. If a bid is sent for a goal that is no longer available, a **withdrawn** message is sent. Bidders without granted bids, receive **no communication** and may exit the protocol after length of time. If no bids are received by the d_i, the auctioneer assumes no solution exists for the goal, and thus requires **backtracking** or can accept the current solution (see step 5). If this occurs with the initial goal $\boldsymbol{G_1}$, then no solution exists, so take path (c) and exit. Otherwise, take path (d) to step 5.

In **step 4**, the bidder may **accept the provisional grant**, committing to the bid, and specifies a *confirm deadline* cd_i by which the auctioneer must confirm grant the bid, otherwise the bidder may de-commit. If the goal is completely achieved, take path (f) to step 5. Otherwise, take path (e) to step 1, and the new goal to announce is the portion of the goal not achieved by the bid, e.g. the new goal $\boldsymbol{G_2} = \boldsymbol{G_1} \setminus \boldsymbol{b_1} = \{2, 3, 5\}$. The bidder may not accept the provisional grant, sending a **provisionally withdrawn** or **withdrawn** message, informing the auctioneer that the bid is not available but may become available later or is not available now or later, respectively. Proceed to step 2 for the bidder to submit an updated bid.

In **step 5**, if arrived from step 3 (backtracking), due to free disposal (the goal need not be fully achieved), the auctioneer may (i) accept the current solution and **confirm grant** all the provisionally granted bids, securing the bids; (ii) if the solution is not suitable, backtrack by **provisionally rejecting** the bid for the previous goal (e.g. if backtracking $\boldsymbol{G_2}$, then reject b_1 for $\boldsymbol{G_1}$), and proceed to step 2 and 3, where the bidder may send an updated bid, and the auctioneer can select a new bid for the previous goal (e.g. $\boldsymbol{G_1}$). If arrived from step 4 (goal fully achieved), either (i) **confirm grant** all the provisionally granted bids; (ii) if unsatisfied with the solution, then backtrack.

In the current PAP implementation, rejected and withdrawn bids are deleted and not used again, which ensures convergence [1]. The confirm deadline is large enough to allow the auctioneer to complete its auction (planning).

2.3. Bids and Dependencies

Sending f enables bidders to determine the auctioneer's preferred bid. This is beneficial as only bidders may understand the potentially complex dependencies between their bids. Therefore, bidders do not need to send all their bids and bid dependencies, which can be complex, difficult to define, and private information.

2.4. Implementation

Over 1000 scenarios with up to 10 auctioneers and 100 bidders were executed. To reduce execution time, rather than fix d_i, planning continued once all bids were received. Given free disposal, when backtracking is not used, the first (greedy) solution found in which no more bids can be allocated is taken as the solution.

The data was generated by CATS [5]. CATS produces various types of data. Paths and scheduling data were arbitrarily selected. Data had between 10 to 1000 goods, and 10 to 1000 bids. A bidder may have more than one bid, and cannot be allocated more than one of each good.

The auctioneers used a simple heuristic for f, similar to Dang and Jennings greedy one-shot approach [6]:

$$f = \textit{number of goods in bid / price of bid}.$$

Due to limited space, graphs of results are not presented.

3. Single Auction, Static Bids

Scenarios comprise one auctioneer and bids distributed to various bidders. Results show that time scales linearly with the number of bids and goods. Running bidders on one processor caused extra computational overhead as more bidders were introduced. Ideally, the time to find a solution should decrease. Distributing the same number of bids to more bidders is likely to result in each bidder having less bids to process. Therefore the auctioneer can reduce d. As the number of bidders and goods increase, the maximum number of bids stored in memory increased. If B is the total number of bids held by all bidders, g is the number of goods and n is the number of bidders, then the PAP guarantees to use less memory if $g \cdot n < B$.

Proceedings of the 2005 IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT'05)
0-7695-2416-8/05 $20.00 © 2005 IEEE

Communication required with PAP, without backtracking and bids not rejected, is $g{\cdot}(2{\cdot}n{+}3)$. One-shot approaches require all B bids to be sent and either granted or rejected after the auction completes. PAP requires less communication if $g{\cdot}(2{\cdot}n{+}3) < 2{\cdot}B$, in particular, requires less communication if each bidder possesses a large number of bids. In our transportation application (reverse auction) [1], bidders bid for transportation along routes. There are many routes with various start times, hence many possible bids. In this case, PAP would be beneficial over one-shot approaches. Our experimental results are consistent with the theory. PAP communication improves over one-shot when: the number of bidders decrease for the same number of bids; as goods decrease; the number of bids increases for the same number of bidders.

Backtracking was used, using the heuristic: if <70%, 70% - 79%, 80% - 89% or 90% - 99% allocation is found initially, then backtrack until >70%, >80%, >90% or 100% allocation is found, respectively. Out of 41 scenarios, 37 (90.2%) produced a better solution by backtracking, which on average, was 9.4% better. Of the 4 (9.8%) which produced a worse solution, it was only 2% worse. From our results, backtracking was useful – it was likely to produce a better solution, and if not, the solution was not significantly worse. Rather than focus on the suitability of the heuristic employed, we are examining the case of a heuristic that utilises backtracking and how PAP supports this. In the case of no free disposal, backtracking can be used until all goods are allocated. There is a communication and time overhead with backtracking. In some cases, there were no solutions with a greater allocation. Therefore, in trying to find a better solution, the auctioneer was left with no allocation of goods. In this situation, it may be beneficial for the auctioneer to revert back to provisionally rejected bids, which are *currently* discarded. This issue is under investigation.

4. Single Auction, Dynamic Bids

We ran 16 scenarios with up to 50% of the bidders' bids *delayed* – available after depth 1, 2, 5 and 10 in the auction (table 1, values are percentage of the optimal with *all* bids). We ran centralised auctions without delayed bids, to simulate one-shot auction approaches that collect bids once and process them. Greedy centralised runs the same heuristic, and hence search, as the PAP and [6].

On average, PAP did better than centralised greedy. PAP took advantage of new (better) bids introduced *during* the auction. The later in planning (increasing depth) bids became available, the smaller the improvement. PAP with delayed bids available at depth 1 did better than centralised optimal. Thus, a greedy search that takes advantage of a changing environment can produce a better solution than the optimal solution that does not. It was not necessary to perform experiments with bids being retracted because if the centralised approach found a solution that contained a retracted bid, then the solution would be infeasible. PAP allows bids to be retracted. Therefore, PAP's ability for auctions to interact with the changing environment, taking advantage of new bids and acknowledging retracted bids, can improve the quality of solutions over one-shot (centralised) approaches.

Table 1. PAP vs centralised with delayed bids.

	Centralised		PAP – delayed bids at depth:			
	Optimal	*Greedy*	*1*	*2*	*5*	*10*
Mean	83.2	78.0	83.9	82.9	79.6	79.5
Std Dev	13.0	15.4	12.2	14.6	14.3	15.6

5. Multiple Auctions

Scenarios comprise up to 10 auctioneers (*same* goal) and 50 bidders. We varied the number of bidders and auctioneers (paths data, 100 goods and 250 bids) and compared the global price (sum of auctioneer's prices) versus number of bidders. More bidders is equivalent to more bids, and hence *resources* available to auctioneers, because even if a bidder has many bids, it can only be allocated at most one of each good.

As resources increase, auctioneers approach their globally, and thus locally, optimal price, but as resources become scarce, the global price decreases. This occurs for two reasons. First, due to the lack of bids, auctioneers may only get a small, or no, allocation of bids. Second, competition increases with decreasing resources as auctioneers must fight for the same bids. The auctioneer's locally optimal allocation of bids *conflicts* with others, reducing its chances of obtaining the optimal. Worse still, competition may result in a globally inefficient allocation of resources [1], as one auctioneer may obtain a bid that another requires. The PAP is likely to perform better locally and globally when resources are plentiful, reducing the chances of conflicts between required bids.

It is known that "auctions are used to allocate scarce resources". Resources in this context are item(s) that the auctioneer is auctioning, and hence, scarce resources imply there are many bids for the auctioneer's item(s). Therefore, PAP is likely to perform well when applied to auctions as there are likely to be many bids.

As resources (*bids per auctioneer*) become scarce, the number of (provisionally) withdrawn messages per total allocation increases, indicating that the environment is more dynamic (bids are being retracted).

Multiple auctioneers backtracked (paths data and 100 goods) to find an allocation that has one or more goods than the initial solution. As resources become scarce, a greater number of auctioneers did not find an allocation. There are four reasons. First, there may not be a solution with a better allocation – less resources implies less possible solutions. Due to partial observability (do not have all bids), the auctioneer does not know if a better solution exists. Second, there may not be enough bids for all the auctioneers. Third, due to increased dynamism from scarce resources, options that could potentially provide a better solution may no longer be available during backtracking. Fourth, auctioneers hold bids by provisionally granting them, preventing other auctioneers from using them, which the bid is later released. Additionally, auctioneers that try to provisionally grant these (provisionally withdraw) bids will discard that option, which they could use it soon after. This also prevents the bidder from having its bid allocated. This issue is under investigation.

We ran scenarios where only half the auctioneers backtracked (paths data, 100 goods, 10 auctioneers). Auctioneers that did not backtrack when resources were scarce were better off than those that did backtrack, since they all found a solution. Once they obtained (provisionally granted) scarce resources (bids), they held on to them. Auctioneers that backtracked released the resources and were unable to find a better, or any, replacement.

Thus, backtracking with scarce resources can be detrimental, and a greedy approach is more suited. This seems counter intuitive as one would expect backtracking to always provide a better solution. Due to partial observability, it is difficult to determine whether better solutions are possible (applies to single auctions with static bids). Even if a better solution is known, it may no longer be available during backtracking. The solution that was given up as a result of backtracking may also be unavailable when an auctioneer tries to regain it.

6. Related work

One-shot approaches, such as [3, 4, 7], require all bids to be sent to a centralised agent for processing. PAP is decentralised, and we have presented benefits of this. [6] uses a greedy approach, like PAP, but is also one-shot and does not allow backtracking. Combinatorial approaches, such as iBundle, use ascending auctions [8-10]. These are suited to domains where bids for auctioned items are dynamically priced. In our domain, bidders have fixed (true) valuations (prices) for their bids – as is the case in many reverse auctions – and therefore a problem of allocation rather than *price determination*. Additionally, they do not consider the multiple auction case, where each auctioneer allocates multiple goods. Double auctions [11, 12], which involve multiple auctioneers, require both auctioneers and bidders to submit goals and bids to a mediator that matches them (e.g. stock market). This may not be practicable as bidders may have many bids, they may not know what to bid until a goal is presented, or with reverse auctions, bids (services offered) are tailored to suit the goal at hand. Thus, we look at single sided auctions. [13, 14] investigates the problem of which auctions bidders should bid in, and at what price, in order to obtain a good at the best price. Again, they assume dynamically priced bids. In PAP, bidders are allowed to bid in all auctions (fixed price) until the bid is allocated.

7. Conclusion and Acknowledgements

PAP was used as a new approach to combinatorial auctions. Benefits over one-shot (centralised) auctions include: bidders not required to submit all bids and their dependencies; reduced communication if bidders possess many bids; an improved solution in a dynamic environment as PAP allows auctions to interact with the changing environment *during* the auction. PAP was able to facilitate multiple dynamic combinatorial auctions. We found that as resources (bids per auctioneer) became scarce, dynamism and competition increased. PAP is likely to perform better when resources are plentiful, which are when auctions are useful anyway. PAP backtracking allowed a better solution to be found than the first (greedy) solution in single auctions, but can be detrimental with multiple auctions with limited resources. PAP scales well, and applied to a second domain shows its generality.

We would like to thank Don Gossink, Steve Wark, Andrew Zschorn, and Kevin Leyton-Brown for their assistance.

8. References

[1] Perugini, D., et al. *Provisional Agreement Protocol for Global Transportation Scheduling*. in *Int Wksh ATT (AAMAS04), to appear in Whitestein Agent Technology series*. 2004. New York, U.S.: IEEE.

[2] Perugini, D., et al. *A Distributed Agent Approach to Global Transportation Scheduling*. in *IAT 2003*. 2003. Halifax, Canada.

[3] Sandholm, T., *Algorithm for optimal winner determination in combinatorial auctions.* Artificial Intelligence, 2002. **135**: p. 1-54.

[4] Nisan, N., *Bidding and Allocation in Combinatory Auctions.* EC, 2000: p. 1-12.

[5] Leyton-Brown, K., M. Pearson, and Y. Shoham. *Towards a Universal Test Suite for Combinatorial Auction Algorithms*. in *In Proceedings of ACM Conference on Electronic Commerce*. 2000.

[6] Dang, V.D. and N.R. Jennings. *Polynomial algorithms for clearing multi-unit single item and multi-unit combinatorial reverse auctions*. in *Proc. 15th European Conf. on AI (ECAI-2002)*. 2002. Lyon, France.

[7] Walsh, W.E., M.P. Wellman, and F. Ygge. *Combinatorial Auctions for Supply Chain Formation*. in *Second ACM Conf on Elec Com*. 2000.

[8] Wurman, P.R. and M.P. Wellman. *AkBA: A Progressive, Anonymous-Price Combinatorial Auction*. in *Second ACM Conference on Electronic Commerce*. 2000. Minneapolis: ACM.

[9] Parkes, D.C. and L.H. Ungar. *Iterative Combinatorial Auctions: Theory and Practice*. in *In Proc of the 17th Nat Conf on AI*. 2000.

[10] Ausubel, L.M. and P. Cramton, *Auctioning Many Divisible Goods.* Journal of the European Economic Association, 2004. **2**: p. 480-493.

[11] Wellman, M.P., *A Market-Oriented Programming Environment and its Application to Distributed Multicommodity Flow Problems.* Journal of Artificial Intelligence Research, 1993. **1**: p. 1-23.

[12] He, M., H. Leung, and N.R. Jennings, *A Fuzzy-Logic Based Bidding Strategy for Autonomous Agents in Continuous Double Auctions.* IEEE Trans on Knowledge & Data Eng., 2003. **15**(6): p. 1345-1363.

[13] Anthony, P. and N.R. Jennings, *Developing a Bidding Agent for Multiple Heterogeneous Auctions.* ACM Transactions on Internet Technology, 2003. **3**(3): p. 185-217.

[14] Shehory, O., *Optimal Bidding in Multiple Concurrent Auctions.* Int. J. of Cooperative Information Sys., 2002. **11**(3 & 4): p. 315-327.