Negotiated Path Planning for Non-Cooperative Multi-Robot Systems

Anna Gautier[1], Bruno Lacerda[1], Nick Hawes[1], and Michael Wooldridge[2]

[1]Oxford Robotics Institute, University of Oxford
[2]Department of Computer Science, University of Oxford

Abstract

As robots becomes more prevalent, systems owned and operated by competing organisations will be required to interact in the same space. Interactions in such settings will be inherently *non-cooperative*. We therefore address the problem of non-cooperative multi-agent path finding. We design an auction mechanism to maximise the total number of tasks completed by a group of independent agents, while incentivising rational agents to participate. Our mechanism is a modified combinatorial Vickrey-Clarke-Groves auction, which aims to maximise social welfare whilst remaining computationally efficient. Our approach uses the privileged knowledge of the auctioneer to identify and solve path conflicts, whilst maintaining the autonomy of individual agents. Our results on synthetic data outperform existing work on this problem.

1 Introduction

When multiple autonomous systems interact in shared spaces, the interface between them must allow each system to achieve as many of their goals as possible, while respecting their autonomy. Consider a large box store, with aisles of groceries, clothing, and electronics. The store does not have the technology to build a fleet of robots, so they contract out individual tasks. One company might be hired to deploy a robot to move around the store examining shelves to track inventory. Another company may be hired to deploy cleaning robots. These robots have misaligned incentives; while they both have to interact in the space, they only care about their own contract with with the store, and so their only goal is to complete their own task. They may be penalised for failing to complete their tasks either with a built in monetary penalty or the eventual loss of their contract. Because there is no centralised system to direct them to complete their goals, conflicts are likely. Conflicts occur when multiple robots try traverse the same area of the supermarket at the same time. While these robots have collision avoidance for non-stationary objects (like humans), relying on collision avoidance for robot-robot interaction places an unnecessary burden on low level controllers and can cause significant delays or even collisions [Street *et al.*, 2020].

As a result, in this paper we consider an offline planning problem where all agents have the ability to make their own choices and are not controlled by a central system. We assume agents are *non-cooperative*. Agents could be indifferent to the success of other agents or may even want to stop other agents from achieving their own goals. Additionally, we assume all agents are *rational*: all agents are interested maximising their own reward given imposed constraints.

While multi-robot path finding problems are well studied [Stern *et al.*, 2019], very little research to date has considered the non-cooperative case. One notable exception is the work of Amir *et al.* [2015], which introduced a mapping between the multi-agent path finding problem (MAPF) and combinatorial auctions (CA). Furthermore, they discuss how the use of combinatorial Vickrey-Clarke-Groves (VCG) auctions can provide a mechanism robust to manipulation attempts by strategic, possibly self-interested agents. They then propose the use of *iBundle* [Parkes, 1999], an iterative CA algorithm with equivalent solutions to VCG, to find solutions to the MAPF problem. However, iBundle requires agents to evaluate an unnecessarily large number of paths. It relies on decentralised conflict resolution such that agents must continue to submit and value more bundles if the price of their high value bundles becomes too large due to conflicts. iBundle also occasionally fails to find solutions; in traditional CAs, it is possible that an agent is allocated no items. Furthermore in MAPF, it is impossible to remove an agent from the space, they must at least remain in the same place, potentially causing conflicts for other agents. Finally, because iBundle finds the optimal allocation and the winner determination subproblem of VCG is NP-complete, its computation time is large.

In this paper, we build on [Amir *et al.*, 2015] and tackle the limitations listed above. We do this by formally defining feasible allocations for CAs in the specific context of MAPF, and designing a mechanism for non-cooperative planning that exploits centralised conflict resolution via privileged knowledge (PK). This mechanism is designed to incentivise agents to both participate and to truthfully report their own interests. Our mechanism is tailored for the MAPF problem, allowing agents to submit a smaller number of paths to the auction. The auctioneer takes on the burden of removing conflicts, and uses a method specific to MAPF to do so. Our mechanism relaxes guarantees and optimality to improve computation time. This results in our computation time being lower than iBundle. We also show that in cases where it is infeasible to evaluate all possible paths (the number of which are exponential in the size of the graph), our mechanism also has higher utility.

2 Related Work

Auctioning approaches have been widely used in robotics, particularly for cooperative coordination of teams of robots. Early work on this topic used CAs [Hunsberger and Grosz, 2000] and first-price one-round auctions [Gerkey and Mataric, 2002] to distribute tasks across a set of agents. Later, in [Lagoudakis *et al.*, 2005], a sequential single item (SSI) auctioning mechanism was proposed for multi-robot routing and task allocation. This approach combines the advantages of parallel single-item auctions and combinatorial auctions, achieving good quality task allocation with low computational effort [Koenig *et al.*, 2006]. SSI auctioning has also been extended in [Nunes and Gini, 2015] to handle temporal constraints. From the perspective of the agents participating in SSI auctions, [Tovey *et al.*, 2005] investigated the problem of generating different bids taking into account different global objectives to be achieved. There has also been work on using auctions to distribute parts of a global task across a team of robots, in the context of planning under uncertainty. [Capitan *et al.*, 2013] does so considering role policies for partially observable Markov decision processes, whilst [Schillinger *et al.*, 2018] considers Markov decision process, auctioning subtasks such that a global linear temporal logic task is achieved by the team. In contrast to these works, we use auctioning to allocate a set of shared resources (more specifically, points in time and space) across a set robots. Furthermore, in our work the robots are *competing* for the use of these resources rather than *cooperating* to achieve a set of tasks.

There are examples of game theoretic non-cooperative planning algorithms, but they are ill suited to our scenario. [Bererton *et al.*, 2004] present a market-based approach for the management of shared resources for a multi-robot team. However, they assume loosely-coupled problems, i.e. problems that can be modelled using few interactions between agents. This is not the case in our work, where all locations yield a potential conflict between the robots. Stochastic games are a model-based method that has been applied to reinforcement learning [Littman, 1994] and planning [Kearns *et al.*, 2000; Niu and Clark, 2019]. This has been adapted to deal with a wide array of scenarios, but relies on a large amount of shared knowledge, for instance knowledge of the other agent's reward function, that can not be assumed in our scenario. [Brafman *et al.*, 2009] introduce a game theoretic model that can be adopted to path planning via STRIPS, but it assumes that agents are willing to form coalitions to achieve goals.

The closest related work to ours is [Amir *et al.*, 2015]. This was the first application of mechanism design to non-cooperative path planning problems. In particular, they reduce the problem of MAPF to a CA. They then demonstrate how a CA mechanism called VCG (see Section 3.2) can be used in MAPF with strategic agents. They provide experimental results using iBundle, an iterative CA with equivalent solutions to VCG [Parkes, 1999]. We build on this work, introducing a privileged knowledge mechanism that allows us to find solutions with less computational effort, as demonstrated in Section 6.

3 Preliminaries

Multi-Agent Pathfinding. In this paper, we tackle a non-cooperative variation of the multi-agent path finding problem, commonly referred to as MAPF. In MAPF, agents act on a graph $\mathcal{G} = (V, E)$, where V is a set of vertices and $E \subseteq V \times V$ is a set of edges. We assume each vertex $v \in V$ represents coordinates in a planar environment, i.e. $v \in \mathbb{R}^2$. Each of n agents starts at their own unique start vertex and at each discrete timestep, agents can wait in their current vertex or travel to another vertex that is connected to their current position by some edge in $\mathcal{G}$. Each agent has a unique goal location, and MAPF aims to find *paths* for each agent to get to their required goal, such that no agents are in *conflict*.

We define a path for an agent as a sequence of vertices $v_1 v_2 \dots v_m \in V^*$ such that $(v_i, v_{i+1}) \in E$ for all $i \in \{1, \dots, m\}$. While many different types of conflict have been explored in the MAPF literature, in this paper we consider vertex and edge conflicts. A *vertex conflict* occurs when two agents are at the same vertex at the same time. An *edge conflict* occurs when two agents are on the same edge at the same time. Note, that for discrete timesteps, these are only distinct when we allow edges to be bidirectional. We consider a MAPF variant where there might not be a global solution, i.e. it may be impossible for all agents to reach their goals. Thus, we consider the problem of allowing as many agents as possible to reach their goal, while minimising cost.

Combinatorial Auctions. A combinatorial auction (CA) contains agents $\mathcal{I} = \{1, 2, \ldots, n\}$ and a set of items $\mathcal{Z} = \{z_1, z_2, \ldots, z_m\}$. Each subset of items $A \in 2^{\mathcal{Z}}$ is known as a *bundle*. Each agent i is rational, so they are assumed to have a well-defined value for every bundle $A \in 2^{\mathcal{Z}}$, determined by function $v_i : 2^{\mathcal{Z}} \to \mathbb{R}$. The agents are asked to report a bid $b_i(A)$ to the auctioneer for every subset $A \in 2^{\mathcal{Z}}$. Ideally, this would be their value for that bundle, but rational agents are not necessarily truthful; if it is in their best interest to lie they will. More specifically, each agent has a strategy $\sigma_i : \mathbb{R} \to \mathbb{R}$ that transforms their valuations to bids. Each agent's bid is uniquely defined by $b_i(A) := \sigma_i(v_i(A))$ for all $A \in 2^{\mathcal{Z}}$. The auctioneer must use these bids to decide what items are distributed to what agent. We call a set $\{S_j\}_{j \in \mathcal{I}}$ where each $S_j \in 2^{\mathcal{Z}}$ an *allocation*.

Definition 1 (Feasible Allocation)**.** *A feasible allocation $\{S_j\}$ is one where each agent i is allocated a bundle of items $S_i \in 2^{\mathcal{Z}}$ such that $S_i \cap S_j = \emptyset$ for all agents $i \neq j$. We denote the set of all feasible allocations by* Γ.

The ultimate job of the auctioneer is to define a function that maps bids to outcomes. An outcome consists of a feasible allocation $\{S_j\}$ and a set of prices $\{p_j\}$, which each agent must pay to the auctioneer.

The auctioneer usually chooses some optimisation function, which can be to maximise properties such as *utility* or revenue.

3.1 MAPF as CA

The next important step is to understand how we auction off a path. The overall idea is considering paths as bundles. We start by noting that a path $w = v_1 v_2 \ldots v_m$ can be interpreted as a bundle of vertex-timestep pairs, i.e. $\{(v_1, 1), (v_2, 2), \ldots, (v_m, m)\} \subseteq V \times \mathbb{N}$. Thus, for notational convenience, for the remainder of the paper we will denote paths interchangeably as either sequences or bundles, i.e. we will denote a path for agent i as $w_i = v_1 v_2 \ldots v_m \in V^*$ or as $S_i = \{(v_1, 1), (v_2, 2), \ldots, (v_m, m)\} \subseteq V \times \mathbb{N}$. Agents value these bundles based on how effective the corresponding path is in achieving their goal and what cost it takes them to traverse that path. We define the best MAPF solution as one in which the sum of agents' costs is as low as possible. If we assume uniform reward R for each agent, then the minimum cost allocation is equivalent to the maximum value allocation, e.g.

$$\begin{aligned}\operatorname*{arg\,min}_{\{S_j\} \in \Gamma} \sum_{i \in \mathcal{I}} \text{cost}(S_i) &= \operatorname*{arg\,max}_{\{S_j\} \in \Gamma} \sum_{i \in \mathcal{I}} -\text{cost}(S_i) \\ &= \operatorname*{arg\,max}_{\{S_j\} \in \Gamma} \sum_{i \in \mathcal{I}} R - \text{cost}(S_i) \\ &= \operatorname*{arg\,max}_{\{S_j\} \in \Gamma} \sum_{i \in \mathcal{I}} v_i(S_i).\end{aligned} \tag{1}$$

Thus, maximising the sum of all values over all agents, known as the *social welfare*, results in a solution to the MAPF problem in which as many agents reach their goals with as little cost as possible.

A high level overview of the parallel between MAPF and Combinatorial Auctions (CA) can be seen in Table 1 from [Amir *et al.*, 2015], who initially introduced this reduction.

Table 1: A high level overview of the parallel between MAPF and CA from [Amir *et al.*, 2015].

MAPF	**CA**
Vertex-time pair	Item
Path (set of items)	Bundle
Derived from the path cost	Valuation function
Minimal sum of path costs	Maximal social welfare

Determining feasible allocations via a CA allows us to make sure that agents do not collide. If two agents' bundles S_i and S_j have no intersection (which is true of all feasible allocations), then they will never be at the same vertex at the same time. This framework can be extended to include edge conflicts, as seen in Section 4.1.

3.2 Combinatorial VCG Auction

Framing non-cooperative path planning as a CA leads us to the problem of designing an auction mechanism that maximises social welfare, or the sum of all valuations. However the auctioneer only has access to the bids the agents report, not their true valuation for the items. Therefore the auctioneer cannot naively add up the bids of agents to determine the social welfare of an allocation. Doing so is highly susceptible to manipulation by agents. *Incentive compatible mechanisms* are one such class of auctions that solve this problem.

Definition 2 (Incentive Compatibility)**.** *A mechanism is incentive compatible if every agent receives their highest possible utility when they report their valuation truthfully. More formally, this means that all agents' strategies are defined by $\sigma_i(v_i) = v_i$.*

Because, by definition, $b_i := \sigma_i(v_i)$, the auctioneer can assume that each agent's valuation for a bundle is equivalent to their bid. In this case, maximising the total utility becomes equivalent to the maximising the total of the bids.

One CA that satisfies the property of incentive compatibility is the combinatorial Vickrey-Clarke-Groves (VCG) auction (named after ideas combined from [Vickrey, 1961], [Clarke, 1971] and [Groves, 1973]). The descriptions in this section are based on Cramton *et al.* [2006].

In VCG the auctioneer allocates bundles based on the welfare maximising outcome:

Definition 3 (Welfare Maximising)**.** *An allocation* $\{S_j\} \in \Gamma$ *is welfare maximising if, for every feasible allocation* $\{T_j\} \in \Gamma$,

$$\sum_{i\in\mathcal{I}} b_i(S_i) \geq \sum_{i\in\mathcal{I}} b_i(T_i). \tag{2}$$

In VCG, prices p_i are set as follows. Let $\{S_j\} \in \Gamma$ be the welfare maximising outcome above and the $\{S_j^{-i}\} \in \Gamma$ be the welfare maximising outcome if agent i was not involved in the auction. Then,

$$p_i = \sum_{k\in\mathcal{I}\setminus\{i\}} b_k(S_k^{-i}) - \sum_{k\in\mathcal{I}\setminus\{i\}} b_k(S_k). \tag{3}$$

This payment represents the amount that agent i has perturbed the system, as agent i pays the difference between the total welfare of all other agents if they had not been in the auction minus the welfare of all other agents when they are included in the auction. Note that if an agent's presence in the system does not change anyone else's path, or only changes another agent's path to one of equal value, they pay nothing. This can be thought of as a way to ensure agents don't take up desirable space in the map (like nodes with many edges, or nodes that connect otherwise unconnected portions of the graph) unless it has high value to them.

There is a final important property that combinatorial VCG auctions satisfy.

Definition 4 (Individual Rationality)**.** *A mechanism is individually rational if each agent has a non-negative outcome from the auction [Krishna, 2009], meaning that their true value from participating minus the price they pay is non-negative.*

This is important because agents will be willing to agree participate in a mechanism if they are guaranteed to get a non-negative utility.

Unfortunately determining the welfare maximising outcome for VCG is NP-complete [Dobzinski and Nisan, 2007], so solving this problem optimally is time consuming. Additionally, using combinatorial VCG auctions for MAPF require agents to give a value for every possible path, which is an unrealistic computational burden on the part of the agent. In a graph with V nodes, valuing all possible bundles means valuing $O(2^{V\times T})$, where T is the length path that takes the longest amount of time. While the number of bundles that actually correspond to a path will be lower (and dependent on the sparsity of the graph), it will still be prohibitively large.

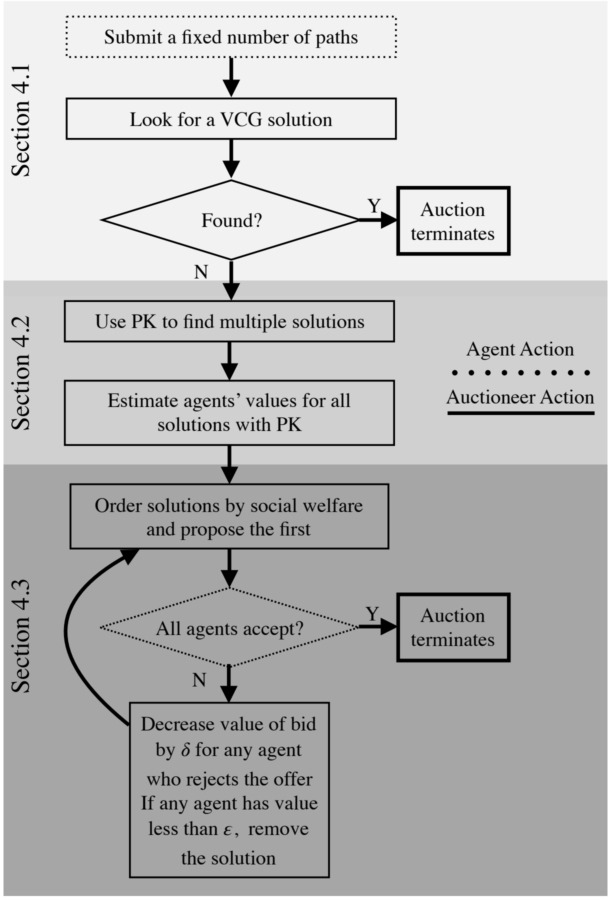

Figure 1: A high level of our privileged knowledge (PK) protocol.

4 A Protocol with Privileged Knowledge

We now introduce a new mechanism which deals with the problem of enumerating all possible bids, by allowing agents to submit bids only for a subset of the bundles. As with a VCG auction, our aim is to create a system in which agents are incentivised to truthfully participate because it allows them to have autonomy over their plans and avoid conflicts. Because we assume agents will submit less bids, we need to account for the case where none of those bids determine a solution. Figure 1 provides an overview of our proposed mechanism, where we allow the auctioneer to use *privileged knowledge* (in particular the knowledge it can obtain from the agents' bids) to find a solution outside of the submitted proposals if none fit within them.

4.1 Initial Bidding and VCG Auctioning

Agents propose timed paths in the form $\{(x_t, t)\}_{t=1}^{m}$. In the first step of the mechanism, they propose a fixed number of paths to the auctioneer. This number is ideally small (i.e. < 10) to cope with the scalability issues of VCG and is chosen as a design parameter.

For each agent i, the auctioneer automatically adds the path that corresponds to staying still in their start position s_i (represented by $\{(s_i, 1), (s_i, 2), \ldots\}$) with value 0. The auctioneer then searches for a potential feasible allocation with the same winner determination and pricing as combinatorial VCG described in Section 3.2. Previous work in CA for MAPF has consisted of off-the-shelf algorithms and does not take into account specificities of the MAPF domain. Here, we extend the definition of feasible allocation to better suit MAPF.

Definition 5 (Feasible Allocation for MAPF)**.** *A feasible allocation for MAPF is one where each agent i is allocated a bundle of items $S_i \in 2^{\mathcal{Z}}$ such that:*

1. $S_i \cap S_j = \emptyset$ *for all agents i, j such that $i \neq j$;*
2. $S_i \neq \emptyset$ *for all agents i; and*
3. *For any agent i, if $(t, a), (t+1, b) \subseteq S_i$, then $(t, b), (t+1, a) \nsubseteq S_j$ for all agents j such that $i \neq j$.*

Point 1 ensures the allocation is feasible in general CA terms, and from the MAPF perspective ensures no vertex conflicts. Point 2 ensures all agents get a path. This is unlike classic VCG where an agent can be allocated the empty set. Because we cannot remove an agent from the physical world, such allocation does not make sense from a MAPF perspective. At the very least, each agent must be allocated the path that amounts to staying still. Point 3 ensures no edge conflicts.

At this stage of the auction, agents can only be allocated bundles which they themselves proposed. As a result, all allocations will correspond to actual paths. While agents are, in theory, allowed to bid on bundles which are not paths, any such bundle will be impossible to achieve and thus have value $-\infty$.

It is possible that the final allocation will result in one or more agents staying still. If this occurs, we move into the second part of our protocol.

4.2 Deconflicting

Now, the auctioneer uses the privileged knowledge gained in the bidding stage to find a better solution.

The auctioneer first finds multiple solutions that allow all agents to reach their goal. Our mechanism is agnostic to the method used to find such solutions, the suitability of each possible method being tied with the particular environment. In situations where the auctioneer believes value is closely tied to the shortest path to the goal, classical MAPF cooperative algorithms (modified to return multiple solutions) are a suitable choice. In our experiments, we used Hierarchical Cooperative A* [Silver, 2005]. We leave comparing different solution methods to future work. Ideally, we desire at least two solutions from the MAPF solver but if there is only one possible solution, the auctioneer will proceed with only that one solution.

Assuming that the auctioneer has a set of alternate solutions $\mathcal{AS} = \{\{S_j^1\}, ..., \{S_j^m\}\}$, it is necessary to order them from most desirable to least desirable. The goal of the entire mechanism is to elicit truthful values, so of course we cannot find the exact social welfare of these solutions with just a handful of bids, but we can develop a heuristic, and then use that to elicit true preferences by asking the agents for more information (Section 4.3).

To approximate the social welfare of a specific alternate solution $\{S_j\}$, we need to approximate the value of each path S_i to each agent i.

We define the approximate value of a path by how different it is from a path that we know agent i's true value of, e.g. a path that agent i bid on the first stage of the mechanism. In particular we chose the path $\tilde{S}_i$ to be the 'closest' bid upon path to our solution path S_i.

Our heuristic for 'closeness' is defined by the following distance function between two bundles $S, T \in 2^{\mathcal{Z}}$:

$$d_i(S, T) = \lambda ||S - T|| + (1 - \lambda)|t_{i,S} - t_{i,T}|, \qquad (4)$$

where $|| \cdot || = \sum ||v_t||_2$ and $t_{i,S}$ is the time at which path S reached the goal of agent i.

This distance function measures two important factors that would change an agents value as a path changes. The first term accounts for physical locations (like rough terrain) and the second accounts for a difference in makespan for the individual robot. λ is thus a parameter that allows us to trade off importance between these two factors.

With this distance function in mind, we can formally define the path $\tilde{S}_i$ which is the 'closest' bid upon path to our solution path S_i as

$$\tilde{S}_i = \underset{S \in \text{Bids}_i}{\arg\min}\, d_i(S_i, S). \qquad (5)$$

Finally, we define approximate value of solution $\{S_j\}$ to agent i as:

$$\tilde{v}_i(\{S_j\}) = b_i(\tilde{S}_i) - d_i(\tilde{S}_i, S_i). \qquad (6)$$

After computing each agent's approximate value in each solution, we can build an approximate ordering of solutions by social welfare and proceed to the next step of the mechanism.

4.3 Descending Auction

Once we have an heuristic for social welfare of our proposed solutions $\mathcal{AS}$, we sort $\mathcal{AS}$ in decreasing order of approximate social welfare, then we carry out a simultaneous descending auction to elicit the true values of each solution to each agent. Thus, the auctioneer first offers the solution with best approximates social welfare, $\{S_1\}$, to each agent at prices equal to $\tilde{v}_i(\{S_1\})$. If all agents accept the offer, this solution is chosen. Otherwise, the approximate value for any agent i that did not accept the offer is reset to $\tilde{v}_i(\{S_1\}) = \delta\tilde{v}_i(\{S_1\})$, for some weighting factor $0 < \delta < 1$, and $\mathcal{AS}$ is reordered. For small δ, the auction will execute quicker, but agents are likely to be proposed offers below their value for the item. For large δ, the descending auction will occur in finer intervals, thus taking longer but gaining a more accurate representation of agent valuations. If at any point an agent accepted a solution, they are assumed to have accepted it in the future and not offered a new value. If at anytime an agent rejects an offer with value $\tilde{v}_i(\{S_1\}) < \epsilon$ for some small $\epsilon > 0$, this solution is assumed to be infeasible and removed from the set.

This process is repeated until all agents accept the same solution or all solutions have been made infeasible. If all agents accept the same solution, then this becomes the final solution $\{\tilde{S}_j\} \in \mathcal{AS}$ and the payment for agent i is

$$p_i = \sum_{j \in \mathcal{I} \setminus \{i\}} b_j(S_j^{-i}) - \sum_{j \in \mathcal{I} \setminus \{i\}} \tilde{v}_j(\tilde{S}_j) \tag{7}$$

where $\{S_j^{-i}\} \in \Gamma$ are the hypothetical solutions described in Section 4.1 that would have existed before perturbation.

If there are no solutions left because all have been deemed infeasible, all agents will be told to stay still.

4.4 Analysis

We discuss how our modifications effect the important properties of individual rationality and incentive compatibility, and present arguments on how these are preserved by our approach.

Individual Rationality. First, the mechanism is individually rational, i.e. each agent will take their assigned path. Suppose that an agent is considering deviating from their assigned path w, because there is some other path w' that has higher value for them. We make the assumption that if they know that path w' is an option with higher value, they would have submitted it. If they submitted it, they know that if it did not conflict with the other agents, they would have been allocated it. Thus, if they choose to follow path w' instead of w, they are guaranteed to be in conflict with at least one other robot, which will cause w' to be infeasible.

Incentive Compatibility. We have not yet proved our mechanism to be incentive compatible, and this may not be feasible due to our approximations being suboptimal. However, our design decisions prevent easy manipulation. If the algorithm does not enter the privileged knowledge sub-protocol then it remains in a VCG auction which is incentive compatible. If it does enter the sub-protocol, truthtelling means accepting the first value that is lower than your actual value for a path. Clearly, agents are not incentivised to accept a path that is higher than their value because they may be charged for it, netting a negative utility from the process. Additionally, we need to show they will not turn down any price that underestimates their value. Suppose they do not, then there is a chance they will instead get offered a deal next which underestimates their value less, or a deal in which they have lower value. In the first case, they will end up with a worse utility. In the second case, they could end up staying still (a worse outcome).

5 Agent Decision Making

Our mechanism assumes agents have the ability to submit a fixed (but small) number of paths to the auctioneer. We also assume that agents have knowledge of their true valuation for these paths. While these valuations and paths can be derived in any number of ways, and do not effect the properties of the mechanism, we outline the strategies we used in our experiments in Section 6.

5.1 Valuing Paths

Given a bundle of items, it is important that an agent understands what value to assign it. If a bundle of items is not a path as described in Section 3.1 then an agent should value it as $-\infty$ as it is impossible to accomplish. If a bundle is a path, the value of that path to an agent is the reward R minus the cost of the path. The cost of the path should be defined for each agent by the distance travelled and the time it takes. This will depend on the specifications of the robot, but an increase in both distance and time will be assumed to increase costs and thus decrease utility. In practice, this can be determined by a shortest path algorithm so that shorter paths have higher value.

Note that the method described in Section 4 does not assume uniform rewards. In a case with non-uniform rewards, the mechanism optimises directly for social welfare. A designer can thus use rewards as a proxy for prioritisation. In uniform reward cases, all agents are treated non-discriminatory, and only the total cost of agents is considered, as shown above. If rewards vary, agents with higher reward will have higher possible value for success, and thus will be prioritised in the auction if possible.

5.2 Agent Planning

We propose a method to quickly generate suboptimal paths, which is used in our simulations below. First, the agent uses the Floyd–Warshall algorithm to generate the shortest

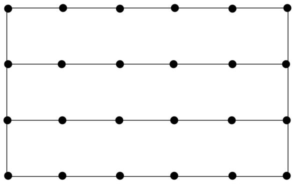

Figure 2: An example of a supermarket with $k = 3$ and $l = 5$.

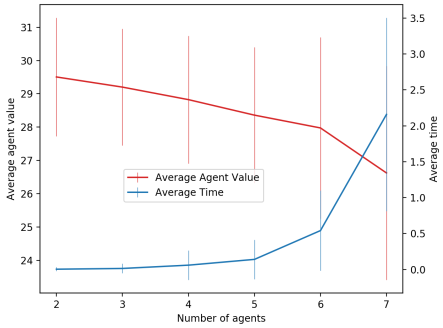

Figure 3: An example of how our mechanism scales with respect to agents on a $(10, 10)$ supermarket with five initial bids.

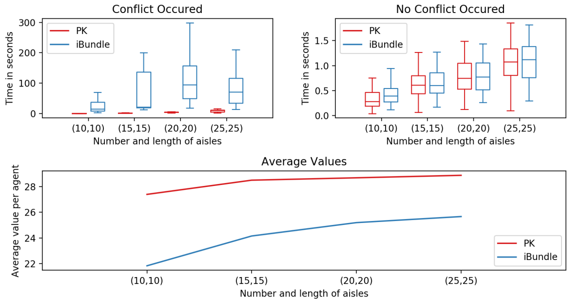

Figure 4: An example of how our mechanism scales in comparison to iBundle on changing supermarket size with 6 agents and 5 initial bids. The top left graph shows instances where conflict resolution is necessary. The top right graph shows instances where no conflict resolution was necessary, and the algorithm terminated in the first stage. The bottom graph show the average value per agent over all trials. Table 2 list these frequencies, along with the frequency in which iBundle failed to find a solution.

path between all pairs of vertices [Floyd, 1962]. The optimal path is the shortest path between the start and finish, say $w^* = sv_1v_2...v_kg$. To generate suboptimal paths, we examine the neighbours of some node v_i in path $\{v_j\}_{j=1}^k$ and choose any neighbour $\tilde{v} \notin \{v_j\}_{j=1}^k$. If the shortest path between $\tilde{v}$ and the goal is $\tilde{v}\tilde{v_1}\tilde{v_2}...\tilde{v_{k_1}}g$, then the new suboptimal path is $\tilde{w} = sv_1v_2...v_i\tilde{v}\tilde{v_1}\tilde{v_2}...\tilde{v_{k_1}}g$. This can be repeated for any v_i and any valid neighbour $\tilde{v}$, or the entire process can be repeated on suboptimal paths instead of the optimal paths, depending on how many paths the agent would like to submit to the auctioneer.

6 Evaluation

To analyse the performance of our modified privileged knowledge auction, we simulated its performance on maps designed to look like supermarkets. All maps of size (k, l) are constructed from k aisles of length l units. An example grap $\mathcal{G}$ is shown in Figure 2. This approach is similar to warehouse environments used in MAPF experiments, e.g. [Ma *et al.*, 2017]. Aisles are connected to each other only at the right and left corridors which are the most likely points of conflict in the maps.

For the following sections, each data point was calculated through 100 random trials. Paths were generated as described in Section 5.2. For simplicity, only vertex conflicts are considered. We do not expect edge conflicts to change the results qualitatively since both methods presented below would encounter the same additional conflicts. Start and goal vertexes for each agent are generated randomly, with no two agents sharing a start or goal vertex. Agents receive a reward of 30 if they reach their goal.

6.1 Scalability in Number of Agents

In Figure 3, we observe how both computation time and social welfare suffer from an increase in agents on a fixed size map. Map size is set to $(10, 10)$ and agents all submit 5 initial bids. Because VCG is exponential in the number of agents, the protocol will also scale exponentially. It will, however, do so to a much lower power, because there are fewer bundles submitted as bids. As expected, the average value obtained by each agent decreases as conflicts become more likely. This is because agents are forced to take longer paths to avoid conflicts, resulting in lower utility (reward minus path cost).

6.2 Comparison to iBundle

Next, we compare our mechanism to the iBundle-based mechanism of Amir *et al.* [2015]. In our iBundle implementation, agents propose same number of bids as in our mechanism: 5 paths. This means if our auction completes after finding a conflict-free solution during the initial VCG process (Section 4.1), then iBundle will converge after 2 iterations. For our comparison we separate the results where a non-conflicting solution is found in this initial phase from those where one is not. This allows us to more clearly analyse the difference in computation time, as seen in Figure 4. We also remove cases where iBundle fails to find a solution after 500 iterations (at least two minutes depending on map). These cases are summarised in Table 2.

Table 2: Conflict and failure frequencies over 100 random trials with 6 agents and 5 initial bids.

Graph Size	Conflict Occurs	iBundle Failure
(10, 10)	24%	6%
(15, 15)	16%	4%
(20, 20)	20%	5%
(25, 25)	20%	6%

Normally, iBundle requires that agents have knowledge of an ordered list of allocations. We do not have this knowledge here because it is prohibitively costly to value all such paths on a large graph. Instead we generate paths as described in Section 5.2. This leads to below optimal solutions for iBundle, so our mechanism is able to perform better, as seen in Figure 4. This shows that iBundle is less robust to partial information on path valuations that our approach, only providing optimal results when one is able to evaluate all paths to the goal (including suboptimal ones), which is infeasible in most scenarios. Note that the optimal allocation (given full information of agents valuations) would, at best, have an average value per agent of 30.

7 Conclusions

We have presented a novel method for non-cooperative multi-agent path finding. In order to solve MAPF problems as CAs, we modify the concept of a feasible allocation in CAs to be more suitable for multi-agent path finding domains. Because valuing every path is impractical, we have proposed a modified offline auction protocol that allows agents to value fewer paths by exploiting the privileged knowledge of the central planner and the physical layout of the planning space. Empirical results show that our mechanism outperforms the state-of-the-art approach for MAPF as CA. Future work includes exploring different methods of deconflicting paths, and formal proofs on the properties of the mechanisms.

References

Ofra Amir, Guni Sharon, and Roni Stern. Multi-agentpathfinding as a combinatorial auction. *AAAI*, 2015.

Curt Bererton, Geoffrey J Gordon, and Sebastian Thrun. Auction mechanism design for multi-robot coordination. In *Advances in Neural Information Processing Systems*, pages 879–886, 2004.

Ronen I Brafman, Carmel Domshlak, Yagil Engel, and Moshe Tennenholtz. Planning games. In *Twenty-First International Joint Conference on Artificial Intelligence*, 2009.

Jesus Capitan, Matthijs T.J. Spaan, Luis Merino, and Anibal Ollero. Decentralized multi-robot cooperation with auctioned pomdps. *The International Journal of Robotics Research*, 32(6):650–671, 2013.

Edward H. Clarke. Multipart pricing of public goods. *Public Choice*, 11(1):17–33, September 1971.

Peter Cramton, Yoav Shoham, and Richard Steinberg. *Combinatorial Auctions*. The MIT Press, 2006.

Shahar Dobzinski and Noam Nisan. Limitations of vcg-based mechanisms. In *Proceedings of the Thirty-Ninth Annual ACM Symposium on Theory of Computing*, STOC '07, page 338–344, New York, NY, USA, 2007. Association for Computing Machinery.

Robert W. Floyd. Algorithm 97: Shortest path. *Commun. ACM*, 5(6):345, June 1962.

Brian P Gerkey and Maja J Mataric. Sold!: Auction methods for multirobot coordination. *IEEE transactions on robotics and automation*, 18(5):758–768, 2002.

Theodore Groves. Incentives in Teams. *Econometrica*, 41(4):617–631, 1973.

Luke Hunsberger and Barbara J Grosz. A combinatorial auction for collaborative planning. In *Proceedings fourth international conference on multiagent systems*, pages 151–158. IEEE, 2000.

Michael Kearns, Yishay Mansour, and Satinder Singh. Fast planning in stochastic games. In *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence*, UAI'00, page 309–316, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc.

Sven Koenig, C Tovey, M Lagoudakis, V Markakis, David Kempe, Pinar Keskinocak, A Kleywegt, Adam Meyerson, and Sonal Jain. The power of sequential single-item auctions for agent coordination. In *Proceedings of the AAAI conference on artificial intelligence*, volume 21, 2006.

Vijay Krishna. *Auction Theory*. Academic Press, September 2009.

Michail G Lagoudakis, Evangelos Markakis, David Kempe, Pinar Keskinocak, Anton J Kleywegt, Sven Koenig, Craig A Tovey, Adam Meyerson, and Sonal Jain. Auction-based multi-robot routing. In *Robotics: Science and Systems*, volume 5, pages 343–350. Rome, Italy, 2005.

Michael L. Littman. Markov games as a framework for multi-agent reinforcement learning. In William W. Cohen and Haym Hirsh, editors, *Machine Learning Proceedings 1994*, pages 157–163. Morgan Kaufmann, San Francisco (CA), January 1994.

Hang Ma, T. K. Satish Kumar, and Sven Koenig. Multi-agent path finding with delay probabilities. 2017.

Luyao Niu and Andrew Clark. Optimal secure control with linear temporal logic constraints. *IEEE Transactions on Automatic Control*, 2019.

Ernesto Nunes and Maria Gini. Multi-robot auctions for allocation of tasks with temporal constraints. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

David C. Parkes. Ibundle: An efficient ascending price bundle auction. In *Proceedings of the 1st ACM Conference on Electronic Commerce*, EC '99, page 148–157, New York, NY, USA, 1999. Association for Computing Machinery.

Philipp Schillinger, Mathias Bürger, and Dimos V Dimarogonas. Auctioning over probabilistic options for temporal logic-based multi-robot cooperation under uncertainty. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7330–7337. IEEE, 2018.

David Silver. Cooperative pathfinding. *AIIDE*, 1:117–122, 2005.

Roni Stern, Nathan Sturtevant, Ariel Felner, Sven Koenig, Hang Ma, Thayne Walker, Jiaoyang Li, Dor Atzmon, Liron Cohen, T. K. Satish Kumar, Eli Boyarski, and Roman Bartak. Multi-agent pathfinding: Definitions, variants, and benchmarks. In *SoCS*, 2019. arXiv: 1906.08291.

C Street, B Lacerda, M Mühlig, and Nicholas Hawes. Multi-robot planning under uncertainty with congestion-aware models. *Proceedings of AAMAS 2020*, 2020.

Craig Tovey, Michail G Lagoudakis, Sonal Jain, and Sven Koenig. The generation of bidding rules for auction-based robot coordination. In *Multi-Robot Systems. From Swarms to Intelligent Automata Volume III*, pages 3–14. Springer, 2005.

William Vickrey. Counterspeculation, Auctions, and Competitive Sealed Tenders. *The Journal of Finance*, 16(1):8–37, 1961.