

# Bounded Suboptimal Multi-Agent Path Finding Using Highways \*

Liron Cohen and Sven Koenig

Department of Computer Science, University of Southern California  
 {lironcoh,skoenig}@usc.edu

## 1 Background

The multi-agent path finding (MAPF) problem is defined as follows: Given a graph and a set of agents with unique start and goal vertices, find collision-free paths for all agents from their respective start vertices to their respective goal vertices. Our objective is to minimize the the total arrival time. MAPF has many applications such as video games, traffic control and robotics. Solving the MAPF problem optimally, whether the objective is to minimize the makespan, the total distance traveled or the total arrival time, is NP-hard [Yu and LaValle, 2013]. Optimal MAPF approaches, such as Conflict-Based Search (CBS) [Sharon *et al.*, 2015] and  $M^*$  [Wagner and Choset, 2015], therefore can be used only for low numbers of agents. Yet, finding low-cost solutions is important because the same throughput can then be achieved with fewer agents. It is therefore common to use bounded-suboptimal<sup>1</sup> MAPF approaches for higher numbers of agents. Existing bounded-suboptimal MAPF approaches, such as Enhanced-CBS (ECBS) [Barer *et al.*, 2014] and inflated- $rM^*$  [Wagner and Choset, 2015], find solutions quickly when the MAPF instances are not tightly-coupled (that is, when ECBS does not need to resolve many conflicts or inflated- $rM^*$  does not need to expand large subdimensions). However, in many application domains this is not the case, and the runtime of these approaches degrades severely. For example, ECBS was shown to scale poorly on a model of an automated warehousing system used by Amazon (formerly called Kiva) with many agents and long narrow corridors [Cohen *et al.*, 2015].

## 2 Using Highways

One of the pitfalls of the aforementioned bounded-suboptimal MAPF approaches is that they use the distances to the goal vertex of an agent as heuristic values to guide the search. These “shortest-path” heuristic values are computed for every agent independently, taking into account obstacles but abstracting away all other agents. They are more informed than the Manhattan distances and can be computed fast compared to solving a MAPF instance. However, using the distances

as heuristic values does not necessarily help the MAPF approaches to find a solution quickly. In light of this observation, we have developed a new bounded-suboptimal MAPF approach, called ECBS+HWY [Cohen *et al.*, 2015]. It exploits the problem structure of a given MAPF instance by finding paths for the agents that include edges from user-provided sets of edges (called highways). ECBS+HWY uses highways in the context of a simple inflation scheme based on the ideas behind experience graphs [Phillips *et al.*, 2012] to derive “highway” heuristic values that encourage the search to return paths that include the edges of the highways. These heuristic values correspond to distances, but for different edge costs. Specifically, we inflate the cost of each non-highway edge by an inflation factor  $w > 1$  (a user specified constant) when calculating the heuristic values.

Figure 1(a) shows a MAPF instance with two agents as an example. The shortest-path heuristic values for the agents are shown in (c) and (d), and the highway heuristic values for the agents are shown in (e) and (f) for  $w = 2$  and the highway shown in (b). For instance, the shortest-path heuristic value for agent 1 and the cell in row 2 and column 3 is 2 (corresponding to moving east and south, each with cost one) while the highway heuristic value is 3 (corresponding to moving east with cost two and then following the highway south with cost one). Highways are a convenient way for users to influence the heuristic values so that the paths of agents follow the given highways. This global behavior, for a suitable highway, helps to reduce head-on collisions. Since the runtimes of CBS and ECBS increase with the number of collisions, using highways has the potential to speed them up. For example, the paths shown in Figure 1(g) were found by CBS for the objective of minimizing total arrival time, using the shortest-path heuristic values. They are an optimal solution (with cost 9) and required 277 node expansions. This is compared to only 31 node expansions when using the highway heuristic values (albeit with a higher cost of 12). In [Cohen *et al.*, 2015], we prove that CBS with highway heuristic values, computed with an inflation factor  $w$ , is  $w$ -suboptimal. We also demonstrate experimentally that ECBS+HWY outperforms ECBS (in terms of runtime and solution cost) in Kiva-like domains with many agents if the highway’s layout captures the domain structure well (that is, the highway edges are such that agents have incentives to move in only one direction in each corridor). Figures 3 and 4 show some experimental results.

\*Our research was supported by NSF under grant numbers 1409987 and 1319966.

<sup>1</sup>Search algorithms are called  $w$ -suboptimal for a user-provided suboptimality bound  $w$  if and only if they return solutions of cost at most  $w \cdot opt$ , where  $opt$  is the cost of an optimal solution.

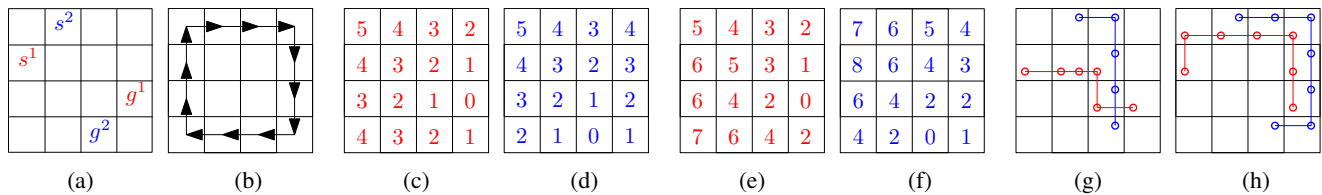


Figure 1: Differences between using CBS with the shortest-path and the highway heuristic values in a pathological domain where CBS is extremely inefficient. (a) shows the start and goal vertices for the two agents. (b) shows the highway. (c) and (d) show the shortest-path heuristic values for agents 1 and 2, respectively. (e) and (f) show the highway heuristic values for  $w = 2$  for agents 1 and 2, respectively. (g) and (h) shows the paths found by CBS using the shortest-path and the highway heuristics, respectively. Each dot represents a time step. Two dots in the same cell thus represent a wait action.

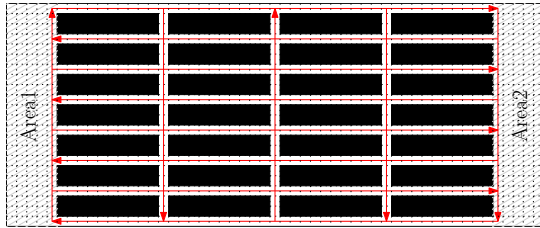


Figure 2: Our model of a Kiva-like domain. Highway edges are represented by red arrows. For a given number of agents, we generate instances by assigning half of the agents a randomly chosen start location in Area1 and a randomly chosen goal location in Area2, and vice-versa for the other half.

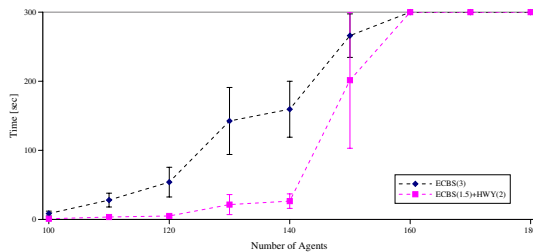


Figure 3: Runtimes of ECBS and ECBS+HWY on the domain in Figure 2 with varying number of agents. Each data point is the median runtime over 10 trials. A vertical bar around a data point indicates the median absolute deviation.

### 3 Generating Highways Automatically

Intuitively, ECBS+HWY outperforms ECBS because it leverages high-level guidance from highways. So far, the highways used with ECBS+HWY have been human-generated. Currently, we are developing automatic ways of generating highways for a given MAPF instance to reduce the dependency of ECBS+HWY on human expertise. We aim to show that ECBS+HWY that uses our automatically generated highways is competitive with ECBS+HWY that uses human-generated ones. We are exploring different methods for generating highways automatically based on the following intuition: While solving the MAPF problem optimally is NP-hard, the shortest path for each agent independently can be computed fast. Although this does not solve a MAPF instance, the cumulative information in these shortest paths can

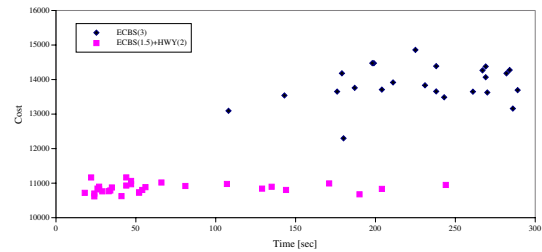


Figure 4: Performances of ECBS and ECBS+HWY on the domain in Figure 2. Each data point is an instance with 150 agents solved within 5 minutes.

be used to design effective highways quickly. First results can be found in [Cohen *et al.*, 2016].

### References

- [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 7th Annual Symposium on Combinatorial Search*, 2014.
- [Cohen *et al.*, 2015] Liron Cohen, Tansel Uras, and Sven Koenig. Feasibility study: Using highways for bounded-suboptimal multi-agent path finding. In *Proceedings of the 8th Annual Symposium on Combinatorial Search*, 2015.
- [Cohen *et al.*, 2016] Liron Cohen, T. K. Satish Kumar, Tansel Uras, Hong Xu, Nora Ayanian, and Sven Koenig. Improved bounded-suboptimal multi-agent path finding solvers. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence*, 2016.
- [Phillips *et al.*, 2012] Michael Phillips, Benjamin J. Cohen, Sachin Chitta, and Maxim Likhachev. E-graphs: Bootstrapping planning with experience graphs. In *Proceedings of the 8th Robotics: Science and Systems Conference*, 2012.
- [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.
- [Wagner and Choset, 2015] Glenn Wagner and Howie Choset. Subdimensional expansion for multirobot path planning. *Artificial Intelligence*, 219:1–24, 2015.
- [Yu and LaValle, 2013] Jingjin Yu and Steven M. LaValle. Structure and intractability of optimal multi-robot path planning on graphs. In *Proceedings of the 27th AAAI Conference on Artificial Intelligence*, 2013.