Springer Lecture Notes in Computer Science, Vol. 7154, R. Ramanujam and S. Ramaswamy (Eds.), pgs. 35-49, Proceedings of 8th International Conference on Distributed Computing and Internet Technology (ICDCIT 2012), Bhubaneswar, India, Feb. 2-4, 2012.

Decision Making as Optimization in Multi-Robot Teams

Lynne E. Parker

University of Tennessee, Knoxville, TN 37996-3450, USA
Email: LEParker@utk.edu, URL: http://web.eecs.utk.edu/~parker

Abstract. A key challenge in multi-robot teaming research is determining how to properly enable robots to make decisions on actions they should take to contribute to the overall system objective. This article discusses how many forms of decision making in multi-robot teams can be formulated as optimization problems. In particular, we examine the common multi-robot capabilities of task allocation, path planning, formation generation, and target tracking/observation, showing how each can be represented as optimization problems. Of course, globally optimal solutions to such formulations are not possible, as it is well-known that such problems are intractable. However, many researchers have successfully built solutions that are approximations to the global problems, which work well in practice. While we do not argue that all decision making in multi-robot systems should be based on optimization formulations, it is instructive to study when this technique is appropriate. Future development of new approximation algorithms to well-known global optimization problems can therefore have an important positive impact for many applications in multi-robot systems.

Keywords: Distributed robots, multi-robot systems, decision making

1 Introduction

The topic of multi-robot systems[1] has been extensively studied for the past two decades. The typical design objective in creating multi-robot teams is to enable the group of robots to solve physical tasks in a manner that is superior to single-robot systems. Many advances have been made in this field (e.g., see [32] for an overview), with some systems beginning to work their way into practical applications (e.g., for warehousing operations [45]).

These advances are possible because of the successful development of solutions to many challenges, including: (a) the design of sophisticated robot hardware that can physically achieve demanding tasks, (b) the development of advanced sensors and perception systems that can provide robots with detailed

[1] For our purposes, we define *multi-robot systems* to be groups of autonomous mobile robots that operate simultaneously in a shared workspace.

knowledge of their environment and the state of their mission, (c) the development of robust and reliable communications systems that allow robots to share information when distributed across a (potentially large) workspace, and (d) the design of intelligent robot control software that enables robots to achieve globally coherent results from individual local control actions. While all of these challenges are equally important, in this article we focus on the latter challenge — the development of intelligent software control for multi-robot teaming.

In a broad sense, robot control software could be considered equivalent to *decision making* in multi-robot teams. In some sense, every action that a robot takes is indeed based on a decision that the robot has made. However, not all multi-robot systems are typically characterized as decision making systems. The concept of decision making usually connotes a *cognitive* mental process, involving processes such as understanding, attention, reasoning, judgment, and so forth. Many multi-robot systems — especially the *swarm robotics* approaches (e.g., [20, 26, 27]) — do not make use of such cognitive processes, and instead incorporate relatively simple control laws that result in emergent group behavior. Since the type of control implemented in these systems does not involve cognitive processes, most robotics researchers would not call this control *decision making.*

On the other hand, a different class of multi-robot systems involves more direct and purposeful interaction (e.g., [30, 34, 38, 40]). These systems consist of robots with higher-level reasoning capabilities, and with possibly varying sensory, computational, and physical skills. In these systems, robots must decide how to coordinate their actions in a more deliberative manner. Thus, these multi-robot systems could indeed be accurately described as decision making systems.

Interestingly, many forms of decision making in *intentionally cooperative* multi-robot systems can be formulated as optimization problems. While we do not claim that all decision making in these domains can, or should, be viewed in this manner, it is instructive to review some representative techniques that have been demonstrated to enable robots to successfully work together in a variety of applications. This article reviews some common optimization formulations that have been presented in the literature for this domain. However, we do not attempt a complete survey of the literature on this topic, since that is beyond the scope of this article.

We begin in Section 2 by providing additional background material for understanding decision making in multi-robot teams. Then, in Section 3, we look at some common multi-robot capabilities that are suitable for formulation as optimization problems. Sections 4 through 7 examine each of these capabilities in more detail. Since the globally optimal solution typically requires excessive computation (i.e., the global problem formulations are NP-complete), distributed approaches that approximate the global solution are typically pursued; example techniques are also discussed in these sections. We offer concluding remarks in Section 8.

2 Background: Types of Multi-Robot Interaction

To place multi-robot decision making in context, we briefly introduce the most common forms of interaction in multi-robot teams (see [33] for a more detailed discussion of these types of systems). These types of interaction are characterized as follows:

- *Collective* multi-robot teams typically involve simple robots that are not aware of other robots on the team, even though they share common goals. In these systems, individual actions are typically beneficial to the team as a whole, and contribute to the team-level objective.
- *Cooperative* multi-robot teams involve robots that are aware of other robots on the team, and share goals with other teammates. Individual robot actions are usually beneficial to the objectives of the team as a whole.
- *Collaborative* multi-robot teams consist of robots that are aware of each other, and have individual goals. However, even though their individual goals may not be identical, they are willing to work with others when needed to help them achieve their individual goals.
- *Coordinative* multi-robot teams are composed of robots with individual goals, but which work together with other robots in a shared workspace to minimize inter-robot interference. In these systems, robots do not actively try to help other robots, but instead work to actively avoid interference.

Decision making most commonly occurs in the latter three types of systems, since the first (collective) type of multi-robot system does not typically include cognitive robot skills. Note that none of these types of interaction incorporates *adversarial* robots that work actively against each other. While this is a popular subject of study in multi-robot systems, especially for the domain of multi-robot soccer (e.g., [6, 17, 44]), we presume for the purposes of this article that all robots in the workspace share the objective of minimizing negative interference with other robots. In the sections that follow, we refer back to these types of interactions, noting which type of interaction is commonly studied in different multi-robot applications.

3 Decision Making as Optimization

To understand decision making in multi-robot teams, it is instructive to consider the types of applications and tasks to which multi-robot teams are applied. As outlined in [32] in more detail, some common applications for multi-robot teams include foraging, coverage, search, warehouse management, surveillance and security, construction and assembly, cooperative manipulation, search and rescue, and soccer. Each application area has its own unique challenges. However, many of these domains make use of solutions to some fundamental multi-robot interaction skills, including task allocation, path planning, formation control, and target tracking or observation. Thus, we examine these latter four areas in more detail in this article, since they have broad relevance to many multi-robot applications.

In all of these multi-robot interaction capabilities, the decision making process can be formulated as an optimization problem. Typically, these are formulated as combinatorial optimization [29] or as convex optimization [5] problems, in order to take advantage of the many tools available for these type of optimization. Importantly, however, these problems are typically not treated as *global* optimization problems for multi-robot applications, since such problems are known to be NP-complete. Since most robotic applications require real-time robot response, there is insufficient time to calculate globally optimal solutions for most applications; such solutions are only possible for very small-scale problems. Instead, typical solutions use distributed methods that incorporate only local cost/utility metrics. While such approaches can only achieve approximations to the global solution, they often are sufficient for practical applications.

As previously mentioned, the four representative multi-robot interaction skills that are often framed as optimization problems are as follows:

- *Task allocation*: Optimize a combination of robot cost and task utility in mapping a set of robots to a set of tasks [13].
- *Path planning*: Generate paths for multiple robots that minimize a performance metric [35]. Typical performance metrics include combined robot path lengths, combined travel times for robots to reach their respective goals, and combined energy use.
- *Formations*: Enable robots to move into a desired formation, or to maintain a specified formation, while moving through the environment. A common quality metric is to minimize the error between each current robot position and that robot's assigned position in the formation [28].
- *Target tracking or observation*: Control cooperative robot motions to ensure that a group of targets remains under observation by the robots. The typical metric is to optimize a combination of the time targets are under observation and a robot cost function [31].

The following sections present formulations of these capabilities as optimization problems. Examples are also given of approximation approaches that have been proposed for these problems.

4 Optimization in Task Allocation

Simply put, the *task allocation* problem in multi-robot systems is determining the proper mapping of a set of robots to a set of tasks, so as to maximize the total utility of the system. The task allocation problem arises frequently in *cooperative*, *collaborative*, and *coordinative* types of multi-robot teams.

In [13], Gerkey and Mataric define a taxonomy that covers several variations of the multi-robot task allocation problem. This taxonomy is defined on three axes, each of which has two possible settings: (1) robots — single-task (ST) versus multi-task (MT), (2) tasks — single-robot (SR) versus multi-robot (MR), and (3) assignments — instantaneous (IA) versus time-extended (TA). A particular task allocation problem is denoted by one choice from each list, such as ST-MR-IA, which is the "easiest" of the task allocation problems.

Each variant requires a different formulation of the optimization problem. For example, if solved in a centralized manner, Gerkey notes that the ST-MR-IA problem can make use of the Hungarian method [21] to find an optimal solution in polynomial time. Distributed variants make use of auction algorithms (e.g., [4]), which are based on economics-inspired metaphors, in which tasks are put up for bid by robots, who then propose their cost (or utility) in performing the task. Robots are awarded tasks based on maximizing the utility of the system. Many implementations of this market-based approach have been developed; an overview of this literature is given in [9].

Other variants of the problem are related to well-known NP-complete problems such as the Set Covering Problem (SCP), and the Set Partitioning Problem (SPP). For example, as noted in [13], viewing the ST-MR-IA multi-robot task allocation problem as an instance of SPP can be stated as:

Definition 1. *Given a finite set of robots R, a family F of acceptable subsets of R that represent all feasible robot coalition-task pairs, and a utility function $u : F \to \mathbb{R}_+$, find a maximum-utility family X of elements in F such that X is a partition of R.*

Some proposed task allocation approaches (e.g., [43]) have adapted existing SCP and SPP approximation algorithms (e.g., [7, 15]), making them relevant to the multi-robot domain. Example results from [47], which make use of market-based techniques to enable multi-robot teams to achieve an exploration task, are shown in Figure 1.

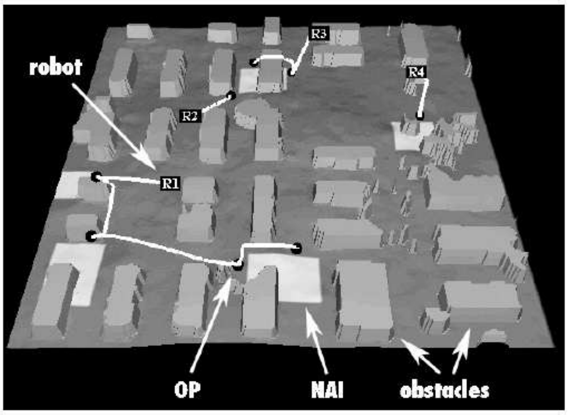

Fig. 1. Illustration of task allocation results in an exploration problem. From [47].

To provide a more detailed example of an approximation approach to task allocation, we briefly overview the ASyMTRe work of [34], which addresses the ST-MR-IA task allocation variant of the Gerkey taxonomy [13]. In this variant, the objective is to achieve task allocation for single-task robots (ST) performing multi-robot tasks (MR) using instantaneous assignment (IA). This problem variant is also called the *coalition formation* problem. While this problem has been addressed extensively in the multi-agent community (e.g., [18, 36, 37]), it has been noted by Vig [42] that most of the multi-agent approaches to coalition formation cannot be directly transferred to multi-robot applications, since robot capabilities and sensors are situated directly on the robots and are not transferable between robots.

The ASyMTRe approach is aimed at enabling sensor-sharing across robots for the purpose of forming coalitions to solve single multi-robot tasks. This method defines basic building blocks of robot capabilities to be collections of environmental sensors (ESs), perceptual schemas (PSs), motor schemas (MSs), and communication schemas (CSs). A robot, R_i, can be represented by $R_i = (ES^i, S^i)$, where ES^i is a set of environmental sensors installed on R_i, and S^i is the set of schemas that are pre-programmed into R_i. According to a set of rules, connections are created among the schemas on the robots to allow information to flow through the system. A set of information types $F = \{F_1, F_2, ...\}$ is introduced to label the inputs and outputs of each schema. (Information types differ from data types (e.g., the data type *integer*) in that they have semantic meanings (e.g., a robot's global position)). A schema can be activated if its inputs are satisfied either by sensors or the outputs of other schemas with the same information types. The ultimate goal is to activate the required MSs on the robot coalition team members to accomplish the task.

For reasoning about coalitions, ASyMTRe uses an anytime algorithm to search the entire solution space and return the best solution found so far according to predefined cost measures. One of the most important contributions of ASyMTRe is that it enables a finer resource sharing by dividing robot capabilities into smaller chunks (i.e., schemas), and reasons about how these schemas can be connected. Information can flow through the system to where it is required such that capability sharing is implicitly enabled through communication. ASyMTRe effectively manages the search space by reducing the solution space to an equivalence class that is smaller in practice, although in theory it can still be of exponential size. It also orders the search through the solution space via a *maximum cooperation size* constraint, preferring smaller-sized coalition solutions over larger ones. These techniques enable ASyMTRe to quickly find good (although not optimal) solutions that work well in practice.

ASyMTRe has been proven to be sound and complete, and has been shown to provide more flexibility for achieving tightly-coupled multi-robot tasks. Figure 2 illustrates ASyMTRe performing dynamic coalition formation for a cooperative navigation task. In this example, a coalition is formed with two robots to reach a goal position, but during task execution, a failure of one of the robots occurs.

This causes ASyMTRe to search for an alternative solution, which is found, resulting in a new coalition of robots.

Extensions to ASyMTRe have also been developed [46] that introduce an information quality based approach to model sensor constraints explicitly, and to provide a general method for maintaining the constraints during the task execution.

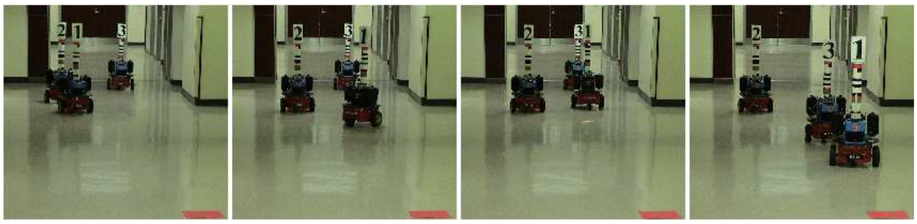

Fig. 2. Illustration of ASyMTRe dynamically forming coalitions for a cooperative navigation task. From [34].

5 Optimization in Path Planning

Another common multi-robot coordination problem is that of planning paths of multiple robots through a workspace. This type of challenge arises most commonly in *coordinative* types of multi-robot teams. Typically, these problems require individual robots to move appropriately through the workspace in order to achieve their own individual task objectives. Ideally, each robot moves as optimally as possible, but it is constrained by other robots also moving in the same workspace. Thus, the robots must work together to ensure that interference is minimized while individual paths are optimized, to the extent possible.

This challenge is also frequently formulated as an optimization problem. As described in [35], the multi-robot path planning can be formulated as an optimization problem as follows (using the notation of [22, 23]):

Definition 2. *Let $\mathcal{A}$ be a rigid robot in a static workspace $\mathcal{W} = \mathbb{R}^k$, where $k = 2$ or $k = 3$. The workspace is populated with obstacles. A configuration $\mathbf{q}$ is a complete specification of the location of every point on the robot geometry. The configuration space $\mathcal{C}$ represents the set of all the possible configurations of $\mathcal{A}$ with respect to $\mathcal{W}$. Let $\mathcal{O} \subset \mathcal{W}$ represent the region within the workspace populated by obstacles. Let the closed set $\mathcal{A}(\mathbf{q}) \subset \mathcal{W}$ denote the set of points occupied by the robot when it is in the configuration $\mathbf{q} \in \mathcal{C}$. Then, the C-space obstacle region, C_{obs}, is defined as: $\mathcal{C}_{obs} = \{\mathbf{q} \in \mathcal{C} | \mathcal{A}(\mathbf{q}) \cap \mathcal{O} \neq \emptyset\}$. The set of configurations that avoid collision (called the free space) is: $\mathcal{C}_{free} = \mathcal{C} \setminus \mathcal{C}_{obs}$. A free path between two obstacle-free configurations c_{init} and c_{goal} is a continuous map: $\tau[0, 1] \rightarrow \mathcal{C}_{free}$ such that $\tau(0) = c_{init}$ and $\tau(1) = c_{goal}$.*

For a team of m robots, define a state space that considers the configurations of all the robots simultaneously: $X = \mathcal{C}^1 \times \mathcal{C}^2 \times \cdots \times \mathcal{C}^m$. The C-space obstacle region must now be redefined as a combination of the configurations leading to a robot-obstacle collision, together with the configurations leading to robot-robot collision. The subset of X corresponding to robot $\mathcal{A}^i$ in collision with the obstacle region, $\mathcal{O}$, is $X^i_{obs} = \{\mathbf{x} \in X | \mathcal{A}^i(\mathbf{q}^i) \cap \mathcal{O} \neq \emptyset\}$. The subset of X corresponding to robot $\mathcal{A}^i$ in collision with robot $\mathcal{A}^j$ is $X^{ij}_{obs} = \{\mathbf{x} \in X | \mathcal{A}^i(\mathbf{q}^i) \cap \mathcal{A}^j(\mathbf{q}^i) \neq \emptyset\}$. The obstacle region in X is then defined as the combination of these latter two equations, resulting in:

$$X_{obs} = \left(\bigcup_{i=1}^{m} X^i_{obs} \right) \bigcup \left(\bigcup_{ij, i \neq j} X^i_{obs} \right). \tag{1}$$

The planning process for multi-robot systems treats X the same as $\mathcal{C}$, and X_{obs} the same as $\mathcal{C}_{obs}$, where c_{init} represents the starting configurations of all the robots, and c_{goal} represents the desired goal configurations of all the robots.

The optimization criteria that are typically used in multi-robot path planning are the minimization of total robot path lengths, the minimization of time for all robots to reach their goals, and the minimization of combined energy for robots to reach their goals. Other constraints can be added to guide the solution search, such as the incorporation of navigation restrictions (e.g., maximum slope, inability to traverse rough terrains).

As with other formulations, the centralized, global optimization problem cannot be solved in real-time. Thus, approximation techniques are used, such as decoupling the planning problem into independent components. One common decoupled approach is to divide the problem into planning independent robot paths, and then coordinating robot velocities along the paths in order to avoid collisions (e.g., [14, 16]). Another common technique is *prioritized planning*, in which robot paths are planned in a priority order, with robots later in the order treating robots earlier in the order as moving obstacles (e.g., [3, 10]). Figure 3 illustrates example results from the prioritized multi-robot path planning work of [3].

To provide more detail regarding an approximation technique to multi-robot path planning, we outline the work of [14], which proposes a decentralized motion planning algorithm for multiple robots. This approach incorporates optimal solutions to subcomponents of the path planning problem. The computationally expensive problem is decomposed into two modules – path planning and velocity planning. Each robot plans its own path independently using the D^* search algorithm [39], which it then broadcasts to all other robots. The D^* search algorithm produces an optimal path from the start position to the goal that minimizes a pre-defined cost function. The cost function used in this work is: $f_{pp} = \rho + \alpha_1 d + \alpha_2 s + \alpha_3 t$, where ρ is a large value if there is any obstacle penetrated by the path, and 0 otherwise; d is the geometric distance; s is the slope of the terrain; t is the penalty for turning; and $\alpha_1, \alpha_2, \alpha_3$ are positive weighting

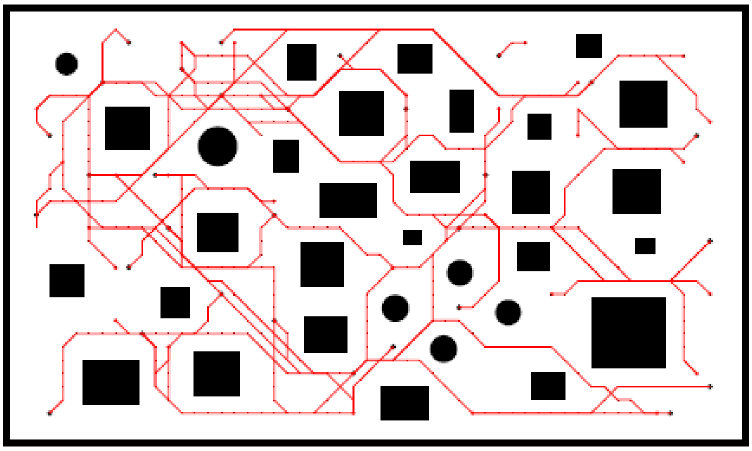

Fig. 3. Illustration of results of prioritized multi-robot path planning for 30 robots. From [3].

factors, where $\|(\alpha_1, \alpha_2, \alpha_3)\| = 1$. Such a cost function guarantees that D^* returns an optimal path that avoids static obstacles, and is the shortest, flattest, smoothest possible path if one exists.

After robot R_i obtains its own path P_i and all other paths $P_j, (j = 1, 2, \ldots, N)$, $j \neq i$, it executes a collision check procedure, which returns all collision regions. Since the configuration space is on a regular grid representation, the collision region is represented by sets of (x, y) pairs at which path intersections occur. Each path P_i can be seen as a continuous mapping $[0, l] \rightarrow \mathcal{W}^{\varepsilon}_{free}$, where l is the path length. Without loss of generality, one can assume that the parameterization of P_i is of constant speed. Then, define $\mathcal{S}_i = [0, l]$ to denote the set of parameter values that place the robot along the path P_i. The path coordination space is defined as $\mathcal{S} = \mathcal{S}_1 \times \mathcal{S}_2 \times \ldots \times \mathcal{S}_N$, and the *coordination diagram* (CD) is an N-dimensional diagram representing the path coordination space.

D^* then searches for a free trajectory in the CD by first mapping the collision regions into the path coordination space as static obstacles. As the path coordination space is parameterized by the non-decreasing path length, any possible movement in CD should be non-decreasing. Thus, the search objective is to find a non-decreasing curve that connects the lower left corner of the diagram $(0, 0, \ldots, 0)$ to the top right corner $(l_1, l_2, \ldots, l_N)$ avoiding penetration into the static obstacles. Such a free curve is called a *trajectory.* The computational expense is reduced by the non-decreasing constraint of the search. At each grid point, $2^N - 1$ action combinations are considered. Although the complexity is still exponential in the number of robots, the algorithm is efficient for a fixed N.

The trajectory is then converted into a velocity profile for each robot, and the performance index of the current trajectory solution is calculated. Since

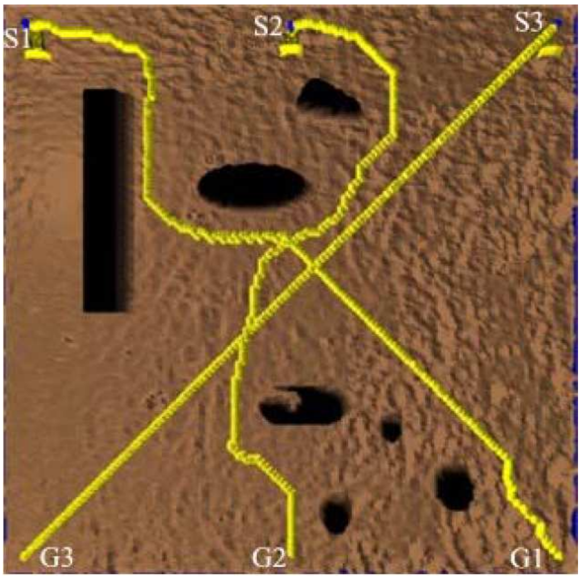

Fig. 4. Illustration of multi-robot path planning technique, which decouples path planning and velocity planning. Shown are the planned paths for three robots. From [14].

searching in the CD is distributed across the robots, each search can minimize a different cost function. The cost function for D^* velocity planning is chosen to be: $f_{vp} = \varrho + \beta_1 d + \beta_2 t_{idle} + \beta_3 p$, where ϱ is a large value if there are any collision regions penetrated by the trajectory, and 0 otherwise; d is the N-dimensional Euclidean distance; t_{idle} is the total idle time for all robots; p is the penalty if robot $\mathcal{A}_i$ has to give way to others; and $\beta_1, \beta_2, \beta_3$ are positive weighting factors, where $\|(\beta_1, \beta_2, \beta_3)\| = 1$.

The performance index and velocity profile are then broadcast to all other robots. An evaluation is performed to obtain a minimum value of the performance index, and the corresponding velocity profile is chosen. Figure 4 gives example results from this technique for three robots.

6 Optimization in Formation Control

The *formation control* problem in multi-robot systems addresses the challenge of moving robots into a desired formation shape and/or having the robots move in a coordinated manner while maintaining a desired shape. This problem arises in *cooperative* multi-robot teams, since all the robots share the same objective of maintaining the specified formation. Many researchers address this problem from control theoretic principles, focusing especially on proving stability and convergence properties (e.g., [2, 11, 12, 41]). The problem of generating an initial desired formation can be formulated in many ways as an optimization problem.

For example, the work of [8] presents the formation problem as a convex optimization problem, showing that certain forms can be solved in real time for large-scale multi-robot teams. This formulation of [8] is as follows:

Definition 3. *Let $P = (p_1, \ldots, p_m)^T$ denote the concatenated coordinates of the m robots in their current pose. Let S represent the equivalence class of similarity transformations of the desired formation. The objective is to obtain a new formation pose $Q = (q_1, \ldots, q_M)^T$, where Q has the same shape as S under an equivalence relation, and either the maximum distance between the respective positions in P and Q are minimized, or the sum of the distances is minimized. The set of nonlinear equality constraints that formulate this problem is given by:*

$$\|s_2\|_2(q_i^x - q_1^x) = (s_i^x, -s_i^y)^T(q_2 - q_1)$$
$$\|s_2\|_2(q_i^y - q_1^y) = (s_i^y, -s_i^x)^T(q_2 - q_1)$$

for $i = 3, \ldots, m$. For the sum of the distances optimization criteria, the constrained optimization problem becomes: $min_q \sum_{i=1}^{m} \|q_i - p_i\|_2$, subject to $Aq = 0$.

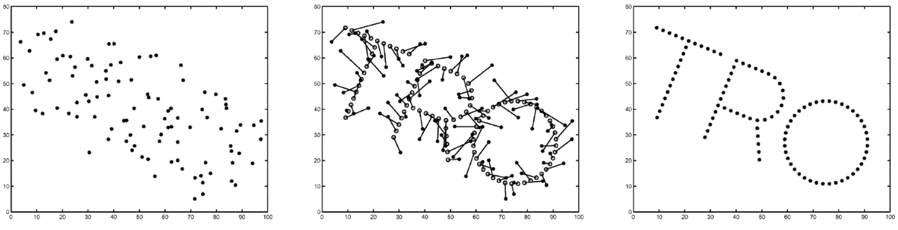

Fig. 5. Illustration of multi-robot formation generation. From [8].

The authors in [8] illustrate this technique for 100 simulated robots, as shown in Figure 5.

7 Optimization in Target Tracking/Observation

The domain of multi-target tracking and observation requires multiple robots to observe multiple targets moving through the environment. The objective is to keep as many of the targets within view by some robot on the team. This is a problem for *cooperative* and *collaborative* multi-robot teams. As pointed out in [31], this task is useful for studying *strong cooperation* in multi-robot teams, since the actions of each robot directly affect the performance of the others on the observation task.

The two-dimensional (planar) version of this task was introduced in [31] as the CMOMMT (Cooperative Multi-robot Observation of Multiple Moving Targets) problem. CMOMMT is formulated as an optimization problem as follows:

Definition 4. *Define the following: $\mathcal{S}$ is a two-dimensional, bounded, enclosed spatial region; $\mathcal{V}$ is a team of m robot vehicles $v_i, i = 1, 2, ...m$; $\mathcal{O}(t)$ is a set of n targets $o_j(t)$, $j = 1, 2, ..., n$, such that target $o_j(t)$ is located within region $\mathcal{S}$ at time t. We say that a robot, v_i, is observing a target when the target is within v_i's sensing range. Define an $m \times n$ matrix $B(t)$, as:*

$$B(t) = [b_{ij}(t)]_{mxn} \text{ such that } b_{ij}(t) = \begin{cases} 1 \text{ if robot } v_i \text{ is observing target } o_j(t) \\ \quad \text{in } \mathcal{S} \text{ at time } t \\ 0 \text{ otherwise} \end{cases}$$

Then, the objective is to maximize the metric $A = \sum_{t=1}^{T} \sum_{j=1}^{n} \frac{g(B(t),j)}{T}$, where:

$$g(B(t), j) = \begin{cases} 1 \text{ if there exists an } i \text{ such that } b_{ij}(t) = 1 \\ 0 \text{ otherwise} \end{cases}$$

Similar problems have been studied by many researchers, including more complex versions in three dimensions (e.g., for aerial vehicles) and with more complex topography. This domain is related to problems in art gallery algorithms, pursuit evasion, and sensor coverage, and has practical relevance in security and surveillance applications. Example research in this domain includes [1, 19, 24, 25].

For the multi-robot target observation problem, the approximation solution that is proposed in [31] is a weighted local force vector approach that attracts robots to nearby targets and repels them from nearby robots. The weights are computed in real-time, and are based on the relative locations of the nearby robots and targets. In this approach, each robot broadcasts to its teammates the position of all targets within its field of view. For all known targets, robots then perform a predictive tracking of that target's location, assuming that the target will continue linearly from its current state. Weights associated with known targets are decreased if other robots are known to be nearby. Setting the weights in this manner aims at generating an improved collective behavior across robots when utilized by all robot team members. Example results from this approach are illustrated in Figure 6.

8 Conclusions

While not all multi-robot systems make use of decision making, it is common in more intentional types of interaction that involve cooperative, collaborative, and coordinative multi-robot teams. These types of teams are applied to a wide variety of applications, many of which involve common multi-robot capabilities such as task allocation, path planning, formation control, and target tracking/observation. Researchers often formulate these common capabilities by defining them as optimization problems. In this paper, we have discussed some common formulations for the four main multi-robot capability areas, and discussed some examples of approximation algorithms that are guided by the optimization formulation. While exact global solutions are not possible due to the

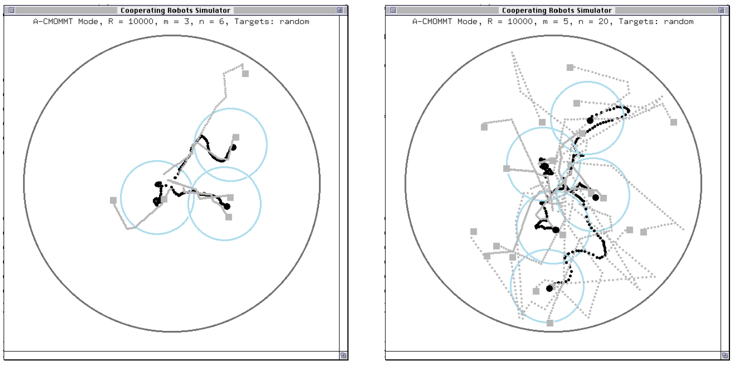
Cooperating Robots Simulator
A-CMOMMT Mode, R = 10000, m = 3, n = 6, Targets: random
Cooperating Robots Simulator
A-CMOMMT Mode, R = 10000, m = 5, n = 20, Targets: random

Fig. 6. Illustration of simulation results of multi-robot target observation, with (left) 3 robots and 6 targets, and (right) 5 robots and 20 targets. From [31].

intractability of the formulations, much research has shown that the approximate techniques work well in practice. These results show, therefore, that much of multi-robot decision making can be successfully viewed, and approximated, as optimization problems. The further development and application of distributed, approximate solutions to optimization problems is therefore expected to be beneficial for generating more effective decision making in multi-robot teams.

References

1. Beard, R.W., McLain, T.W., Goodrich, M.: Coordinated target assignment and intercept for unmanned air vehicles. In: Proceedings of IEEE International Conference on Robotics and Automation. IEEE (2002)
2. Belta, C., Kumar, V.: Abstraction and control for groups of robots. IEEE Transactions on Robotics 20(5), 865–875 (2004)
3. Bennewitz, M., Burgard, W., Thrun, S.: Optimizing schedules for prioritized path planning of multi-robot systems. In: Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on. vol. 1, pp. 271–276 (2001)
4. Bertsekas, D.: The auction algorithm for assignment and other network flow problems: A tutorial. Interfaces pp. 133–149 (1990)
5. Boyd, S., Vandenberghe, L.: Convex optimization. Cambridge Univ Pr (2004)
6. Browning, B., Bruce, J., Bowling, M., Veloso, M.: STP: Skills, tactics and plays for multi-robot control in adversarial environments. IEEE Journal of Control and Systems Engineering 219, 33–52 (2005)
7. Chvatal, V.: A greedy heuristic for the set-covering problem. Mathematics of operations research pp. 233–235 (1979)
8. Derenick, J., Spletzer, J.: Convex optimization strategies for coordinating large-scale robot formations. Robotics, IEEE Transactions on 23(6), 1252–1259 (2007)
9. Dias, M., Zlot, R., Kalra, N., Stentz, A.: Market-based multirobot coordination: A survey and analysis. Proceedings of the IEEE 94(7), 1257–1270 (2006)
10. Erdmann, M., Lozano-Perez, T.: On multiple moving objects. Algorithmica 2, 477–521 (1987)

11. Fax, J.A., Murray, R.M.: Information flow and cooperative control of vehicle formations. IEEE Transactions on Automatic Control 49(9) (2004)
12. Ge, S.S., Fua, C.H.: Queues and artificial potential trenches for multirobot formations. IEEE Transactions on Robotics 21(4), 646–656 (2005)
13. Gerkey, B., Mataric, M.J.: A formal analysis and taxonomy of task allocation in multi-robot systems. International Journal of Robotics Research 23(9), 939–954 (2004)
14. Guo, Y., Parker, L.E.: A distributed and optimal motion planning approach for multiple mobile robots. In: Proceedings of IEEE International Conference on Robotics and Automation (2002)
15. Hoffman, K., Padberg, M.: Solving airline crew scheduling problems by branch-and-cut. Management Science pp. 657–682 (1993)
16. Kant, K., Zucker, S.W.: Toward efficient trajectory planning: the path-velocity decomposition. The International Journal of Robotics Research 5(3), 72–89 (1986)
17. Kitano, H., Kuniyoshi, Y., Noda, I., Asada, M., Matsubara, H., Osawa, E.: Robocup: A challenge problem for AI. AI Magazine 18(1), 73–85 (Spring 1997)
18. Klusch, M., Gerber, A.: Dynamic coalition formation among rational agents. IEEE Intelligent Systems 17(3), 42–47 (2002)
19. Kolling, A., Carpin, S.: Multirobot cooperation for surveillance of multiple moving targets – a new behavioral approach. In: Proceedings of the IEEE International Conference on Robotics and Automation. pp. 1311–1316. IEEE (2006)
20. Kube, C.R., Zhang, H.: Collective robotics: From social insects to robots. Adaptive Behavior 2(2), 189–219 (1993)
21. Kuhn, H.: The hungarian method for the assignment problem. Naval research logistics quarterly 2(1-2), 83–97 (1955)
22. Latombe, J.C.: Robot Motion Planning. Kluwer Academic Publishers, Norwell, MA (1991)
23. LaValle, S.M.: Planning Algorithms. Cambridge University Press (2006)
24. LaValle, S.M., Gonzalez-Banos, H.H., Becker, C., Latombe, J.C.: Motion strategies for maintaining visibility of a moving target. In: Proceedings of the 1997 IEEE International Conference on Robotics and Automation. pp. 731–736. IEEE (1997)
25. Luke, S., Sullivan, K., Panait, L., Balan, G.: Tunably decentralized algorithms for cooperative target observation. In: Proceedings of the fourth international joint conference on Autonomous Agents and Multiagent Systems. pp. 911–917. ACM Press (2005)
26. Mataric, M., Nilsson, M., Simsarian, K.: Cooperative multi-robot box pushing. In: Proceedings of IEEE International Confereence on Intelligent Robots and Systems (IROS). pp. 556–561 (1995)
27. McLurkin, J.: Stupid robot tricks: Behavior-based distributed algorithm library for programming swarms of robots. In: M.S. Thesis, Massachusetts Institute of Technology (2004)
28. Michael, N., Zavlanos, M., Kumar, V., Pappas, G.: Distributed multi-robot task assignment and formation control. In: Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on. pp. 128–133 (2008)
29. Nemhauser, G., Wolsey, L.: Integer and combinatorial optimization, vol. 18. Wiley New York (1988)
30. Parker, L.E.: ALLIANCE: An architecture for fault-tolerant multi-robot cooperation. IEEE Transactions on Robotics and Automation 14(2), 220–240 (1998)
31. Parker, L.E.: Distributed algorithms for multi-robot observation of multiple moving targets. Autonomous Robots 12(3), 231–255 (2002)

32. Parker, L.E.: Chapter 40: Multiple mobile robot systems. In: Siciliano, B., Khatib, O. (eds.) Springer Handbook of Robotics. Springer (2008)
33. Parker, L.E.: Distributed intelligence: Overview of the field and its application to multi-robot systems. Journal of Physical Agents 2(1), 5–14 (2008)
34. Parker, L.E., Tang, F.: Building multi-robot coalitions through automated task solution synthesis. Proceedings of the IEEE, special issue on Multi-Robot Systems 94(7), 1289–1305 (2006)
35. Parker, L.: Path planning and motion coordination in multiple mobile robot teams. In: Meyers, R.A. (ed.) Encyclopedia of complexity and system science. Springer (2009)
36. Sandholm, T., Larson, K., Andersson, M., Shehory, O., Tohme, F.: Coalition structure generation with worst case guarantees. Artificial Intelligence 111(1-2), 209–238 (1999)
37. Shehory, O.: Methods for task allocation via agent coalition formation. Artificial Intelligence 101(1-2), 165–200 (1998)
38. Simmons, R., Singh, S., Hershberger, D., Ramos, J., Smith, T.: First results in the coordination of heterogeneous robots for large-scale assembly. In: Proc. of the ISER '00 Seventh International Symposium on Experimental Robotics (2000)
39. Stentz, A.: Optimal and efficient path planning for unknown and dynamic environments. International Journal of Robotics and Automation 10, 89–100 (1993)
40. Sukhatme, G., Montgomery, J.F., Vaughan, R.T.: Experiments with cooperative aerial-ground robots. In: Balch, T., Parker, L.E. (eds.) Robot Teams: From Diversity to Polymorphism. pp. 345–368. A K Peters (2002)
41. Tabuada, P., Pappas, G., Lima, P.: Motion feasibility of multi-agent formations. IEEE Transactions on Robotics 21(3), 387–392 (2005)
42. Vig, L., Adams, J.A.: Issues in multi-robot coalition formation. In: Parker, L.E., Schultz, A., Schneider, F. (eds.) Multi-Robot Systems Volume III: From Swarms to Intelligent Automata. Kluwer (2005)
43. Vig, L., Adams, J.A.: Multi-robot coalition formation. IEEE Transactions on Robotics 22(4), 637–649 (2006)
44. Weigel, T., Gutmann, J.S., Dietl, M., Kleiner, A., Nebel, B.: CS Freiburg: coordinating robots for successful soccer playing. IEEE Transactions on Robotics and Automation 5(18), 685–699 (2002)
45. Wurman, P., D'Andrea, R., Mountz, M.: Coordinating hundreds of cooperative, autonomous vehicles in warehouses. AI Magazine 29(1), 9 (2008)
46. Zhang, Y., Parker, L.: Iq-asymtre: Synthesizing coalition formation and execution for tightly-coupled multirobot tasks. In: Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on. pp. 5595–5602 (2010)
47. Zlot, R., Stentz, A.: Complex task allocation for multiple robots. In: Proceedings of IEEE International Conference on Robotics and Automation (2005)