Scalable Mechanism Design for Multi-Agent Path Finding

Paul Friedrich[1,2] , **Yulun Zhang**[3] , **Michael Curry**[1,2,4] , **Ludwig Dierks**[5] , **Stephen McAleer**[3] , **Jiaoyang Li**[3] , **Tuomas Sandholm**[3,6] and **Sven Seuken**[1,2]
[1]ETH AI Center
[2]University of Zurich
[3]Carnegie Mellon University
[4]Harvard University
[5]University of Illinois at Chicago
[6]Strategy Robot, Inc., Strategic Machine, Inc., Optimized Markets, Inc.
paul.friedrich@uzh.ch, yulunzhang@cmu.edu

Abstract

Multi-Agent Path Finding (MAPF) involves determining paths for multiple agents to travel simultaneously and collision-free through a shared area toward given goal locations. This problem is computationally complex, especially when dealing with large numbers of agents, as is common in realistic applications like autonomous vehicle coordination. Finding an optimal solution is often computationally infeasible, making the use of approximate, suboptimal algorithms essential. Adding to the complexity, agents might act in a self-interested and strategic way, possibly misrepresenting their goals to the MAPF algorithm if it benefits them. Although the field of mechanism design offers tools to align incentives, using these tools without careful consideration can fail when only having access to approximately optimal outcomes. In this work, we introduce the problem of scalable mechanism design for MAPF and propose three strategyproof mechanisms, two of which even use approximate MAPF algorithms. We test our mechanisms on realistic MAPF domains with problem sizes ranging from dozens to hundreds of agents. We find that they improve welfare beyond a simple baseline.

1 Introduction

There are numerous emerging domains of large-scale resource allocation problems, such as allocating road capacity to cars or 3D airspace to unmanned aerial vehicles (UAVs), where *multi-agent path finding (MAPF)* is required to calculate a valid allocation. In realistically-sized instances of such problems, non-colliding paths must be calculated for large numbers of agents simultaneously. Such computationally challenging domains are increasingly of real-world importance. Future UAV traffic in urban areas from parcel deliveries alone is projected to require managing tens of thousands of concurrent flight paths [Doole *et al.*, 2020].

Most MAPF research assumes that the true agent *preferences* (e.g., cost for moving on the map) are known to (or even set by) the allocating system and focuses on finding computationally efficient ways to obtain solutions that minimize an aggregated agent cost measure, called *optimal* solutions. However, if agents are *self-interested* and independently report their preferences to the system, it may be beneficial for them to *misreport*. For example, an agent may wish to exaggerate her cost of a delay to ensure that the algorithm favors her over an agent that reported a lower cost when resolving a potential conflict. In this *non-cooperative* setting, the system may not know the agents' true preferences since they could be misreported. In that case, there is no guarantee that its MAPF algorithm finds a solution that maximizes true social welfare.

The field of *mechanism design* deals with exactly this problem: designing systems that efficiently allocate scarce resources while ensuring that they are *individually rational* and *strategyproof*. These desirable properties guarantee that participating in the mechanism always benefits each agent and that agents are incentivized to report their private information truthfully to the mechanism, respectively. The celebrated *Vickrey-Clarke-Groves (VCG)* mechanism [Vickrey, 1961; Clarke, 1971; Groves, 1973] achieves this goal. It chooses an allocation that perfectly maximizes welfare given agents' reported preferences and then charges side payments to each agent. Via a careful choice of payments, VCG ensures that there is no incentive for agents to lie about their preferences.

However, mechanism design has so far either not been applied to large-scale routing problems or has bypassed path finding, e.g., via congestion pricing in road domains [Beheshtian *et al.*, 2020]. The main reason is that even classic, cooperative MAPF problems are computationally extremely challenging. Finding an optimal solution is NP-hard [Yu and LaValle, 2013]. The state-of-the-art MAPF algorithm producing optimal solutions is *conflict-based search (CBS)* [Sharon *et al.*, 2015]. CBS first generates a shortest path for each agent and then resolves conflicts (path collisions) iteratively, building a constraint tree in a best-first-search manner. Its worst-case runtime scales exponentially in the number of agents, and identifying why certain scenarios are computationally feasible to optimally solve while others

Code at https://github.com/lunjohnzhang/MAPF-Mechanism.

are not is an open area of research [Gordon *et al.*, 2021].

Much of the MAPF literature therefore employs (bounded) suboptimal algorithms or even counts finding any feasible path assignment as a success. While there exist a number of suboptimal MAPF algorithms that work very well in practice as long as agent preferences are fully known [Barer *et al.*, 2014; Li *et al.*, 2021], most are unsuitable for a mechanism with self-interested agents. This is because VCG-based mechanisms rely on finding an optimal assignment – if the assignment is only approximately optimal, the resulting mechanism is generally no longer strategyproof [Sandholm, 2002].

In this paper, we show how we can design suboptimal but more scalable MAPF algorithms that are made strategyproof through VCG-based payments by ensuring that they choose optimally from some restricted, fixed set of outcomes. This is called the *maximal-in-range (MIR)* property [Nisan and Ronen, 2007], which guarantees that charging VCG payments results in no incentive to lie.

The rest of this paper is organized as follows. After giving an overview of related work (Section 2), we introduce our formulation of the non-cooperative MAPF problem and illustrate the approach of pairing an optimal MAPF algorithm with VCG payments to achieve strategyproofness (Section 3).

In Section 4, we first show how this approach applied to the optimal CBS algorithm results in a welfare-maximizing and strategyproof mechanism, which we call the *payment-CBS (PCBS)* mechanism.

As we are interested in significantly larger problems, we successively restrict the search space to increase scalability. We focus on the class of *prioritized* MAPF algorithms [Silver, 2005]. Instead of searching for a jointly optimal allocation, these algorithms search through a space of *priority orderings*. They plan paths for agents sequentially, such that each planned path avoids all paths that belong to agents with higher priority. They are fast and efficient in practice but have no (bounded) suboptimality guarantees, and one can construct MAPF instances that cannot be solved by them.

With this restriction, we first introduce the *exhaustive-PBS (EPBS)* mechanism, which uses an allocation rule based on an adapted version of *priority-based search (PBS)* [Ma *et al.*, 2019]. It constructs a priority ordering, that is, a strict partial order on the set of agents, by iteratively adding a binary priority relation and re-planning accordingly for pairs of agents that conflict. By exhaustively expanding the PBS search tree, EPBS fulfills the MIR property. EPBS illustrates how a suboptimal MAPF algorithm can be made strategyproof using MIR and our variation of VCG payments, which can be calculated without additional computational cost.

In a second step, we further restrict the search space to a fixed number of randomly drawn *total* priority orderings. The resulting mechanism, which we call *Monte-Carlo prioritized planning (MCPP)*, chooses the welfare-maximizing allocation among those resulting from applying *prioritized planning (PP)* [Erdmann and Lozano-Perez, 1987] to the randomly drawn orderings. In addition to being strategyproof through the combination of the MIR property with our VCG-based payments, MCPP can trade off optimality and scalability by varying the number of sampled priority orderings.

In summary, we present three strategyproof path allocation mechanisms that do not rely on computationally costly combinatorial auctions and model self-interested agents in the standard MAPF setting. We prove that applying the maximal-in-range property enables the use of suboptimal MAPF algorithms in strategyproof mechanisms. In experiments across large MAPF benchmarks, we show that such mechanisms show a significant computational speedup compared to solutions relying on optimal MAPF algorithms and outperform a naïve strategyproof baseline in terms of agent welfare.

2 Related Work

Mechanism design is used in a wide range of domains where goods must be allocated to self-interested agents. Example domains which have seen high-profile implementations include electricity markets [Cramton, 2017], spectrum auctions [Cramton, 1997], course allocation at universities [Budish, 2011; Othman *et al.*, 2010], vaccine allocation [Castillo *et al.*, 2021] and large-scale sourcing auctions, for instance for transportation services [Sandholm, 2013]. Given some chosen allocation rule, depending on the rule's structure, it may be possible to design a payment rule such that every agent has a dominant strategy to report her preferences truthfully, no matter what the other agents report [Myerson, 1981].

Mechanism design where the allocated goods are paths usually fails to capture important characteristics of the MAPF domain. Ridesharing [Ma *et al.*, 2022] matches riders looking to go from A to B with drivers and aligns incentives for both sides with payments. Road pricing [Beheshtian *et al.*, 2020] facilitates the optimally efficient movement of agents through a transportation network by charging higher payments for travel in areas with high congestion. Both approaches leave the decision on what route to take up to individual agents, meaning that collisions are not considered and cannot be prevented. *UAV traffic management (UTM)* [Seuken *et al.*, 2022] is an emerging domain that combines heterogeneous, self-interested agents with explicit path allocations. Recent work in UTM focuses on resolving conflicts between agents, but offloads computational cost to the agent by requiring her to produce and submit her intended movement paths to the mechanism beforehand [Duchamp *et al.*, 2019].

Prior research that explicitly models the MAPF domain has mostly focused on the *cooperative* case [Stern *et al.*, 2019], with work on the *non-cooperative* case just emerging and focusing on auction-based mechanisms. Das *et al.* [2021] assume a 2D grid topology with intersection vertices and solve the online problem by running separate VCG auctions at each intersection. Amir *et al.* [2015] model paths as *bundles* of vertices, and use an iterative combinatorial auction mechanism called iBundle to sell non-colliding paths to self-interested agents in a way that achieves strategyproofness at the cost of poor scalability. Gautier *et al.* [2022] build on their work with a suboptimal mechanism that achieves a computational speed-up at the cost of losing strategyproofness. Chandra *et al.* [2023][1] provide a planning mechanism that simpli-

[1] While the paper claims strategyproofness, the relevant analysis is restricted to myopic agents only considering the next edge. Given that they employ sequential auctions for moving multiple edges, incentives do not translate to the full MAPF problem.

fies the strategy space by only having conflicting agents bid for movement priority in decentralized, repeated VCG auctions. We go a step further by reducing bid complexity for agents from valuing paths or priorities to merely providing values for a successful arrival and for their time. Furthermore, we improve on the high computational complexity of (VCG) auction-based approaches while still achieving their desired strategyproofness by using more scalable, established MAPF algorithms. Lastly, we use the classic MAPF formulation for the allocation problem and do not assume any additional topology on our graphs. Research in prioritized-based methods for MAPF has used randomization to generate initial orderings that are then modified by the algorithm [Bennewitz *et al.*, 2002]. However, unlike ours, this line of research does not consider incentives or produce strategyproof mechanisms.

3 Problem Setup

We consider the problem of assigning paths through a graph (V, E) to n self-interested agents. For example, for UAV airspace assignment, each agent is an individual UAV operator, while the graph represents the assignable airspace.

Each agent i has a start vertex s_i and a goal vertex g_i that she wants to reach as quickly as possible. Time is discretized into timesteps indexed by t. At each timestep, every agent can either move to an adjacent vertex or wait at its current vertex, both of which incurs *cost* c_i. Additionally, each agent has some *value* v_i for arriving at her goal vertex g_i. We also call the 4-tuple $\tau_i = (s_i, g_i, c_i, v_i)$ the agent's *type*. A *path* π_i for agent i is a sequence of vertices that are pairwise adjacent or identical. At the start, agents can wait at a private location (the *garage vertex*) for a number of timesteps where they do not conflict with other agents and from which the only movement option is to their start vertex s_i. A path begins at the garage or the agent's start vertex s_i and ends at her goal vertex g_i. Once an agent reaches her goal, she disappears, and her goal vertex becomes free for other agents. The cost of a path $c(\pi_i)$ is defined as $c_i|\pi_i|$. The *welfare* of an agent represents her satisfaction with a path which she is assigned and is given by her value for arrival minus cost of traveling along the path, $w_i(\pi_i) = v_i - c_i(\pi_i)$. The sum of all agents' welfare is called *social welfare*. To represent the welfare of all agents excluding i, we write $w_{-i} = \sum_{j \neq i} w_j(\pi_j)$. We further allow for the assignment of an empty path $\pi_i = \emptyset$ to any agent, which does not carry any value or cost.

Our theoretical results require the ability of our mechanisms to find feasible solutions for all instances and priority orderings given. Assuming either "empty paths" or "garage vertex & disappear-at-target" is sufficient. The latter assumption makes the MAPF instance *well-formed* [Čáp *et al.*, 2015], guaranteeing that prioritized algorithms (like EPBS and MCPP) are complete, i.e. always find feasible solutions. One of the two assumptions can be dropped at the cost of individual rationality or lower empirical scalability, respectively.

Our mechanisms can work with cost functions that use information other than $|\pi_i|$, as long as costs c_i are independent of other agents' assignments. If there is such a dependence, all proposed algorithms stop being well-defined as they use single-agent best paths that no longer exist in isolation.

We define two types of conflict: A *vertex conflict* $\langle i, j, v, t\rangle$ occurs when agents i and j attempt to occupy vertex $v \in V$ at the same timestep t; and an *edge conflict* $\langle i, j, u, v, t\rangle$ occurs when agents i and j attempt to traverse the same edge $(u, v) \in E$ in opposite directions at the same time (between $t-1$ and t). A *feasible assignment* is a set $d = \{\pi_1, \ldots, \pi_n\}$ of conflict-free paths, one for each agent. An *optimal assignment* is the feasible assignment d^* that produces the maximum social welfare, $d^* = \operatorname{argmax}_d \sum_i w_i(\pi_i^d)$. Thus we enhance the standard MAPF problem formulation, which aims to minimize *flowtime* or *makespan*, meaning the sum or the maximum of the arrival times of all agents at their goal vertices. Modeling all agents as having a value of zero and cost for time of one turns our social welfare objective into flowtime minimization, and further changing $\Sigma_i w_i$ to $\min_i w_i$ results in makespan minimization, both at no runtime cost. Since both also remove the ability for agents to misreport and thus the need for strategyproofness, we omit their discussion.

In contrast to the classic MAPF problem, we assume that agents' types are private and not *a priori* known to the mechanism that assigns the paths. Instead, each agent reports her type to the mechanism. We denote by $\hat{\tau} = (\hat{\tau}_1, \ldots, \hat{\tau}_n)$ a set of reported types that serve as the mechanism input, and by $\pi(\hat{\tau})$ the path assignment that the mechanism outputs. We assume agents are self-interested and may misreport a type $\hat{\tau}_i \neq \tau_i$ in such a way as to ensure that the mechanism selects an outcome that is favorable to them. A (social) welfare optimizing mechanism, for instance, finds the outcome which maximizes the sum of *reported agent welfare*. This outcome may be arbitrarily suboptimal for other individual agents (hurting egalitarian social welfare, that is, fairness) or the population as a whole (hurting social welfare, that is, efficiency). We assume that agents do not misreport their start and goal vertices, as they would have negative infinite welfare for receiving a path that does not take them from their start to their goal, reducing the agent's strategic choices to c_i and v_i.

To align incentives, we assume the mechanism may charge *payment* $p_i(\hat{\tau})$ to agent i upon assigning them a path. Consequently, an agent's *utility* is given by the welfare of her assignment minus the payment she is charged,

$$u_i(\hat{\tau}) = v_i - c_i\big(\pi_i(\hat{\tau})\big) - p_i(\hat{\tau}).$$

In a slight abuse of notation, we also write $u_i(\hat{\tau}_i, \hat{\tau}_{-i})$ for the utility resulting from agent i reporting $\hat{\tau}_i$ if the set of reports from the remaining agents is $\hat{\tau}_{-i}$. We assume that given any $\hat{\tau}_{-i}$, any agent i will report the type $\hat{\tau}_i$ that maximizes her utility $u_i(\hat{\tau}_i, \hat{\tau}_{-i})$. To facilitate our analysis, we restrict our attention to *strategyproof (SP)* mechanisms, where it is a dominant strategy for agents to report their true type τ_i, independent of the reports of the other agents $\hat{\tau}_{-i}$.

Additionally, we focus on *individually rational (IR)* mechanisms, that is, mechanisms that guarantee every agent (weakly) positive utility $u_i(\hat{\tau}) \geq 0$. Such mechanisms guarantee that agents are incentivized to participate in the mechanism, as they cannot lose utility doing so [Krishna, 2009].

Summary We aim to find a mechanism that, given a set of reports, outputs an assignment with a valid path (or no path) for each agent, such that it *maximizes welfare* and that the mechanism is *strategyproof* and *individually rational.*

3.1 Mechanism Design Desiderata for MAPF

We now explore how to achieve two key desiderata: strategyproofness and individual rationality.

For mechanisms such as our payment-CBS (PCBS) which find optimal allocations, strategyproofness can be guaranteed by charging agents VCG payments. However, for suboptimal mechanisms such as our exhaustive-PBS (EPBS) and Monte-Carlo prioritized planning (MCPP), an additional property called *maximal-in-range* has to be satisfied. We show that this property is sufficient to guarantee strategyproofness when combined with a variation on VCG payments, which we call *VCG-based payments.*

Definition 1. An allocation rule f satisfies the *maximal-in-range property (MIR) with range* S, if there exists a fixed subset S of all possible allocations $\mathcal{D}$ such that for all possible reports of misreportable information $\hat{\tau} = (\hat{\tau}_1, \ldots, \hat{\tau}_n)$, the allocation rule chooses the outcome in the range S which results in the highest reported welfare *("the outcome which maximizes reported welfare over S")*. In other words, iff

$$\exists S \subseteq \mathcal{D} : \forall \hat{\tau} : f(\hat{\tau}) = \operatorname*{argmax}_{d \in S} \sum_{i=1}^{n} \hat{w}_i(d).$$

Our EPBS and MCPP mechanisms use VCG payments with one variation. Let d^* be the assignment which maximizes social welfare within the mechanism's allocation rule's range S, that is, $d^* := \mathrm{argmax}_{d \in S} \sum_i \hat{v}_i - \hat{c}_i(\pi_i^d)$. The VCG approach also requires calculating the *counterfactual* assignment d^*_{-i} which maximizes social welfare in the hypothetical problem instance where agent i is not present.

Definition 2. The *VCG-based payment rule* charges to agent i the difference in total reported welfare of all agents excluding i of the factual and counterfactual welfare-maximizing assignment: $p_i(\hat{\tau}) := \hat{w}_{-i}(d^*_{-i}) - \hat{w}_{-i}(d^*)$.

Intuitively, the payment represents the amount by which agent i's presence has caused a worse assignment and lowered social welfare for all other agents, also called agent i's *externality*. Typically, calculating all d^*_{-i} requires solving the underlying optimization problem once per agent for an altered problem instance where that agent is not present. This is extremely costly in the MAPF domain. Our EPBS and MCPP mechanisms use a slightly different d^*_{-i}, which sidesteps this requirement and ensures that payments are non-negative. They take all assignments within their range S, set agent i's value and cost to 0 but leave the assignment's paths unchanged (i.e., agent i's path is still present), and select the so altered assignment with the highest welfare. Equivalently, $d^*_{-i} = \mathrm{argmax}_{d \in S} \sum_{j \neq i} \hat{v}_j - \hat{c}_j(\pi_j^d)$. All counterfactual assignments are thus drawn from the range that was already explored when finding d^*, avoiding costly recalculation.

To ensure *individual rationality*, or non-negative utilities for all agents, our mechanisms will not assign paths to agents if the path's cost exceeds the agent's value. If an agent is assigned a path whose cost is greater than the value the agent would gain from completing it, i.e., if $c_i|\pi_i| > v_i$, we cap the agent's cost for that path at v_i and assume that agent i will choose not to move at all. Formally, we define $c_i(\pi_i) := \min(v_i, c_i|\pi_i|)$ such that an agent's welfare becomes $w_i := v_i - c_i(\pi_i) = \max(0, v_i - c_i|\pi_i|)$. If an agent is assigned a path for which she would have negative welfare and decides not to take that path due to individual rationality, our adjustment sets her welfare to zero.[2] Then

$$d^* = \operatorname*{argmax}_{d \in S} \sum_i \hat{w}_i(d) = \operatorname*{argmax}_{d \in S} \sum_{j \neq i} \hat{w}_i(d) = d^*_{-i},$$

which implies that $p_i = 0$, and since $w_i = 0$ also $u_i = 0$.

Lemma 1. *Let (f, p) be a mechanism consisting of a path allocation rule f that satisfies MIR with range S and the VCG-based payment rule p. Then (f, p) is strategyproof.*

The proof is in the appendix and follows Nisan and Ronen's (2007) argumentation using our domain-specific adaptations.

Lemma 2. *Let (f, p) be a mechanism consisting of a path allocation rule f and the VCG-based payment rule p. Let $c_i(\pi_i) = \max(v_i, c_i|\pi_i|)$ as above. Then*

(i) The mechanism (f, p) is individually rational.

(ii) For all agents i and reported types $\hat{\tau}$, $p_i(\hat{\tau}) \geq 0$.
That is, the mechanism has no negative payments.

The proof is in the appendix.

A welfare-maximizing MAPF algorithm fulfills MIR as long as two conditions are met. First, an agent cannot change the range of assignments explored by the algorithm by misreporting. Second, the algorithm breaks ties between equal-length paths in a way that is independent of the agents' reports (which we assume). Together, they ensure that the mechanism constructs the same range for all possible agent reports.

Standard suboptimal MAPF algorithms like EECBS [Li *et al.*, 2021] may be a natural first target in search of scalability. Yet, none, to our knowledge, fulfill the MIR property, as their search tree expansion makes use of heuristics that agents can influence by misreporting their cost. Both of our suboptimal mechanisms use priority-based allocation rules. We show that this class of MAPF algorithms can be altered in natural ways to achieve MIR and thus strategyproofness.

4 Mechanisms

We present three mechanisms that solve the classical MAPF problem formulation for self-interested agents. All three fulfill the mechanism design desiderata of *strategyproofness*, *individual rationality* and *no negative payments*.

4.1 Payment-CBS (PCBS)

PCBS constitutes our optimal benchmark mechanism. Given reported agent types, it first finds a reported social welfare maximizing allocation $d^* \in \mathcal{D}$ using *conflict-based search (CBS)* [Sharon *et al.*, 2015] and then charges agents VCG payments. As it fulfills the definition of a VCG mechanism, it is *strategyproof* (see Proof of Lemma 1), and by Lemma 2 it is *individually rational* and charges *no negative payments*.

While EPBS and MCPP compute all outcomes within their range before selecting the welfare-maximal one, PCBS does

[2] MCPP and EPBS treat empty, 0-welfare paths as space unavailable to lower priority agents, a necessary compromise to avoid positive externalities and enable computational cheapness of payments.

not. To find the VCG payment for an agent, CBS must be run on the setting excluding that agent in order to obtain the counterfactual d^*_{-i}. In total, the mechanism runs CBS once to find the optimal allocation d^* and again for each $i = 1, \dots, n$ to compute d^*_{-i} and thus p_i. However, these runs could occur in parallel, reducing PCBS's real runtime to that of regular CBS' (given sufficiently many CPUs). While it thus shares a level of limited scalability with the main optimal, strategyproof benchmark of iBundle [Amir *et al.*, 2014], PCBS eliminates iBundle's prohibitive communication complexity.

4.2 Exhaustive-PBS (EPBS)

Our EPBS mechanism, derived from *priority-based search (PBS)* [Ma *et al.*, 2019], illustrates how a suboptimal MAPF algorithm can be adapted to fulfill the MIR property. Using a two-stage algorithm it constructs a *priority ordering*, i.e. a strict partial order on the set of agents. Starting from an empty priority ordering and a set of individually optimal but colliding paths, it builds a search tree where each node contains a priority ordering and a movement plan with its cost and list of collisions. It iteratively expands a node, picks a collision between two agents and resolves it by expanding into two child nodes, which each add one of the two possible priority relations between the two agents to its priority ordering, ensuring that the priority ordering stays transitive. Paths are re-planned using the new priority ordering with a single-agent low-level search such as A^*, collisions are tracked, welfare is calculated, and the next node is expanded. Once all nodes are expanded to collision-free leaves, EPBS terminates and returns the highest welfare leaf node's allocation. In contrast, PBS uses depth-first search, expanding the highest-welfare child and terminating once a single node is collision-free.

To achieve welfare-maximizing allocations with certainty, we would in principle need to search through, and construct assignments for, all possible priority orderings. Using the MIR property, we reduce the size of our search space to just the leaves of EPBS's search tree. The search tree's depth is $\mathcal{O}(n^2)$, as each node expansion adds one pair of agents to the priority ordering and no branch can add the same pair twice.

While finding the allocation d^* in the worst case thus scales exponentially with the number of agents, by fully expanding its search tree EPBS calculates all outcomes within its range in one go. Due to the variation we make in defining VCG-based payments, the range includes all counterfactuals d^*_{-i}. Thus, to run the mechanism, meaning to find d^* and calculate all payments p_i, the search tree needs to be constructed only once.[3] Building this tree could be parallelized to some degree [Świechowski *et al.*, 2023] (although we have not done so in this work), as each new child node's path replanning task based on its unique ordering requires only the information in its parent node and does not need to access a data structure that is shared with nodes other than its own future children.

Proposition 1. *EPBS is strategyproof.*

Proof. EPBS uses a lexicographic ordering of agents (which we assume to be fixed) and their reports of start- and goal vertices (which we assume to not be misreportable) for exploring assignments. Given a set of agent types that vary only in misreportable information, namely costs c_i and values v_i, EPBS constructs the same search tree from each set, resulting in the same set of leaf nodes that contain collision-free assignments $\mathcal{D}_{\text{PBS}}$. Therefore this set of assignments fulfills the definition of EPBS's *range*. Since EPBS chooses the welfare maximizing assignment over $\mathcal{D}_{\text{PBS}}$, it fulfills *MIR* and is thus made strategyproof by its payments according to Lemma 1. □

Corollary 1. *EPBS is individually rational and has non-negative payments.*

The proof follows directly from Lemma 2.

4.3 Monte-Carlo Prioritized Planning (MCPP)

MCPP demonstrates how MIR allows for creating strategyproof MAPF mechanisms that scale to the theoretical optimum, at the cost of suboptimality. Our solution utilizes *prioritized planning (PP)* [Erdmann and Lozano-Perez, 1987]. In contrast to (E)PBS, PP takes as an input a *total priority ordering* $\succ$ and expands it into a set of conflict-free paths by sequentially assigning each agent (starting with the highest priority one) the shortest possible path that avoids all agents of higher priority. In this way, calculating one assignment takes only n low-level path-finding searches.

Our MCPP mechanism (see Mechanism 1) samples a subset O of all possible total priority orderings in a way that is independent of agent reports, then runs prioritized planning on each ordering in O yielding a range of path assignments $\mathcal{D}_O$, selects the welfare-maximizing assignment $d^* \in \mathcal{D}_O$ and charges VCG-based payments. Concretely, MCPP randomly samples m total priority orderings from the space of $n!$ possible ones. We can directly trade off computational efficiency and optimality by changing the number of samples m. Increasing m increases the probability of O containing orderings that generate near-optimal assignments, and thus the expected welfare of MCPP. We avoid the usually costly computation of counterfactual outcomes for the VCG payments, as the counterfactual d^*_{-i} is selected from the range $\mathcal{D}_O$, which is already fully explored during the finding of d^*. The computational complexity of MCPP therefore scales linearly with the number of agents and the number of samples m that are selected, a drastic improvement over the exponential scaling of PCBS, EPBS and auction-based strategyproof MAPF mechanisms. As the sampled orderings are independent, their m outcomes could be calculated in parallel, resulting in a real runtime for MCPP of just n calls of A^*.

Proposition 2. *MCPP is strategyproof.*

Proof. MCPP generates a subset O of priority orderings without considering agent reports. Since each ordering $\succ$ in O corresponds to a unique path assignment $d_\succ$, the set $\mathcal{D}_O = \{d_\succ | \succ \in O\}$ fulfills the definition of MCPP's *range*. As MCPP chooses the welfare-maximising assignment over $\mathcal{D}_\succ$, it fulfills *MIR* and is thus made strategyproof by its payments according to Lemma 1. □

[3] Here we touch on a fundamental question in *algorithmic mechanism design* posed by [Nisan and Ronen, 2001]: How often does one need to solve the underlying optimization problem to obtain n agents' VCG payments? Hershberger and Suri [2001] prove that the answer is one in the domain of finding a shortest path in a network where edges with costs represent self-interested agents.

Mechanism 1: MCPP

```
Input: Agent reports $(\tau_1, \dots, \tau_n)$, No. of samples $m$
Output: Conflict-free paths $(\pi_1, \dots, \pi_n)$ and
        payments $(p_1, \dots, p_n)$
Allocation rule: Prioritized Planning
Draw a set $O$ of $m$ total priority orderings $\succ$
  independently from agent reports.
for $\succ \in O$ do
    $d_{\succ} \leftarrow PP(\succ)$
Result: set of assignments $\{d_{\succ} \mid \succ \in O\} = \mathcal{D}_O$, each
  assignment contains conflict-free paths $(\pi_1, \dots, \pi_n)$
$w(d) \leftarrow \sum_i \big(v_i - c(\pi_i^d)\big)$ for all $d \in \mathcal{D}_O$
$d^* \leftarrow \operatorname{argmax}_{d \in \mathcal{D}_O} w(d)$, the welfare maximising
  assignment over $\mathcal{D}_O$
Payment rule: VCG-based payments
for agents $i = 1, \dots, n$ do
    $w_{-i}(d) \leftarrow \sum_{j \neq i} \big(v_j - c(\pi_j^d)\big)$ for all $d \in \mathcal{D}_O$
    $d^*_{-i} \leftarrow \operatorname{argmax}_{d \in \mathcal{D}_O} w_{-i}(d)$
    $p_i \leftarrow w_{-i}(d^*_{-i}) - w_{-i}(d^*)$
Assign each agent $i$ the path $\pi_i \in d^*$
Charge each agent $i$ the payment $p_i$
```

Corollary 2. *MCPP is individually rational and has non-negative payments.*

The proof follows directly from Lemma 2.

Remark. If there is public knowledge about agents, we can also incorporate it. For example, if it is known that an agent generally has high costs across all problem instances, we can improve the expected welfare of the mechanism by restricting O to total orders that assign that agent a high priority.

5 Experiments

Our experiments verify that our mechanisms' behavior follows our claims on optimality, scalability, and payments. Our benchmarks are strategyproof mechanisms whose allocation rule solves classical MAPF, iBundle [Amir *et al.*, 2015] being the most advanced competition. We could not obtain its reference implementation to be able to accurately compare runtimes. In the original benchmarks [Amir *et al.*, 2014], iBundle shows similar scalability as CBS. As our PCBS, if parallelized, has identical time complexity as CBS, the comparison can be extrapolated. We also do not compare scalability against SOTA approximate MAPF algorithms, as they are not strategyproof and feature structures unsuited for agent cost-weighted social welfare objectives.

5.1 Experiment Setup

We evaluate payment-CBS (PCBS), exhaustive-PBS (EPBS) and Monte-Carlo prioritized planning (MCPP) on four 2D maps selected from the commonly used MAPF benchmarks by Stern *et al.* [2019]. The appendix contains evaluations on derived 3D maps. We include the so-called First-Come-First-Serve (FCFS) allocation rule, which runs prioritized planning on a single randomly sampled priority ordering as a lower bound for achieved welfare and runtime, and does not include payments. While suboptimal, it is the best pre-existing mechanism that scales to larger instances and is used in practice, e.g., in the EU's regulation on assigning airspace to drone traffic [The European Commission, 2021]. We independently sample each agent's value from $\mathcal{U}([0,1])$ and cost from $\mathcal{U}([0, v_i/\text{dist}(s_i, g_i)])$, where $\text{dist}(s_i, g_i)$ is the length of the shortest path from s_i to g_i in isolation. This ensures that near-optimal paths are likely to result in positive welfare. MCPP uses a sample size of $m = 100$, except in Figure 1. Each run uses a set of 100 instances. We set a runtime limit for all mechanisms at 3600 seconds for the *random-32-32-20* and *dens312d* maps and 5400 seconds for the *dens520d* and *Paris-1-256* maps. Success rate refers to the fraction of runs that finish within the time limit. The runtime for individual runs that do not finish within the limit is set to the limit. We did not use parallelized implementations of our mechanisms. Our code was implemented in C++ and ran on a 2x12-core Intel Xeon E5-2650v4 machine with 128 GB of RAM.

5.2 Results

Figure 1 compares the success rate, runtime and welfare over FCFS of MCPP with different sample sizes m on the *random-32-32-20* map. It demonstrates MCPP's ability to trade off scalability with solution quality as a function of sample size. As the number of agents and thus congestion increases, the average runtime of finding single shortest paths does too (for all mechanisms solving MAPF!), causing MCPP's apparent superlinear scaling.

Figure 2 compares our three mechanisms and FCFS on the standard MAPF metrics of runtime and success rate, and shows their achieved mean welfare over all instances as the ratio when divided over FCFS's welfare. If a mechanism for a given map and number of agents fails to solve all instances, we do not plot their welfare. All mechanisms, including FCFS which does not use payments, are strategyproof. FCFS, being essentially 1-sample MCPP without payments, serves as the baseline with fastest runtime and lowest welfare. PCBS, using the optimal CBS algorithm, shows the highest welfare but fails to scale to larger instances within our time limit. EPBS, using the suboptimal PBS algorithm, scales similarly. We would expect to see more differentiation in achieved welfare of these two mechanisms at settings with more agents and thus more collisions, where allocation decisions have a bigger impact on welfare. To observe this behaviour, one could parallelize PCBS, EPBS and MCPP (see Section 4) or increase the runtime limit. The figure demonstrates the exceptional scalability of MCPP compared to the search-based PCBS and EPBS while improving over the welfare baseline. The magnitude of welfare difference between the mechanisms depends on the choice of cost and value distributions and can be increased at will by e.g. employing higher variance distributions (see appendix).

Figure 3 shows the size of the VCG-based payments for a fixed mechanism, map and number of agents. We observe that payments are indeed non-negative, and overwhelmingly zero or very small. This indicates that in this setting, agents cause few externalities to others.

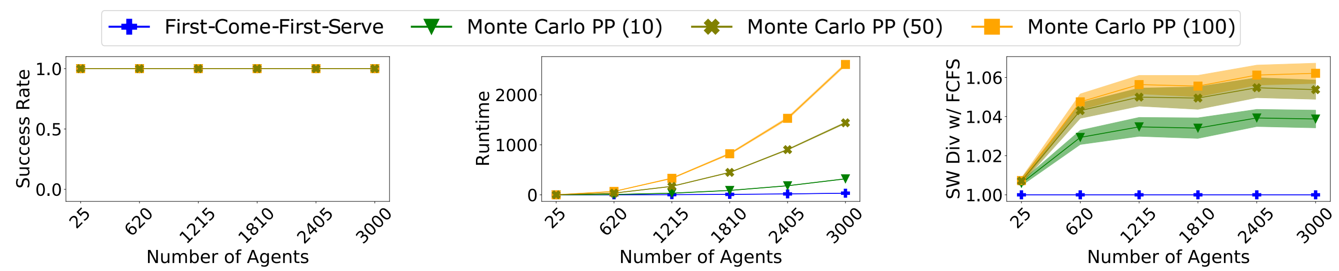
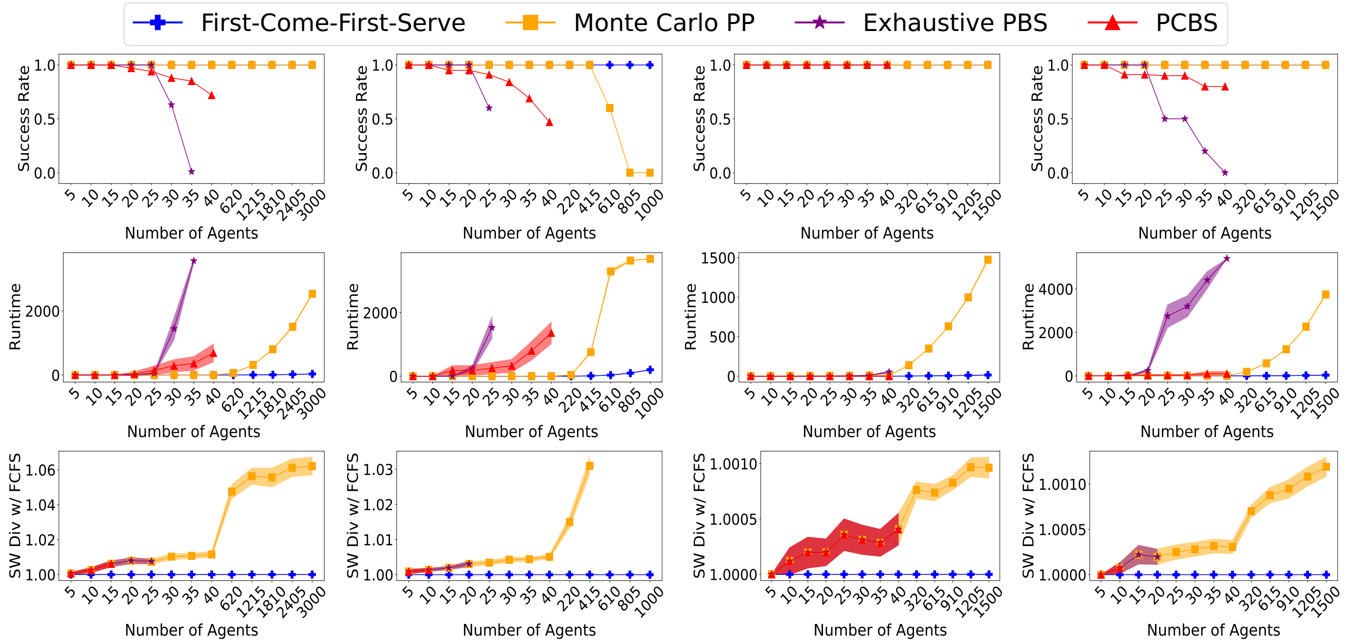


Figure 1: Success rate, runtime, and ratio-to-baseline of social welfare (SW) of Monte Carlo PP with $m \in \{10, 50, 100\}$ samples.



random-32-32-20 ***den312d*** ***Paris_1_256*** ***den520d***

Figure 2: Success rate, runtime, and ratio-to-baseline of social welfare (SW) for PCBS, EPBS, MCPP and FCFS (the baseline). Solid lines indicate the average value over 100 instances, while the shaded area is the 95% confidence interval. Maximum agent numbers are 3000 for *random-32-32-20*, 1000 for *den312d* and 1500 for *Paris_1_256* and *den520d*. Agent numbers between 5 and 40 are shown in higher granularity than between 40 and the maximum to illustrate scaling differences.

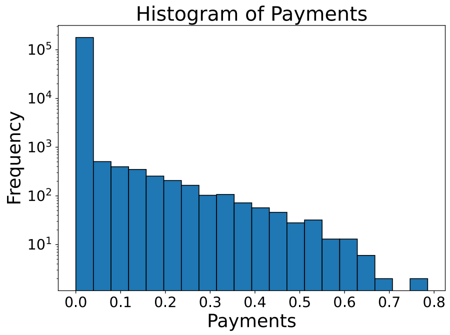


Figure 3: Size of payments for the MCPP mechanism, on the *random-32-32-20* map at $n = 1810$ agents. Frequency in the y-axis is scaled logarithmically.

6 Conclusion

We introduced a domain that bridges the gap between *mechanism design* and MAPF, where paths are allocated in the standard MAPF formulation but agents might make self-interested misreports of their costs and values.

We presented three mechanisms – PCBS, EPBS and MCPP – that are *strategyproof*, *individually rational* and have *no negative payments*, and which combine MAPF allocation algorithms with a VCG-based payment rule.

We showed how using the *maximal-in-range (MIR)* property, suboptimal MAPF allocation mechanisms like EPBS and MCPP can be made strategyproof.

Future research could find more scalable MAPF algorithms that fulfil MIR or investigate improving the welfare of suboptimal MIR-fulfilling MAPF algorithms by adjusting the range using non-misreportable agent type information.

Contribution Statement

Paul Friedrich and Yulun Zhang share equal contribution.

Acknowledgements

Tuomas Sandholm is supported by the Vannevar Bush Faculty Fellowship ONR N00014-23-1-2876, National Science Foundation grant RI-2312342, ARO award W911NF2210266, NIH award A240108S001, and CRA CI Fellow postdoctoral funding for Stephen McAleer. This paper is part of a project that has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (Grant agreement No. 805542).

References

[Amir *et al.*, 2014] Ofra Amir, Guni Sharon, and Roni Stern. An empirical evaluation of a combinatorial auction for solving multi-agent pathfinding problems. *Technical Report TR-05-14, Harvard University*, 2014.

[Amir *et al.*, 2015] Ofra Amir, Guni Sharon, and Roni Stern. Multi-agent pathfinding as a combinatorial auction. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pages 2003–2009, 2015.

[Barer *et al.*, 2014] Max Barer, Guni Sharon, Roni Stern, and Ariel Felner. Suboptimal variants of the conflict-based search algorithm for the multi-agent pathfinding problem. In *Proceedings of the International Symposium on Combinatorial Search (SoCS)*, pages 19–27, 2014.

[Beheshtian *et al.*, 2020] Arash Beheshtian, R Richard Geddes, Omid M Rouhani, Kara M Kockelman, Axel Ockenfels, Peter Cramton, and Wooseok Do. Bringing the efficiency of electricity market mechanisms to multimodal mobility across congested transportation systems. *Transportation Research Part A: Policy and Practice*, 131:58–69, 2020.

[Bennewitz *et al.*, 2002] Maren Bennewitz, Wolfram Burgard, and Sebastian Thrun. Finding and optimizing solvable priority schemes for decoupled path planning techniques for teams of mobile robots. *Robotics and Autonomous Systems*, 41(2):89–99, 2002.

[Budish, 2011] Eric Budish. The combinatorial assignment problem: Approximate competitive equilibrium from equal incomes. *Journal of Political Economy*, 119(6):1061–1103, 2011.

[Čáp *et al.*, 2015] Michal Čáp, Peter Novák, Alexander Kleiner, and Martin Selecký. Prioritized planning algorithms for trajectory coordination of multiple mobile robots. *IEEE Transactions on Automation Science and Engineering*, 12(3):835–849, 2015.

[Castillo *et al.*, 2021] Juan Camilo Castillo, Amrita Ahuja, Susan Athey, Arthur Baker, Eric Budish, Tasneem Chipty, Rachel Glennerster, Scott Duke Kominers, Michael Kremer, Greg Larson, Jean Lee, Canice Prendergast, Christopher M. Snyder, Alex Tabarrok, Brandon Joel Tan, and Witold Wiecek. Market design to accelerate COVID-19 vaccine supply. *Science*, 371(6534):1107–1109, 2021.

[Chandra *et al.*, 2023] Rohan Chandra, Rahul Maligi, Arya Anantula, and Joydeep Biswas. SOCIALMAPF: Optimal and efficient multi-agent path finding with strategic agents for social navigation. *IEEE Robotics and Automation Letters*, 2023.

[Clarke, 1971] Edward H Clarke. Multipart pricing of public goods. *Public Choice*, pages 17–33, 1971.

[Cramton, 1997] Peter Cramton. The FCC spectrum auctions: An early assessment. *Journal of Economics & Management Strategy*, 6(3):431–495, 1997.

[Cramton, 2017] Peter Cramton. Electricity market design. *Oxford Review of Economic Policy*, 33(4):589–612, 2017.

[Das *et al.*, 2021] Sankar Narayan Das, Swaprava Nath, and Indranil Saha. OMCoRP: An online mechanism for competitive robot prioritization. In *Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS)*, pages 112–121, 2021.

[Doole *et al.*, 2020] Malik Doole, Joost Ellerbroek, and Jacco Hoekstra. Estimation of traffic density from drone-based delivery in very low level urban airspace. *Journal of Air Transport Management*, 88:101862, 2020.

[Duchamp *et al.*, 2019] Vincent Duchamp, Billy Josefsson, Tatiana Polishchuk, Valentin Polishchuk, Raúl Sáez, and Richard Wiren. Air traffic deconfliction using sum coloring. In *Proceedings of the IEEE/AIAA Digital Avionics Systems Conference (DASC)*, pages 1–6, 2019.

[Erdmann and Lozano-Perez, 1987] Michael Erdmann and Tomas Lozano-Perez. On multiple moving objects. *Algorithmica*, 2:477–521, 1987.

[Gautier *et al.*, 2022] Anna Gautier, Alex Stephens, Bruno Lacerda, Nick Hawes, and Michael Wooldridge. Negotiated path planning for non-cooperative multi-robot systems. In *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 472–480, 2022.

[Gordon *et al.*, 2021] Ofir Gordon, Yuval Filmus, and Oren Salzman. Revisiting the complexity analysis of conflict-based search: New computational techniques and improved bounds. In *Proceedings of the International Symposium on Combinatorial Search (SoCS)*, pages 64–72, 2021.

[Groves, 1973] Theodore Groves. Incentives in teams. *Econometrica: Journal of the Econometric Society*, pages 617–631, 1973.

[Hershberger and Suri, 2001] J. Hershberger and S. Suri. Vickrey prices and shortest paths: What is an edge worth? In *Proceedings of the IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 252–259, 2001.

[Krishna, 2009] Vijay Krishna. *Auction theory*. Academic press, 2009.

[Li *et al.*, 2021] Jiaoyang Li, Wheeler Ruml, and Sven Koenig. EECBS: A bounded-suboptimal search for multi-agent path finding. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pages 12353–12362, 2021.

[Ma *et al.*, 2019] Hang Ma, Daniel Harabor, Peter J. Stuckey, Jiaoyang Li, and Sven Koenig. Searching with consistent prioritization for multi-agent path finding. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pages 7643–7650, 2019.

[Ma *et al.*, 2022] Hongyao Ma, Fei Fang, and David C. Parkes. Spatio-temporal pricing for ridesharing platforms. *Operations Research*, 70(2):1025–1041, 2022.

[Myerson, 1981] Roger B Myerson. Optimal auction design. *Mathematics of Operations Research*, 6(1):58–73, 1981.

[Nisan and Ronen, 2001] Noam Nisan and Amir Ronen. Algorithmic mechanism design. *Games and Economic Behavior*, 35(1-2):166–196, 2001.

[Nisan and Ronen, 2007] Noam Nisan and Amir Ronen. Computationally feasible VCG mechanisms. *Journal of Artificial Intelligence Research*, 29:19–47, 2007.

[Othman *et al.*, 2010] Abraham Othman, Tuomas Sandholm, and Eric Budish. Finding approximate competitive equilibria: Efficient and fair course allocation. In *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 873–880, 2010.

[Sandholm, 2002] Tuomas Sandholm. eMediator: A next generation electronic commerce server. *Computational Intelligence*, 18(4):656–676, 2002.

[Sandholm, 2013] Tuomas Sandholm. Very-large-scale generalized combinatorial multi-attribute auctions: Lessons from conducting $60 billion of sourcing. In Nir Vulkan, Alvin E. Roth, and Zvika Neeman, editors, *The Handbook of Market Design*, chapter 14. Oxford University Press, 2013.

[Seuken *et al.*, 2022] Sven Seuken, Paul Friedrich, and Ludwig Dierks. Market Design for Drone Traffic Management. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pages 12294–12300, 2022.

[Sharon *et al.*, 2015] Guni Sharon, Roni Stern, Ariel Felner, and Nathan R. Sturtevant. Conflict-based search for optimal multi-agent pathfinding. *Artificial Intelligence*, 219:40–66, 2015.

[Silver, 2005] David Silver. Cooperative pathfinding. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE)*, pages 117–122, 2005.

[Stern *et al.*, 2019] Roni Stern, Nathan R. Sturtevant, Ariel Felner, Sven Koenig, Hang Ma, Thayne T. Walker, Jiaoyang Li, Dor Atzmon, Liron Cohen, T. K. Satish Kumar, Roman Barták, and Eli Boyarski. Multi-agent pathfinding: Definitions, variants, and benchmarks. In *Proceedings of the International Symposium on Combinatorial Search (SoCS)*, pages 151–158, 2019.

[Świechowski *et al.*, 2023] Maciej Świechowski, Konrad Godlewski, Bartosz Sawicki, and Jacek Mańdziuk. Monte carlo tree search: A review of recent modifications and applications. *Artificial Intelligence Review*, 56(3):2497–2562, 2023.

[The European Commission, 2021] The European Commission. Commission implementing regulation (EU) 2021/664 of 22 April 2021 on a regulatory framework for the U-space (text with EEA relevance). *Official Journal of the European Union*, 64(L139):161–186, 2021.

[Vickrey, 1961] William Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *The Journal of Finance*, 16(1):8–37, 1961.

[Yu and LaValle, 2013] Jingjin Yu and Steven LaValle. Structure and intractability of optimal multi-robot path planning on graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pages 1443–1449, 2013.