Noname manuscript No.
(will be inserted by the editor)

Auctions for Multi-Robot Task Allocation in Communication Limited Environments

Michael Otte · Michael Kuhlman · Donald Sofge

Received: date / Accepted: date

Abstract We consider the problem of multi-robot task allocation using auctions, and study how lossy communication between the auctioneer and bidders affects solution quality. We demonstrate both analytically and experimentally that even though many auction algorithms have similar performance when communication is perfect, different auctions degrade in different ways as communication quality decreases from perfect to nonexistent. Thus, if a multi-robot system is expected to encounter lossy communication, then the auction algorithm that it uses for task allocation must be chosen carefully.

We compare six auction algorithms including: standard implementations of the Sequential Auction, Parallel Auction, Combinatorial Auction; a generalization of the Prim Allocation Auction called G-Prim; and two multi-round variants of a Repeated Parallel Auction. Variants of these auctions are also considered in which award information from previous rounds is rebroadcast by the auctioneer during later rounds. We consider a variety of valuation functions used by the bidders, including: the total and maximum distance traveled (for distance based cost functions), the expected profit or cost to a robot (assuming robots' task values are drawn from a random distribution). Different auctioneer objectives are also evaluated, and include: maximizing profit (max sum), minimizing cost (min sum), and minimizing the maximum distance traveled by any particular robot (min max). In addition to the cost value functions that are used, we are also interested in fleet performance statistics such as the expected robot utilization rate, and the expected number of items won by each robot.

Experiments are performed both in simulation and on real AscTec Pelican quad-rotor aircraft. In simulation, each algorithm is considered across communication qualities ranging from perfect to nonexistent. For the case of the distance-based cost functions, the performance of the auctions is compared using two different communication models: (1) a Bernoulli model and (2) the Gilbert-Elliot model. The particular auction that performs the best changes based on the the reliability of the communication between the bidders and the auctioneer. We find that G-Prim and its repeated variant perform relatively well when communication is poor, and that re-sending winner data in later rounds is an easy way improve the performance of multi-round auctions, in general.

Keywords Multi-Robot · Multi-Agent · Auction · Any-Com · Task Allocation · Prim Allocation · G-Prim · Sequential Auction · Parallel Auction · Combinatorial Auction

This work was performed at the Naval Research Laboratory and was funded by the Office of Naval Research under grant numbers N0001416WX01271 and N0001416WX01272. The views, positions and conclusions expressed herein reflect only the authors' opinions and expressly do not reflect those of the Office of Naval Research, nor those of the Naval Research Laboratory.

Michael Otte
University of Maryland
College Park, MD 20742
E-mail: otte@umd.edu

Michael Kuhlman and Donald Sofge
U.S. Naval Research Laboratory
Washington D.C. 20375
Tel.: +202-767-0806
Fax: +202-767-0806
E-mail: michael.kuhlman@nrl.navy.mil
E-mail: donald.sofge@nrl.navy.mil

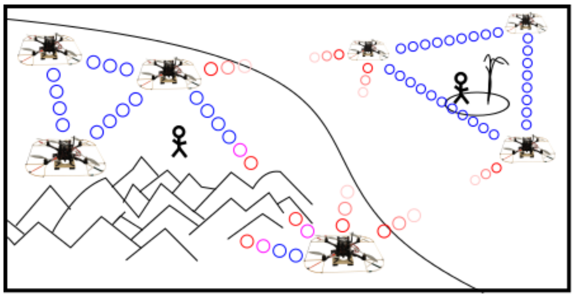

Fig. 1: Auctions in communication limited environments. Distance and terrain (depicted here), as well as other factors (not shown), can cause messages to be dropped in environments where autonomous multi-robot teams are deployed.

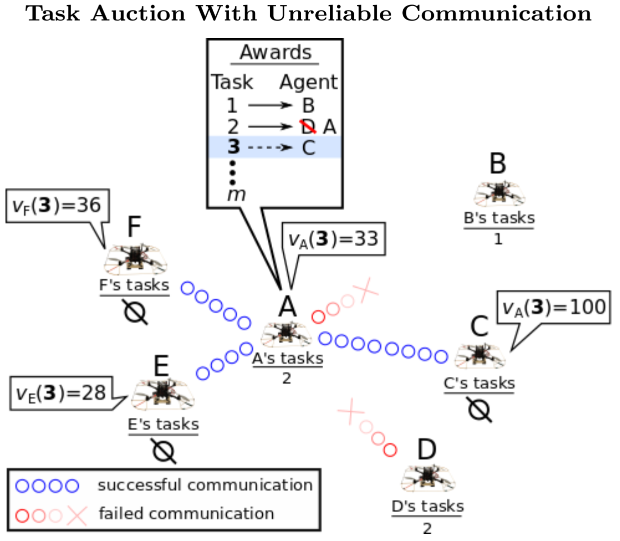

Fig. 2: We study how imperfect communication affects autonomous auctions. In this example, the third of m tasks is advertised and sold to the highest bidder. The message from the auctioneer (A) advertising the sale of task 3 did not reach robots D and B, and they do not bid as a result. Robots A, C, E, and F bid. C wins with the highest bid of $V_C(3) = 100$. A previous message from robot D, acknowledging D's acceptance of task 2, was dropped; thus, the auctioneer must also perform task 2 to ensure that task 2 is completed. Task 1 was successfully awarded to robot B.

1 Introduction

Multi-robot teams are often deployed in environments where communication is unreliable, yet they may also face problems that require dividing a set of tasks between the different members of the team. Communication may be unreliable due to environmental factors such as weather and obstacles, the distance between robots, and interference (see Figure 1). Communication may also be intentionally withheld in the interest of maintaining stealth. Auctions and other market-based approaches are widely used for task allocation because they provide a distributed mechanism to partition tasks in a way that allows different robots to have different costs or values over the same set of tasks (items).

In an auction, each robot determines its own bid for each item based on an internal valuation or cost function. The auctioneer then sells items to the highest or lowest bidder, depending on if the auctioneer is seeking to maximize value or minimize cost, respectively.

A valuation function is used to account for factors such as task preference and the ability to complete a task, while a cost function often accounts for the resources that are required to complete a task; e.g., the distance that must be traveled or the fuel that must be used. Auction algorithms differ based on how many rounds are required to sell a particular number of items, how many items each robot can bid for each round, and how many items are sold each round. Most auction algorithms have both a maximum valuation and a minimum cost formulation; the difference between the two is often as simple as swapping minimization operations for maximization operations, or *vice versa*.

Auctions necessarily involve communication between the auctioneer and the bidders (see Figure 2). Yet, while a variety of auctions have been proposed and/or used for multi-robot task allocation, the effects of lossy communication on auction performance are not usually reported. Indeed, a direct comparison of how communication degradation affects the relative performance of different auctions has not previously been performed.

A main contribution of this research is an improved understanding of how a variety of popular auctions perform in communication degraded scenarios. We perform both analytical and experimental comparisons of how six different auctions perform at different points along the communication quality spectrum in a variety of different scenarios. Three of these six are standard implementations of well known auctions including: the Sequential Auction, Parallel Auction, and Combinatorial Auction. Another is a generalization of Prim Allocation called *G-Prim* that extends the basic dynamics of Prim Allocation to work with any cost or valuation function[1]. The final two auctions that we compare are multi-round ("repeated") variations of the Parallel Auction; one is a standard implementation, and the other shares similarities with G-Prim.

For auctions that require more than a single round of bidding to sell the entire set of items, we also investigate a simple but effective modification in which the auctioneer rebroadcasts award information from earlier rounds along with each round's (new) award information. The performance measures that we consider include: the total and maximum distance traveled (for distance based cost functions), the expected profit or cost of the group solution (assuming robots' task values are drawn from a random distribution), the expected robot utilization rate, and the expected number of items won by each robot.

A small additional contribution of this work is a mixed integer programming formulation for a Combinatorial Auction that minimizes the maximum distance traveled by any robot.

Experiments are performed both in simulation and on real AscTec Pelican quad-rotor aircraft. In simula-

[1] Prim Allocation was originally designed to use a specific cost function that is based on a bounded approximation to the multi-agent version of the traveling salesperson problem Lagoudakis et al (2004).

tion, each algorithm is considered using a variety of valuation functions, and across communication qualities ranging from perfect to nonexistent using two different communication models: (1) a Bernoulli model and (2) the Gilbert-Elliot model.

In general, we find that methods rank differently depending on both the quality of the communication channel and the cost function that is minimized. We also find that the relative performance of different auctions change as communication degrades. Both G-Prim, and the repeated parallel variant that is based on G-Prim perform relatively well when communication is poor. Re-sending winner data is an easy way to improve the performance of multi-round auctions, in general.

This paper is organized as follows: Related work appears in Section 2 and preliminaries such as algorithms and notation appear in Section 3. The auction algorithms themselves appear in Section 4. The analytical solutions for expected agent utilization statistics are derived in Section 5, and section 6 contains analysis for straightforward extensions to the basic methods. Section 7 describes experiments that we run including: a comparison of the analytical solutions to the average results observed over Monte Carlo simulations, a comparison of how different communication models affects auction performance while using distance based cost functions, and hardware experiments using auctions in a test bed using three agents: two AscTec Pelican Quadrotors and a stationary auctioneer. Discussion and conclusions appear in Sections 8 and 9, respectively. Finally, an appendix contains the derivations of a few numerical calculations and additional plots from experiments.

2 Related Work

A variety of auction techniques have previously been applied to problems of distributed task allocation in autonomous robotics. Surveys of previous work can be found by Dias et al (2006) and Koenig et al (2010).

Defining one robot to be the auctioneer is a common approach, and the one that we explore in this paper. That said, a number of alternative organizations appear in the literature. For example, all robots may broadcast their bids over a fully connected communication graph so that winners can be calculated in parallel on all robots (Vail and Veloso, 2003). Robots may initiate new auctions to re-sell tasks they wish to drop (Guerrero and Oliver, 2003). Robots may also share the role of auctioneer. For example, in the CNET protocol (Smith, 1980) idle contractor nodes bid for jobs from manager "nodes" in a sealed bid first price auction, and "nodes" switch between contractor and manager as necessary. A similar idea is used for multi-robot teams by Caloud et al (1990); such teams are deployed in a simulated environment across multiple networked computers by Botelho and Alami (1999). The CNET idea is extended to "TraderBots" in Dias et al (2004).

A variety of different auction mechanisms have been used for task allocation in multi-robot teams. In a *Sequential Auction*, the auctioneer sells a sequence of items, one item at a time, in an order selected by the auctioneer. Simmons et al (2000) have robots bid to visit frontier nodes after map updates. Wei et al (2015) have robots bid to search unvisited points for targets, and also to transport targets to a goal. A sequence of one-round auctions is used by Rekleitis et al (2008) for dividing a Boustrophedon multi-robot coverage sweep, by Vail and Veloso (2003) for assigning roles in robot soccer, and by Pippin and Christensen (2011) for assigning target detection tasks. Vail and Veloso (2003) use a sequence of one-round auctions that are distributed in the sense that robots share information, and each auction is calculated separately on all robots.

MURDOCH uses a similar sequence of distributed one-round auctions to allocate a variety of tasks including: box pushing, century duty, and object tracking (Mataric and Sukhatme, 2001; Gerkey and Mataric, 2002). Nanjanath and Gini (2010) use several Sequential Auctions, that are run in parallel, and auctions repeat during task execution. Guerrero and Oliver (2003) have robots bid for participation in a variety of multi-robot tasks, and an auction is repeated as necessary whenever a robot abandons its task.

A multi-round ϵ rising price auction[2] is analyzed in Bertsekas and Castañon (1991), and found to converge to a solution within ϵn of optimal, where n is the number of bidders. We note that rising price auctions are not commonly used for task allocation in scenarios where we control all bidding agents, because the outcome of this auction is identical to that of a sealed-bid second price auction[3]. Rising price auctions require multiple rounds to sell each item, in general, while sealed-bid second price auctions only require one round to sell each item. In Berhault et al (2003) sealed bid single round Combinatorial Auctions are solved approximately using a primal-dual algorithm from Zurel and Nisan (2001), where tasks are visiting points of interest in an unknown environment.

In *Parallel Auctions* all items are bid on simultaneously, such that the auction lasts one round.

[2] Rising price auctions are also known as English Auctions.

[3] Second price sealed-bid auctions are also know as a Vickrey auctions. In this auction the highest bidder wins, but pays the second-highest bid price; essentially outbidding the second highest bidder by $\epsilon \to 0$.

Combinatorial Auctions are the only type of auction that are guaranteed to find an optimal solution in the case that item valuations are not independent[4]. Each agent submits a bid for every possible subset of items, and the auctioneer awards sets of items to agents such that the best overall allocation of items to agents is achieved. Combinatorial Auctions are NP-hard to solve in practice, and Sandholm (2002) show them to be exponentially complex. Combinatorial Auctions are used for multi-robot task assignment by Hunsberger and Grosz (2000). They are studied, in general, in Parkes and Ungar (2000); Andersson et al (2000); Sandholm (2002), and a survey of Combinatorial Auction techniques and applications can be found in De Vries and Vohra (2003). The mixed integer programming based solution that we use for the minimum summed path length objective is presented in Andersson et al (2000). An *open iterative Combinatorial Auction* is shown to approach optimality as the minimum bid increment approaches 0, assuming honest bidding, in Parkes and Ungar (2000).

The use of optimal Combinatorial Auctions is impractical whenever an auction involves more than a handful of items, thus other work has focused on quickly finding solutions with bounded suboptimality. A number of different distributed bidding mechanisms are compared by Lagoudakis et al (2005) and a variant of Sequential Auctions are shown to provide solutions that are provably close to optimal. We note that Lagoudakis et al (2005) performs a thorough analysis of using auctions for multi-robot task allocation in terms of computation, distance, and time. A refinement of one of these methods, called "*Prim Allocation*" is presented by Lagoudakis et al (2004). Assuming that the objective is to minimize the summed lengths of all robots' paths, then G-Prim achieves solutions with twice the optimal summed length in the worst case. PRIM is extended by Sariel and Balch (2006) by the addition of a number of heuristics.

Dias and Stentz (2000) explore an auction variant that uses a two phase approach; in the first phase a sequence of $\lceil m/n \rceil$ Parallel Auctions are run (in the first Parallel Auction the most lucrative n of m tasks are awarded such that each of the n robots gets one task, and then the k-th Parallel Auction follows the same procedure for the $m - kn$ remaining tasks); later, in the second phase, robots are free to greedily trade tasks for "money." The first phase of the auction variant explored by Dias and Stentz (2000) is interesting in that it greedily assigns items in a number of rounds such that each robot wins one item in each round (except possibly the last round). In the current paper we refer to this variant of a repeated auction as *Repeated G-Prim*; though we note that the work by Dias and Stentz (2000) which first used a variant of this idea predated that by Lagoudakis et al (2004) which described Prim Allocation. The work by Dias and Stentz (2000) is extended to region exploration in Zlot et al (2002).

Non-auction market-based task-exchanges have also been used as a mechanism of performing greedy gradient ascent to improve the initial solutions found using auctions. These are used in simulated search-and-track/destroy missions in Chandler and Pachter (2001); sub-teams of agents may elect to auction off the targets and/or their agent workers to increase global utility.

Experimental comparisons between different auctions, market approaches, and other multi-robot coordination mechanisms can be found in Cavalcante et al (2013); Schneider et al (2014); Mataric and Sukhatme (2001); Berhault et al (2003). Approximation methods for Combinatorial Auctions are compared to sequential single item auctions by Cavalcante et al (2013), greedy sequential and Parallel Auctions are evaluated by Schneider et al (2014), assuming robots visit points in the order in which they are won. Single round distributed auctions are compared to heuristic methods by Mataric and Sukhatme (2001). Berhault et al (2003) observe that Combinatorial Auctions outperform single item auctions in simulation.

Cooperation in situations where communication is either unreliable or high cost has been investigated by: Trawny et al (2009); Otte (2018); Stone and Veloso (1998); Hoeing et al (2007); Castelpietra et al (2001); Parker (1998); Gerkey and Matarić (2001); Zlot et al (2002); Dias and Stentz (2000); Rekleitis et al (2008); Beard and McLain (2003); Otte et al (2017a). Collective localization despite communication constraints have been investigated by Trawny et al (2009). Otte (2018) shows how a collective neural network can be trained across a swarm of robots despite unreliable communication. In the robot soccer domain, *a priori* "locker-room-agreements" are augmented by opportunistic sharing of real-time data in Stone and Veloso (1998), and critical messages are re-sent until they are acknowledged in Castelpietra et al (2001). Robots that discover new tasks auction them to neighbors within communication range in Hoeing et al (2007). The ALLIANCE framework assumes that communication may not be available (Parker, 1998), and MURDOCH assumes that communication is not necessarily perfect, but "reasonable" in that messages are successful most of the time and provide the minimum bandwidth required by the

[4] Item valuations are not independent when there are dependencies between item valuations such that there are or extra costs or values associated with owning different subsets of items.

algorithm (Gerkey and Matarić, 2001). Zlot et al (2002) assume imperfect communication and note that a market approach based on work by Dias and Stentz (2000) is capable of functioning without communication, i.e., all regions will eventually be explored. Communication delays in task assignment via an asynchronous distributed auction algorithm, where m tasks are sold to $n < m$ and each agent may only win a single task, are studied by Moore and Passino (2004). Line-of-sight communication for market-based allocation of tasks in cooperative Boustrophedon—i.e., lawn-mower—sweep is used by Rekleitis et al (2008). Beard and McLain (2003) investigate how limits on communication range affect collective search in an environment with obstacles. In previous work, Otte et al (2017a), we investigated how the presence or absence of communication affects a search game where two teams compete to locate a lost target.

Alighanbari and How (2005) use a centralized (non-auction) algorithm that is run in parallel on each member of a team, robots exchange information and use an unreliable communication channel to reach solution consensus. This idea is applied to distributed auctions by Choi et al (2009), who report that the resulting method outperforms the Prim allocation algorithm. Greedy post-hoc trading is part of the method by Choi et al (2009) but not used in the Prim allocation algorithm it was compared to. Therefore, in our opinion, an alternative interpretation of this result is that the performance difference between the two methods can be traced to the use of post-hoc trading.

Non-auction-based centralized methods that require consensus are studied in situations with intermittent and asynchronous communications and moving targets by Dionne and Rabbath (2007). Agents only communicate when local information is sufficient to change the global decision. Beard and Stepanyan (2003) show that a communication spanning tree is necessary for asymptotic convergence between shared state in a multi-robot team.

The re-planning and other adaptability benefits of market-based approaches have been touted as being robust to agent loss by Botelho and Alami (1999); Mataric and Sukhatme (2001). Communication degradation and partial/total failure of robots are studied by Dias et al (2004) using an implementation of "TraderBots". The dynamic re-assignment problem (in response to new obstacles or loss of agents) is explored by Castanon and Wu (2003); the method is arguably market-based in that prices are assigned to both tasks and agents, however the robots strive to reach consensus on an optimal reassignment using a distributed shortest augmenting path (SAP) algorithm (Bertsekas and Castañon, 1993) and auction and trade mechanisms are not used directly. Schneider et al (2015) is an extension of Schneider et al (2014) to dynamic environments, and considers practical evaluation metrics, including: total mission time computed as a combination of execution time and deliberation time, as well as distance actually traveled by the robots.

Closely related work to our own includes work by Choi et al (2009) and Lagoudakis et al (2005). Unreliable communication is assumed by Choi et al (2009), in which Prim Allocation is compared to a distributed technique that uses post-hoc greedy trading for solution improvement; in contrast, we consider different algorithms and valuation functions, and analyze agent participation. Bounds on G-Prim's solution length, when using each of six distance-based heuristics, are reported by Lagoudakis et al (2005), assuming perfect communication.

The particular auction algorithms that we study include:

– The Parallel Auction
– The Sequential Auction (Mataric and Sukhatme, 2001)
– A generalization of Prim Allocation (Lagoudakis et al, 2004) that we call G-Prim.
– The Repeated Parallel Auction used in the first phase of (Dias and Stentz, 2000).
– A modification of the Repeated Parallel Auction that borrows ideas from G-Prim (and which was previously used as the first phase of a more general market approach by Dias and Stentz (2000)), which we call Repeated G-Prim.
– The Combinatorial Auction (Parkes and Ungar, 2000; Andersson et al, 2000; Sandholm, 2002).

The original Prim Allocation Algorithm assumes tasks are locations, and defines an item's cost as the minimum edge length required to add that item to a spanning tree rooted at the bidding agent. In contrast, G-Prim allows robots to use any valuation or cost function but retains Prim's bidding mechanics: m items are sold over m rounds, and in each round each agent i bids on the single unsold item that i values the most. These mechanics enable G-Prim to have nice properties when communication is poor. The auctions we study are detailed in Section 3.

The main difference between our work and all previous work is that we evaluate how performance changes as communication shifts across the quality spectrum from perfect to nonexistent. We evaluate six different auction techniques and four different valuation functions (two for each cost function we consider). A secondary contribution of our work is an integer-programming solution to a Combinatorial Auction in the case that the objective is to minimize the maximum path length that any robot needs to travel.

A preliminary version of this work was presented at the International Symposium on Multi-Robot and Multi-Agent Systems (Otte et al, 2017b). The preliminary version focused only on the Sequential, Parallel, and G-Prim Auctions, while the current paper additionally considers two versions of a Repeated Parallel Auction (one based on the Parallel Auction, and another based on G-Prim), as well as the Combinatorial Auction. The current extended journal version also includes additional analysis and experiments for the case that the auctioneer re-sends the current winner list during subsequent rounds of multi-round auctions; this is a straightforward modification that can improve performance. Finally, in addition to the Bernoulli communication model considered in (Otte et al, 2017b), we also perform experiments assuming the Gilbert Elliott Communication model, and report results for additional metrics including the summed path length of the robots and a min-max criteria, as well as approximations to these quantities that may be easier to calculate for large systems in practice.

3 Preliminaries

In this section we define our nomenclature; discuss various item valuation functions, item cost functions, and auction objectives; and define the communication models used in this paper.

3.1 Nomenclature

The number of agents and auction items are denoted n and m, respectively. The number of rounds in an auction depends on the particular type of auction being performed. Subscripts i, j, and r denote association with a particular item, agent, or round, respectively. a denotes the auctioneer's ID; $a = 1$ by convention. Set Θ contains all items and θ_j is the j-th item.

A particular item is denoted θ_j and the set of items is $\Theta = \{1, \dots, m\}$. It is often convenient to denote the k-th subset of Θ as $S_k \subset \Theta$ (this is particularly useful for Combinatorial Auctions). The set of items that has already been sold is Θ_{sold}, while the set of unsold items is $\Theta_{\text{unsold}} = \Theta \setminus \Theta_{\text{sold}}$.

The winning robot of a particular item is denoted $\hat{i}$, and the particular item that is won is denoted $\hat{\theta}_j$.

A bid is denoted b, a bid from robot i for item θ_j is denoted b_{i,θ_j}. In Combinatorial Auctions robots may bid on sets of items, in which case $b_{i,k}$ represents a bid from robot i on the subset S_k. The set of winning bids over the items (or sets of items, in the case of a Combinatorial Auction) in a particular auction is B. During round r, robot i calculates a bid for item θ_j using the valuation function $V_{i,r}(\theta_j)$. Set B_i is agent i's self-maintained set of bids, $\tilde{B}_i$ is the set of i's bids of which the auctioneer is aware.

The winning bid set that results in the globally optimal partitioning of items among robots is denoted B^*. In the best case, the team wins the most economical distribution of tasks, i.e., to find B^*. In general, there may be benefits to owning multiple items in particular subsets of items; for example, if tasks represent visiting locations in the environment, then visiting two nearby locations will often cost less than the sum of visiting each location separately. The only auction that is guaranteed to find B^* is a Combinatorial Auction—at great computational expense that is often impractical. In most cases the auctioneer estimates the most economical distribution of tasks using a greedy procedure. Let $\hat{B}$ denote the set of winning bids that leads to this greedy solution.

The set $\tilde{W}_i$ contains the items the auctioneer awards to agent i, while W_i is the set of items robot i knows it has won. Dropped messages may cause $B_i \neq \tilde{B}_i$ and $W_i \neq \tilde{W}_i$. An acknowledgment is denoted C. Depending on algorithm, different subscripts are used to indicate which item is being acknowledged (e.g., C_j) and which robot the acknowledgment came from (e.g., $C_{i,j}$)., The cardinality of set '$\cdot$' is denoted $|\cdot|$. The probability of an event '$\cdot$' is $\mathbb{P}(\cdot)$ and the expectation of a value '$\cdot$' is $\mathbb{E}(\cdot)$; probabilities and expectations are defined over the event space of all communication histories and all realizations of valuations (or costs).

3.2 Valuation, Cost Functions, Objectives

We now discuss the various item cost and valuation functions we consider, as well as the objectives used by the auctioneer to determine winners.

3.2.1 Scenario 1, Maximization or Minimization of Random Valuations

In Scenario 1 we assume all valuations are absolutely continuous independent and identically distributed random variables (i.i.d.) drawn at random from the *same* probability density function $f_X(x)$.

The proportion of (a very large) agent population that values item j at value $V_{i,r}(j) = X$ such that $a \leq X \leq b$ is given by:

$$\mathbb{P}(a \leq X \leq b) = \int_a^b f_X(x)dx. \tag{1}$$

The assumption that Equation 1 is an accurate model for valuations is valid if both: (i) different agents have different preferences and skills that are independently determined, but (ii) the preferences and skills of the agent population as a whole can be described by well behaved distributions.

Our analysis of Scenario 1 assumes that the valuations of all items are associated with the *same* underlying $f_X(x)$, we do not require that the form of $f_X(x)$ be known[5]. Having all agents draw valuations from $f_X(x)$ guarantees that each possible *ordering* of agent-item valuations is equally likely. In this case, each agent is equally likely to value θ_j the most; thus, each agent is equally likely to win item θ_j when communication is perfect.

Finally, for Scenario 1 we assume that valuations are constant over time,

$$V_{i,\hat{r}}(\theta_j) = V_{i,r}(\theta_j) \text{ for all } \hat{r}, r$$

which is a valid assumption if valuations are mutually independent, e.g., there are no synergies between items.

Scenario 1 is used because it both (1) provides a scenario motivated by different bidders having unique costs or valuations for each particular item but also where (2) the overall preferences of a population of bidders is well understood (enabling straightforward analysis in many cases).

3.2.2 Scenario 2, Minimization of Multi-TSP Cost

In Scenario 2 we assume items are locations that must be visited. Cost $V_{i,r}(\theta_j)$ is defined by the extra distance agent i must travel to visit θ_j in addition to the locations $W_{i,r-1}$ that i has already accepted in rounds $1 \ldots r-1$. In other words, cost is calculate by subtracting traveling salesperson (TSP) length over i's current solution from that of the TSP solution that also includes visiting the new item θ_j. Let $\ell_{\mathrm{TSP}}(\cdot)$ denote the length of the TSP solution over item set '$\cdot$'. Formally,

$$V_{i,r}(\theta_j) = \ell_{\mathrm{TSP}}(W_{i,r-1} \cup \{\theta_j\}) - \ell_{\mathrm{TSP}}(W_{i,r-1}). \qquad (2)$$

In this paper we explore using both the *true* multi-TSP length as well as Christofides TSP approximation algorithm[6] (Christofides, 1976) that was used in the original Prim Allocation Algorithm. Christofides TSP approximation assumes a metric space; thus, its use in task allocation problems is limited to scenarios where the cost function takes the form of a distance metric (or is approximated by a metric well enough for practical purposes). Dynamic constraints may break these assumptions if the environment is very small with respect to the maneuver distance (e.g., the minimum turning radius) of the robots. However, in large environments (where task locations are likely to be far apart with respect to maneuver distance) such issues can usually be ignored[7]. Using the true multi-TSP length is only viable for small numbers of items; the cost metric from Prim Allocation provides a practical alternative for problems involving many items.

Given TSP based bid costs, it is possible to have the auctioneer use a variety of different objective functions to determine the winning bid. The first is the summed distance traveled by all robots. Let ξ_i denote the path taken by robot i to visit the items that it has won. Let $\|\xi_i\|$ denote the distance traveled by a robot following ξ_i. The summed distance is calculated

$$\text{summed path length} = \sum_{i \in R} \|\xi_i\|$$

and is the cost function that is minimized in the multiple traveling salesperson problem. When using this cost function we are using the *min-sum* objective.

The second distance-based cost function we evaluate is the maximum distance traveled by any robot, which is calculated:

$$\text{maximum path length} = \max_{i \in R} \|\xi_i\|.$$

If all robots travel at the same velocity (and we assume instantaneous change of speed) then this is equivalent to the make-span. Minimizing the maximum path length essentially balances, as much as possible, the workload that is performed by all robots. When using this cost function we are using the *min-max* objective.

3.3 Communication Models

There are at least two ways to handle poor communication: (1) the auctioneer can wait until it hears from

[5] Using the same $f_X(x)$ for all agent-item pairs is only a requirement for the G-Prim Auction and the Repeated G-Prim Auction. For the Parallel Auction, Sequential Auction, and Repeated Parallel Auction this assumption can be relaxed such that each item j is associated with its own $f_{X,j}(x)$ that is shared by all agents.

[6] Prim Allocation (which uses a multi-TSP variant of Christofides TSP approximation algorithm) estimates multi-TSP length by building an incremental spanning forest from agents' locations. A post-processing step is used to obtain a multi-TSP approximation from the spanning forest. This heuristic yields a solution no worse than 1.5 times the optimal length and is calculated in polynomial time. While the true TSP-based solution we consider provides a closer estimate of the optimal multi-TSP path, its runtime is super-polynomial with respect to item number.

[7] If the environment is cluttered with static obstacles then Christofides TSP approximation requires a preprocessing step to find the minimum length path between different task locations. Static obstacles are beyond the scope of this work.

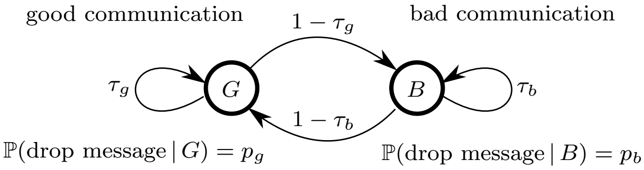

Fig. 3: The Gilbert-Elliot Communication Model assumes a good state G and a bad state B. The probability of dropping a message in G and B is p_g and p_b, respectively, where $p_g < p_b$. The probability of remaining in G and B is τ_g and τ_b, respectively, while the probability of leaving G and B is $1-\tau_g$ and $1-\tau_g$, respectively.

each agent every round, or (2) the auctioneer can assume that rounds last for a specific duration and that agents that are unable to successfully communicate do not submit bids. The former shares similarities with the TCP message passing protocol, and the latter with UDP. In practice it is impossible to know if a particular robot has moved beyond communication range (or become disabled, etc.). Therefore, we study case (2).

In Section 7 we perform experiments in simulation and with real robots. In simulation we control how messages are dropped, and we perform experiments using both (i) the Bernoulli model and (ii) the Gilbert-Elliot model.

The Bernoulli model assumes that there is a probability p that each communication message is successful, where $0 \leq p \leq 1$ and communication attempts are i.i.d. The Bernoulli model is a very simple model that ignore many aspects of the real world such as ranged effects, directional effects, and bandwidth saturation. However, it is a useful model for cases in which robots are relatively close to each other vs. their communication ranges, relatively little bandwidth is used, and messages are dropped due to external events present in the environment. The simplicity of the Bernoulli model makes it useful as an analytical tool, enabling the derivation for a number of auction performance measures.

The Gilbert-Elliot Communication Model assumes a good state G and a bad state B. The probability of dropping a message in G and B is p_g and p_b, respectively, where $p_g < p_b$. The probability of remaining in G and B is τ_g and τ_b, respectively, while the probability of leaving G and B is $1-\tau_g$ and $1-\tau_g$, respectively (see Figure 3). Like the Bernoulli model, the Gilbert-Elliot model does not explicitly account for ranged effects or directional effects. However, the Gilbert-Elliot model can model nonstationary effects that occurs from the robots collectively being in good or bad communication states. Indeed, Gilbert-Elliot has been shown to more accurately approximate the communication breakdown that occurs in practice over a channel that experiences bandwidth saturation (Elliott, 1963).

Table 1: High-Level Comparison of Different Auction Types

Auction type	rounds	number of items advertised in round r	number of items sold per round
Parallel	1	m	m
Sequential	m	1	1
G-Prim	m	$m-r$	1
R. Parallel	$\lceil m/n \rceil$	n	n
R. G-Prim	$\lceil m/n \rceil$	$m-rn$	n
Combinatorial	1	m	m

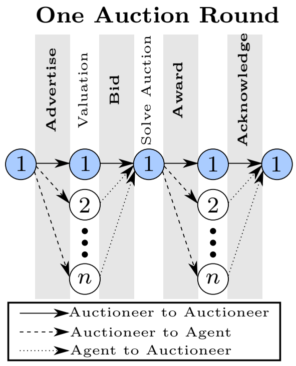

Fig. 4: Phases of an auction round, and the messages sent between them. **Round phases appear boldfaced**, computations during phases appear as plain text. Messages from auctioneer $(a = 1)$ to auctioneer are always received. Assuming a Bernoulli model, messages from auctioneer to non-auctioneer agents $(2 \ldots n)$ and from non-auctioneer agents to auctioneer are sent with probability p and dropped with probability $q = 1 - p$.

4 Auction Algorithms

In this section we describe algorithms for the various auctions that we compare, including: Sequential, Parallel, G-Prim, Repeated Parallel, Repeated G-Prim, and Combinatorial Auctions. Pseudocode describing the auctions appears in Algorithms 1-6, respectively. Note that computations on the auctioneer are printed blue, while calculations on all agents (nonauctioneers and auctioneers) are printed black. The pseudocode in Algorithms 1-6 assumes a profit *maximization* objective. The alternative cost *minimization* objective can be achieved by replacing arg max by arg min and setting unreceived bids to ∞ instead of $-\infty$.

Table 1 compares auctions based on the number of rounds they require and the number of items bid for and sold each round.

Figure 4 shows the phases, messages, and communication graph of a single auction round (identical for all auctions). Each message is either broadcast by the auctioneer or sent once by each non-auctioneer. A broadcast is equivalent to sending the same message once to each agent; we assume the success or failure of a broadcast message reaching different agents is independent. *We assume messages sent from the auctioneer to itself are never dropped.*

In a Sequential Auction there are m different auction rounds, one each for the m items, and each message requires $\mathcal{O}(1)$ space. In a Parallel Auction there is a single round (in which each of the m items are simultane-

Table 2: Auction Message Space and Count Requirements

	message size	messages/robot	total messages
sequential	$\Theta(1)$	$\Theta(m)$	$\Theta(nm)$
parallel	$\Theta(m)$	$\Theta(1)$	$\Theta(n)$
G-Prim	$\Omega(1)$, $\mathcal{O}(m)$	$\Theta(m)$	$\Theta(nm)$
R. parallel naive	$\Theta(n)$	$\Omega(1)$, $\mathcal{O}(m/n)$	$\Omega(n)$, $\mathcal{O}(m)$
R. parallel G-Prim	$\Omega(1)$, $\mathcal{O}(m)$	$\Omega(1)$, $\mathcal{O}(m/n)$	$\Omega(n)$, $\mathcal{O}(m)$
combinatorial	$\Omega(1)$, $\mathcal{O}(m!)$	$\Theta(1)$	$\Theta(n)$

Table 3: Runtime Required for Auction (with perfect communication)

	time per round	rounds	total time
sequential	$\Theta(n)$	$\Theta(m)$	$\Theta(nm)$
parallel	$\Theta(nm)$	$\Theta(1)$	$\Theta(nm)$
G-Prim	$\Theta(n+m^2)$	$\Theta(m)$	$\Theta(m(n+m^2))$
R. parallel naive	$\mathcal{O}(n^2 \log n)$	$\Theta(m/n)$	$\mathcal{O}(nm \log n)$
R. parallel G-Prim	$\mathcal{O}(n^2 \log n)$	$\Theta(m/n)$	$\mathcal{O}(nm \log n)$
combinatorial	$\Omega(2^m)$, $\mathcal{O}(3^m)$	$\Theta(1)$	$\Omega(2^m)$, $\mathcal{O}(3^m)$

ously bid for), and messages between agents (and to any manager) require space $\mathcal{O}(m)$. In the G-Prim algorithm the manager advertises all unsold items each round and agent bids for their top choice each round. This requires managers to send messages of size $\mathcal{O}(m)$, while agents send messages of size $\mathcal{O}(1)$. In the repeated variant of the Parallel Auction, each agent wins a single item each round and $\mathcal{O}(m/n)$ rounds are required. Each round the auctioneer and the agents send messages of size $\mathcal{O}(n)$. The Repeated G-Prim Auction is similar, except that the auctioneer advertises all unsold items each round using messages of size $\mathcal{O}(m)$ (then agent bids for their top n choices of the remaining items each round). In a Combinatorial Auction there is a single round (wherein each of the $\mathcal{O}(m!)$ possible combinations of items are bid for by each agent), and messages between agents (and to any manager) require space $\mathcal{O}(m!)$ in the worst case. Tables 2 and 3 summarize the message sizes associated with each type of auction, and the runtimes required to determine a winner assuming perfect communication.

Algorithm 1 Parallel Auction

On auctioneer a, 1: a.Broadcast(Θ)
On each agent i, 2: **if** i.Receive(Θ) **then**
On each agent i, 3: **for all** $\theta_j \in \Theta$ **do**
On each agent i, 4: $B_i[j] \leftarrow V_{i,1}(\theta_j)$
On each agent i, 5: i.Send($B_i[1 \dots m]$)
On auctioneer a, 6: **while** time left **do**
On auctioneer a, 7: **if** a.Receive($B_i[1 \dots m]$) **then**
On auctioneer a, 8: $\tilde{B}_{i,1\dots m} \leftarrow B_i[1 \dots m]$
On auctioneer a, 9: **for** $j \in \{1 \dots m\}$ **do**
On auctioneer a, 10: $i \leftarrow \arg\max_i \tilde{B}_{1\dots n,m}$
On auctioneer a, 11: $\tilde{W}_i \leftarrow \tilde{W}_i \cup \{\theta_j\}$
On auctioneer a, 12: a.Broadcast($\tilde{W}_{1\dots n}$)
On each agent i, 13: **if** i.Receive($\tilde{W}_{1\dots n}$) **then**
On each agent i, 14: $W_i \leftarrow \tilde{W}_i$
On each agent i, 15: i.Send(C_i)
On auctioneer a, 16: wait appropriate amount of time
On auctioneer a, 17: **for all** $i \in [1 \dots n]$ s.t. **not** a.Receive(C_i) **do**
On auctioneer a, 18: $W_a \leftarrow W_a \cup W_i$

Algorithm 2 Sequential Auction

1: **for** $j \in \{1 \dots m\}$ **do**
On auctioneer a, 2: a.Broadcast(θ_j)
On each agent i, 3: **if** i.Receive(θ_j) **then**
On each agent i, 4: $B_i[j] \leftarrow V_{i,j}(\theta_j)$
On each agent i, 5: i.Send($B_i[j]$)
On auctioneer a, 6: **while** time left **do**
On auctioneer a, 7: **if** a.Receive($B_i[j]$) **then**
On auctioneer a, 8: $\tilde{B}_{i,j} \leftarrow B_i[j]$
On auctioneer a, 9: $h \leftarrow \arg\max_i \tilde{B}_{i,j}$
On auctioneer a, 10: $\tilde{W}_h \leftarrow \tilde{W}_h \cup \{\theta_j\}$
On auctioneer a, 11: a.Broadcast(h, j)
On each agent i, 12: **if** i.Receive(h, j) **and** $i = h$ **then**
On each agent i, 13: $W_i \leftarrow W_i \cup \{\theta_j\}$
On each agent i, 14: i.Send($C_{h,j}$)
On auctioneer a, 15: wait appropriate amount of time
On auctioneer a, 16: **if not** a.Receive($C_{h,j}$) **then**
On auctioneer a, 17: $W_a \leftarrow W_a \cup \{\theta_j\}$

4.1 Parallel Auction

A Parallel Auction (Algorithm 1) has a single round in which all items are auctioned simultaneously. The auctioneer broadcasts the item list Θ to all robots including itself (line 1). Each robot i calculates a bid for each item θ_j using its valuation function $V_{i,1}(\theta_j)$, and sends its resulting bid list B_i to the auctioneer (lines 3-5). The auctioneer waits for a predetermined length of time to receive bids (lines 6-8), awards each θ_j to the agent that sent the best bid for θ_j (lines 9-11), and then broadcasts the award list W (line 12). Winning robots that receive W return an acknowledgment C_i (lines 13-15). Finally, the auctioneer takes responsibility for tasks with unacknowledged sales (lines 17-18). Note that while the auctioneer broadcasts $\tilde{W}_{1\dots n}$, each robot i only needs to know the subset $\tilde{W}_i \subset \tilde{W}_{1\dots n}$ (the items that i won).

4.2 Sequential Auction

A Sequential Auction (Algorithm 2) sells 1 item per round for m rounds. During *each round* of a Sequential Auction the auctioneer chooses (e.g., randomly) an unsold item θ_j and sells it using the same phases as a Parallel Auction—advertisement, bidding, winner determination, winner announcement, and acknowledgment.

4.3 G-Prim Auction

The G-Prim Auction (Algorithm 3) is similar to a Sequential Auction in that one item is sold per round

Algorithm 3 G-Prim Auction

for $r \in \{1 \ldots m\}$ **do**
On auctioneer a, $\tilde{B}_{i \ldots n, 1 \ldots m} \leftarrow -\infty$
On auctioneer a, $\Theta_r \leftarrow \Theta \setminus \Theta_{\text{sold}}$
On auctioneer a, a.Broadcast(Θ_r)
On each agent i, **if** i.Receive(Θ_r) **then**
On each agent i, **for all** $\theta_j \in \Theta_r$ **do**
On each agent i, $B_i[j] \leftarrow V_{i,r}(\theta_j)$
On each agent i, $\hat{b}_{i,j} \leftarrow \arg\max_j B_i[j]$
On each agent i, i.Send($\hat{b}_{i,j}$)
On auctioneer a, **while** time left **do**
On auctioneer a, **if** a.Receive($\hat{b}_{i,j}$) **then**
On auctioneer a, $\tilde{B}_{i,j} \leftarrow \hat{b}_{i,j}$
On auctioneer a, $(h, j) \leftarrow \arg\max_{(i,j)} \tilde{B}_{i,j}$
On auctioneer a, $\tilde{W}_h \leftarrow \tilde{W}_h \cup \{\theta_j\}$
On auctioneer a, a.Broadcast(h, j)
On each agent i, **if** i.Receive(h, j) **and** $i = h$ **then**
On each agent i, $W_i \leftarrow W_i \cup \{\theta_j\}$
On each agent i, i.Send($C_{i,j}$)
On auctioneer a, wait appropriate amount of time
On auctioneer a, **if not** a.Receive($C_{i,j}$) **then**
On auctioneer a, $W_a \leftarrow W_a \cup \{\theta_j\}$
On auctioneer a, $\Theta_{\text{sold}} \leftarrow \Theta_{\text{sold}} \cup \{\theta_j\}$

for m rounds; it is also similar to a Parallel Auction in that multiple items are up for sale each round. During round r, each agent bids for the unsold item that it values the most, and the auctioneer awards the single item that received the best bid, where $\Theta_{\text{unsold}} = \Theta \setminus \Theta_{\text{sold}}$. During round r, the unsold item valued most by i is denoted $\hat{b}_{i,r} \in \Theta_{\text{unsold}}$. The number of unsold items is denoted $|\Theta_{\text{unsold}}|$, and $|\Theta_{\text{unsold}}| = m - r + 1$ at the start of round r. Each round has the same phase order as the other two Auctions.

4.4 Repeated Parallel Auction

A naive implementation of a Repeated Parallel Auction (Algorithm 4) is another way to combine elements of the parallel and Sequential Auctions. During the r-th round, the auctioneer advertises the first n items that have not yet been sold or all unsold items if less than n items remain unsold (lines 4-7). Each agent that receives the advertise message performs a valuation for each items up for bid, and then returns this information to the auctioneer (lines 8-13). The auctioneer greedily allocates items to the bidders as follows: the first unsold item is sold to the agent with the best bid for that item, the c-th unsold item is sold to the agent in $\tilde{R}$ that has the highest bid for that item, where $\tilde{R}$ contains the agents that have not yet won an item in round r (lines 17-22). As with the other auctions, agents must acknowledge that they have received the award message, and if they do not then the auctioneer assumes responsibility for the tasks (lines 17-22). The items acquired by each agent is simply the union of the items they required in each round (line 30).

Algorithm 4 Repeated Parallel Auction

1: $r \leftarrow 0$
2: **while** $\Theta_{\text{unsold}} \neq \emptyset$ **do**
3: $r \leftarrow r + 1$
On auctioneer a, 4: $\check{j} \leftarrow m - |\Theta_{\text{unsold}}|$
On auctioneer a, 5: $\hat{j} \leftarrow \max(m, \check{j} + n - 1)$
On auctioneer a, 6: $S_r \leftarrow \{\theta_{\check{j}}, \ldots, \theta_{\hat{j}}\}$
On auctioneer a, 7: a.Broadcast(S_r)
On each agent i, 8: **if** i.Receive(S_r) **then**
On each agent i, 9: $c \leftarrow 0$
On each agent i, 10: **for all** $\theta_j \in S_r$ **do**
On each agent i, 11: $c \leftarrow c + 1$
On each agent i, 12: $B_i[c] \leftarrow V_{i,r}(\theta_j)$
On each agent i, 13: i.Send($B_i[1 \ldots c]$)
On auctioneer a, 14: **while** time left **do**
On auctioneer a, 15: **if** a.Receive(B_i) **then**
On auctioneer a, 16: $\tilde{B}_{i,r} \leftarrow B_i$
On auctioneer a, 17: $\tilde{R} \leftarrow \{1, \ldots, n\}$
On auctioneer a, 18: **for** $\theta_j \in S_r$ **do**
On auctioneer a, 19: $h \leftarrow \arg\max_{i \in \tilde{R}} \tilde{B}_{i,r}$
On auctioneer a, 20: $\tilde{W}_{h,r} \leftarrow \tilde{W}_{h,r} \cup \{\theta_j\}$
On auctioneer a, 21: $\tilde{R} \leftarrow \tilde{R} \setminus \{h\}$
22: $\Theta_{\text{unsold}} \leftarrow \Theta_{\text{unsold}} \setminus \{\theta_j\}$
On auctioneer a, 23: a.Broadcast($\tilde{W}_{1 \ldots n, r}$)
On each agent i, 24: **if** i.Receive($\tilde{W}_{1 \ldots n, r}$) **then**
On each agent i, 25: $W_{i,r} \leftarrow W_{i,r} \cup \tilde{W}_{i,r}$
On each agent i, 26: i.Send(C_i)
On auctioneer a, 27: wait appropriate amount of time
On auctioneer a, 28: **for all** $i \in [1 \ldots n]$ s.t. **not** a.Receive(C_i) **do**
On auctioneer a, 29: $W_a \leftarrow W_a \cup W_{i,r}$
On each agent i, 30: $W_i \leftarrow \bigcup_{c \in [1,r]} W_{i,r}$

The number of rounds that the auction takes depends on the number of items that are sold each round (line 2). The auctioneer itself will receive one item per round and so the auction will be over in at most m, if all agents are able to bid every round then the auction will be over in $\lceil m/n \rceil$ rounds.

4.5 Repeated G-Prim Auction

A more sophisticated batch method exists that is motivated by the G-Prim method (Algorithm 5). This batch algorithm is similar to the naive version in that up to n items are sold each round (each item to a different agent). However, it is similar to the G-Prim algorithm in that the auctioneer advertises all unsold items each round (lines 5-6), and each agent bids for the items that they value the most. During each round each agent bids for the n unsold items that they value the most — or all remaining items if less n items remain unsold (lines 7-11). Agents must acknowledge that they have received the award message (lines 23-25), and if they do not

Algorithm 5 Repeated G-Prim Auction

1: $r \leftarrow 0$
2: **while** $\Theta_{\text{unsold}} \neq \emptyset$ **do**
3: $\quad r \leftarrow r + 1$
On auctioneer a, 4: $\quad \tilde{B}_{i\ldots n,1\ldots m} \leftarrow -\infty$
On auctioneer a, 5: $\quad \Theta_r \leftarrow \Theta \setminus \Theta_{\text{sold}}$
On auctioneer a, 6: $\quad a.\text{Broadcast}(\Theta_r)$
On each agent i, 7: $\quad$ **if** $i.\text{Receive}(\Theta_r)$ **then**
On each agent i, 8: $\quad\quad B_{i,r} \leftarrow \emptyset$
On each agent i, 9: $\quad\quad$ **while** $|B_{i,r}| < \min(n, |\Theta_{\text{unsold}}|)$ **do**
On each agent i, 10: $\quad\quad\quad B_{i,r} \leftarrow B_{i,r} \cup \{\arg\max_{\theta_j} V_{i,r}(\theta_j)\}$
On each agent i, 11: $\quad\quad i.\text{Send}(B_{i,r})$
On auctioneer a, 12: $\quad$ **while** time left **do**
On auctioneer a, 13: $\quad\quad$ **if** $a.\text{Receive}(B_{i,r})$ **then**
On auctioneer a, 14: $\quad\quad\quad \tilde{B}_{i,r} \leftarrow B_{i,r}$
On auctioneer a, 15: $\quad \tilde{R} \leftarrow \{1, \ldots, n\}$
On auctioneer a, 16: $\quad$ **while** $\tilde{R} \neq \emptyset$ **do**
On auctioneer a, 17: $\quad\quad h \leftarrow \arg\max_{i \in \tilde{R}} \max(\tilde{B}_{i,r} \cap \Theta_{\text{unsold}})$
On auctioneer a, 18: $\quad\quad \theta_j \leftarrow \max(\tilde{B}_{h,r} \cap \Theta_{\text{unsold}})$
On auctioneer a, 19: $\quad\quad \tilde{W}_{h,r} \leftarrow \tilde{W}_{h,r} \cup \{\theta_j\}$
On auctioneer a, 20: $\quad\quad \tilde{R} \leftarrow \tilde{R} \setminus \{h\}$
21: $\quad\quad \Theta_{\text{unsold}} \leftarrow \Theta_{\text{unsold}} \setminus \{\theta_j\}$
On auctioneer a, 22: $\quad a.\text{Broadcast}(\tilde{W}_{1\ldots n,r})$
On each agent i, 23: $\quad$ **if** $i.\text{Receive}(\tilde{W}_{1\ldots n,r})$ **then**
On each agent i, 24: $\quad\quad W_{i,r} \leftarrow W_{i,r} \cup \tilde{W}_{i,r}$
On each agent i, 25: $\quad\quad i.\text{Send}(C_i)$
On auctioneer a, 26: $\quad$ wait appropriate amount of time
On auctioneer a, 27: $\quad$ **for all** $i \in [1 \ldots n]$ s.t. **not** $a.\text{Receive}(C_i)$ **do**
On auctioneer a, 28: $\quad\quad W_a \leftarrow W_a \cup W_{i,r}$
On each agent i, 29: $W_i \leftarrow \bigcup_{c \in [1,r]} W_{i,r}$

Algorithm 6 Combinatorial Auction

On auctioneer a, $a.\text{Broadcast}(\Theta)$
On each agent i, **if** $i.\text{Receive}(\Theta)$ **then**
On each agent i, $\quad$ **for all** $\theta_j \in \Theta$ **do**
On each agent i, $\quad\quad B_i[j] \leftarrow V_{i,1}(\theta_j)$
On each agent i, $\quad i.\text{Send}(B_i[1 \ldots m])$
On auctioneer a, **while** time left **do**
On auctioneer a, $\quad$ **if** $a.\text{Receive}(B_i[1 \ldots m])$ **then**
On auctioneer a, $\quad\quad \tilde{B}_{i,1\ldots m} \leftarrow B_i[1 \ldots m]$
On auctioneer a, **for** $j \in \{1 \ldots m\}$ **do**
On auctioneer a, $\quad$ **if** $\theta_j \in b_{i,k} \in B^*$ **then**
On auctioneer a, $\quad\quad \tilde{W}_i \leftarrow \tilde{W}_i \cup \{\theta_j\}$
On auctioneer a, $a.\text{Broadcast}(\tilde{W}_{1\ldots n})$
On each agent i, **if** $i.\text{Receive}(\tilde{W}_{1\ldots n})$ **then**
On each agent i, $\quad W_i \leftarrow \tilde{W}_i$
On each agent i, $\quad i.\text{Send}(C_i)$
On auctioneer a, wait appropriate amount of time
On auctioneer a, **for all** $i \in [1 \ldots n]$ s.t. **not** $a.\text{Receive}(C_i)$ **do**
On auctioneer a, $\quad W_a \leftarrow W_a \cup W_i$

then the auctioneer assumes responsibility for the tasks (lines 27-28).

4.6 Combinatorial Auction

In a Combinatorial Auction each agent bids on all possible sets of items and the auctioneer calculates the optimal way to divide the sets among teams (Algorithm 6). Assuming individual items are labeled $1, 2, \ldots,$ then it is convenient to denote each possible set of items using a unique binary integer k. Formally, we define set S_k such that $\theta_j \in S_k$ if and only if the θ_j-th bit of k is 1. For example, the binary representation of 9 is 1001 and so we define $S_9 \equiv \{1, 4\}$. In a full Combinatorial Auction over m items each agent bids on $2^m - 1$ different sets of items. We use mixed integer programming formulations to solve the Combinatorial Auctions. Let $c_{i,k}$ represent the path length of agent i to visit all items in set S_k and for all i, k let $x_{i,k} \in \{0, 1\}$ represent whether or not agent i wins set S_k (i.e., $x_{i,k} = 1$ if it does). A mixed integer program that solves for the min-sum objective is defined as follows: minimize $\sum_{i=1}^{n} \sum_{k=1}^{2^m-1} x_{i,k} c_{i,k}$ Subject to the m constraints (i.e., for all j): $\sum_{k|j \in S_k} x_{i,k} = 1$ to ensure that each item is visited by a single agent.

Note that because the cost function is added across all agents, and only one agent can win any particular set, we can ignore all but the lowest bid for any particular set and solve the smaller mixed integer program: minimize $\sum_{k=1}^{2^m-1} \hat{x}_k \hat{c}_k$ Subject to the m constraints (i.e., for all j): $\sum_{k|j \in S_k} \hat{x}_k = 1$, where $\hat{c}_k = \min_i(c_{i,k})$ for all k, and $\hat{x}_k \in \{0, 1\}$ represents whether or not subset S_k is sold to agent $\arg\min_i(c_{i,k})$.

If we instead wish the auction to use the min-max objective then we want to minimize $\left[\max_i \sum_{k=1}^{2^m-1} x_{i,k} c_{i,k}\right]$ which can be solved with the following mixed integer program that uses the auxiliary variable v: minimize $[v]$ subject to n constraints (one for each agent i): $v \geq \sum_{k=1}^{2^m-1} x_{i,k} c_{i,k}$ as well as the m constraints (i.e., for all j): $\sum_{k|j \in S_k} x_{i,k} = 1$.

5 Analysis

In this section we analyze agent utilization in the auctions described in Section 4 assuming a Bernoulli model of communication. We begin by considering bid valuations that are random variables in Sections 5.1-5.6, followed by how these can be used to bid results for a TSP bid cost metric in Section 5.7. For both scenarios we consider the expected number of items that are won by any particular agent, treating the cases of the auctioneer and non-auctioneer agents separately. We also derive calculations for the probability that any particular nonauctioneer agent ends up doing at least one task. The reason that we are interested in these statistics is that they show how agent utilization changes as a function of communication quality.

5.1 Analysis of Parallel Auction when Bids are Realizations of Random Variables

Communicating a bid to the auctioneer requires receiving the advertisement list and sending a bid message. The probability exactly $k-1$ non-auctioneers communicate a bid to the auctioneer is $p^{2(k-1)}(qp+q)^{n-k}\binom{n-1}{k-1}$. Given our assumptions, the probability the auctioneer wins θ_j given k bids are communicated to the auctioneer (by both auctioneer and non-auctioneers) is $1/k$.

The expression $p^{2(k-1)}(qp+q)^{n-k}\binom{n-1}{k-1}$ comes from the facts that: For each of $k-1$ non-auctioneers to submit a bid, advertise messages must be passed successfully from the auctioneer to $k-1$ non-auctioneers (which happens with probability p^{k-1}) and bid messages successful returned by them (which also happens with probability p^{k-1}). The events that the remaining $n-k$ non-auctioneers do not submit bids require that *either* (1) an advertise message is successful but the bid message is dropped or (2) the advertise message is dropped (the compound event that either one or the other of these things happen to exactly $n-k$ agents has probability $(qp+q)^{n-k}$). Finally, there are $\binom{n-1}{k-1}$ different ways to assign the auctioneers such that $k-1$ non-auctioneers successfully bid and $n-k$ non-auctioneers do not.

Thus, the expected number of items won outright by the auctioneer is:

$$\mathbb{E}\Big(|\tilde{W}_a|\Big) = \sum_{k=1}^{n} \frac{m}{k} p^{2(k-1)}(qp+q)^{n-k}\binom{n-1}{k-1}.$$

The probability a particular $i \neq a$ plus $k-2$ other non-auctioneers communicate a bid message to the auctioneer is $p^{2(k-1)}(qp+q)^{n-k}\binom{n-2}{k-2}$, and so the expected number of items awarded to $i \neq a$ by the auctioneer is:

$$\mathbb{E}\Big(|\tilde{W}_{i\neq a}|\Big) = \sum_{k=2}^{n} \frac{m}{k} p^{2(k-1)}(qp+q)^{n-k}\binom{n-2}{k-2}$$

and the expected number awarded to all non-auctioneers is:

$$\mathbb{E}\Big(\sum_{i\neq a} |\tilde{W}_i|\Big) = (n-1)\mathbb{E}\Big(|\tilde{W}_{i\neq a}|\Big).$$

Taking responsibility for (or *adopting*) an item requires both winning that item and also receiving the award message. The expected number of items adopted by a single non-auctioneer and the set of all non-auctioneers are, respectively:

$$\mathbb{E}\Big(|W_{i\neq a}|\Big) = p\mathbb{E}\Big(|\tilde{W}_{i\neq a}|\Big)$$

$$\mathbb{E}\Big(\sum_{i\neq a} |W_i|\Big) = (n-1)\mathbb{E}\Big(|W_{i\neq a}|\Big).$$

The expected number of items adopted by the auctioneer includes the items it wins plus all unacknowledged sales:

$$\mathbb{E}(W_{i\neq a}) = \mathbb{E}(\tilde{W}_a) + (pq+q)\mathbb{E}\Big(\sum_{i\neq a} |\tilde{W}_i|\Big).$$

The expected number of items adopted twice, i.e., by the auctioneer as well as a non-auctioneer, is:

$$\mathbb{E}(|W_a \cap (\cup_{i\neq a} W_i)|) = pq\mathbb{E}\Big(\sum_{i\neq a} |\tilde{W}_i|\Big).$$

The probability a non-auctioneer does *not* win item θ_j in a Parallel Auction *assuming its bid is received* is:

$$\mathbb{P}(\theta_j \notin \tilde{W}_{i\neq a} \,|\, \zeta) = \textstyle\sum_{k=2}^{n} \frac{k-1}{k} p^{2(k-2)}(qp+q)^{n-k}\binom{n-2}{k-2}$$

where ζ is the event "i's bid is received." Thus, the probability a non-auctioneer wins zero items, assuming ζ, is:

$$\mathbb{P}(\tilde{W}_{i\neq a} = \emptyset \,|\, \zeta) = \mathbb{P}(\theta_j \in \tilde{W}_{i\neq a})^m$$

The probability a non-auctioneer adopts at least one task, i.e., $i \neq a$ wins at least one item and gets the award message, is:

$$\mathbb{P}(W_{i\neq a} \neq \emptyset) = 1 - \Big(q + pq + p^2 q + p^3 \mathbb{P}(\tilde{W}_{i\neq a} = \emptyset \,|\, \zeta)\Big)$$

5.2 Analysis of Sequential Auction when Bids are Realizations of Random Variables

We now switch our focus to the Sequential Auction. The expected number of items won outright by the auctioneer is:

$$\mathbb{E}\Big(|\tilde{W}_a|\Big) = m\sum_{k=1}^{n} \frac{1}{k} p^{2(k-1)}(qp+q)^{n-k}\binom{n-1}{k-1}$$

and the number of items awarded to each non-auctioneer is:

$$\mathbb{E}\Big(|\tilde{W}_{i\neq a}|\Big) = m\sum_{k=2}^{n} \frac{1}{k} p^{2(k-1)}(qp+q)^{n-k}\binom{n-2}{k-2}.$$

These are equivalent to the expectations derived for the Parallel Auction. Thus, all remaining expected values of the Sequential Auction are identical to their Parallel Auction counterparts; we do not repeat them to save space.

The probability that an agent does at least one task in a Parallel Auction is different than in a Sequential Auction, despite the fact that expected number of items visited by a particular agent is identical for the two auctions. In a Sequential Auction the probability a non-auctioneer $i \neq a$ wins θ_j conditioned on the event ζ_j (its bid is received) is:

$\mathbb{P}(\theta_j \in \tilde{W}_{i\neq a} \,|\, \zeta_j) = \sum_{k=2}^{n} \frac{1}{k} p^{2(k-2)}(qp+q)^{n-k}\binom{n-2}{k-2}$

and the probability a non-auctioneer does at least one task:

$$\mathbb{P}(W_{i\neq a} \neq \emptyset) = 1 - (1 - p^3\mathbb{P}(\theta_j \in \tilde{W}_{i\neq a} \,|\, \zeta_j))^m.$$

5.3 Analysis of G-Prim Auction when Bids are Realizations of Random Variables

We now consider G-Prim. For rounds $r < m$, an agent often gets multiple auction rounds to bid for items it values more than other agents. Let $\neg\zeta_{h,j,r}$ denote the event "agent h's bid for j was not received in round r".

Lemma 1 *Given $\theta_j \in \Theta_{\text{unsold}}$ at the beginning of round r and $V_{h,r}(\theta_j) > V_{i,r}(\theta_j)$ for all $i \neq h$ and where $r < m$; then $\mathbb{P}_{r+1}(\theta_j \in \Theta_{\text{unsold}} \,|\, \neg\zeta_{h,j,r}) > 0$.*

Proof By construction $|\Theta_{\text{unsold}}| > 2$ at the beginning of round r when $r < m$. The probability all agents $i \neq h$ bid on some other item $\theta_k \neq \theta_j$ is nonzero and thus $\mathbb{P}_r(\theta_j = \hat{b}_{i,j} \,|\, \neg\zeta_{h,j,r}) < 1$ for all $i \neq h$. It follows that $\mathbb{P}_r(\theta_j \in \tilde{B}_{i,j} \,|\, \neg\zeta_{h,j,r}) < 1$ for all $i \neq h$. If no agents bid on θ_j then θ_j is not sold; i.e., $\theta_j \notin \tilde{B}_{i,j}$ implies that $\theta_j \notin \Theta_{\text{sold}}$ at the end of round r and the beginning of round $r+1$.

Corollary 1 *For all θ_j such that $h = \arg\max_i V_i(\theta_j)$, $\mathbb{P}_{G\text{-}Prim}(\theta_j \in \tilde{W}_{i\neq a}) > \mathbb{P}_{Sequential}(\theta_j \in \tilde{W}_{i\neq a})$.*

In other words, Corollary 1 states that G-Prim increases $\mathbb{P}(\theta_j \in W_{i\neq a})$ for any θ_j that i values more than any other agent. Theorem 1 leverages Corollary 1 to bound the probability an agent wins zero items in G-Prim based on the Sequential Auction:

Theorem 1 *If $m > 1$ and $p < 1$ then:*
$$\mathbb{P}_{G\text{-}Prim}(W_{i\neq a} \neq \emptyset) \geq \mathbb{P}_{Sequential}(W_{i\neq a} \neq \emptyset).$$

Proof By construction, when $m > 1$ and $n > 1$ there is a greater than $1/n$ probability that an item θ_j exists such that agent h values θ_j more than any other agent; formally, $\mathbb{P}(\exists\theta_j \,|\, h = \arg\max_i V_i(\theta_j)) > 1/n$ for all agents h. Corollary 1 finishes the proof.

The effect described in Lemma 1 also increases the expected number of items visited by non-auctioneers because it reduces the auctioneer's advantage of self communication.

Corollary 2 $\mathbb{E}_{G\text{-}Prim}(|W_{i\neq a}|) \geq \mathbb{E}_{Sequential}(|W_{i\neq a}|)$.

Because there are only m items, the number of items visited by the auctioneer must decrease to maintain balance.

Corollary 3 $\mathbb{E}_{G\text{-}Prim}(|W_a|) \leq \mathbb{E}_{Sequential}(|W_a|)$.

5.4 Analysis of Repeated Parallel Auction when Bids are Realizations of Random Variables

When there is perfect communication, then the number of rounds in a Repeated Parallel Auction is $\ell = \lceil m/n \rceil$ and n items are sold in the each of the first $\ell - 1$ rounds and $m \mod n$ items are sold in the final round ℓ. In contrast, if communication is nonexistent then the auctioneer is the only bidder in each round, one item is sold per round (to the auctioneer), and the auction takes $\ell = m$ rounds. If communication is somewhere between perfect and nonexistent, then the length of the auction ℓ is a function of how many robots were able to get messages to the auctioneer in rounds $1, \ldots, \ell - 1$.

We are unable to express the probability of agent participation in this type of auction in a single closed form expression. However, we are able to compute the solution by iteratively tracking how a number of intermediate quantities change from round to round. The basic idea is to track how the probability that each particular number of items remains unsold at the end of each round, conditioning our calculation on an event that we are interested in (for example, assuming that a particular nonauctioneer has not yet acquired any items is useful for calculating the probability that a nonauctioneer acquires at least one item by the end of the auction). The auction is Markov by the definition of Scenario 1—the probability that an agent has won an item in a previous round does not change the probability that it will win an item in the current round. This enables us to calculate statistics for the auction state at the end of round r as a function of those that existed at the end of round $r - 1$.

In the Repeated Parallel Auction sells a number of items each round that is equal to the number of agents that were able to get a bid message to the auctioneer. This means that the iterative calculation needs to be slightly different for the case that the number of items remaining is less than the number of agents and the number of remaining items is at least the number agents.

We use $\Theta_{\text{r,unsold}}$ to represent the set of unsold items at the end of round r, and $W_{r,i\neq a}$ is the set of items won by a nonauctioneer agent $i \neq a$ by the end of round r. Let $P_{r,h}$ denote the probability that h items are left unsold at the end of round r, assuming that agent $i \neq a$ has not won anything by the end of round r:

$$P_{r,h} = \mathbb{P}_{\text{R-Parallel}}(|\Theta_{\text{r,unsold}}| = h \,|\, W_{r,i\neq a} = \emptyset).$$

By convention we allow round 0 to represent the time before the first round of the auction, thus

$$P_{0,h} = \begin{cases} 1 & \text{if } h = m \\ 0 & \text{if } h \neq m \end{cases}$$

which can also be interpreted as the initialization before the iterative calculation. The iterative calculation itself is presented in the following algorithms. The iterative calculation of the intermediate quantity $P_{r,h}$ is shown in Algorithm 7.

Algorithm 7 Iterative Calculation of $P_{r,h}$

for $r \in \{1 \dots m\}$ **do**
  $P_{r,0:m} \leftarrow 0$
  for $j \in \{0 \dots m\}$ **do**
    for $b \in \{1 \dots n\}$ **do**
      $h \leftarrow \max(0, j-b)$
      $A \leftarrow B \leftarrow C \leftarrow D \leftarrow 0$
      if $b < n$ **then**
        $A \leftarrow p^{2(b-1)}(pq+q)^{n-b}\binom{n-2}{b-1}$
      if $j \geq b$ **then**
        if $b \geq 2$ **then**
          $B \leftarrow qp^{2(b-1)}(pq+q)^{n-b}\binom{n-2}{b-2}$
      else
        $C \leftarrow \frac{b-j}{b}p^{2(b-1)}(pq+q)^{n-b}\binom{n-2}{b-2}$
        $D \leftarrow \frac{j}{b}qp^{2(b-1)}(pq+q)^{n-b}\binom{n-2}{b-2}$
      $P_{r,h} \leftarrow P_{r,h} + (A+B+C+D)P_{r-1,j}$
  $P_{r,0} \leftarrow P_{r,0} + P_{r-1,0}$

The outer most loop calculates the probabilities $P_{r,0:m}$ based on the probabilities $P_{r-1,0:m}$, the dynamics of the auction, and the Bernoulli model.

For each round r, the inner two loops calculate the various probabilities of the various events that can lead to agent $i \neq a$ not being assigned an item in round r. The inner two loops iterate over all possibilities that j items exist unsold at the beginning of the round (end of the previous round), and the different possibilities for the different numbers b of bids that can be received by the auctioneer. Using j and b we calculate h which is the number of items that are left after the current round r, and then add the relevant probability mass to $P_{r,h}$. There are four nontrivial events that transfer probability mass; these are, respectively:

A Nonauctioneer agent $i \neq a$ is unable to send a bid message to the auctioneer.

B Nonauctioneer agent $i \neq a$ is one of $b-1$ nonauctioneers that get a bid to the auctioneer, there are more than b items left, but agent i does not receive the award message.

C Nonauctioneer agent $i \neq a$ is one of $b-1$ nonauctioneers that get a bid to the auctioneer, there are less than b items left for sale, and agent $i \neq a$ is outbid for all items.

D Nonauctioneer agent $i \neq a$ is one of $b-1$ nonauctioneers that get a bid to the auctioneer, there are less than b items left for sale, agent $i \neq a$ wins one of the remaining items, but the award message is dropped.

The conditional probabilities A, B, C, and D are multiplied by their preconditions for different combinations of starting item count j and ending item count m. This has the effect of summing up the probabilities of all probability mass associated with agent $i \neq a$ continuing to not receive an item (at least from agent i's point-of-view). The probability mass associated with the trivial event that all items have already been sold is also carried forward each round, and added to the probably that the auctioneer sold out during that particular round.

Given the iterative calculation, the probability that a nonauctioneer $i \neq a$ participates (agent $i \neq a$ knows it is allocated at least one item) is calculated:

$$\mathbb{P}(W_{i \neq a} \neq \emptyset) = 1 - P_{m,0}$$

which happens because, by the end of the iterative calculation, $P_{m,0}$ has accumulated all probability mass associated with all event sequences in which agent $i \neq a$ never gets allocated an item (to its knowledge), by the end of round m — and all items are necessarily sold by then (due to the fact that the auctioneer wins one item per round).

The expectations of the number of items won by each nonauctioneer, the auctioneer, etc. can be calculated if we know $\hat{P}_{r,h}$, the probability that a particular number of items exists unsold over all possible event histories for each possible number of items left unsold h at each round r. It is possible to compute each $\hat{P}_{r,h}$ iteratively given all $\hat{P}_{r-1,h}$. Again, by convention we allow round 0 to represent the time before the first round of the auction, thus

$$\hat{P}_{0,h} = \begin{cases} 1 & \text{if } h = m \\ 0 & \text{if } h \neq m \end{cases}$$

The iterative calculation of $\hat{P}_{r,h}$ is shown in Algorithm 8.

Algorithm 8 Iterative Calculation of $\hat{P}_{r,h}$

for $r \in \{1 \dots m\}$ **do**
  $\hat{P}_{r,0:m} \leftarrow 0$
  for $j \in \{0 \dots m\}$ **do**
    for $b \in \{1 \dots n\}$ **do**
      $h \leftarrow \max(0, j-b)$
      $\hat{P}_{r,h} \leftarrow \hat{P}_{r,h} + p^{2(b-1)}(pq+q)^{n-b}\binom{n-1}{b-1}\hat{P}_{r-1,j}$
  $\hat{P}_{r,1} \leftarrow \hat{P}_{r,1} + \hat{P}_{r-1,1}$

Given $\mathbb{E}\Big(|\tilde{W}_{i \neq a}|\Big)$ the expected number of items acquired by each nonauctioneer is calculated:

$$\mathbb{E}\Big(|W_{i \neq a}|\Big) = p\mathbb{E}\Big(|\tilde{W}_{i \neq a}|\Big).$$

Algorithm 9 Iterative Calculation of $\mathbb{E}\left(|\tilde{W}_a|\right)$

$F \leftarrow 0$
for $r \in \{1 \dots m\}$ **do**
  for $j \in \{0 \dots m\}$ **do**
    for $b \leftarrow 1 \dots n$ **do**
      if $b \leq j$ **then**
        $F \leftarrow F + p^{2(b-1)}(pq+q)^{n-b}\binom{n-1}{b-1}\hat{P}_{r-1,j}$
      else
        $F \leftarrow F + \frac{j}{b}p^{2(b-1)}(pq+q)^{n-b})\binom{n-1}{b-1}\hat{P}_{r-1,j}$
$\mathbb{E}\left(|\tilde{W}_a|\right) \leftarrow F$

Algorithm 10 Iterative Calculation of $\mathbb{E}\left(|\tilde{W}_{i\neq a}|\right)$ for Batch Auction

$E \leftarrow 0$
for $r \in \{1 \dots m\}$ **do**
  for $j \in \{0 \dots m\}$ **do**
    for $b \leftarrow 2 \dots n$ **do**
      if $b \leq j$ **then**
        $E \leftarrow E + p^{2(b-1)}(pq+q)^{n-b}\binom{n-2}{b-2}\hat{P}_{r-1,j}$
      else
        $E \leftarrow E + \frac{j}{b}p^{2(b-1)}(pq+q)^{n-b}\binom{n-2}{b-2}\hat{P}_{r-1,j}$
$\mathbb{E}\left(|\tilde{W}_{i\neq a}|\right) \leftarrow E$

The derivations of other expected quantities in term of $\mathbb{E}\left(|\tilde{W}_{i\neq a}|\right)$ follow the same progression as for the Parallel Auction.

5.5 Analysis of Repeated G-Prim Auction when Bids are Realizations of Random Variables

The relationship between Repeated G-Prim and Repeated Parallel is analogous to that between the Parallel Auction and the G-Prim Auction—as long as $m > n$ (when $n \leq m$ Repeated G-Prim is equivalent to Repeated Parallel because all agents bid on all items in a single round). Thus, when $m > n$ the closed form solution for the expected number of items won by nonauctioneers for Repeated Parallel becomes a lower bound for the expected number of items won by nonauctioneers for Repeated G-Prim (and the reverse is true for the expected number of items won by the auctioneer). This happens because each agent has, on average, more than a single round to attempt to bid for items that it highly values (increasing the chances that it wins them). Similarily, the probability that a nonauctioneer wins any item in a Repeated Parallel Auction is a lower bound on the same quantity for the Repeated G-Prim Auction.

5.6 Analysis of Combinatorial Auction when Bids are Realizations of Random Variables

The Combinatorial Auction only lasts one round, similar to the Parallel Auction. Moreover, Given the independence of task values that is assumed in Scenario 1, the event that an agent is awarded any particular task does not change that agent's value of any other task. Therefore, in Scenario 1 the Sequential Auction and the Combinatorial Auction will yield identical results (assuming the same communications between auctioneers and nonauctioneer are successful or dropped). Thus, trivially, for Scenario 1 the expected number of items visited by agents is the same for the Parallel Auction as for the Sequential Auction. The probability that a nonauctioneer is awarded at least one item is also the same. It is important to note that the similarity of results between the Sequential Auction and th Parallel Auction is only valid for Scenario 1; this analysis does not hold in the general case that the cost or value of a set is not simply the cost or value of its individual items—for example, in Scenario 2.

5.7 Analysis of Auctions When Item Costs are Distance-Based

Costs in Scenario 2 are defined by the extra TSP length required to visit a new location (Equation 2). When i wins θ_j, the multi-TSP path and its sub-length over i's tasks cannot shorten; indeed, it lengthens almost surely[8]. This is formalized in Proposition 1.

Proposition 1 $\mathbb{P}(\ell_{\mathrm{TSP}}(W_{i,r} \cup \{\theta_j\}) > \ell_{\mathrm{TSP}}(W_{i,r})) = 1.$

Lengthening i's multi-TSP path causes i to visit more of the environment (due to the triangle inequality), and decreases i's cost of visiting other locations with a higher probability than it increases it[9]. This is formalized in Proposition 2.

Proposition 2 *Assuming* $W_{i,r} = W_{i,r-1} \cup \{\theta_j\}$*, then* $\ell_{\mathrm{TSP}}(W_{i,r}) > \ell_{\mathrm{TSP}}(W_{i,r-1}) \implies \mathbb{P}(V_{i,r}(\theta_k) < V_{i,r-1}(\theta_k)) > 1/2$*, for all* $\theta_k \in \Theta_{\mathrm{unsold}}$ *at round* r*.*

[8] This statement makes the implicit assumption locations are initially chosen by a random process that would sample the environment densely, in the limit, if the number of locations were allowed to go to infinity. The "almost surely" refers to the fact that, given randomly chosen locations, the chances a new location lies along the old multi-TSP (in which case the multi-TSP length remains the same) is zero.

[9] Note that, this assumes item locations are initially chosen by a random process that would sample the environment densely, in the limit, if the number of locations were allowed to go to infinity.

A lower cost of visiting θ_k increases the chances that an agent will win θ_k. This is formalized in Proposition 3.

Proposition 3 *When* $p > 0$ *and* $\theta_k \in \Theta_{\text{unsold}}$ *in* $r-1$, $V_{h,r}(\theta_k) < V_{i,r-1}(\theta_k) \implies \mathbb{P}(\theta_k \in W_{h,r}) > \mathbb{P}(\theta_k \in W_{i,r-1})$.

Combining Propositions 2 and 3 yields Corollary 4.

Corollary 4 *When* $p > 0$ *and* $\theta_j, \theta_k \in \Theta_{\text{unsold}}$ *at the beginning of round* $r-1$ *and* $W_{h,r} = W_{h,r-1} \cup \{\theta_j\}$, *then* $\mathbb{P}(\theta_k \in W_{h,r+1}) > \mathbb{P}(\theta_k \in W_{i,r})$.

Corollary 4 states that if h wins any task θ_j then the probability h wins another task θ_k increases (in Scenario 2). The following Corollary 5 holds because if agent h's chances of winning θ_k increase, the chance that agents $i \neq h$ wins θ_k must decrease.

Corollary 5 *When* $p > 0$ *and* $\theta_j, \theta_k \in \Theta_{\text{unsold}}$ *in round* $r-1$ *and* $W_{h,r} = W_{h,r-1} \cup \{\theta_j\}$, *then* $\mathbb{P}(\theta_k \in W_{i\neq h,r+1}) < \mathbb{P}(\theta_k \in W_{i\neq h,r})$.

Given randomly distributed start and item locations, the costs in *round 1* of Scenario 2 meet all assumptions required by the analysis of Scenario 1 (agents draw values from the same distribution and the maximum value objective of Scenario 1 is met by negating Scenario 2's costs). This leads to Proposition 4.

Proposition 4 *The probability that i wins round 1 of an auction in Scenario 1 is equal to the probability that i wins Round 1 of the same auction type in Scenario 2.*

Corollary 6 holds because Parallel Auctions have one round.

Corollary 6 *For the Parallel Auction, all results derived in Scenario 1 are valid for Scenario 2.*

We now prove that the equations derived for the Sequential Auction in Scenario 1 become inequalities that provide bounds on the Sequential Auction in Scenario 2 (Lemma 2 and Corollaries 7-10).

Lemma 2 *When* $p < 1$ *and for all* θ_j $\mathbb{P}_{S2,\ Sequential}(\theta_j \in \tilde{W}_{a,r}) \geq \mathbb{P}_{S1,\ Sequential}(\theta_j \in \tilde{W}_{a,r})$.

Proof The auctioneer has an advantage over nonauctioneers when $p < 1$ because the auctioneer always has perfect communication with itself. Consequently, a is more likely to win round $r \geq 1$ than $i \neq a$ when $p < 1$; and thus a has an increased chance of winning rounds $r > 1$ by Corollary 4.

Lemma 2 has Corollaries 7 and 8, regarding the probability that agents adopt at least one task; and Corollaries 9 and 10, regarding the expected number of tasks adopted by agents.

Corollary 7 *When* $p < 1$, $\mathbb{P}_{S2,\ Sequential}(\theta_j \in W_{a,r}) \geq \mathbb{P}_{S1,\ Sequential}(\theta_j \in W_{a,r})$.

Corollary 8 *When* $p < 1$, $\mathbb{P}_{S2,\ Sequential}(\theta_j \in W_{i\neq a,r}) \leq \mathbb{P}_{S1,\ Sequential}(\theta_j \in W_{i\neq a,r})$.

Corollary 9 *When* $p < 1$, $\mathbb{E}_{S2,\ Sequential}(|W_a|) \geq \mathbb{E}_{S1,\ Sequential}(|W_a|)$.

Corollary 10 *When* $p < 1$, $\mathbb{E}_{S2,\ Sequential}(|W_{i\neq a}|) \leq \mathbb{E}_{S1,\ Sequential}(|W_{i\neq a}|)$,

G-Prim's Scenario 2 performance is similarly bounded by G-Prim's Scenario 1 performance. Formal proofs follow the same logic as for the Sequential Auction, but are less useful because they go in the opposite direction of G-Prim's closed-form bounds for Scenario 1. Nonetheless, averaging over repeated trials of G-Prim in Scenario 1 provides a means of obtaining a numerical bound on its performance in Scenario 2.

Similar reasoning to Corollaries 7-10 can be used to show that the iterative calculations that we derive for the Repeated Parallel Auction for Scenario 1 are bounds on those of Repeated Parallel Auction of Scenario 1; and that the performance of Repeated Parallel Auction of Scenario 1 bounds that of the Repeated G-Prim Auction in Scenario 2 (but in the opposite direction of what would be required to use the performance of Repeated Parallel Auction for Scenario 1 as a bound on G-Prim in Scenario 2).

6 Analysis of Straightforward Extensions

The auction algorithms that have been presented up to this point represent the most basic implementations of each idea. However, there are two simple modifications that may significantly improve performance in many cases. (1) Re-sending each message multiple times, e.g., in the event that its receipt has not been acknowledged. (2) Having the auctioneer send the winner lists of all previous rounds along with the winner list of the current round. We investigate each of these ideas in the remainder of this section.

6.1 (Naively) Re-sending each message up to c_t times

A simple way to increase the probability that important data gets to a destination is to re-send data multiple times. In the most naive implementation each messages is simply sent c_t times. We note that this is only useful if the communication requirements of the auction are well below the limits of the communication channel; otherwise, naively sending more messages will likely cause further communication degradation.

The analysis that we have done up to this point can be modified to handled this special case (in which channel capacity itself is not an issue) as follows: we redefine q and p to be the probabilities that a particular *piece of data* never gets delivered or eventually gets delivered, respectively. We let $\hat{q}$ and $\hat{p}$ be the probabilities that a particular message never gets delivered or gets delivered, $\hat{q} = 1 - \hat{p}$. The probability a piece of data never gets sent is equal to the probability that each of c_t messages is dropped,

$$q = \hat{q}^{c_t}$$

and the probability a message eventually gets delivered is the compliment event

$$p = 1 - \hat{q}^{c_t} = 1 - (1 - \hat{p})^{c_t}$$

Indeed, this also handles the closely related idea in which an acknowledgment based protocol is used. In an acknowledgment protocol each messages is re-sent until the sender receives an acknowledgment from the receiver, and the total number of times a piece of data may be transmitted is limited to c_t times.

It is important to note that this idea has limited practical application if the communication required by the auction already represents a significant amount of the communication channel's capacity; in which case the Bernoulli model is a poor approximation due to the fact that sending more data may simply decrease communication quality.

6.2 Rebroadcasting the winners of previous rounds during the current round

With the exceptions of the Parallel Auction and the Combinatorial Auction, all auctions presented in Section 4 require more than a single round. Another simple way to increase agent participation is to rebroadcast the winners of rounds $1, \ldots, r-1$ along with the winners of round r. That way, if one (or some) award messages are dropped, then agents have additional chances to be informed of the items they have won in earlier rounds.

Having the auctioneer resend the list of previous winners along with the new winners does not increase the *number* of messages that must be sent, only the size of the award message. This strikes a balance between adding additional communication onto the channel and helping agents know which tasks they have won when communication is imperfect. In the case of G-Prim and the batch G-Prim Auctions, this increases the total bandwidth sent from auctioneer to agents by no more than a small constant factor (less than 2). In G-Prim and batch G-Prim the auctioneer already advertises all items that are still unsold during each round, which requires bandwidth $\mathcal{O}(m)$. However, the relative effects are larger for the sequential algorithm and the naive batch method, which will have increased bandwidth sent from auctioneer to agents by $\mathcal{O}(m)$ and $\mathcal{O}(m/n)$, respectively.

As a rough approximation, if there is enough bandwidth to use G-Prim or Batch G-Prim, then there is likely enough bandwidth to use the modified version of these algorithms that resend all winners each round. Whether or not using these ideas with the sequential algorithm or the naive batch method makes sense will depend problem specifics such as the system being used and environmental factors. As with the previous calculation, this analysis is also useful because it provides an upper bound on how performance may be improved by simply sending more messages.

We now we discuss how the performance of the Sequential Auction can be improved by this technique, assuming Scenario 1 and a Bernoulli distribution. We are unable to calculate a complete closed form solutions for the quantities that we are interested in; however, we are able to describe an iterative algorithm that can be used to obtain the desired quantity. A similar iterative calculation for the Repeated Parallel Auction appears in the appendix. As with the versions of these algorithms that do not resend winners, the results for the Sequential Auction represent a bound on the G-Prim Auction, and results for the Repeated Parallel Auction represent a bound on the Repeated G-Prim, and performance in Scenario 1 is a bound on performance in Scenario 2 (but in the opposite direction of what would be required to apply the results of the iterative calculations to G-Prim and Repeated G-Prim in scenario 2).

We observe that most of the auction dynamics of the Sequential Auction with Winner Re-sends remain unchanged with respect to the basic Sequential Auction. However, if an agent wins a particular item in round r of a multi-round auction, then it has $m - r + 1$ chances to receive the award message from the auctioneer. Assuming that the winning agent receives its earliest award message in $\hat{r}$ (where $\hat{r} \geq r$) then the auctioneer has $m - \hat{r} + 1$ chances to receive the acknowledgment before it assumes responsibility for the item.

The expected number of items won outright by the auctioneer and each individual non-auctioneer remain the same. Respectively:

$$\mathbb{E}\left(|\tilde{W}_a|\right) = m \sum_{k=1}^{n} \frac{1}{k} p^{2(k-1)} (qp + q)^{n-k} \binom{n-1}{k-1}$$

and

$$\mathbb{E}\left(|\tilde{W}_{i \neq a}|\right) = m \sum_{k=2}^{n} \frac{1}{k} p^{2(k-1)} (qp + q)^{n-k} \binom{n-2}{k-2}.$$

Taking responsibility for (or *adopting*) an item requires both winning that item and also receiving the award message; in this case the latter depends on the round in which the item was won. Assuming that a nonauctioneer wins an item in round r, then the probability that agent does not receive an award message for that item by the end of the auction is $q^{m-\hat{r}+1}$, and the probability that an award message is received is $1 - q^{m-\hat{r}+1}$. Summing over the expected items won in each round gives us the expected number of items adopted by a single non-auctioneer:
$\mathbb{E}\Big(|W_{i\neq a}|\Big) = \sum_{r=1}^{m}(1-q^{m-r+1})\sum_{k=2}^{n}\frac{1}{k}p^{2(k-1)}(qp+q)^{n-k}\binom{n-2}{k-2}$.

The expected number of items adopted by a single non-auctioneer and the set of all non-auctioneers is

$$\mathbb{E}\Big(\sum_{i\neq a}|W_i|\Big) = (n-1)\mathbb{E}\Big(|W_{i\neq a}|\Big).$$

The probability that the auctioneer never receives an acknowledgment message for an item sold to some non-auctioneer in round r is dependent on both the probability an agent itself receives (the first) successful transmission of the award message, and the probability of a successful acknowledgment. The number of new items that an agent becomes aware of winning upon receiving an award message is dependent on the transmission history. This makes closed form solutions for both (A) the number of items visited by the auctioneer, and (B) the number of items visited twice difficult to calculate in closed form. However, these quantities can be calculated in an iterative fashion according to algorithm 11. The basic idea is to track how the expected quantities of items change after every round, which is illustrated in Algorithm 11.

We note that intermediate quantities that we track in Algorithm 11 and subsequent algorithms are labeled using capital letters; different algorithms use different modifiers on capitol letters to help denote the fact that the temporary quantities in different algorithms are unrelated to each other. C/C++ language style comments appear in our presentation to help communicate what each quantity represents. We use the word "adopted" to denote the case when a nonauctioneer *knows* that it has won an item—which is different from the case that the nonauctioneer has been awarded the item (since, in the latter case, the nonauctioneer may not have yet received this knowledge in a successfully transmitted award message).

The probability that a nonauctioneer agent does at least one task in a Parallel Auction with the modification we are considering can be calculated using a similar iterative calculation that is displayed in Algorithm 12.

Algorithm 11 Iterative Calculation of $\mathbb{E}\Big(|W_a|\Big)$ given p, q, n, and m.

$\acute{Y} \leftarrow 0$; $\acute{R} \leftarrow 0$; $\acute{L} \leftarrow 0$; $\acute{J} \leftarrow 0$; $\acute{N} \leftarrow 0$; $\acute{B} \leftarrow 0$
for $r \in \{1 \dots m\}$ **do**
  $\acute{S} \leftarrow \sum_{b=1}^{n}\frac{1}{n}p^{2(b-1)}(pq+q)^{n-b}\binom{n-1}{b-1}$
  // $\acute{S} \equiv \mathbb{E}$(num items won by auctioneer during round r)
  $\acute{K} \leftarrow \sum_{b=2}^{n}\frac{1}{n}p^{2(b-1)}(pq+q)^{n-b}\binom{n-2}{b-2}$
  // $\acute{K} \equiv \mathbb{E}$(num items won by each nonauctioneer during
  // round r)
  $\acute{Y} \leftarrow \acute{Y} + \acute{S}$
  // $\acute{Y} \equiv \mathbb{E}$(num items won by auctioneer through round r)
  $\acute{R} \leftarrow \acute{R} + (n-1)\acute{K}$
  // $\acute{R} \equiv \mathbb{E}$(num items won by all nonauctioneers through r)
  $\acute{Q} \leftarrow (n-1)rpq^{m-r}\acute{K}$
  // $\acute{Q} \equiv \mathbb{E}$(num items adopted, all nonauctioneers, end of r)
  $\acute{E} \leftarrow \acute{B} + \acute{J} + \acute{K}$
  // $\acute{E} \equiv \mathbb{E}$(non acked items in r [before awards sent])
  $\acute{L} \leftarrow \acute{L} + (\acute{J} + \acute{K})p$
  // $\acute{L} \equiv \mathbb{E}$(num items each nonauctioneer adopts by the end
  // of round r)
  $\acute{J} \leftarrow (\acute{J} + \acute{K})q$
  // $\acute{J} \equiv \mathbb{E}$(num items won but not adopted, by a particular
  // nonauctioneer, after r)
  $\acute{N} \leftarrow \acute{N} + (n-1)(p^2\acute{E} + q\acute{B})$
  // $\acute{N} \equiv \mathbb{E}$(num items acked by all nonauctioneers by
  // the end of round r)
  $\acute{B} \leftarrow qp\acute{E} + q\acute{B}$
  // $\acute{B} \equiv \mathbb{E}$(num items adopted but not acked by a particular
  // nonauctioneer)
$\acute{X} = \acute{R} - \acute{Q}$
// $\acute{X} \equiv \mathbb{E}$(num items won but not adopted, by all
// nonauctioneers)
$\acute{B}_{all} = (n-1)\acute{B}$
// $\acute{B}_{all} \equiv \mathbb{E}$(num items adopted but not acked by all nonauctioneers)
$\mathbb{E}\Big(|W_a|\Big) \leftarrow \acute{Y} + \acute{X} + \acute{B}_{all}$
$\mathbb{E}\Big(|W_{i\neq a}|\Big) \leftarrow \acute{L}$

Algorithm 12 Iterative Calculation of $\mathbb{P}_{\text{Seq resend}}(W_{i\neq a} \neq \emptyset)$ given p, q, n, and m.

$A \leftarrow 0$; $B \leftarrow 0$; $C \leftarrow 1$; $D \leftarrow 0$
for $r \in \{1 \dots m\}$ **do**
  $E \leftarrow \sum_{b=2}^{n}\frac{1}{n}p^{2(b-1)}(pq+q)^{n-b}\binom{n-2}{b-2}$
  // $E \equiv \mathbb{P}$(auctioneer wins in round r)
  $A \leftarrow A + (B + CE)p$
  // $A \equiv \mathbb{P}$(nonauctioneer won before r and knows it won)
  $B \leftarrow (B + CE)q$
  // $B \equiv \mathbb{P}$(nonauctioneer won by r but is ignorant of win)
  $C \leftarrow C(1 - E)$
  // $C \equiv \mathbb{P}$(nonauctioneer did not win by round r)
  $D \leftarrow 1 - C$
  // $D \equiv \mathbb{P}$(nonauctioneer won before round r)
$\mathbb{P}_{\text{Seq resend}}(W_{i\neq a} \neq \emptyset) \leftarrow 1 - A$

7 Experiments

Simulation experiments, which are presented in Sections 7.1-7.3 are run on a Dell Precision computer with 64GB RAM and an Intel Core i7 processor (note only one core is used at a time). TSP and mixed integer programming solutions were found using Google's Operations Research Toolbox. We are able to run simulations much more quickly than real-time. Experiments involving AscTec Pelican UAVs are presented in Section 7.4 (See Section 7.4 for a full description of the system used for these hardware experiments).

7.1 Agent Utilization Assuming Random Valuations and Bernoulli distribution

In this section we present the results of an extensive comparison of agent utilization given the various auctions for the case that valuations are random variables representing cost, and where the auctioneer is attempting to minimize the sum of costs over all agents. The Bernoulli communication model is assumed, which enables us to compare the results of Monte Carlo simulations (1000 for each data-point) to our analytical results. For the Monte Carlo simulations, every robot determines a unique valuation for each item by drawing a random number from the range [0 1], and we simulate the nine cases where 3, 30, and 300 agents divide 10, 100, and 1000 tasks. Communication is varied across the range $p \in [0\ 1]$.

We note that results for the Combinatorial Auction are identical to those for the Parallel Auction in the case presented in this section, and plots for the Combinatorial Auction are omitted to save space. In particular, the parallel and Combinatorial Auctions are identical in the special case that there are no symbioses between any items (and this happens here because valuations are determined by random variables that are independent of the other items owned or not owned by each agent). Both the Parallel Auction and Combinatorial Auction last a single round.

Data is presented in two different formats as follows:

1. Figures 5-8 present agent utilization statistics for different auctions on the same plots to facilitate the comparison of different auctions. Each figure contains nine plots, one for each combination of n and m, where $n \in \{3, 30, 300\}$ is agent count and $m \in \{10, 100, 1000\}$ is item count.
   - Figure 5: each plot shows the probability that any particular nonauctioneer agent takes possession of at least one item, i.e., does at least one task. This is useful for determining how likely nonauctioneer agents are to contribute to the solution in the various scenarios.
   - Figure 6: each plot shows the expected number of items that any particular nonauctioneer takes possession of. This is useful for determining the expected workload of nonauctioneers in different scenarios.
   - Figure 7: each plot shows the expected number of items the auctioneer takes possession of. Note that the auctioneer takes possession of any items that it wins outright, as well as any items awarded to other agents but not acknowledged by the end of the auction. This is useful for determining the expected workload of the auctioneer in different scenarios.
   - Figure 8: each plot shows the expected number of items that are possessed by more than one agent, i.e., the auctioneer and some nonauctioneer. In a task allocation scenario, double visits represent wasted effort that happens when a nonauctioneer takes possession of a task, but fails to acknowledge its acceptance of that task to the auctioneer (so the auctioneer assumes it must take possession of the task).
   - Figure 9 shows the expected solution cost, where cost is calculated by summing the cost of all tasks that are done to the agent(s) that do them. When the auctioneer and a nonauctioneer both do a task, then the costs to both agents are included in the summation.
2. Figures 17-24 Appear in the appendix. Each figure displays on the performance of a single auction type (either the basic version, or versions where winner data is re-sent in subsequent rounds), and show all information about the expected number of items won by a any particular nonauctioneer, all nonauctioneers, and the auctioneer; as well as the expected number of items visited twice. Again, each figure contains nine plots, one for each combination of $n \in \{3, 30, 300\}$ and $m \in \{10, 100, 1000\}$. Seeing the different expected item number statistics for a particular auction type on the same plot makes it easier to see how item allocation shifts from the auctioneer to the nonauctioneers as communication improves.

From looking at Figure 5-8 and 17-24 it is apparent that auction variants that resend data each round have better agent utilization than those that do not. We also find that G-Prim and Repeated G-Prim have the best agent utilization of all auctions, with G-Prim outperforming Repeated G-Prim when communication is particularly poor. A more detailed discussion of the results from this experiment appears later in Section 8.

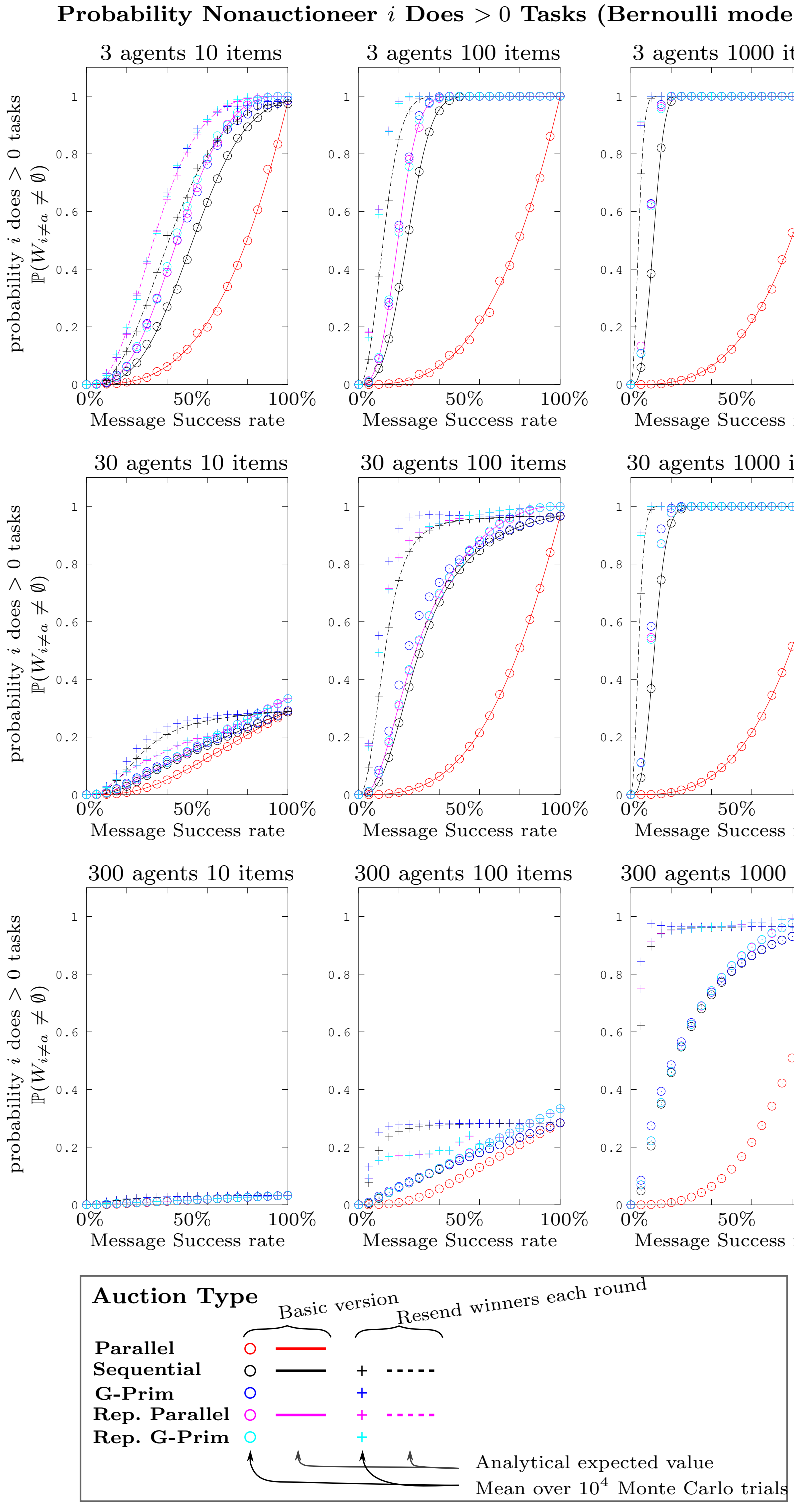

Fig. 5: The probability that a non-auctioneer agent ($i \neq a$) takes position of at least one item (in other words, does at least one task), given different team sizes (rows), numbers of items (columns), and assuming various communication qualities in a Bernoulli communication model (horizontal axes). Analytical values (lines) vs. results from simulations (markers) for various auction algorithms (colors). Auction algorithms that resend award data in subsequent rounds are drawn as dashed lines and with '+' symbols, while auction algorithms that do not are drawn with solid lines and 'o' symbols. Note that in the lower left plots there are more agents than items, and so most agents do not do a task. The analytical solutions are not depicted for cases we are unable to calculate them, e.g., due to numerical limitations or the computational complexity of the iterative calculations.

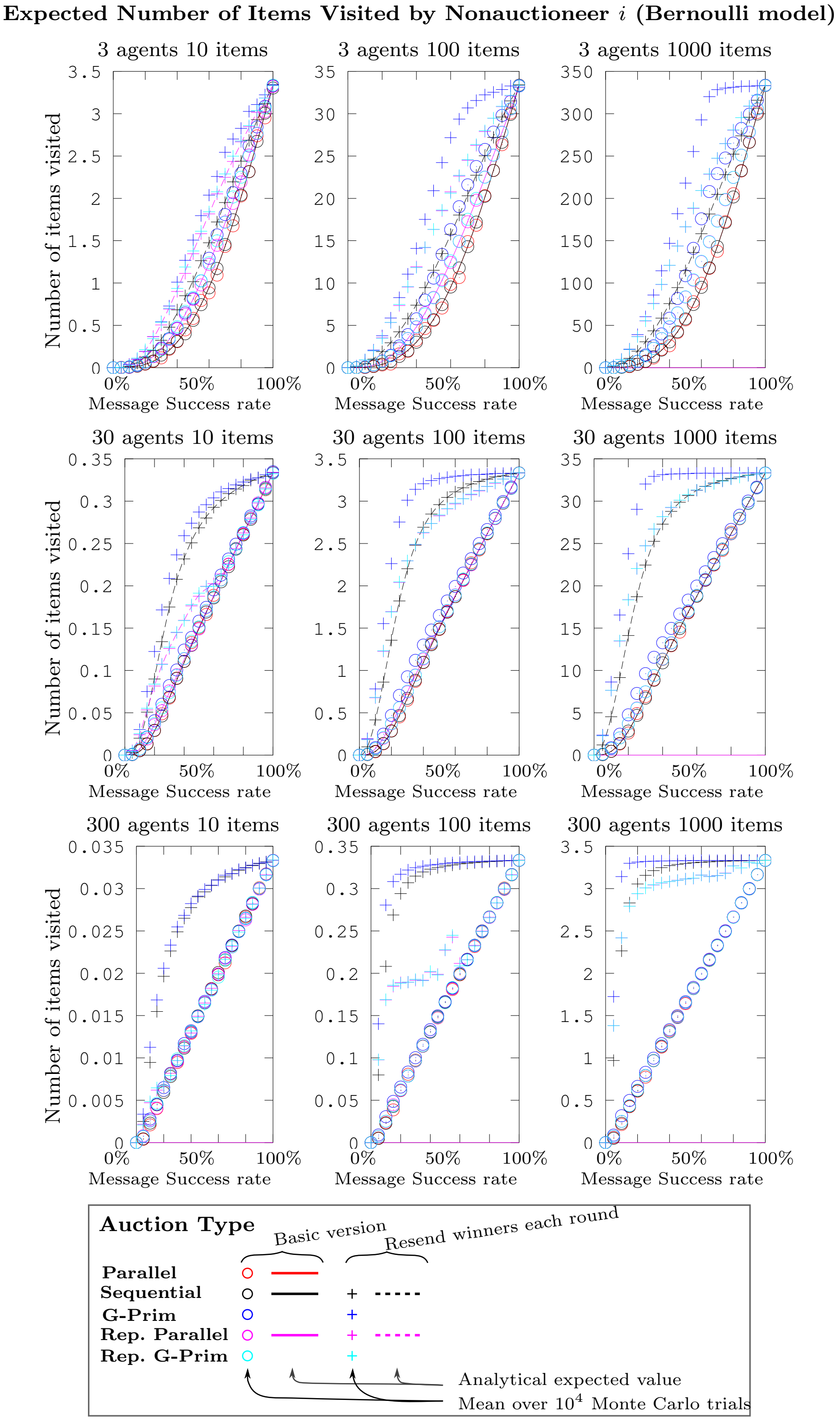

Fig. 6: The expected number of items visited by each non-auctioneer ($i \neq a$), given different team sizes (rows), numbers of items (columns), and assuming various communication qualities in a Bernoulli communication model (horizontal axes). Analytical values (lines) vs. results from simulations (markers) for various auction algorithms (colors). Auction algorithms that resend award data in subsequent rounds are drawn as dashed lines and with '+' symbols, while auction algorithms that do not are drawn with solid lines and 'o' symbols. The analytical solutions are not depicted when we are unable to calculate them, e.g., due to numerical limitations or the computational complexity of the iterative calculations.

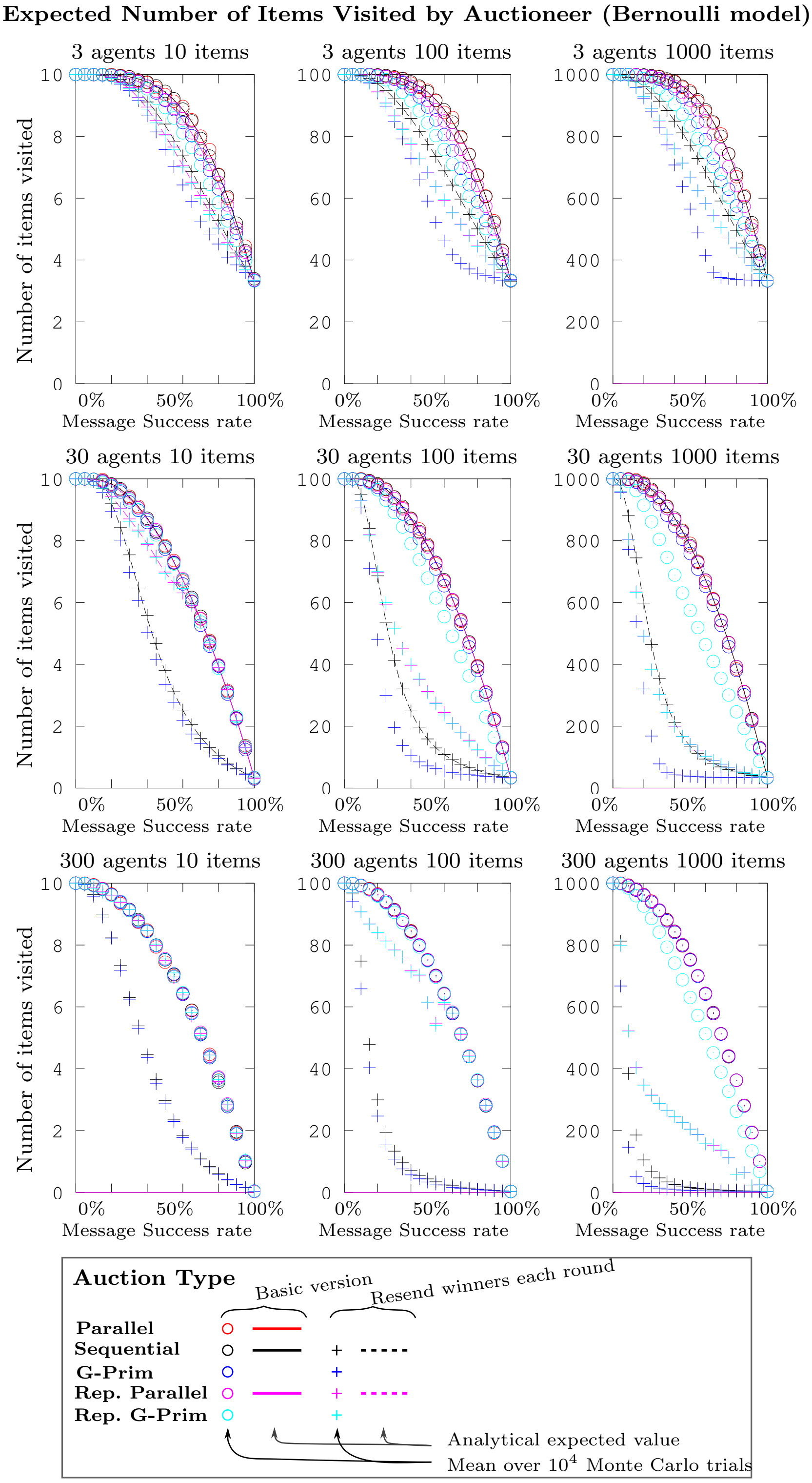

Fig. 7: The expected number of items visited the auctioneer, given different team sizes (rows), numbers of items (columns), and assuming various communication qualities in a Bernoulli communication model (horizontal axes). The auctioneer visits items that it wins as well as all unacknowledged sales. Analytical values (lines) vs. results from simulations (markers) for various auction algorithms (colors). Auction algorithms that resend award data in subsequent rounds are drawn as dashed lines and with '+' symbols, while auction algorithms that do not are drawn with solid lines and 'o' symbols. The analytical solutions are not depicted when we are unable to calculate them, e.g., due to numerical limitations or the computational complexity of the iterative calculations.

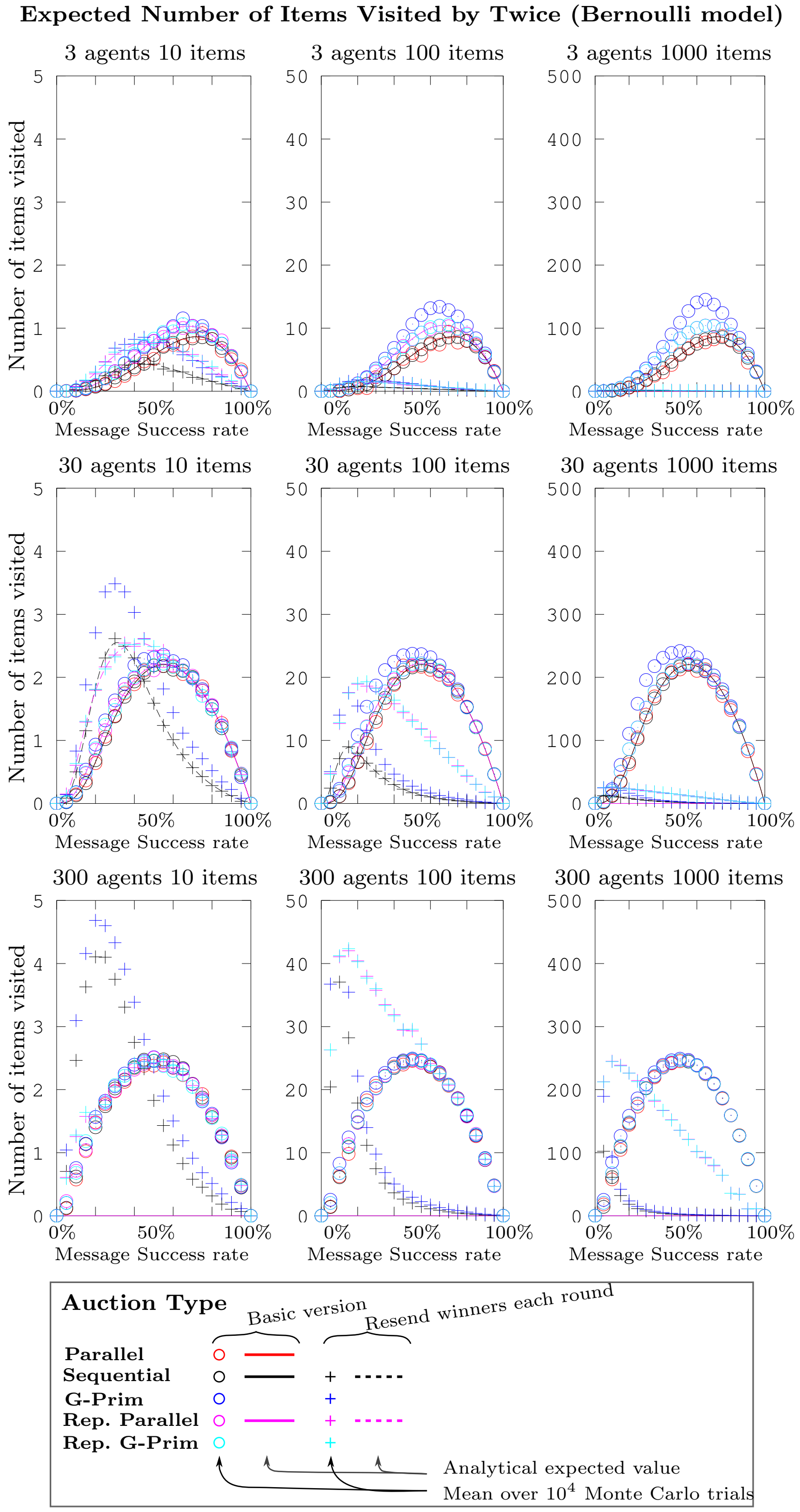

Fig. 8: The expected number of items visited twice (that is, by the auctioneer and some other agent), given different team sizes (rows), numbers of items (columns), and assuming various communication qualities in a Bernoulli communication model (horizontal axes). Double visits represent items that were received by a nonauctioneer, but not successfully acknowledged. Analytical values (lines) vs. results from simulations (markers) for various auction algorithms (colors). Auction algorithms that resend award data in subsequent rounds are drawn as dashed lines and with '+' symbols, while auction algorithms that do not are drawn with solid lines and 'o' symbols. The analytical solutions are not depicted when we are unable to calculate them, e.g., due to numerical limitations or the computational complexity of the iterative calculations.

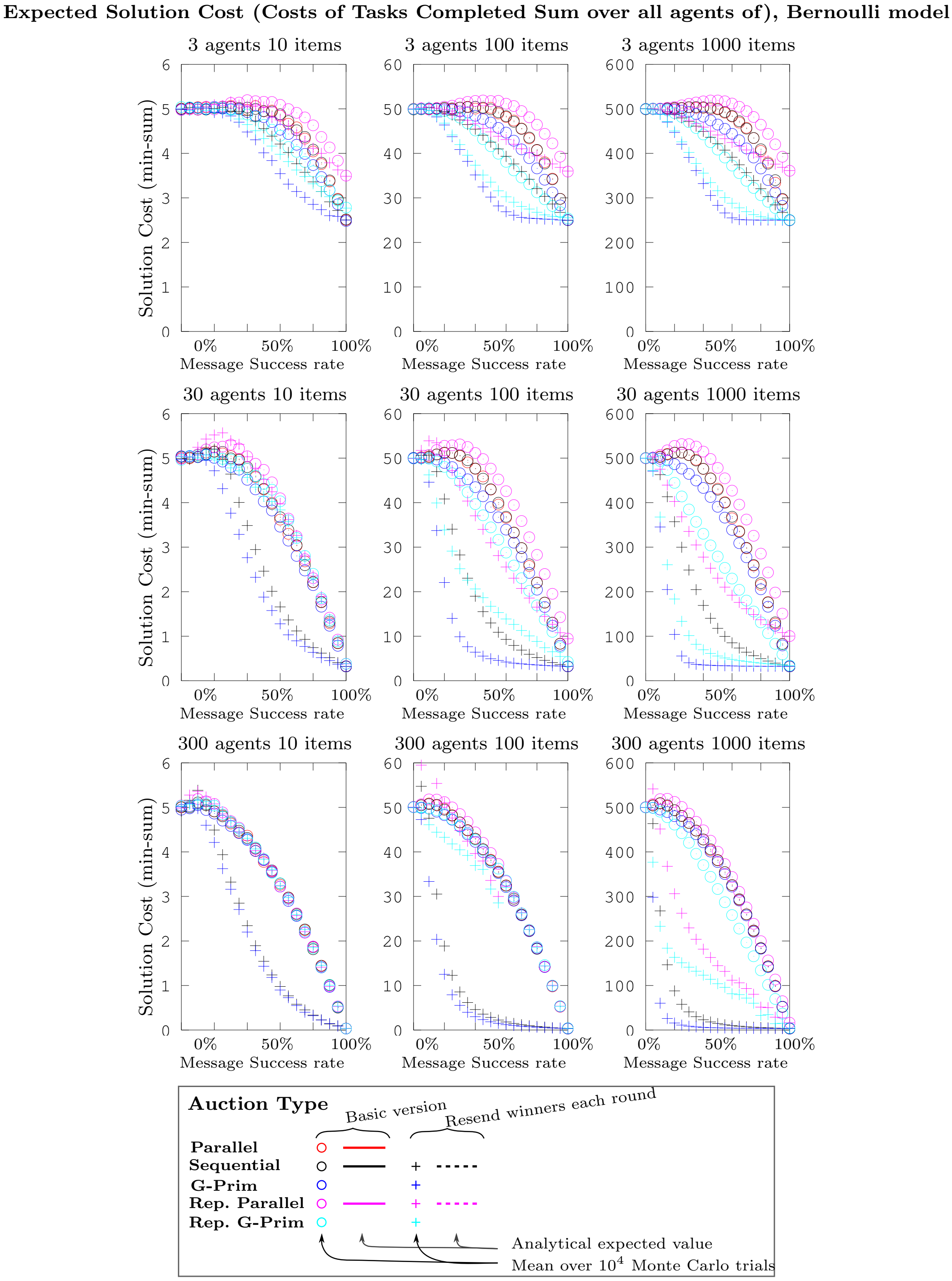

Fig. 9: The expected solution cost (summed cost of tasks accepted, given random cost valuation on interval [01] for each task by each robot), given different team sizes (rows), numbers of items (columns), and assuming various communication qualities in a Bernoulli communication model (horizontal axes). Analytical values (lines) vs. results from simulations (markers) for various auction algorithms (colors). Auction algorithms that resend award data in subsequent rounds are drawn as dashed lines and with '+' symbols, while auction algorithms that do not are drawn with solid lines and 'o' symbols. The analytical solutions are not depicted when we are unable to calculate them, e.g., due to numerical limitations or the computational complexity of the iterative calculations. If a task is done by both the auctioneer and a nonauctioneer (due to failed communication) then its cost to both agents is included in the solution cost.

7.2 Comparison of agent utilization between random valuation and min TSP valuation.

In this section we compare agent allocation statistics for the random valuation (that was used in the previous section) to the scenario where valuations are based on the minimum length solutions to traveling salesperson problems. We look at the case where 5 agents participate in auctions for 10 locations, and start and item locations are drawn uniformly at random from a 100 by 100 kilometer square. The TSP-based costs are recalculated for an agent i after rounds in which i wins an item. We run 1000 trials per data-point and plot the mean values from experiments vs. the expected values predicted by our analysis in Figure 10. We limit our focus to the parallel, sequential, and G-Prim Auctions. Figure 10 displays information for each auction and scenario in its own plot. Different auction types appear as columns, and different scenarios appear as rows.

Figure 11 shows both the probability that each particular nonauctioneer takes possession of at least one item, and the resulting summed TSP lengths over all agents for the 5 agent, 10 item, min TSP valuation case. Of note is the non-tightness of the analytical bound for the case of TSP valuation (which is derived from the analysis of the random variable case).

An interesting result from this experiment is that the analytical bounds are looser for the TSP-metric than for the random valuation metric. Additional results are discussed in detail in Section 8.

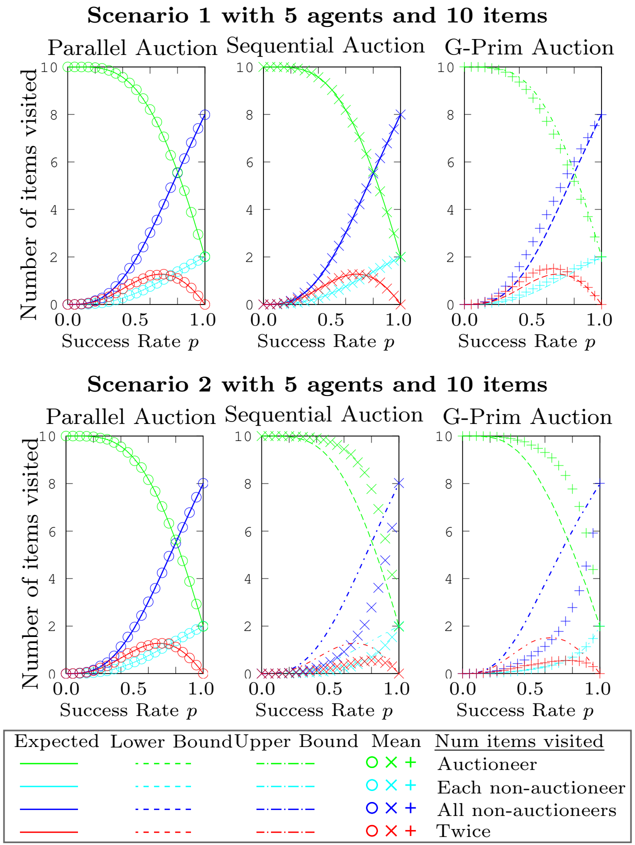

Fig. 10: The average number of items each agent visits over various communication qualities, auctions, and for Scenarios 1 and 2. In Scenario 1 bids are realizations of random variables, in Scenario 2 bids are based on the extra distance required to visit task locations. Items are visited twice if an agent receives an award message from the auctioneer but fails to send an acknowledgment message back to the auctioneer (and so the auctioneer also visits the item).

7.3 Monte Carlo Simulations Evaluating Path Length and Computation Time

In this experiment we evaluate how the various auctions described in the previous section perform when used at different levels of communication channel reliability—using both the Bernoulli communication model as well as the Gilbert Elliott communication models.

In particular, we focus on distance valuation functions that are motivated by multi robot task allocation problems, and consider two different objectives that the auctioneer could attempt to optimize: min-sum TSP lengths over all robots, and min-max TSP lengths over all robots. For each objective (min-sum TSP and min-max TSP) we evaluate the use of both the actual TSP length as well as a commonly used approximation to the TSP length that is much easier to compute, in practice.

- **Actual TSP Valuation**: When using the actual TSP lengths, agents value new tasks by computing the TSP length of visiting all items they currently own plus each additional item up for auction. For min-sum and min-max this involves selling the item(s) that result in the smallest actual min-sum and min-max values, respectively.
- **Approximate TSP Valuation**: The TSP approximation is based on the minimum spanning forest length, where each robot contributes its own tree in the forest. For min-sum this involves selling the item(s) that results in the smallest new edge being added to the spanning forest, while for min-max this involves selling the item that results in the smallest new spanning forest length. Once a spanning forest has been computed, it is possible (with more post processing) to find an approximation to each robot's actual TSP that is no greater than 1.5 times the length of that robots actual TSP over the locations that it owns[10].

[10] Indeed, the original Prim Allocation algorithm leveraged the fact that this approximation method was used for valuations.

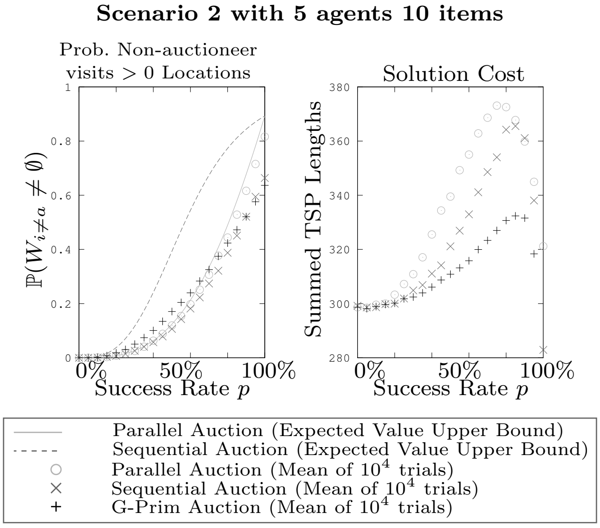

Fig. 11: Left: the probability that a non-auctioneer ($i \neq a$) does at least one task, analytical value vs. results from experiments over various communication qualities for different auction types. Right: Solution quality over various communication qualities.

In Section 7.1 we observed that multi-round auctions had much better performance for variants that resend winner lists each round (verses the basic implementations that do not). Therefore, we limit our experiments in this section to (A) multi-round variants that resend winner lists each round, and (B) single round auctions.

For the Bernoulli communication model a message is dropped based on a weighted coin flip. For the Gilbert-Elliot communication model we assume perfect communication in the good state ($p_g = 0$) and nonexistent communication in the bad state ($p_b = 1$); we fix the expected duration of a communication blackout (100 messages), and then calculate transition probabilities (τ_g and τ_b) that yield the desired expected communication quality that we wish to test. The initial state is determined such that the expectation of starting in each state of the model is equal to its stationary probability.

This experiment shows that the overall ranking of auctions differs depending on the level of communication that exists, the valuation function that is used, and also between computation time and solution quality. Additional discussion is postponed until Section 8.

Results for the four valuation metrics appear in Figures 12-13, respectively. Each figure shows side-by-side results for the Bernoulli model and the Gilbert-Elliot communication model, and each data-point represents the mean result over 50 trials. For these experiments we assume 5 robots that bid to visit 10 points in the environment. Robot start locations and points are chosen from a uniform random distribution in a square search space that spans $[-100, 100] \times [-100, 100]$ kilometers. We assume that robots have enough fuel to complete their missions.

7.4 Hardware Experiments

Hardware experiments involve three agents: two AscTec Pelican quad-rotor UAVs with Odroid single board computers, and an auctioneer with a Dell Precision computer (64GB RAM and an Intel Core i7 processor). The two UAVs receive position measurements from a Vicon motion capture system in an indoor space that spans 8 by 10 meters. We use the ETH-Zurich modular sensor fusion framework for state estimation by Lynen et al (2013). Robot Operating System (ROS) is used on all computers for local inter-process communications. All three agents communicate using the UDP-based Lightweight Communications and Marshaling protocol by Huang et al (2010), over (lossy) 802.11n wireless Ethernet.

We run 10 trials each of the G-Prim auction and the Repeated G-Prim auction (for a total of 20 trials). In both sets of experiments each robot used the TSP valuation metric and the auctioneer (one of the robots) used the min-max objective function. Tests take approximately 2-5 minutes to run including auctions and task execution. Figure 14 depicts actual paths taken by robots as they visit the items they won in two typical auctions. The mean communication quality observed over all experiments was 91.1% (i.e., the message drop rate was 8.9%) and had a standard deviation of 10.9%. The mean min-max path length found by the G-Prim Auction was 23.811 meters (standard deviation 0.253 meters), and by the Repeated G-Prim Auction was 24.0 meters (standard deviation 0.895 meters).

Our Pelican robots are not equipped with sensors that can be used to detect and avoid possible local collisions. Therefore, we instead use NRL's centralized *Puppeteer* framework to ensure that the robots do not collide when following their respective paths. We note that the auction process is completely decentralized and runs over 802.11n wireless Ethernet using UDP across a busy network. In a field deployment in a larger environment it is reasonable to expect that robot-robot encounters would be rare. More sophisticated on-board sensors such as radar, camera, etc. could also be used to prevent local collisions even when communication is poor.

The main result of this experiment is a demonstration that G-Prim and Repeated G-Prim can be used on real UAVs that communicate over a lossy channel.

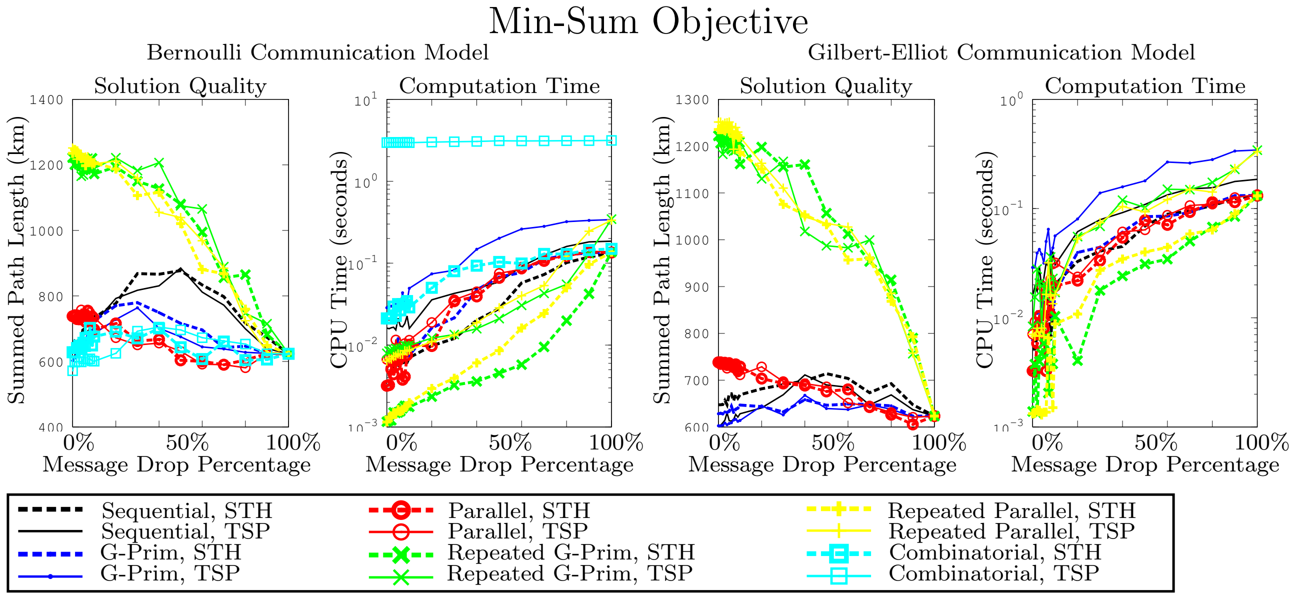

Fig. 12: Using auctions over a lossy communication channel, 5 robots seek to minimize their summed path length (min-sum objective) when visiting 10 target locations. Data-points represent mean results over 50 trials. Different auction methods and valuation functions (spanning tree heuristic and traveling sales person) are compared.

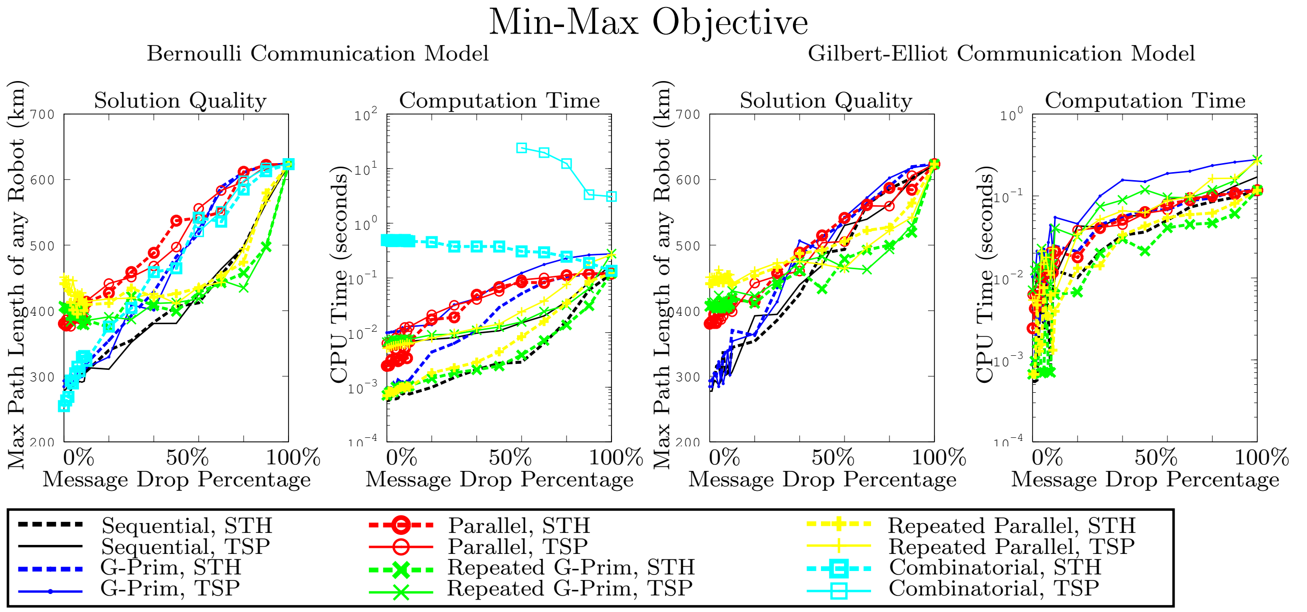

Fig. 13: Using auctions over a lossy communication channel, 5 robots use the min-max objective when dividing the work of visiting 10 target locations. Data-points represent mean results over 50 trials. Different auction methods and valuation functions (spanning tree heuristic and traveling sales person) are compared.

8 Discussion of Results

8.1 G-Prim and Repeated G-Prim

G-Prim, our generalization of the earlier Prim Allocation Algorithm, was found to work very well in practice; both in terms of agent utilization, but also with respect to solution quality. When $p < 1$ in the Bernoulli model, the G-Prim Auction enables more agents to win tasks than either the Sequential or Parallel Auctions. By having each agent i bid for the item that i values most, G-Prim reduces the chances that an item highly valued by i is sold to some other agent in the event that a message to/from i is dropped. G-Prim also tends to result in better solutions overall (Figure 11-Right). The price of these advantages is that the auctioneer must use advertisement messages of size $\mathcal{O}(m)$ each round, instead

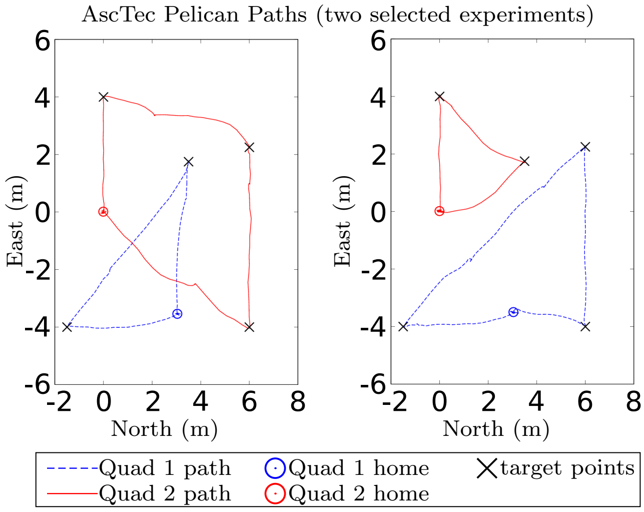

Fig. 14: The paths taken by two AscTec Pelican quad-rotors while visiting five target points. Left and Right show two different experiments. The left solution was the result of the G-Prim auction, the one on the right was the result of a Repeated G-Prim Auction. Observed communication quality was above 90% in both experiments.

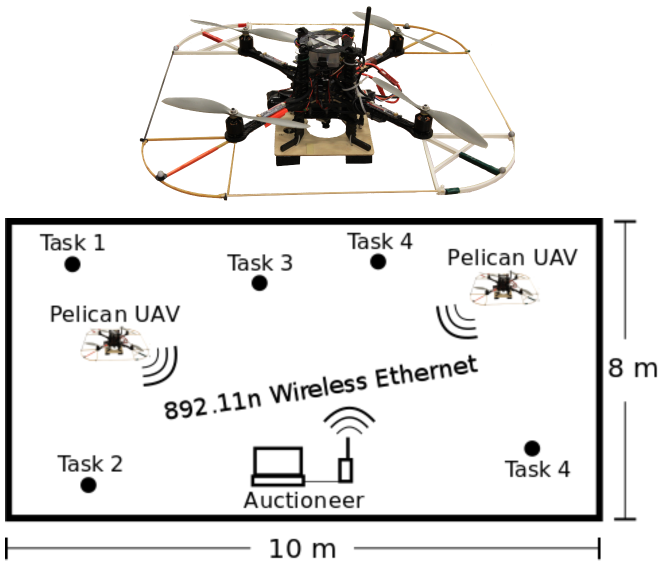

Fig. 15: Top:AscTec Pelican quad-rotors we use in our hardware experiments. Bottom: Experimental Setup with three agents (two AscTec Pelican quad-rotor agents and laptop-based auctioneer) using 802.11n wireless Ethernet.

of the $\mathcal{O}(1)$ sized messages used by the Sequential Auction. Bidders retain the same message sizes in G-Prim as they do in the Sequential Auction.

We believe that the G-Prim Auction may have general applications well beyond multi-robot task allocation. Indeed, the bidding mechanism is straightforward—each agent bids for the item that it wants the most each round, and in each round the auctioneer awards the item with that round's best bid to the agent that bid for it during that round.

The relationship between the Sequential Auction and the G-Prim Auction is somewhat analogous to the relationship between the Repeated Parallel Auction and the Repeated G-Prim Auction. In the repeated auctions each agent that manages to bid is awarded with an item; however, by enabling agents to submit bids for the n items that they most desire, the auctioneer is more likely to award agents items that the agents believe are more beneficial than other remaining options (except during the last round).

8.2 Communication loss affects auctions

The most obvious result from this work is that dropped communication significantly affects the performance of auctions. Moreover, dropped communication affects different auctions differently. While total communication failure results in the auctioneer taking responsibility of all items, partial breakdown effects different auctions in different ways. Indeed, given a particular set of items, valuation function, and objective function, auctions that have similar performance when communication is perfect may have very different performance when the communication between the auctioneers and the bidders partially breaks down.

8.3 Re-sending winners in multi-round auctions

For multi-round auctions we find that re-sending winners of earlier rounds along with the current round's winners is an easy way to improve performance across all multi-round auctions with respect to agent utilization and solution quality. While this result is arguably intuitive, the amount of performance improvement increased with both the number of items and the number of agents. Re-sending winner data does not increase the number of messages that must be sent, but it does increase the size of the winner announcement messages to $\mathcal{O}(m)$, which is on par with other message sizes used for the Parallel, G-Prim, and Repeated G-Prim Auctions, but larger than the $\mathcal{O}(1)$ messages used in the Sequential Auction and the $\mathcal{O}(m/n)$ sized messages used in the Repeated Parallel Auction.

This result is also evident in the repeated auctions Repeated Parallel and Repeated G-Prim) that have more than a single round but fewer rounds than the Sequential and G-Prim Auctions.

8.4 Repeated Auctions

A notable difference between the Repeated Auctions and other auctions is that Repeated Auctions award an item to *each* agent that manages to bid in a particular round (as long as the number of unsold items is greater than the number of agents). Overall, this causes more agents to be used in the task allocation (for any communication level above nonexistent) but also causes solution quality to degrade slightly vs. otherwise similar multi-round auctions. This slight performance degradation tends to vanish as the number of agents and/or items increases.

8.5 Communication models

The Bernoulli communication model is very simple and only scratches the surface of the various ways that communication may be limited in practice. Nonetheless, we believe that it represents an interesting case where the chance of messages being sent between auctioneer and bidders are fixed and i.i.d. Moreover, the straightforward analysis of the Bernoulli model enables us to understand how the underlying dynamics of different auctions break down in different ways communication between agents decreases.

Our results comparing the TSP metrics differ somewhat depending on if the Bernoulli model or the Gilbert-Elliot communication model is used. In general, the rate of packet drop affects different algorithms in different ways. The transition points (where the ranking order of the various algorithms in terms of quality changes) happen at a lower message drop rate if the Bernoulli model is used than if the Gilbert-Elliot model is used.

Regardless of the auction algorithm and communication model that is used, the average time required to solve an auction increases as communication gets worse. This is likely due to a number of reasons, including a more time consuming valuation function and route-planning that must be run on the auctioneer after it assumed responsibility for more items. All algorithms experience the increase in mean CPU time at lower drop rates when the Gilbert-Elliot model is used than when the Bernoulli model is used.

8.6 Analytical Solutions and Numerical Limits

Due to numerical limits of computers, the analytical solutions that we derive for expected agent utilization can only provide accurate calculations, in practice, for limited numbers of agents and items. For the closed form solutions this is due to the appearance of the binomial coefficient within the solution. For the numerical solutions for which we provide pseudocode, another limiting factor is the computational complexity of the resulting calculations.

Despite their limits, we believe that both analytical solutions and numerical limits are useful for providing intuition about how agent utilization may be affected by communication.

The probability that the auctioneer does any task depends on both the number of items it wins, as well as the number of items won by other agents that are unacknowledged by those agents. While it is possible to numerically calculate these quantities, the calculations are more involved than those included here. We note that the auctioneer is responsible for running the auction, regardless of how many items it eventually takes possession of.

8.7 Auctions when TSP metrics are used

The poor performance of the Parallel Auction when communication is good is due to the fact that all agents must bid on all items without accounting for any symbiosis between them. This is a known phenomenon in auctions, in general. The fact that the parallel algorithm improves when communication is poor (at least for the min-sum objective) is due to the artifact that we have a low item to agent ratio. If the min-sum objective is used, then the "start-up" overhead of using an additional agent is exaggerated in such cases because the agent usually needs to travel a relatively great distance to get to any item (this does not happen with the min-max objective because there is no start-up overhead of using an additional agent as long as its path to an item and back is less than the longest tour planned by any robot currently in use). We conjecture that this effect would become less pronounced if there were many more items up for auction.

In the case of the min-max objective, the Repeated Parallel Auctions perform the best when communication is poor. We attribute this to the fact that there are multiple opportunities to have a successful communication, and so the chances that the auctioneer is stuck visiting all the target points itself is reduced vs. the Parallel Auction. Moreover, since there are multiple rounds, it is possible for the bids in later rounds to account for symbiosis with items won in earlier rounds. The repeated Sequential Auctions also require relatively little computation time (an order of magnitude less than the other algorithms).

The "best" auction algorithm depends both on the objective function and the communication quality, but is qualitatively consistent between the different messaging models. If very high loss rates are observed when the min-max objective function is used (e.g., greater than 50%) then we find that the Repeated G-Prim Auctions has the best performance.

8.8 Note on Parallel Auction With Spanning Tree TSP Estimate

When the Parallel Auction is used with the spanning forest approximation to the TSP cost function, then all agents that submit bids are awarded items in the Voronoi cell of a Voronoi graph created from the locations of the bidding agents. This happens because if no items are owned to begin with then an agent's spanning tree is a star graph rooted at the agent, and the TSP estimate cost function for each item is simply the distance from that agent to an item. No items are owned prior to round 1 in a Parallel Auction, and all items are sold in a single round. Thus all bidding agents are awarded the items that are closer to themselves than to any other bidding agent.

8.9 TSP Valuation vs. Their Approximations

A very obvious (and intuitive) trend across all experiments, regardless of communication quality, is that using TSP-based valuation consistently outperforms using valuation based on the spanning-tree heuristic with the same type of auction. That said, the TSP valuation takes longer to calculate, and will become prohibitively expensive long before that spanning-tree valuation does. The relationship between TSP and spanning tree based valuation is to be expected.

9 Conclusion

We evaluate the performance of a variety of auctions for multi-robot task allocation in scenarios where some of the messages between the auctioneer and the bidders may be dropped.

We study six auctions. Three of the auctions we study have been widely used in the past, and include: the Parallel Auction, the Sequential Auction, and the Combinatorial Auction. Another, the G-Prim Auction, is based on a generalization of Prim Allocation. Finally, the remaining two are different variants of an idea in which every agent that bids in a round wins something; we call these Repeated Parallel and Repeated G-Prim, respectively, depending on the way that items up for bid each round are chosen. Finally, variations of the multi-round auctions (Sequential, G-Prim, Repeated Parallel, and Repeated G-Prim) are also tested while using a simple modification in which the auctioneer includes the winners of previous rounds whenever it broadcasts new winners in subsequent rounds.

Assuming a Bernoulli communication model, we derive closed-form solutions for the expected agent utilization in each of the Sequential, Parallel Auctions, and Repeated Parallel Auctions, and bound the performance of G-Prim and Repeated G-Prim in terms of the Sequential Auction.

In simulations and experiments we consider the performance of these auctions in two different Scenarios, and using variety of different valuation/cost functions and approximations. The first scenario involves minimizing or maximizing the value of items sold, where item values are random variables, depending on if bids represent the cost of undertaking a particular task or the value they will receive from doing a particular task, respectively. The second scenario assumes that items are randomly drawn locations, and defines cost as the extra distance required to visit a location. In the second scenario we consider both the min-sum and min-max formulations of the problems (handling the cases where agents seek to minimize total distance traveled over all agents, and agents seek to minimize the furthest distance traveled by any agent, respectively). We test two different ways of calculating/estimating this resulting costs: in the first, a computationally expensive "optimal" TSP-based valuation is used, and in the second a spanning-tree based heuristic approximation is used.

We find that the G-Prim Auction outperforms the other methods with respect to agent utilization across a wide variety of scenarios, and has more optimal solution cost (or value) assuming item valuations are random variables. Results with respect to the TSP based solution quality were mixed, with different auctions performing better or worse at different points along the communication spectrum. That said, more work is necessary to characterize the performance of the TSP based methods for larger numbers of agents and items; as the small number of agents and items evaluated in the TSP solution length comparison may have been too small to illustrate the differences that were seen for larger numbers of agents and items in the random valuation case.

Re-sending winners in subsequent rounds was found to be an easy way to increase the performance of all multi-round auctions. In general, single-round auctions were found to have much worse performance than the multi-round auctions in communication limited scenarios. The main reason for this is that agents that happen to miss the one and only round of bidding must sit idle.

This is a particularly notable for the single round Combinatorial Auction, which provides the best allocation of items to those agents that manage to bid in the general case (multi-round auctions cannot, in general, find an optimal solution in the case that item valuations are dependent on whether or not a bidder wins other items). Multi-round auctions are able to spread the negative effects of lost communication more evenly across different agents, so each agent has a higher probability of eventually submitting bids and winning tasks. Multi-round auctions are also naturally suited to resending winner data across multiple rounds without significantly increasing the number of messages that must be sent.

Acknowledgments

This work was performed at the Naval Research Laboratory and was funded by the Office of Naval Research under grant numbers N0001416WX01271 and N0001416WX01272. The views, positions and conclusions expressed herein reflect only the authors' opinions and expressly do not reflect those of the Office of Naval Research, nor those of the Naval Research Laboratory.

References

Alighanbari M, How JP (2005) Decentralized task assignment for unmanned aerial vehicles. In: Conference on Decision and Control, IEEE, pp 5668–5673, DOI 10.1109/CDC.2005.1583066

Andersson A, Tenhunen M, Ygge F (2000) Integer programming for combinatorial auction winner determination. In: MultiAgent Systems, 2000. Proceedings. Fourth International Conference on, IEEE, pp 39–46

Beard RW, McLain TW (2003) Multiple UAV cooperative search under collision avoidance and limited range communication constraints. In: Conference on Decision and Control, IEEE, vol 1, pp 25–30

Beard RW, Stepanyan V (2003) Information consensus in distributed multiple vehicle coordinated control. In: Conference on Decision and Control, IEEE, vol 2, pp 2029–2034 Vol.2

Berhault M, Huang H, Keskinocak P, Koenig S, Elmaghraby W, Griffin P, Kleywegt A (2003) Robot exploration with combinatorial auctions. In: International Conference on Intelligent Robots and Systems, IEEE/RSJ, vol 2, pp 1957–1962

Bertsekas DP, Castañon DA (1991) Parallel synchronous and asynchronous implementations of the auction algorithm. Parallel Computing 17(6):707–732

Bertsekas DP, Castañon DA (1993) Parallel asynchronous hungarian methods for the assignment problem. ORSA Journal on Computing 5(3):261–274

Botelho SC, Alami R (1999) M+: a scheme for multirobot cooperation through negotiated task allocation and achievement. In: International Conference on Robotics and Automation, IEEE, vol 2, pp 1234–1239

Caloud P, Choi W, Latombe JC, Le Pape C, Yim M (1990) Indoor automation with many mobile robots. In: International Conference on Intelligent Robots and Systems, IEEE/RSJ, pp 67–72

Castanon DA, Wu C (2003) Distributed algorithms for dynamic reassignment. In: Conference on Decision and Control, vol 1, pp 13–18 Vol.1, DOI 10.1109/CDC.2003.1272528

Castelpietra C, Iocchi L, Nardi D, Piaggio M, Scalzo A, Sgorbissa A (2001) Communication and Coordination among heterogeneous Mid-Size Players: ART99, Springer Berlin Heidelberg, Berlin, Heidelberg, pp 86–95. DOI 10.1007/3-540-45324-5_7

Cavalcante RC, Noronha TF, Chaimowicz L (2013) Improving combinatorial auctions for multi-robot exploration. In: Advanced Robotics (ICAR), 2013 16th International Conference on, pp 1–6, DOI 10.1109/ICAR.2013.6766508

Chandler P, Pachter M (2001) Hierarchical control for autonomous teams. In: Guidance, Navigation, and Control Conference, AIAA, pp 632–642

Choi HL, Brunet L, How JP (2009) Consensus-based decentralized auctions for robust task allocation. IEEE Transactions on Robotics 25(4):912–926

Christofides N (1976) Worst-case analysis of a new heuristic for the travelling salesman problem. Technical Report 388, Graduate School of Industrial Administration, Carnegie Mellon University

De Vries S, Vohra RV (2003) Combinatorial auctions: A survey. INFORMS Journal on computing 15(3):284–309

Dias MB, Stentz A (2000) A free market architecture for distributed control of a multirobot system. In: 6th International Conference on Intelligent Autonomous Systems, pp 115–122

Dias MB, Zinck M, Zlot R, Stentz A (2004) Robust multirobot coordination in dynamic environments. In: International Conference on Robotics and Automation, IEEE, vol 4, pp 3435–3442

Dias MB, Zlot R, Kalra N, Stentz A (2006) Market-based multirobot coordination: A survey and analysis. Proceedings of the IEEE 94(7):1257–1270

Dionne D, Rabbath CA (2007) Multi-UAV decentralized task allocation with intermittent communications: the dtc algorithm. In: American Control

Conference, pp 5406–5411, DOI 10.1109/ACC.2007.4282637

Elliott EO (1963) Estimates of error rates for codes on burst-noise channels. The Bell System Technical Journal 42(5):1977–1997

Gerkey BP, Matarić MJ (2001) Principled communication for dynamic multi-robot task allocation. In: Experimental Robotics VII, Springer, pp 353–362

Gerkey BP, Mataric MJ (2002) Sold!: Auction methods for multirobot coordination. IEEE Transactions on Robotics and Automation 18(5):758–768

Guerrero J, Oliver G (2003) Multi-robot task allocation strategies using auction-like mechanisms. Artificial Research and Development in Frontiers in Artificial Intelligence and Applications 100:111—122

Hoeing M, Dasgupta P, Petrov P, O'Hara S (2007) Auction-based multi-robot task allocation in comstar. In: Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS '07, pp 280:1–280:8

Huang AS, Olson E, Moore DC (2010) Lcm: Lightweight communications and marshalling. In: International Conference on Intelligent Robots and Systems, IEEE/RSJ, pp 4057–4062

Hunsberger L, Grosz BJ (2000) A combinatorial auction for collaborative planning. In: MultiAgent Systems, 2000. Proceedings. Fourth International Conference on, IEEE, pp 151–158

Koenig S, Keskinocak P, Tovey CA (2010) Progress on agent coordination with cooperative auctions. In: AAAI, vol 10, pp 1713–1717

Lagoudakis MG, Berhault M, Koenig S, Keskinocak P, Kleywegt AJ (2004) Simple auctions with performance guarantees for multi-robot task allocation. In: International Conference on Intelligent Robots and Systems, IEEE/RSJ, vol 1, pp 698–705 vol.1

Lagoudakis MG, Markakis E, Kempe D, Keskinocak P, Kleywegt AJ, Koenig S, Tovey CA, Meyerson A, Jain S (2005) Auction-based multi-robot routing. In: Robotics Science and Systems

Lynen S, Achtelik MW, Weiss S, Chli M, Siegwart R (2013) A robust and modular multi-sensor fusion approach applied to mav navigation. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp 3923–3929

Mataric MJ, Sukhatme GS (2001) Task-allocation and coordination of multiple robots for planetary exploration. In: International Conference on Advanced Robotics

Moore BJ, Passino KM (2004) Coping with information delays in the assignment of mobile agents to stationary tasks. In: Conference on Decision and Control, IEEE

Nanjanath M, Gini M (2010) Repeated auctions for robust task execution by a robot team. Robotics and Autonomous Systems 58(7):900–909

Otte M (2018) An emergent group mind across a swarm of robots: Collective cognition and distributed sensing via a shared wireless neural network. The International Journal of Robotics Research 37(9):1017–1061, DOI 10.1177/0278364918779704

Otte M, Kuhlman M, Sofge D (2017a) Competitive target search with multi-agent teams: symmetric and asymmetric communication constraints. Autonomous Robots DOI 10.1007/s10514-017-9687-0, URL `https://doi.org/10.1007/s10514-017-9687-0`

Otte M, Kuhlman M, Sofge D (2017b) Multi-robot task allocation with auctions in harsh communication environments. In: International Symposium on Multi-Robot and Multi-Agent Systems, Los Angeles

Parker LE (1998) Alliance: An architecture for fault tolerant multirobot cooperation. IEEE Transactions on Robotics and Automation 14(2):220–240

Parkes DC, Ungar LH (2000) Iterative combinatorial auctions: Theory and practice. In: AAAI

Pippin C, Christensen H (2011) A bayesian formulation for auction-based task allocation in heterogeneous multi-agent teams. In: Proceedings of the SPIE

Rekleitis I, New AP, Rankin ES, Choset H (2008) Efficient boustrophedon multi-robot coverage: an algorithmic approach. Annals of Mathematics and Artificial Intelligence 52(2):109–142, DOI 10.1007/s10472-009-9120-2

Sandholm T (2002) Algorithm for optimal winner determination in combinatorial auctions. Artificial intelligence 135(1):1–54

Sariel S, Balch TR (2006) Efficient bids on task allocation for multi-robot exploration.

Schneider E, Balas O, Ozgelen AT, Sklar EI, Parsons S (2014) Evaluating auction-based task allocation in multi-robot teams. In: AAMAS Workshop on Autonomous Robots and Multirobot Systems (ARMS)

Schneider E, Sklar EI, Parsons S, Özgelen AT (2015) Auction-Based Task Allocation for Multi-robot Teams in Dynamic Environments, Springer International Publishing, Cham, pp 246–257

Simmons R, Apfelbaum D, Burgard W, Fox D, Moors M, Thrun S, Younes H (2000) Coordination for multi-robot exploration and mapping pp 852–858

Smith R (1980) Communication and control in problem solver. IEEE Transactions on computers 29(12):1104–1113

Stone P, Veloso M (1998) Communication in domains with unreliable, single-channel, low-bandwidth communication. In: Collective Robotics, Springer, pp 85–

97

Trawny N, Roumeliotis SI, Giannakis GB (2009) Cooperative multi-robot localization under communication constraints. In: International Conference on Robotics and Automation, IEEE, pp 4394–4400

Vail D, Veloso M (2003) Dynamic multi-robot coordination. In: In Multi-Robot Systems: From Swarms to Intelligent Automata, Volume II, Kluwer Academic Publishers, pp 87–100

Wei C, Hindriks KV, Jonker CM (2015) Auction-Based Dynamic Task Allocation for Foraging with a Cooperative Robot Team, Springer International Publishing, Cham, pp 159–174. DOI 10.1007/978-3-319-17130-2_11

Zlot R, Stentz A, Dias MB, Thayer S (2002) Multi-robot exploration controlled by a market economy. In: International Conference on Robotics and Automation, IEEE, vol 3, pp 3016–3023

Zurel E, Nisan N (2001) An efficient approximate allocation algorithm for combinatorial auctions. In: Proceedings of the 3rd ACM conference on Electronic Commerce, ACM, pp 125–136

A Iterative Calculations for Repeated Parallel Auction with re-sends in Scenario 1

In this section we present the iterative calculation of quantities for the Repeated Parallel Auction with re-sends in Scenario 1, assuming a Bernoulli communication model. Intermediate quantities are denoted with capital letters, and variations of them.

B Additional Auction Performance Curves for Scenario 1

In this section we present additional figures from the first series of experiments. Each figure in this section pertains to a single auction, and shows how agent utilization changes over various swarm sizes and item counts.

Algorithm 13 precomputation one: iterative computation of $\grave{C}[:][:]$ given p, q, n, and m.

$\grave{A}[1:m+1][1:m+1][1:m+1] \leftarrow 0$
// $\grave{A}[r][i][x]$ is the number of scenarios that end during round x
// assuming r items are sold by round j
$\grave{B}[1:m+1][1:m+1][1:m+1] \leftarrow 0$
// $\grave{B}[r][j][x] \equiv \mathbb{P}$(r items sold by round j starting at round x)
$\grave{C}[1:m+1][1:m+1] \leftarrow 0$
// $\grave{C}[x][r] \equiv \mathbb{P}$(auction lasts to round x, assuming it is already
// at round r)
$\grave{A}[1][1][:] \leftarrow 1$
$\grave{B}[1][1][:] \leftarrow 1$
$\grave{C}[1][1] \leftarrow 1$
for $x \in \{0 \ldots m\}$ **do**
  for $r \in \{1 \ldots x\}$ **do**
    for $i \in \{1 \ldots x+1\}$ **do**
      if $\grave{A}[r][i][x+1] > 0$ **then**
        for $k \in \{1 \ldots n\}$ **do**
          $\grave{D} \leftarrow p^{2(k-1)}(pq+q)^{n-k} * \binom{n-1}{k-1}$
          // $\mathbb{P}$(scenario to scenario transition)
          if $i = j$ **then**
            continue
          else if $i + k \leq x + 1$ **then**
            $j \leftarrow i + k$
            $\grave{A}[r+1][j][x+1] \leftarrow \grave{A}[r+1][j][x+1] + \grave{A}[r][i][x+1]$
          else
            $j \leftarrow x + 1$
            $\grave{B}[r+1][j][x+1] \leftarrow \grave{B}[r+1][j][x+1] + \grave{D}\grave{B}[r][i][x+1]$
  $\grave{C}[x+1][:] \leftarrow \grave{B}[:][x+1][x+1]$

Algorithm 14 precomputation two: iterative computation of $\bar{A}[:]$ given p, q, n, and m.

$\bar{A}[1:m+1]$
for $x \leftarrow \{m \ldots 0\}$ **do**
  $j \leftarrow x+1$
  $\bar{C} \leftarrow 0$; $\bar{N} \leftarrow 0$; $\bar{D} \leftarrow 0$
  for $z \in \{x \ldots 0\}$ **do**
    $\bar{F} \leftarrow \bar{C}$
    // $\mathbb{E}$(num items adopted by end of round if award message not received)
    $\bar{G} \leftarrow 1 - \bar{F}$
    // $\mathbb{E}$(num items not adopted by end of round if award message not received)
    $\bar{C} \leftarrow p + q\bar{F}$
    // $\mathbb{E}$(num items adopted by end of round if award message not received)
    $\bar{H} \leftarrow \bar{D} + \bar{G}$
    // $\mathbb{E}$(num items adopted but not acked at middle of round if award message received)
    $\bar{I} \leftarrow p\bar{H}$
    // $\mathbb{E}$(num items first acked this round if award message received)
    $\bar{J} \leftarrow \bar{I} + \bar{N}$
    // $\mathbb{E}$(num items adopted and acked by end of round if award message received)
    $\bar{K} \leftarrow \bar{N}$
    // $\mathbb{E}$(num items adopted and acked by end of round if award message not received)
    $\bar{L} \leftarrow 1 - \bar{J}$
    // $\mathbb{E}$(num items adopted and not acked by end of round if award message received)
    $\bar{M} \leftarrow \bar{F} - \bar{K}$
    // $\mathbb{E}$(num items adopted and not acked by end of round if award message not received)
    $\bar{N} \leftarrow p\bar{J} + q\bar{K}$
    // $\mathbb{E}$(num items acked by end of round)
    $\bar{D} \leftarrow p\bar{L} + q\bar{M}$
    // $\mathbb{E}$(num items adopted and not acked by end of round)
  $\bar{A}[j] \leftarrow \bar{N}$
  // $\mathbb{E}$(num items sold that are eventually acked, given j rounds remain)

Algorithm 15 Finding $\mathbb{P}_{\text{RPNImp}}(W_{i \neq a} \neq \emptyset)$, $\breve{C}[:][:][:]$, $\breve{D}[:][:][:]$ and $\breve{D}[:][:][:]$ given p, q, n, and m.

$\breve{A}[1:m+1][1:m+1] \leftarrow 0$
  // $\breve{A}[r][i]$is number of scenarios that end at round r when // i items are sold in round x)
$\breve{B}[1:m+1][1:m+1] \leftarrow 0$
  // $\breve{B}[r][i] \equiv \mathbb{P}(i$ items are sold by the end of round $r)$
$\breve{C}[1:m+1][1:m+1][1:m+1] \leftarrow 0$
  // $\breve{C}[r][i][x] \equiv \mathbb{P}$(selling item i in round r when x remain)
$\breve{D}[1:m+1][1:m+1][1:m+1] \leftarrow 0$
  // $\breve{D}[r][i][x] \equiv \mathbb{P}$(selling item i to auctioneer in round r when // x items remain)
$\breve{E}[1:m+1][1:m+1][1:m+1] \leftarrow 0$
  // $\breve{E}[r][i][x] \equiv \mathbb{P}$(selling item i to nonauctioneer in round r // when x items remain)
$\breve{F}[1:m+1][1:m+1] \leftarrow 0$
  // $\mathbb{P}$(nonauctioneer has not won by end of round r when // x items remain)
$\breve{G}[1:m+1][1:m+1] \leftarrow 0$
  // $\mathbb{P}$(nonauctioneer has won but not adopted by end of // round r when x items remain)
$\breve{A}[1][1] \leftarrow 1$
$\breve{B}[1][1] \leftarrow 1$
$\breve{F}[1][1] \leftarrow 1$
for $x \in \{1 \ldots m\}$ **do**
  $r \leftarrow x+1$
  for $last \in \{0 \ldots m-1\}$ **do**
    $i \leftarrow last + 1$;
    if $\breve{A}[r-1][i] > 0$ **then**
      for $num\_received \in \{1 \ldots n\}$ **do**
        $k \leftarrow num\_received$
        $\breve{H} \leftarrow p^{2(k-1)}(pq+q)^{n-k}\binom{n-1}{k-1}$
        // $\mathbb{P}$(scenario to scenario transition)
        $first\_item \leftarrow last + 1$
        $f \leftarrow first\_item + 1$
        $\breve{I} \leftarrow (k-1)/(n-1)$
        // $\mathbb{P}$(particular nonauctioneer bid)
        $\breve{J} \leftarrow 1 - \breve{I}$
        // $\mathbb{P}$(particular nonauctioneer did not bid)
        if $last + num\_received \leq m$ **then**
          $last\_item \leftarrow last + num\_received$
          $l \leftarrow last\_item + 1$
          $\breve{A}[r][l] += \breve{A}[r-1][i]$
          $\breve{K} \leftarrow 1$
          // $\mathbb{P}$(particular nonauctioneer wins, if it bid)
        else
          $last\_item \leftarrow m$
          $l \leftarrow last\_item + 1$
          $\breve{K} \leftarrow (last\_item - first\_item + 1)/k$
          // $\mathbb{P}$(particular nonauctioneer wins, if it bid)
        $\breve{B}[r][l] += \breve{H}\breve{B}[r-1][i]$
        $\breve{Q} \leftarrow 1 - \breve{K}$
        // $\mathbb{P}$(particular nonauctioneer wins, if it bid)
        $\breve{R} \leftarrow \breve{J} + \breve{I}\breve{Q}$
        // $\mathbb{P}$(particular nonauctioneer does not win, // if it bid)
        $\breve{F}[r][l] += \breve{H}\breve{R}\breve{F}[r-1][i]$
        $\breve{L} \leftarrow (1-\breve{R})\breve{F}[(][r]-1,i)$
        // $\mathbb{P}$(particular nonauctioneer wins its first item)
        $\breve{G}[r][l] += \breve{H}q(\breve{L} + \breve{G}[r-1][i])$
        $\breve{M} \leftarrow m - last\_item$ // rounds remaining
        $x \leftarrow \breve{M} + 1$
        $\breve{C}[r][f:l][x] += \breve{H}\breve{B}[r-1][i]$
        $\breve{D}[r][f:l][x] += (1/k)\breve{H}\breve{B}[r-1][i]$
        if $num\_received > 1$ **then**
          $\breve{E}[r][f:l][x] += (1-1/k)/(n-1)\breve{H}\breve{B}[r-1][i]$

$\breve{N} \leftarrow sum(\breve{F}[:][end])$ // $\mathbb{P}$(nonauctioneer never wins)
$\mathbb{P}_{\text{RPNImp}}(W_{i \neq a} \neq \emptyset) \leftarrow sum(\breve{G}[:][end]) + \breve{N}$

Algorithm 16 Iterative calculation of $\mathbb{E}\left(|W_a|\right)$ for RPI given p, q, n, m, $\grave{C}[:][:]$, $\bar{A}[:]$, $\breve{C}[:][:][:]$, $\breve{D}[:][:][:]$ and $\breve{D}[:][:][:]$

$\widehat{A} \leftarrow 0$; $\widehat{B} \leftarrow 0$; $\widehat{C} \leftarrow 0$; $\widehat{D} \leftarrow 0$; $\widehat{E} \leftarrow 0$;
for *item* $\in \{1 \dots m\}$ **do**
    $i \leftarrow item + 1$
    $\widehat{F} = 0$
    for *round* $\in \{1 \dots m\}$ **do**
        $r \leftarrow round + 1$
        for *remain* $\in \{0 \dots m\}$ **do**
            $x \leftarrow remain + 1$
            for *rounds_left* $\in \{0 \dots m\}$ **do**
                $j \leftarrow rounds\_left + 1$
                $\widehat{A} \mathrel{+}= \breve{C}[r][i][x]\grave{C}[x][j]$
                // $\mathbb{E}$(num items sold)
                $\widehat{B} \mathrel{+}= \breve{D}[r][i][x]\grave{C}[x][j]$
                // $\mathbb{E}$(num items won by auctioneer)
                $\widehat{C} \mathrel{+}= \breve{E}[r][i][x]\grave{C}[x][j]$
                // $\mathbb{E}$(num items won by nonauctioneer)
                $\widehat{F} \mathrel{+}= (1 - q^j)\breve{E}[r][i][x]\grave{C}[x][j]$
                // $\mathbb{P}$(sold to nonauctioneer and adopted)
                $\widehat{E} \mathrel{+}= \bar{A}[j]\breve{E}[r][i][x]\grave{C}[x][j]$
                // $\mathbb{E}$(num items adopted and acked by
                // nonauctioneer)
        $\widehat{D} \mathrel{+}= \widehat{F}$
        // $\mathbb{E}$(num items awarded to each nonauctioneer)
        $\widehat{H} \mathrel{*}= (1 - \widehat{F})$
        // $\mathbb{P}$(particular nonauctioneer does not participate)
$\mathbb{E}\left(|W_{i \neq a}|\right) \leftarrow \widehat{D}$
$\widehat{J} \leftarrow \widehat{D}\,(n-1)$
// $\mathbb{E}$(num items visited by all nonauctioneers)
$\widehat{I} \leftarrow (\widehat{D} - \widehat{E})(n-1)$
// $\mathbb{E}$(expected num items visited twice)
$\mathbb{E}\left(|W_a|\right) \leftarrow \widehat{B} + (\widehat{C} - \widehat{D})(n-1) + \widehat{I}$

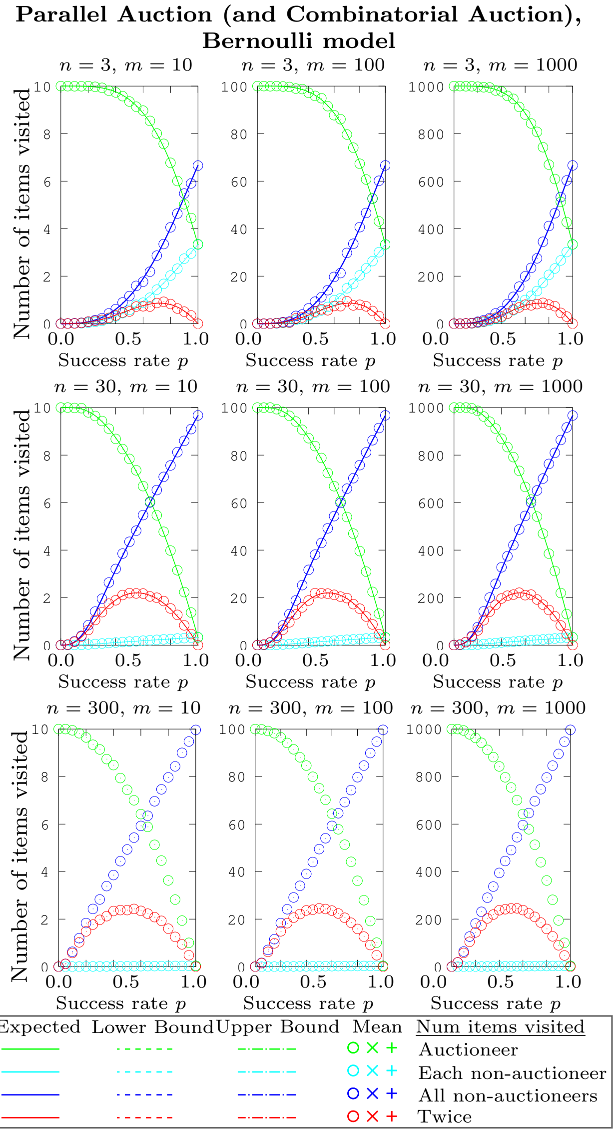

Fig. 16: Parallel Auction Agent Utilization Performance Curves. The average number of items each agent visits over various communication qualities in Scenario 1 (in which bids are realizations of random variables), assuming a Bernoulli communication model. Items are visited twice if an agent receives an award message from the auctioneer but fails to send an acknowledgment message back to the auctioneer (and so the auctioneer also visits the item). In the special case that no symbiosis exist between items (which happens in Scenario 1) the performance of the Parallel Auction and the **Combinatorial Auction** are identical.

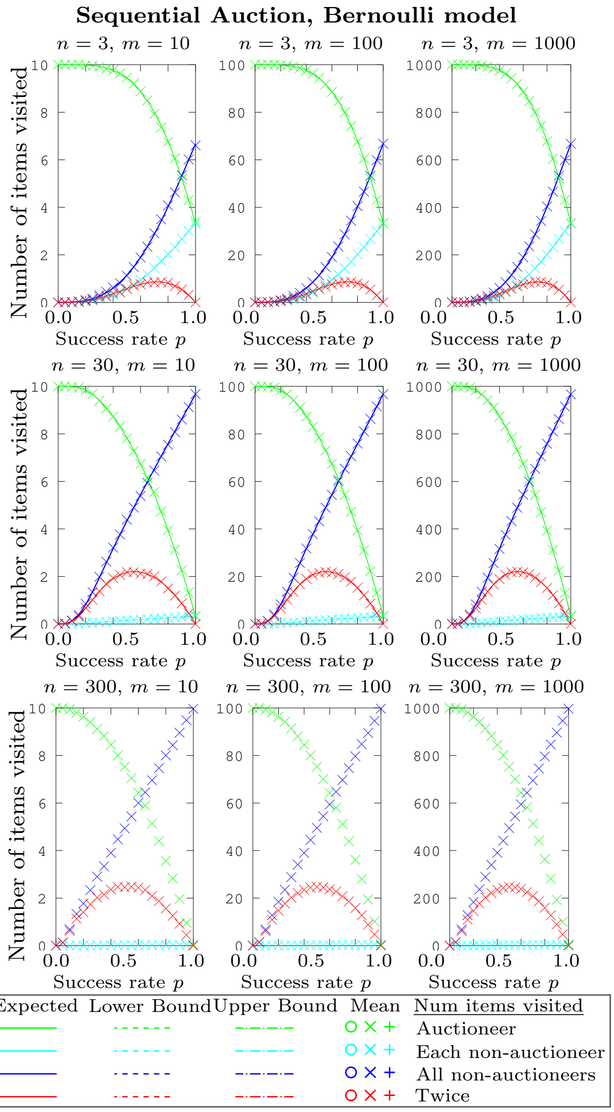

Fig. 17: Sequential Auction Agent Utilization Performance Curves. The average number of items each agent visits over various communication qualities in Scenario 1 (in which bids are realizations of random variables), assuming a Bernoulli communication model. Items are visited twice if an agent receives an award message from the auctioneer but fails to send an acknowledgment message back to the auctioneer (and so the auctioneer also visits the item).

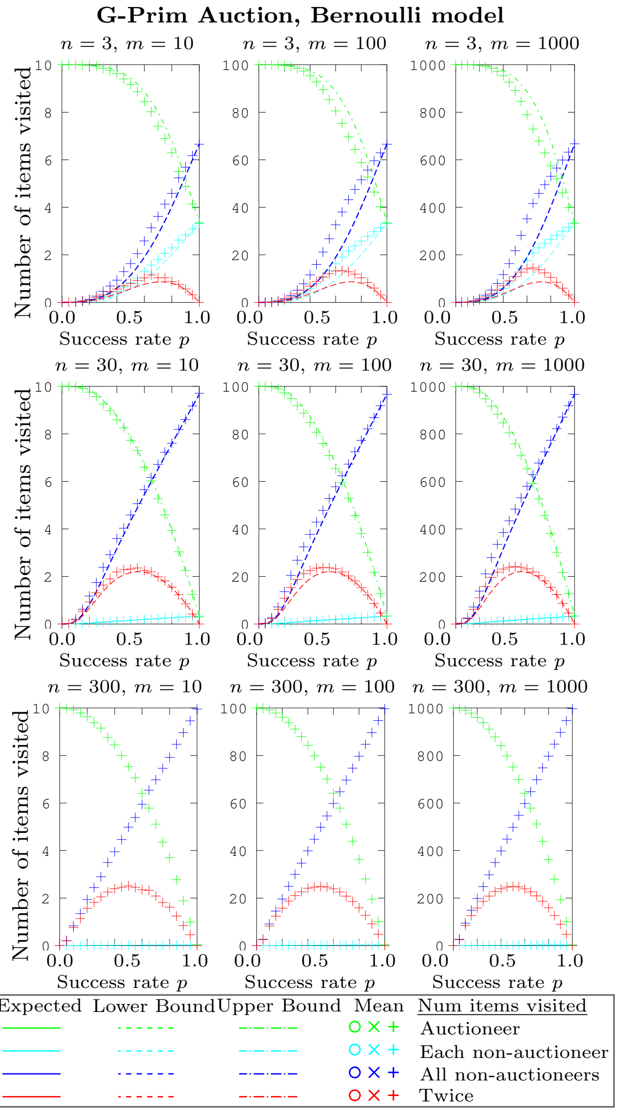

Fig. 18: G-Prim Auction Agent Utilization Performance Curves. The average number of items each agent visits over various communication qualities in Scenario 1 (in which bids are realizations of random variables), assuming a Bernoulli communication model. Items are visited twice if an agent receives an award message from the auctioneer but fails to send an acknowledgment message back to the auctioneer (and so the auctioneer also visits the item).

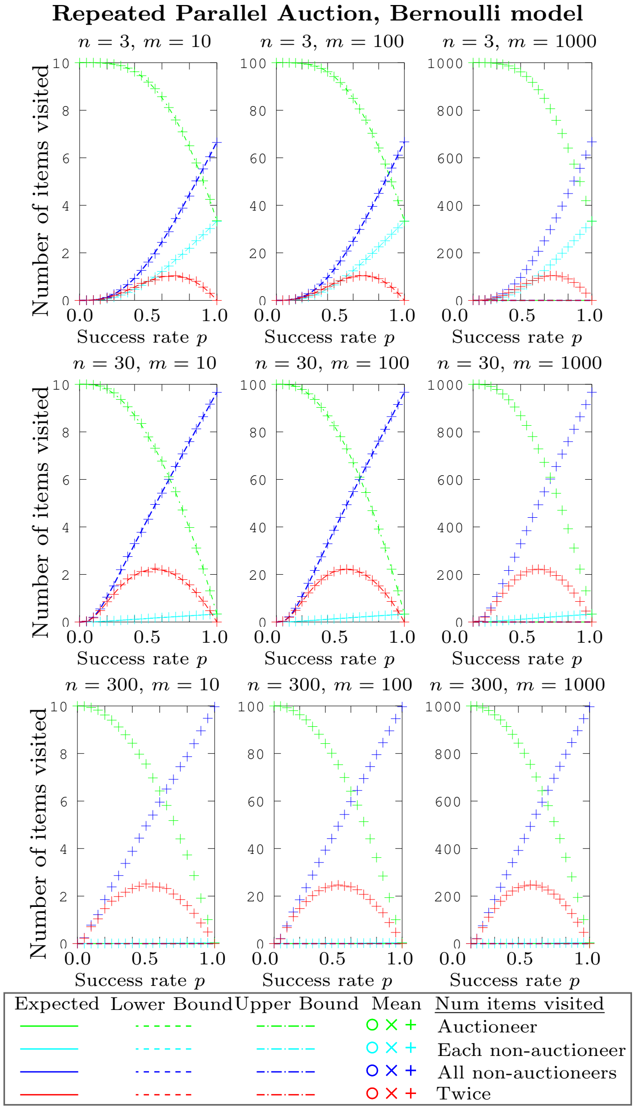

Fig. 19: Repeated Parallel Auction Agent Utilization Performance Curves. The average number of items each agent visits over various communication qualities in Scenario 1 (in which bids are realizations of random variables), assuming a Bernoulli communication model. Items are visited twice if an agent receives an award message from the auctioneer but fails to send an acknowledgment message back to the auctioneer (and so the auctioneer also visits the item).

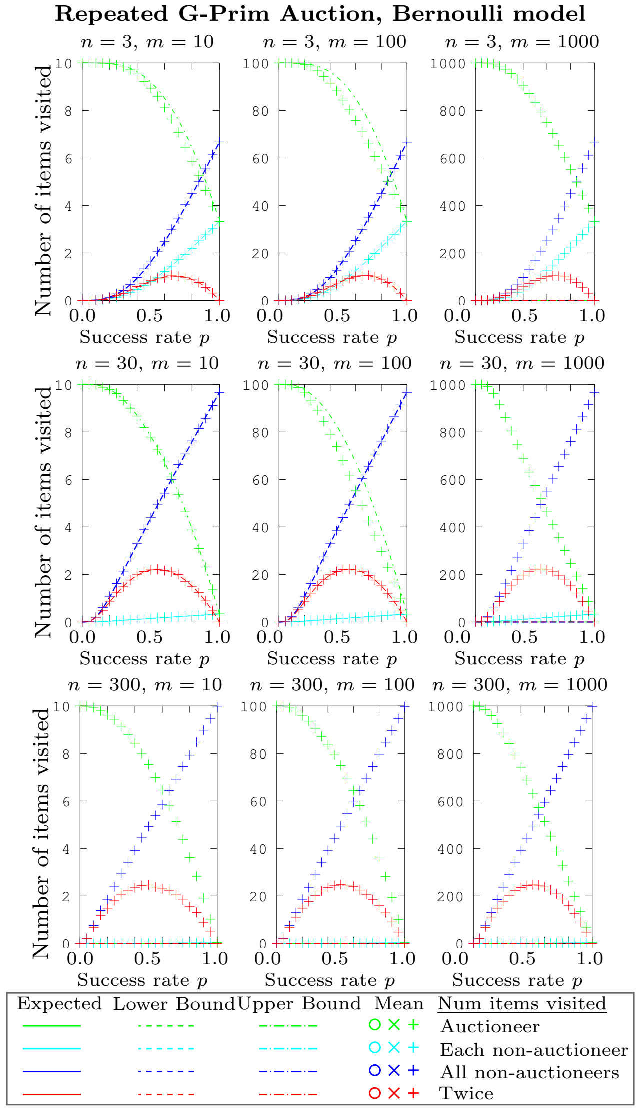

Fig. 20: Repeated G-Prim Auction Agent Utilization Performance Curves. The average number of items each agent visits over various communication qualities in Scenario 1 (in which bids are realizations of random variables), assuming a Bernoulli communication model. Items are visited twice if an agent receives an award message from the auctioneer but fails to send an acknowledgment message back to the auctioneer (and so the auctioneer also visits the item).

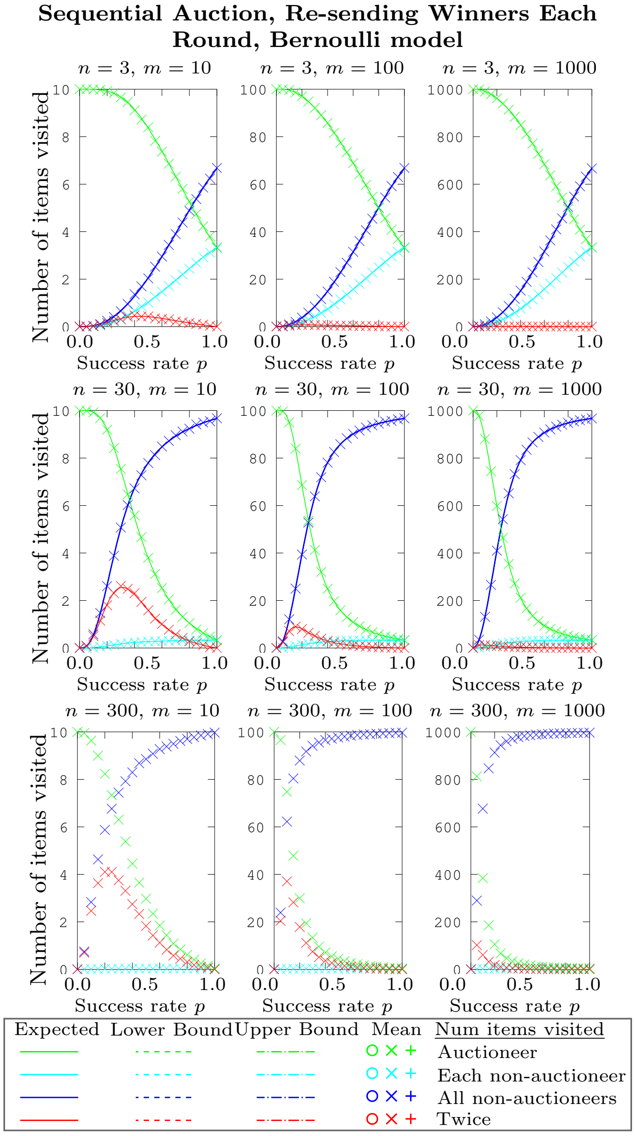

Fig. 21: Sequential Auction With Winner Rebroadcasts Agent Utilization Performance Curves. The average number of items each agent visits over various communication qualities in Scenario 1 (in which bids are realizations of random variables), assuming a Bernoulli communication model. Items are visited twice if an agent receives an award message from the auctioneer but fails to send an acknowledgment message back to the auctioneer (and so the auctioneer also visits the item).

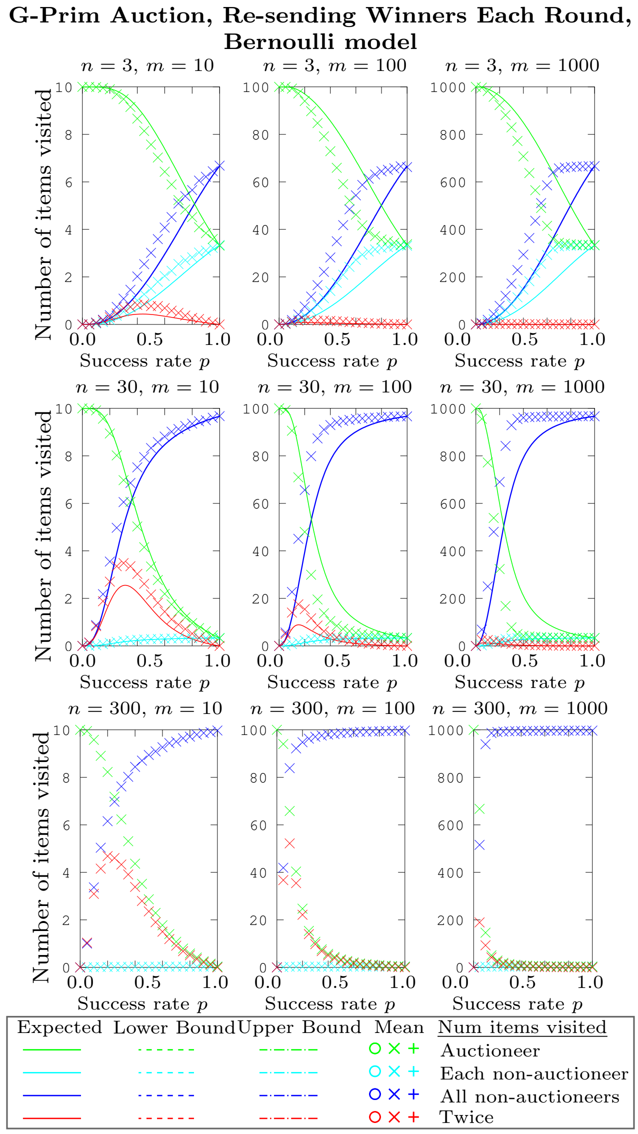

Fig. 22: G-Prim Auction With Winner Rebroadcasts Agent Utilization Performance Curves. The average number of items each agent visits over various communication qualities in Scenario 1 (in which bids are realizations of random variables), assuming a Bernoulli communication model. Items are visited twice if an agent receives an award message from the auctioneer but fails to send an acknowledgment message back to the auctioneer (and so the auctioneer also visits the item).

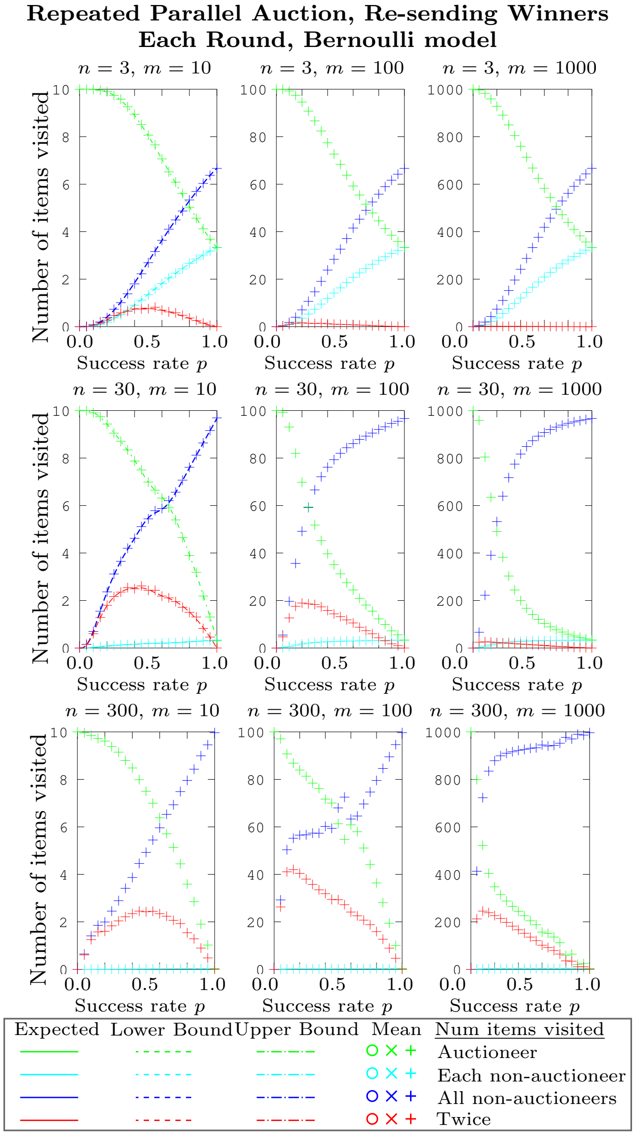

Fig. 23: Repeated Parallel Auction With Winner Rebroadcasts Agent Utilization Performance Curves. The average number of items each agent visits over various communication qualities in Scenario 1 (in which bids are realizations of random variables), assuming a Bernoulli communication model. Items are visited twice if an agent receives an award message from the auctioneer but fails to send an acknowledgment message back to the auctioneer (and so the auctioneer also visits the item).

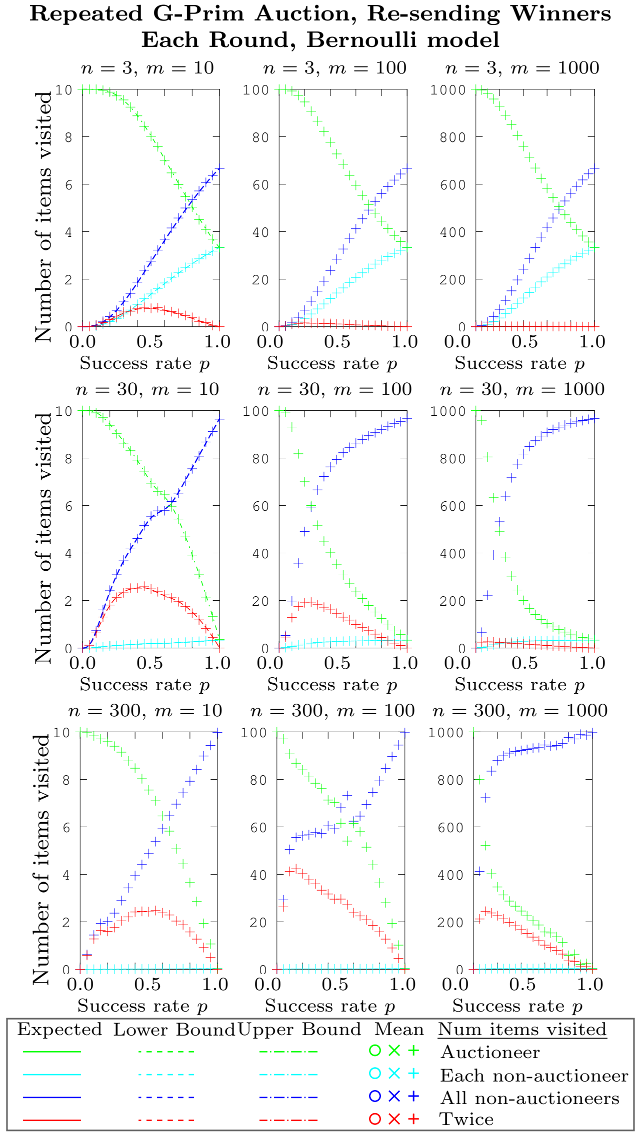

Fig. 24: Repeated G-Prim Auction With Winner Rebroadcasts Agent Utilization Performance Curves. The average number of items each agent visits over various communication qualities in Scenario 1 (in which bids are realizations of random variables), assuming a Bernoulli communication model. Items are visited twice if an agent receives an award message from the auctioneer but fails to send an acknowledgment message back to the auctioneer (and so the auctioneer also visits the item).