

ODrM*

Optimal Multirobot Path Planning in Low Dimensional Search Spaces

Cornelia Ferner, Glenn Wagner, Howie Choset

Abstract—We believe the core of handling the complexity of coordinated multiagent search lies in identifying which subsets of robots can be safely decoupled, and hence planned for in a lower dimensional space. Our work, as well as those of others take that perspective. In our prior work, we introduced an approach called subdimensional expansion for constructing low-dimensional but sufficient search spaces for multirobot path planning, and an implementation for graph search called M*. Subdimensional expansion dynamically increases the dimensionality of the search space in regions featuring significant robot-robot interactions. In this paper, we integrate M* with Meta-Agent Constraint-Based Search (MA-CBS), a planning framework that seeks to couple repeatedly colliding robots allowing for other robots to be planned in low-dimensional search space. M* is also integrated with operator decomposition (OD), an A*-variant performing lazy search of the outneighbors of a given vertex. We show that the combined algorithm demonstrates state of the art performance.

I. INTRODUCTION

Multirobot systems offer flexibility, sensor coverage and redundancy which makes them attractive for tasks such as surveillance, search and rescue and warehouse automation applications. Path planning for such systems is difficult due to the large number of degrees of freedom in multirobot systems. Thus, the primary focus of research in multirobot path planning is exploiting the structure of the problem to allow for exploring a low-dimensional search space, rather than the high-dimensional configuration space.

Polynomial time algorithms have been developed to find suboptimal paths for multirobot systems [5] [6] [14], by constructing paths which either minimize interactions between robots, or restrict such interactions to specific, stereotyped behaviors. Planning cost optimal paths for multirobot systems is significantly more difficult, as there is less freedom to choose paths which minimize conflicts.

A number of algorithms that produce optimal paths exploit inherent sparsity in robot-robot interactions to efficiently find optimal cost paths. When possible, robots are treated separately, or with only weak constraints. When coupling is necessary, the structure inherent to systems of multiple separate agents can be used to efficiently guide search in the joint configuration space. Conflict-Based Search (CBS)

[8] constructs a set of constraints when planning for individual robots to find optimal cost solutions without exploring higher-dimensional spaces. Meta-Agent Conflict-Based Search (MA-CBS) [9] is a variant of CBS that allows for merging robots which frequently interfere with each other, to reduce the proliferation of constraints. Operator decomposition (OD) is a variant of A* which delays consideration of joint actions for which the constituent individual robot actions are unpromising [11]. In our prior work, we introduced subdimensional expansion and its graph search implementation, M* [12], which construct a search space that initially restricts each robot to its individually optimal path. When robot-robot collisions are found, the search space is locally expanded to guarantee that a collision free path can be found. MA-CBS exploits decoupling between robots at the highest level, while OD performs very low-level decoupling. In this paper, we demonstrate that integrating M*, MA-CBS and OD provides state of the art performance in finding optimal cost paths for multirobot systems.

II. PRIOR WORK

In a theoretical study which has recently attracted attention, Kornhauser *et al.* [3] proved the existence of polynomial time algorithms that find suboptimal cost paths for multirobot systems moving in a workspace represented by a graph. Furthermore, given a graph with n vertices, such an algorithm will return a path of at most $O(n^3)$ steps.

A number of such polynomial algorithms have also been developed. Luna and Bekris introduced the Push and Swap algorithm [5], which operates on an undirected graph with at least two unoccupied vertices. Paths are constructed by applying “push” and “swap” primitives on robots where “push” refers to all robots moving towards their goal position and “swap” refers to two agents swapping positions. The resulting solutions require robots to move sequentially and thus are of low quality. A recent improvement [7] enhances the path quality by allowing simultaneous moves of robots.

Wang and Botea [14] introduced a polynomial time algorithm for *slideable* grid worlds. A grid world is slideable if there is an unoccupied cell next to the initial and goal positions of each robot. Moreover, every sequence consisting of three nodes of the initially computed paths requires an alternative of equal cost. Wang and Botea presented a polynomial time algorithm for solving these *slideable* problems.

Peasgood *et al.* described an algorithm that constructs a spanning tree to cover a topological representation of the workspace [6]. Initially, all robots are moved to the leaf nodes of the tree, whereupon robots swap positions until each

Cornelia Ferner is a graduate student at ITS, Salzburg University of Applied Sciences, 5412 Puch/Salzburg, Austria and is funded by the Austrian Marshall Plan Foundation cornelia.ferner@fh-salzburg.ac.at

Glenn Wagner is a graduate student at the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, 15213 gswagner@andrew.cmu.edu

Howie Choset is a professor at the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, 15213 choset@cs.cmu.edu

robot is at its goal location. The algorithm is complete and the complexity increases linearly with the number of robots.

Each of the described polynomial algorithms constructs paths which minimize robot-robot interactions or limit such interactions to specific, stereotyped behaviors, to minimize the cost of finding a path. Such limitations result in paths of suboptimal cost. Algorithms have been introduced which exploit sparsity of robot-robot interactions inherent in the problem definition to efficiently find optimal cost paths. Operator decomposition (OD) [11] and Enhanced Partial Expansion A* (EPEA*) [1] are variants of A* that delay the instantiation of unpromising neighbors to lazily explore the graph representation of the joint configuration space. M* [12], Meta-Agent Conflict-based Search (MA-CBS) [9] and Increasing Cost Search Tree (ICST) [10] all construct a separate search space which reflects sparse robot-robot interactions.

Our previous work introduced *subdimensional expansion*, a method of dynamically constructing a low-dimensional search space as subspace of the full configuration space. An initial search space is constructed by restricting each robot to an individually optimal path. As robot-robot collisions are detected during exploration of the search space by an underlying joint planner, the dimensionality of the search space is locally increased to ensure the presence of an optimal, collision free path. In this manner, a minimal sufficient search space is constructed for the particular problem.

Subdimensional expansion is agnostic to the choice of planning algorithm to be used to explore the constructed search space. Originally, we adapted A* to the subdimensional framework and called the resulting algorithm M^* . [12]. Additionally, probabilistic planners such as rapidly-exploring random trees (RRTs) [4] and probabilistic roadmaps (PRMs) [2] can be used as the coupled planner in subdimensional expansion, resulting in the sRRT and sPRM algorithms, respectively [13].

In this paper, we demonstrate the advantages of combining subdimensional expansion using M^* , with operator decomposition and Meta-Agent Constraint-Based Search (MA-CBS). We first replace A* with operator decomposition as the underlying planning algorithm in M^* , resulting in $ODrM^*$. We then integrate $ODrM^*$ into the MA-CBS framework, resulting in $MA-CBS+ODrM^*$, and show improved performance compared to MA-CBS combined with straight operator decomposition: Of all trials including 40 robots, $MA-CBS+ODrM^*$ solves about 96%, which is 40% more than MA-CBS with OD.

III. PROBLEM DEFINITION

We wish to find an optimal, collision free path for a set of n robots $r^i, i \in \{1, 2, \dots, n\}$ from a specified initial configuration to a goal configuration where all robots are assigned unique start and goal positions. Each robot's configuration space can be represented as directed graph $G^i = \{V^i, E^i\}$. The vertices of G^i represent all possible locations a robot can occupy, while the edges are the possible transitions between

locations. The joint configuration space is represented by the Cartesian product of the individual graphs.

We evaluated our approach in a workspace represented by an eight-connected grid which naturally induces a graph representation: each node is a pixel and neighboring nodes share an edge when robots can move between them. We assume that motion among all robots is synchronous, meaning that all robots move from pixel to pixel at the same time; one possible motion is to just stay in the current pixel, so both moving and waiting is assigned a cost of one. However, a robot waiting at its goal incurs zero cost. The cost of a joint action is the sum of the costs of the actions of the individual robots. Robots are prohibited from simultaneously occupying the same vertex or traversing intersecting edges.

IV. APPROACH

In this paper, we combine subdimensional expansion (i.e. M^*) with other planning approaches to boost the overall performance when planning in large systems. Other algorithms such as Meta-Agent Constraint-Based Search (MA-CBS) or operator decomposition (OD) benefit from the variable dimensionality of the search space constructed by subdimensional expansion which is reflected in lower run time and a higher percentage of solvable problem instances.

With the resulting composite planner $MA-CBS+ODrM^*$ we seek to plan in spaces with as low as possible dimensionality: $MA-CBS+ODrM^*$ starts by planning individually optimal paths for all robots. The idea then is to avoid coupling the planning for multiple robots to keep the dimensionality of the search space as low as possible. Thus, MA-CBS - the highest level of planning - identifies minimal subsets of dependent robots for which a coupled search must be performed. $ODrM^*$ computes the coupled solution whenever necessary. Within $ODrM^*$, a minimal search space is constructed using subdimensional expansion, while operator decomposition is used to explore the search space.

A. Meta-Agent Conflict-Based Search

Meta-Agent Conflict-Based Search (MA-CBS) [9] is a planning framework introduced by Sharon *et. al* based on their *Conflict-Based Search* (CBS) planning algorithm [8]. Conflict-Based Search explores a space of constraints on individual robots, rather than the joint configuration space of the system. Each search node contains a set of constraints and the optimal path for each robot subject to the constraints. The constraints prohibit individual robots from occupying a specific node at a specific time that would lead to interference with another robot.

At each step, the search node with the smallest total path cost is examined for collisions between the individual constrained paths. If no collisions are detected, then the optimal solution has been found. If a collision is found between two robots at position q and time t , the search tree branches. Two new nodes are created, each with a constraint prohibiting one of the involved robots from occupying q at time t . New paths are then computed for each of the involved robots to obey the newly expanded set of constraints. When planning for

an individual robot, paths which do not conflict with the paths of other robots in the search node are preferred, but no additional cost will be incurred to avoid such conflicts.

While the search space for constrained planning is of constant dimensionality, the set of possible constraints grows exponentially. As a result, CBS performs poorly when there are many alternate paths which require a large number of constraints to cover. In such cases, it is more efficient to resort to coupled search to find a path for the effected robots. MA-CBS [9] is an extension of CBS which merges robots into meta-agents when more than a merge threshold B number of conflicts between a given pair of robots has been found. Planning within a meta-agent is conducted using a coupled planning algorithm, which respects the constraints placed on the meta-agent. Internal constraints upon the constituent robots are removed when merged into a meta-agent, although the new meta-agent inherits external constraints that result from collisions with other agents. MA-CBS with a given merge threshold B is denoted as MA-CBS(B). As described in [9], when $B = 0$, robots are merged the first time a collision is detected, and thus MA-CBS(0) is equivalent to Independence Detection [11]. When B is set to infinity, robots are never merged, rendering MA-CBS(∞) equivalent to CBS. Typically, smaller values of B work in more open environments with many alternate paths, resorting to coupled search earlier, while larger values of B work better in more constrained environments.

B. ODrM*

A new algorithm, *ODrM**, is employed for the coupled replanning of all robots within one meta-agent. *ODrM** results from utilizing a variant of A* called operator decomposition (OD) [11] as the underlying planner for exploring a search space constructed by subdimensional expansion [12].

Subdimensional expansion starts planning a path for a system of robots by independently computing an individually optimal path for each robot. The individually optimal paths define a one-dimensional search space embedded in the system's joint configuration space. The search space is then explored by a planning algorithm to find a collision free path. When the underlying planner visits a state at which robot-robot collisions occur, the search space is locally grown along a low dimensional subspace to ensure the presence of a path around the discovered collision (Figure 1). In this manner, a sufficient search space of minimal dimensionality is constructed.

Subdimensional expansion makes use of two fundamental concepts: the *individual policy* and the *collision set*. The individual optimal policy of the i -th robot maps the position of r^i to its optimal action at that position in the absence of other robots. Obeying its individual policy from any position produces an individually optimal path for that robot.

The collision set C_k for a given point q_k in the joint configuration space is the set of robots r^i for which the planner has found a path through q_k to a collision between r^i and another robot. When the planning algorithm extends a path from q_k , every robot not in C_k obeys its individual

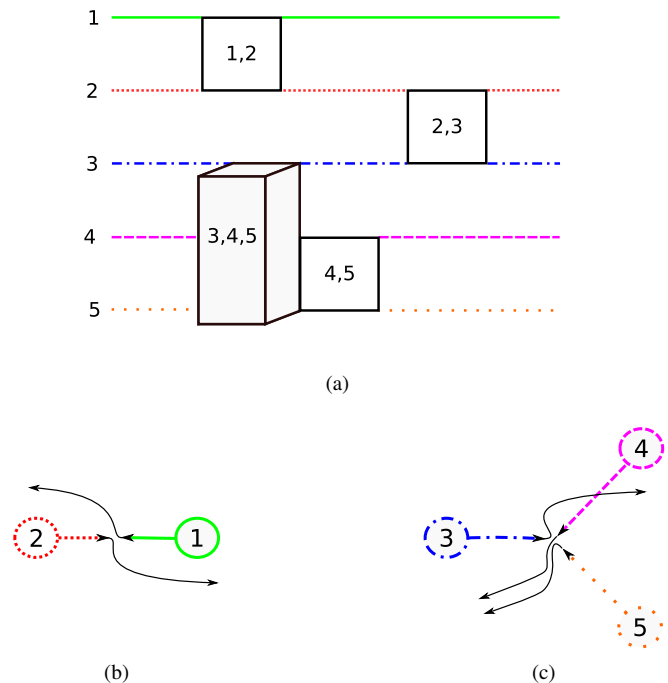


Fig. 1: (a) A conceptual visualization of a variable dimensionality search space for five robots. Initially each robot is constrained to its individual policy, represented by a single line. (b) When robots 1 and 2 collide, the local dimensionality of the search space must be increased, as represented by a square. (c) When three robots collide while following their individual policy, the local dimensionality of the search space must be increased further, represented by the cube, to include all local paths of the three robots. Once robot 3 clears robots 4 and 5, it no longer must remain coupled with them, even though robots 4 and 5 continue to interact.

policy, while all possible actions must be considered for robots in C_k , thus implicitly defining a search space of variable dimensionality.

The M^* algorithm is an implementation of subdimensional expansion for graph search [12]. The configuration space of each robot is represented as directed graph, and A* is used as the underlying path planning algorithm.

Recursive M^* (rM^*) [12] is a variant of M^* which improves the handling of physically separated sets of colliding robots. While basic M^* must couple the planning for all such sets, rM^* plans for each spatially separated set of robots independently. The computational cost is then exponential in the size of the largest set of colliding robots instead of the total number of colliding robots as is the case for basic M^* .

In *ODrM**, operator decomposition replaces A* as the underlying path planner for rM^* . A* has a fundamental flaw for multirobot path planning in that the number of out-neighbors for a single node increases exponentially with the number of robots. A* must add all out-neighbors of a node to the open list, even if many will never be expanded. OD mitigates this problem by procedurally generating the out-neighbors so that low cost neighbors are generated first, and high cost neighbors may never be instantiated. When OD expands a *standard* node, *intermediate* nodes are created to represent all possible actions of the first robot. The

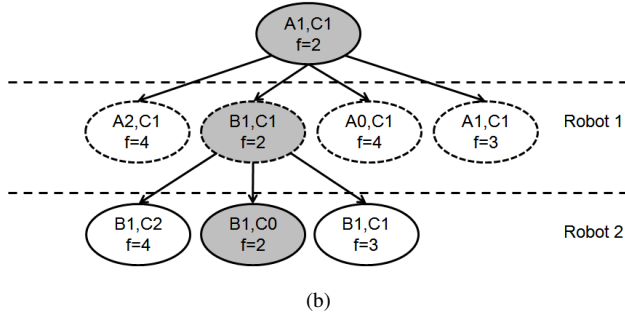
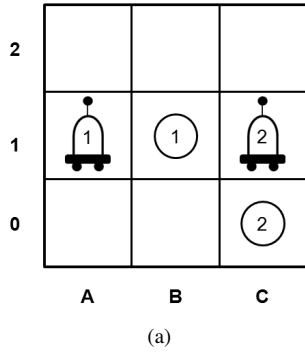


Fig. 2: (a) A simple path planning problem including two robots to be solved by operator decomposition. Nodes $A1$ and $C1$ represent the initial nodes, $B1$ and $C0$ the goals. Every robot has five actions, up, down, wait and left/right. (b) $\{A1, C1\}$ is the root node of the planning problem. Dashed lines symbolize intermediate nodes, solid lines standard nodes. The cost for the optimal solution is 2. After the expansion of the nodes for the first robot, OD only expands successor nodes for the intermediate node with optimal cost - $\{B1, C1\}$. When expanding nodes for the second robot, collisions with the new position of the first robot have to be avoided. Thus, only three nodes are generated for the second robot. Node $\{B1, C0\}$ has lowest cost. As it is the goal node, the planning problem is solved. [10, adapted]

cost and heuristic cost-to-go of the intermediate nodes are updated to reflect the new position of the first robot, then the intermediate nodes are added to the open list. When an intermediate node is expanded, more intermediate nodes are created to represent the possible actions of the next robot. Standard nodes are generated when actions are assigned for the last robot. This procedure results in the creation of standard nodes representing all robots moving towards their goals before initiating less promising nodes. Typically fewer total nodes are created, reducing the computational cost of finding a path.

Figure 2 illustrates the node expansion of operator decomposition for a problem including two robots in detail. When coupled with an optimal heuristic, operator decomposition is complete and optimal with respect to path cost. Thus, also ODrM* is guaranteed to find optimal paths.

ODrM* requires a few modifications to provide coupled planning for meta-agents within the MA-CBS framework. Both the OD planner and the individual policies for each robot must respect the constraints placed on the meta-agent by MA-CBS. Furthermore, preference must be given to equal-cost paths which avoid collision with the paths of robots not in the meta-agent.

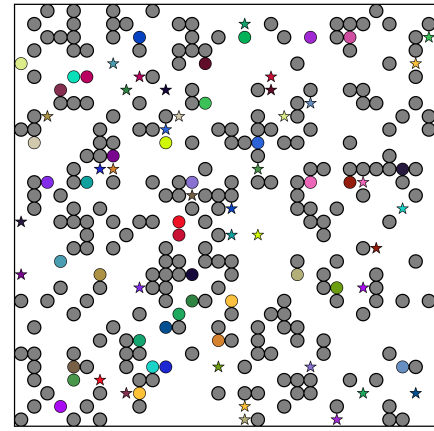


Fig. 3: A typical 8-connected grid world with 32×32 cells for a test run including 40 robots. Colored circles represent the initial positions of the robots, colored stars their goal positions. Gray circles represent the obstacles. 100 random environments were generated for each test point.

V. RESULTS

To test the algorithms, we ran simulations on a Core i5-2500 computer at 3.30 GHz (Turbo Boost disabled) with 8 GB of RAM. All simulations were implemented in python. We chose to use a fixed-size, eight-connected grid of 32×32 cells, with a probability of 20% of a given cell being marked as an obstacle, as in [11]. Initial and goal positions for each robot were chosen randomly within the same connected component of the workspace (Figure 3).

Each trial was given five minutes to find a solution. For each number of robots from 5 to 60, we tested 100 random environments. We recorded the percentage of trials that were successful within 5 minutes as well as the median time required to find solutions. Run time is plotted on a logarithmic axis.

A. OD vs. rM* vs. ODrM*

We first demonstrate that ODrM* outperforms basic rM* and OD. Recall that rM* uses A* as underlying planning algorithm, so that rM* typically expands more nodes than ODrM*. Thus, we expected ODrM* to solve more instances within the given time limit. As shown in Figure 4, both basic rM* and ODrM* substantially outperform OD in all but the easiest test case. Basic rM* solves about 20% -30% fewer trials involving 25 or 30 robots, and 10% less trials than ODrM* for other test conditions. As ODrM* generates fewer nodes than rM*, also the runtime of the algorithm is improved as depicted in the time plots in Figure 4.

As results in Figure 4 show, ODrM* outperforms basic OD as well. OD operates in the joint configuration space that grows exponentially with every additional robot. Running OD in the search space of variable dimensionality of rM* helps to improve the performance of OD by approximately 60% for problems including 20 robots. Moreover, basic OD solves almost none of the instances involving 25 robots and more.

The time to solution plots plateau when a given percentage of trials reach the five minutes limit. Python's CPU time

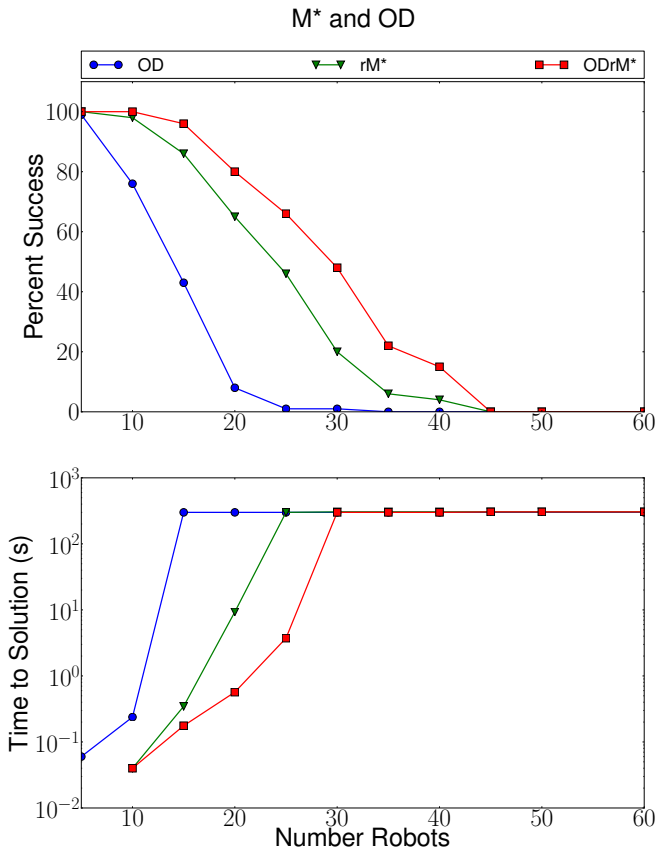


Fig. 4: Results for OD, rM* and ODrM*. The top plot illustrates the percentage of trials in which a solution was found within five minutes. The bottom graph shows the median time to find a solution.

function has a resolution of one millisecond, resulting in faster solutions being reported as taking zero time, which is dropped on a logarithmic plot.

B. CBS vs. MA-CBS(10)+OD vs. MA-CBS(10)+ODrM*

We now demonstrate that in our test environment combining ODrM* with the MA-CBS framework outperforms both basic CBS and MA-CBS combined with the OD coupled planner. MA-CBS is parameterized by a merge threshold which must be tuned to a specific problems characteristics. As mentioned earlier, a lower threshold is more beneficial in more open environments, while higher thresholds are beneficial in more constrained environments. A merge threshold of 10 was selected as appropriate for the test environment.

We compare the results of CBS, equivalent to MA-CBS(∞), with MA-CBS(10)+OD and MA-CBS(10)+ODrM* (Figure 5). MA-CBS(10)+OD outperforms CBS by a modest amount of approximately 5% for systems of 40 robots. Due to the greater power of ODrM* when compared to OD, MA-CBS(10)+ODrM* successfully finds a solution in 96% of 40 robot trials and in almost 80% in 60 robot trials. The performance benefits of MA-CBS(10)+ODrM* are most evident for the hardest problems. Easy problems are those for which few robot-robot collisions are found, resulting in little if any merging, and all algorithms are functionally equivalent

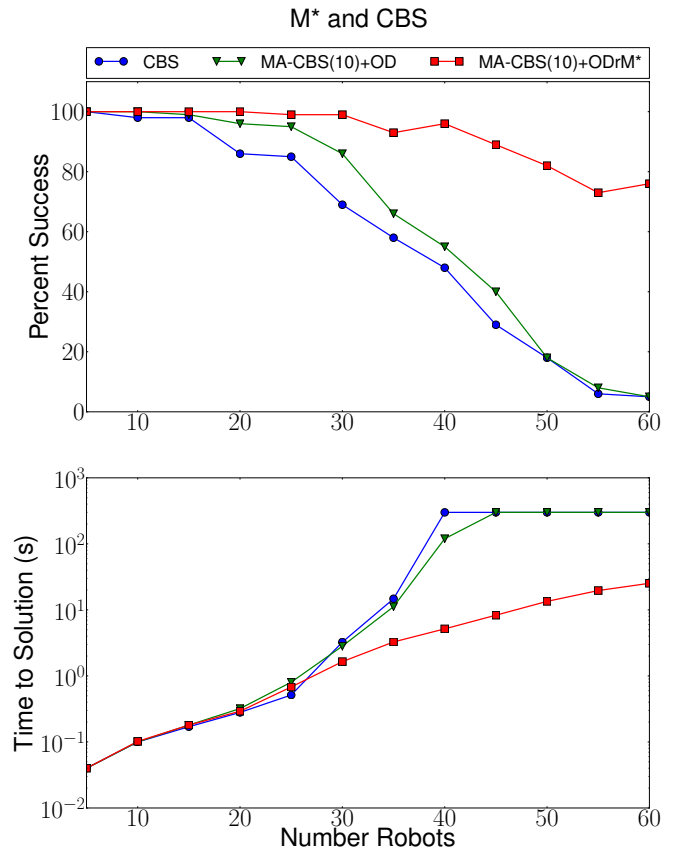


Fig. 5: Results for CBS, MA-CBS(10)+OD and MA-CBS(10)+ODrM*. The top plot illustrates the percentage of trials in which a solution was found within five minutes. The bottom graph shows the median time to find a solution.

to CBS. Harder problems require that many robot-robot conflicts be resolved, and result in the formation of many meta-agents. Under these conditions, the benefits of MA-CBS(10)+ODrM* become most evident.

C. Optimal vs Suboptimal Paths

In the previous section, we demonstrated that MA-CBS(10)+ODrM* scales well with the number of robots in the average case. However, finding optimal paths on graphs for multirobot systems is known to be NP-hard, while suboptimal paths can be found in polynomial time [3]. This raises the question of what the performance tradeoffs of MA-CBS(10)+ODrM* are compared to algorithms which return suboptimal paths.

Inflated M* is one such suboptimal path planning algorithm. Multiplying the heuristic function used by M* by an inflation factor $\epsilon > 1$ often reduces the time required to find a solution [12]. However, the path returned by inflated M* may cost up to a factor of ϵ more than the optimal path. We compare MA-CBS+ODrM* with inflated ODrM* with $\epsilon = 1.1$ (Figure 6).

As expected, inflated ODrM* finds solutions more quickly, and for significantly more problems involving 35 or more robots. The cost penalty incurred by inflated ODrM* depends on the number of robots in the system. For 5 robot systems,

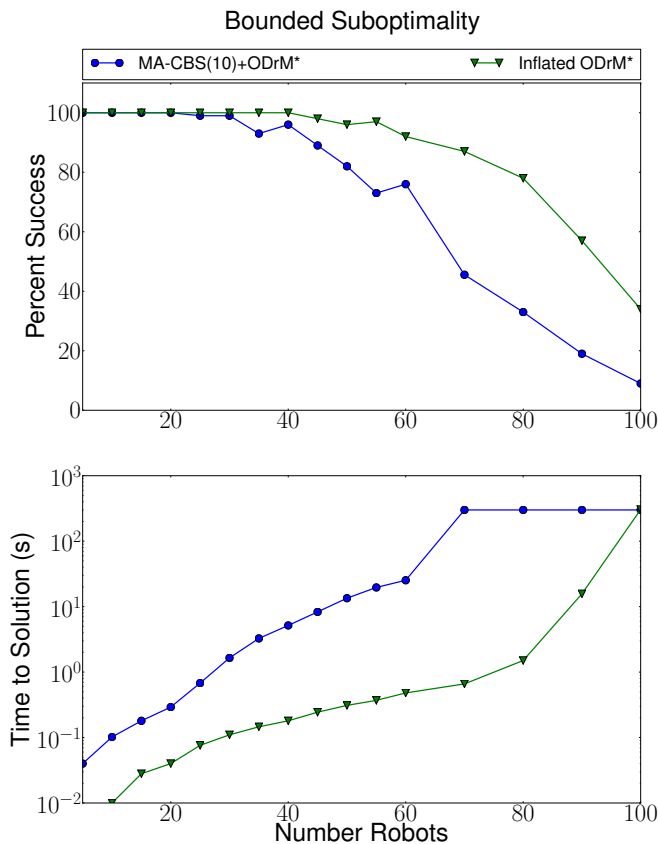


Fig. 6: Results for MA-CBS(10)+ODrM* and ODrM* with a heuristic inflated by 1.1. With the inflated heuristic, ODrM* is guaranteed to find a path that costs no more than 10% more than the optimal path. The top plot illustrates the percentage of trials in which a solution was found within five minutes. The bottom graph shows the median time to find a solution.

the penalty is a negligible 0.06%, rising to 1.9% for 60 robots, and 2.4% for 100 robots.

VI. CONCLUSION AND FUTURE WORK

We present ODrM*, a new algorithm for performing coupled planning and show that it surpasses the state of the art in coupled planning. We then show that integrating ODrM* into the MA-CBS framework provides superior performance to both CBS and MA-CBS with OD. There are two perspectives from which these results may be considered. When focus is placed on MA-CBS, it is clear that the performance of MA-CBS+ODrM* is due to the greater efficiency of ODrM* when compared to OD.

The results when considered with an emphasis on ODrM* tell a more complex story. Subdimensional expansion, the approach underlying ODrM*, draws its strength from restricting each robot to its individually optimal policy, unless a collision is observed. In many environments, such as those considered in this paper, many different optimal policies exist for each robot. The choice of individual policy then has a significant impact on the difficulty of finding a solution, as some policies may entirely avoid collision between robots, while others promote collisions. The individual policies of

ODrM* for a given meta-agent obey the constraints imposed by MA-CBS, and attempt to avoid collisions with robots not in the meta-agent. As a result, MA-CBS can be viewed in part as a method for optimizing the choice of individual policy for ODrM*.

We are interested in applying MA-CBS+ODrM* to unknown environments. ODrM* would be modified to use D* with OD-style node expansion to provide efficient recomputation in coupled planning. The other primary challenge would be to identify portions of the CBS search tree which could be reused. We are also interested in planning under uncertainty, particularly in coordination. Robots can be treated as being probability distributions along their assigned paths during collision checking. The probability of collisions along a given joint path can then be computed. Both MA-CBS and ODrM* are dependent upon discrete collisions for branching and search space expansion respectively, which can be addressed by placing a threshold on the acceptable probability of collisions.

REFERENCES

- [1] A. Felner, M. Goldenberg, G. Sharon, R. Stern, T. Beja, N. Sturtevant, J. Schaeffer, and R.C. Holte. Partial-expansion A* with selective node generation. *Proceedings of the 26th AAAI Conference on Artificial Intelligence (AAAI)*, 2012.
- [2] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configurations spaces. *IEEE Transactions on Robotics and Automation*, 12:566–580, June 1996.
- [3] D.M. Kornhauser, G. Miller, and P. Spirakis. Coordinating pebble motion on graphs, the diameter of permutation groups, and applications. In *Proceedings of the 25th Symposium on Foundations of Computer Science (FOCS)*, pages 241–250, 1984.
- [4] S. M. LaValle and J. J. Kuffner. Randomized kinodynamic planning. In *Proceedings IEE International Conference on Robotics and Automation*, 1999.
- [5] R. Luna and K.E. Bekris. Push and swap: Fast cooperative path-finding with completeness guarantees. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2011.
- [6] Mike Peasgood, John McPhee, and Christopher M. Clark. Complete and scalable multi-robot planning in tunnel environments. In *First IFAC Workshop on Multivehicle Systems*, 2006.
- [7] Q. Sajid, R. Luna, and K.E. Bekris. Multi-agent pathfinding with simultaneous execution of single-agent primitives. In *SoCS*, 2012.
- [8] G. Sharon, R. Stern, A. Felner, and N. Sturtevant. Conflict-based search for optimal multi-agent path finding. *AAAI (to appear)*, 2012.
- [9] G. Sharon, R. Stern, A. Felner, and N. Sturtevant. Meta-agent conflict-based search for optimal multi-agent path finding. In *SoCS*, 2012.
- [10] G. Sharon, R. Stern, M. Goldenberg, A. Felner, and I. Beer-Sheva. The increasing cost tree search for optimal multi-agent pathfinding. In *Twenty-Second International Joint Conference on Artificial Intelligence*, 2011.
- [11] T. Standley. Finding optimal solutions to cooperative pathfinding problems. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence (AAAI)*, 2010.
- [12] Glenn Wagner and Howie Choset. M*: A complete multirobot path planning algorithm with performance bounds. In *Proceedings of IEEE International Conference on Intelligent Robots and Systems 2011*, September 2011.
- [13] Glenn Wagner, Misu Kang, and Howie Choset. Probabilistic path planning for multiple robots with subdimensional expansion. In *Proceedings of IEEE/RSJ International Conference on Robotics and Automation*, May 2012.
- [14] K.H.C. Wang and A. Botea. Tractable multi-agent path planning on grid maps. In *Proceedings of the International Joint Conference on Artificial Intelligence IJCAI-09*, pages 1870–1875, 2009.