A Taxonomy for Task Allocation Problems with Temporal and Ordering Constraints

Ernesto Nunes[a], Marie Manner[a], Hakim Mitiche[a], Maria Gini[a,]

[a] *Department of Computer Science and Engineering, University of Minnesota, 4-192 Keller Hall, 200 Union St, Minneapolis, MN 55455*

Abstract

Previous work on assigning tasks to robots has proposed extensive categorizations of allocation of tasks with and without constraints. The main contribution of this paper is a more specific categorization of problems that have both temporal and ordering constraints. We propose a novel taxonomy that emphasizes the differences between temporal and precedence constraints, and organizes the current literature according to the nature of ordering and temporal constraints of addressed problems. We summarize widely used models and methods from the task allocation literature and related areas, such as vehicle routing and scheduling problems, showing similarities and differences.

Keywords: task allocation, taxonomy, multi-robot coordination

Contents

Email addresses: `enunes@cs.umn.edu` (Ernesto Nunes), `manner@cs.umn.edu` (Marie Manner), `h.mitiche@gmail.com` (Hakim Mitiche), `gini@umn.edu` (Maria Gini)

1. Introduction

What is multi-robot task allocation? Think of a shipping company that sells an item every hour; a robot at the warehouse could receive that order, fetch the item, pack it, and prepare it for pick-up by a postal service. What happens when the company sells 20 items every hour? What about 20 items every minute? What about 20 items a second? Amazon, a popular shopping website, sold 36.8 million items on an especially popular shopping day in 2013. With 426 items ordered per second that day, a single robot would be hard-pressed to keep up with the orders. If the warehouse used a large team of robots, each robot would have to plan an efficient route through the warehouse to fetch items for shipping without colliding with other robots, without taking items that another robot is handling, all while planning its route around fetching items that are out-of-stock but will be restocked soon.

Allocation of tasks with constraints on when, where, and in what order they need to be done by groups of robots is an important class of problems with many real-life applications, such as warehouse automation, pickup and delivery, surveillance at regular intervals, space exploration, and search and rescue.

The nature of the temporal constraints in this class of problems is very broad; for example, in search and rescue domains the tasks are discovered over time and have to be done as quickly as possible. In dynamic environments, robots might end up arriving late to some tasks and might miss some. On the other hand, success in surveillance tasks requires not to arrive late to tasks. Additionally, tasks may need to be executed in a specific order, such as in urban disaster scenarios in which police must clear blockades from roads before ambulances can travel to carry injured people. Some tasks may need to be done synchronously, as in surveillance where robots have to track multiple people at the same time. We explore the following research questions to study the nuances of task allocation problems with temporal and ordering constraints :

- What are the main types of temporal and ordering constraints in multi-robot task allocation?
- What are the most commonly used optimization objectives? Are they predominantly temporal-based, distance-based, or multi-objective?
- What models and methods from related areas can be applied to this class of problems?
- Which questions or variants have been answered well, and which remain largely open in this class of problems?

Our main contribution is a novel taxonomy that divides the literature according to the nature of the temporal and ordering constraints considered. We provide a granular division of the literature based on time windows and precedence/synchronization constraints, while maintaining the subcategories proposed by Gerkey and Matarić [2004]. Gerkey's taxonomy is based on three main characteristics of robots, tasks, and time, as follows:

- *Single-task robots (ST) vs. multi-task robots (MT)*: ST robots can do at most one task at a time, while MT robots can work on multiple tasks simultaneously.
- *Single-robot tasks (SR) vs. multi-robot tasks (MR)*: SR tasks require exactly one robot in order to be completed, while multiple robots are needed to complete an MR task.
- *Instantaneous (IA) vs. time-extended (TA) assignments*: In IA, tasks are allocated as they arrive, while in TA, tasks are scheduled over a planning horizon (defined in Section 2.1).

The iTax taxonomy [Korsah et al., 2013] focuses on interrelated utilities and constraints among tasks, both for individual robots and across robots, and for complex tasks which can be decomposed in many different ways, adding a level above Gerkey's taxonomy, but does not address specifically how to deal with temporal and precedence constraints. Our work attempts to fill this gap.

We extend Gerkey and Matarić [2004]'s taxonomy adding to its time-extended (TA) part temporal constraints either in the form of time windows (TA:TW) or in the form of synchronization and precedence constraints (TA:SP).

1.1. Organization

We begin by defining the class of multi-robot task allocation problems with temporal and ordering constraints (MRTA/TOC) in Section 2. In Section 3 we relate this class of problems to problems in other areas, setting the ground for our exploration of models and methods in those areas. In Section 4 we present commonly used temporal and ordering models. In Section 5 we review the most common optimization objectives considered in the literature. Our taxonomy is introduced in Section 6. Task execution and the dynamics therein are discussed in Section 7. Solutions are introduced in Section 8. We discuss open issues, future directions, and final thoughts in Section 9.

Next, we formally define the task allocation problem with temporal and ordering constraints (MRTA/TOC), and summarize the terminology we use.

2. MRTA/TOC: Multi-robot Task Allocation with Temporal and Ordering Constraints

2.1. Terminology and Abbreviations

We define the terminology we use informally as follows:

- A *robot* is an autonomous agent responsible for performing some actions. Alternative names for robots are physical agents, unmanned vehicles, and rovers. Robots in MRTA/TOC are typically modeled as holonomic or point robots, since the focus is not on low level control of robot motion.
- A *team* is a set of robots that work together. A team is often called a *coalition* when it is dynamic, i.e. formed to do some tasks and disbanded after that [Parker and Tang, 2006].

- A *task* is an action to be performed, also referred to as a work unit, activity, waypoint, or customer request. In some scheduling literature tasks are divided into jobs [Davis and Burns, 2011], but in other cases jobs are made of tasks [Balas et al., 2008].
- A *time window* is a time interval starting with the earliest time a task can start, and ending with the latest time the task can end. If the earliest time is not given, the latest time is referred to as a deadline constraint.
- *Synchronization constraints* specify constraints among tasks that, for instance, have to start or finish at the same time.
- *Precedence constraints* specify partial ordering relationships between pairs of tasks, i.e., a task has to be completed before another task can start.
- A *schedule* is a timetable in which each task has a specific time to start, end, or both. In some cases each robot has its own individual schedule Nunes and Gini [2015], while in others all robots share a single schedule.
- The *scheduling horizon* is the time period for which schedules are created. Alternatively, it is the end time, after which robots are not allowed to start or end tasks.
- The *planning horizon* is the time period over which plans are created.
- The *makespan* is the time difference between the end of the last task and the start of the first task.
- A *route* is a sequence of locations to visit. Routes and schedules are often used interchangeably, but schedules always concern time, while routes concern physical locations.
- A *task release* refers to a task becoming available for execution. Task release can be deterministic if the release time is known upfront, dynamic if the release time is stochastic, or sporadic if it is governed by unknown probabilities; task release is also called periodic when the same task is released at regular intervals.

We use the following acronyms:

- MRTA/TOC for Multi-Robot Task Allocation with Temporal and Ordering Constraints.
- MIP for Mixed Integer Programming, and MILP if the objective function and constraints are linear.
- TOPTW for Team Orienteering Problem (TOP) with Time Windows.
- VRPTW for Vehicle Routing Problem (VRP) with Time Windows.
- JSP for job-shop scheduling problems.

2.2. Problem Formulation

We assume there is a finite set of robots and a set of tasks. A robot may have a location, speed, route, and/or schedule. A task has a subset of the following parameters: location, expected duration, cost, demand, reward, earliest start, and latest finish time.

Ordering constraints express a dependency between tasks, and are usually encoded as directed acyclic graphs. Each node in the graph represents a task, and each edge indicates precedence or synchronization in the execution of the tasks.

The objective is to optimize some function of the cost (or reward) for doing the tasks for all the robots. Cost can be a temporal measure (e.g. makespan), or a spatial measure (e.g. distance traveled). Commonly used optimization functions are more thoroughly described later in Section 5.

3. Connections with Other Problems

Multi-robot task allocation (MRTA) started in earnest in the 90's, when researchers started pulling together teams of robots to accomplish multiple tasks. MRTA draws from a variety of areas in mathematics and operations research as well as computer science and robotics, including assignment problems, distributed computing, distributed AI, and scheduling.

The search for robust approaches to MRTA focused on how the robots perform in complex environments, leading researchers to add features like uncertainty with probabilistic and stochastic models, time windows for tasks, and spatial constraints. Solutions take different approaches, such as auctions, market-based planning, Markov Decision Processes, decentralized scheduling algorithms, and distributed constraint optimization.

In this paper we cover a subset of MRTA problems, which we call MRTA/TOC, to highlight the importance of temporal and ordering constraints among tasks and to shed light on how the inclusion of temporal and ordering constraints increases the complexity of task allocation.

Similar types of problems include the vehicle routing problem [Dantzig and Ramser, 1959], the job shop scheduling problem [Manne, 1960], and the team orienteering problem [Chao et al., 1996]. Overall, multi-robot task allocation diverges from each of these problems on key points, including assumptions on the number of robots, robot and task homogeneity, environment dynamics caused by failures or interference with other robots, and communication restrictions.

We are now prepared to discuss the relationship between MRTA/TOC problems and the vehicle routing problem with time windows (VRPTW), the team orienteering problem with time windows (TOPTW), and the job-shop scheduling problem (JSP).

3.1. MRTA/TOC vs. VRPTW

The vehicle routing problem with time windows (VRPTW) [Kolen et al., 1987, Solomon and Desrosiers, 1988, Desrochers et al., 1988, Toth and Vigo, 2002] studies problems which require solving allocation, routing, and scheduling subproblems simultaneously. Vehicles and robots are often treated as points in space, ignoring kinematic constraints, but kinematic [e.g. Cheng et al., 2008, for unmanned aerial vehicles] and sometimes dynamic [Pecora and Cirillo, 2012, for ground vehicles] constraints can be considered.

The solutions to several variants of VRPTW – such as multi-depot [Kang et al., 2005, Polacek et al., 2004], dynamic and stochastic [Taş et al., 2013, Pavone et al., 2011, Laporte et al., 1992], and precedence and synchronization constrained [Korsah et al., 2012, Bredström and Rönnqvist, 2008] – have been extended to MRTA/TOC settings. An example of VRP similarities is the online pickup and delivery problem with transfers, where a team of vehicles has to pick up a set of items at a location and deliver them to another location [Coltin and Veloso, 2014a]. This problem is a generalization of the pickup and delivery problem [Savelsbergh and Sol, 1995] which is well studied in operations research. However, the proposed solution is a typical MRTA approach. The authors combine a centralized temporal planner, which creates initial schedules, with auctions, which are used to repair the plans when delays or failures occur. In the same vein, Korsah et al. [2012] studied a MRTA problem that can be framed as a vehicle routing problem with temporal, precedence and synchronization constraints. The authors offer a MILP-based model and an optimal Branch-and-Price solution.

Despite their similarities, these problems differ in some ways. First, VRPTW assumes an infinite number of vehicles is always available, with a few exceptions [e.g. Lau et al., 2003]). This assumption is not practical in robotic systems where the number of robots is usually fixed and can even decrease due to failures. VRPTW problems usually assume that all vehicles start from the same depot and return to the depot after work. In MRTA/TOC problems, robots may start at different locations and do not need to return to their initial locations. VRPTW problems mostly assume homogeneous vehicles with respect to their capabilities and capacities [for exceptions, see Bettinelli et al., 2011, Dondo and Cerdá, 2007], while in MRTA/TOC robots are not necessarily homogeneous and their capacities and types can differ [Ponda et al., 2010, Schneider et al., 2005, Xu et al., 2005]. Lastly, unlike VRPTW problems, in MRTA/TOC problems communication is important and often constrained. In [Mercker et al., 2010] the communication graph is unknown (hence the algorithm does not always converge), while in [Ponda et al., 2012a] the communication graph is maintained by using specialized robots or robots not working on a task to act as communication relays. While in the previous two works convergence is guaranteed only for complete communication graphs, Jackson et al. [2013] and Smith and Bullo [2007] proposed distributed algorithms that converge using only local communication.

3.2. MRTA/TOC vs. TOPTW

In the team orienteering problem with time windows (TOPTW), an origin and destination pair is given, and the goal is to search for control points to visit between the origin and destination such that the profit (or score function) is maximized while respecting all constraints. Each control point is associated with a profit (or score), and each edge connecting control points is weighted by the cost of moving between the control points [Labadie et al., 2012]. Control points are equivalent to tasks for robots in MRTA/TOC.

When TOPTW considers origin and destination to be the same point, then we have sub-tours similar to those for VRPTW problems, which can be described as vehicle routing problems with profit [Archetti et al., 2014]. One application of TOPTW problems, dial-a-ride, has gained some popularity in MRTA/TOC [e.g. Coltin and Veloso, 2014a, Rubinstein et al., 2012, Bouros et al., 2011], In dial-a-ride, the problems are over-constrained [Carrabs F., 2007, Cordeau and Laporte, 2007], which means that not all the tasks can be performed, and thus the goal is to find the subset of tasks that maximizes the total profit [Rubinstein et al., 2012].

3.3. MRTA/TOC vs. JSP

The job-shop scheduling problem (JSP) is concerned with allocating groups of activities, called jobs, to a set of machines with the goal of minimizing the cost of completing the jobs, alone or in combination with other objectives [Allahverdi et al., 2008, Graham et al., 1979]. The problem can be decomposed into sequencing the activities and assigning start and end times to them (scheduling), which are solved simultaneously. Certain MRTA/TOC problems can be modeled as job-shop scheduling with setup times, deadlines and precedence constraints [Cesta et al., 2000, Balas et al., 2008, Oddi et al., 2011]; these problems include [Dahl et al., 2009, Gombolay et al., 2013, Nunes and Gini, 2015, McIntire et al., 2016], although [Nunes and Gini, 2015] does not consider precedence constraints. In order to model MRTA/TOC problems as job-shop scheduling problems, tasks are treated as jobs and robots as machines. Simple tasks can be mapped to a job with only one activity, and complex tasks with subtasks of a job with multiple activities.

The mathematical models for JSP do not apply directly to MRTA/TOC problems, because JSP does not account for travel time. When setup times are used in JSP, the setup time typically depends on the machine and not on the time needed for the job to reach the machine. The equivalent of travel time would be to use setup times that depend on the specific job [Korsah et al., 2013].

Modeling MRTA/TOC problems as JSP problems is most useful when models and methods developed for scheduling [Cesta and Oddi, 1996, Cesta et al., 1999, Lee et al., 2009, Shah et al., 2009] are combined with MRTA solution techniques. In [Gombolay et al., 2013] a centralized approach is proposed in which a central temporal network is used and integrated with a MILP-based planner yielding near optimal schedules. In the decentralized approach of Barbulescu et al. [2010] each robot forms its own simple temporal network [Dechter et al., 1991], encoding both temporal and precedence constraints in the network. To enforce precedence constraints a robot has to know which other robots depend on its schedule, so a high communication overhead is required to keep all robots up-to-date. The distributed approach in [Nunes and Gini, 2015] cuts down on communication costs by having each robot keep its own independent local temporal network and uses sequential single-item auction for allocation.

Having outlined the differences and similarities between MRTA/TOC and related problems, we now turn our attention to temporal and ordering constraints on MRTA problems.

4. Temporal Models

Temporal models are outlined in Subsection 4.1 and Subsection 4.2, the nature of ordering constraints is presented in Subsection 4.4, while in Subsection 4.5 we discuss the nature of temporal constraints.

4.1. Relationships between time intervals

In general terms, time can be modeled using points or intervals [Allen, 1983]. An example time point is 10 am, while an interval is a continuous set of values bounded below and above by some time point, for example [10 am-12 pm]. When representing temporal constraints we may use either representation; however, the interval representation is much more common and is referred to as a time window.

The seminal paper by Allen [1983] proposed a set of relationships that hold between any two time intervals, as depicted in Fig. 1.

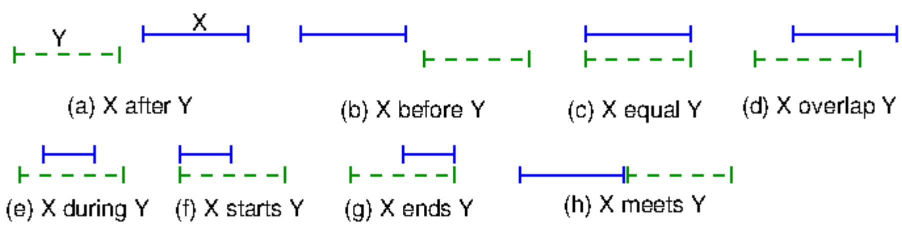


Figure 1: All possible relationships between pairs of time intervals [Allen, 1983]

While the relationships originally were described between qualitative time intervals, they are also useful to describe the ordering between quantitative time intervals. The relationships can be used to model partial or complete ordering constraints between tasks, – for example, task X should be done before, after, or at the same time as task Y. The X before Y operator can be used to describe precedence constraints between tasks, while the X equal Y operator describes a synchronization constraint between the intervals or time points of two tasks.

4.2. Simple Temporal Networks (STN)

Equally influential is Dechter's approach [Dechter et al., 1991], which proposed to represent a class of temporal constraints with a graph, called a simple temporal network (STN). An example is in Fig. 2.

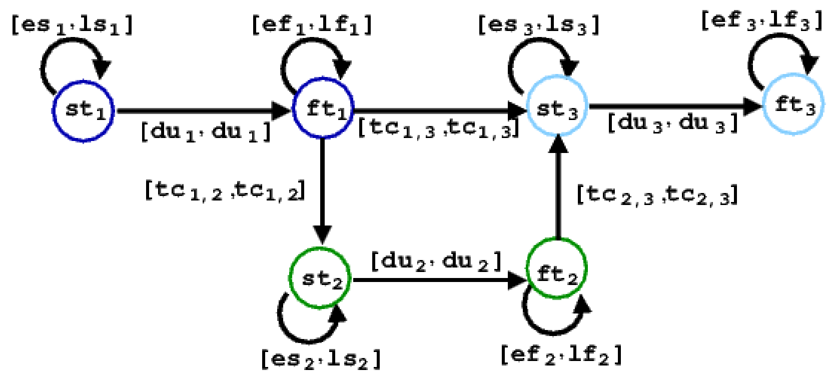


Figure 2: A simple temporal network with three tasks 1, 2 and 3, each with a time window. The self-loops on tasks indicate the absolute start and end times for the task. Task 1 is done first, and then there is a choice of doing 2 and 3 or just 3. st and ft are the actual start and finish times for each task, es and ls are the earliest and latest times tasks can start, similarly ef and lf are the earliest and latest times tasks can finish, and du represent tasks' durations. $tc_{k,k'}$ is a time cost defined as the sum of the travel and wait times, which constrains when the next task k' can be started.

Nodes represent time point variables or time events, and weighted edges represent inequality constraints between time points. To reduce computational complexity, this model requires exactly one constraint between every pair of time point variables. A solution to the scheduling problem can be computed in polynomial time using the Floyd-Warshall algorithm. In an STN the relationship between time windows can be represented by establishing constraints between start and finish times of tasks. While there are more complex models, for instance [Stergiou and Koubarakis, 2000, Block et al., 2006], in general these are NP-hard and can be approximated by solving several simple temporal problems [Boerkoel and Durfee, 2013].

STNs are commonly used in MRTA problems [Nunes and Gini, 2015, Gombolay et al., 2013, Barbulescu et al., 2010] because constraint consistency can be efficiently verified in polynomial time [Planken et al., 2008, Xu and Choueiry, 2003, Dechter et al., 1991]. An important feature of STNs is that new time points and constraints can be dynamically added in polynomial time [Coles et al., 2009, Cesta and Oddi, 1996], which is beneficial in dynamic domains where new tasks can appear and disappear.

STNs have been successfully extended to multi-agent settings [Boerkoel and Durfee, 2012, Boerkoel and Planken, 2012, Hunsberger, 2002] and to scenarios with uncertainties. Vidal [1999] uses set bounded uncertainty to model duration uncertainty of temporal events in an STN, and introduces the STN with uncertainty (STNU). Tsamardinos [2002] and Fang et al. [2014] extend STNUs by modeling uncertainty as probabilities. The former attempts to minimize the risk of temporal inconsistencies occurring, and the latter attempts to bound the probability of not meeting a schedule, respectively.

4.3. Time Window Constraints

A time window constraint is a temporal interval constraint on the start, or finish time of a task, or both. A time window has a lower bound value, usually the task's earliest start time, and an upper bound value, usually the task's latest finish time. A task can also have a latest start time and an earliest finish time, resulting in a time window of the form [earliest start time, latest start time, earliest finish time, latest finish time]. This representation implicitly provides an upper bound to the task duration. When the earliest and latest start times are the same, the time window specifies only a start time. Same for finish time.

Having earliest and latest start or finish times increases the flexibility of task allocation, but increases the search space because there are multiple ways of scheduling a task within its time window. The use of time windows for auction-based task allocation to agents was pioneered in the MAGNET system, which proposed various allocation algorithms [Collins et al., 2000, Collins and Gini, 2006].

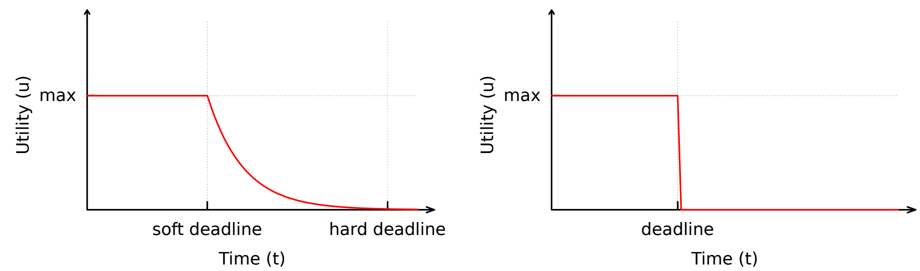
Time window constraints can be used to model many types of temporal relationships among tasks. For instance, deadline constraints [Luo et al., 2015, Amador et al., 2014, Ramchurn et al., 2010b] only impose constraints on the latest time robots can arrive to tasks before the task expires. The flexibility in temporal constraint representation, together with the temporal relationships between time windows (see Section 4) make time windows a powerful temporal constraint modeling tool.

While in many problems time windows are taken as input data, in others (e.g Robocup Search and Rescue – RCSR), the time window is hidden from the algorithm designer. The designer simply has access to a time-varying utility function (e.g left plot in Fig. 3) that returns zero if the robot performs the task past some hard deadline. However, the actual hard deadline is not revealed a priori (see papers [Parker et al., 2015, Scerri et al., 2005] for example). In RCSR problems, the time windows of tasks do not remain constant either. For example, fires growing in nearby buildings might amplify each other, which would cause the deadline for extinguishing a fire to change.

Task allocation problems with time windows are generally (except for a few special cases e.g. [Melvin et al., 2007]) NP-hard [Solomon and Desrosiers, 1988], and even the problem of verifying if feasible solutions exist is NP-complete. The inclusion of time windows makes it hard to design efficient approximation algorithms. Constraint-free task allocation problems accept greedy constant factor approximation for submodular utility functions [Segui-Gasco et al., 2015].

4.4. Precedence and Synchronization Constraints

Precedence constraints specify a partial or total order for the tasks, without providing a time window for each task. Time windows for tasks can be used to specify implicitly precedence or synchronization constraints, but in general they are not sufficient. Two time windows with the same start time do not necessarily indicate a synchronization constraint. Time windows that overlap are not sufficient to specify precedence constraints.



(a) soft deadline utility function (b) hard deadline utility function

Figure 3: Utility of soft deadlines vs. hard deadlines. The maximum utility is earned before the deadline. An exponentially decaying utility can be gained if the task finishes between the soft and hard deadlines (case (a)). No utility is gained after the (hard) deadline (case (b)).

Allocation of tasks with time windows is in general harder than allocation of tasks with only precedence constraints, because the tasks have to be scheduled within their time windows in such a way that there is sufficient time for their execution. If time windows are pairwise disjoint, except possibly for the end-points [Melvin et al., 2007], and robots move in 2D then the tasks are totally ordered and in special cases there is a polynomial solution.

The introduction of temporal and ordering constraints in general increases the complexity of task allocation. Solutions also might contain assignments to different robots of tasks that depend on each other, creating cross-schedule dependencies among robots Jones et al. [2011], Korsah [2011]. This is undesirable because exogenous events during execution affecting one robot will also affect the robots that depend on the affected robot.

4.5. Hard vs. Soft Temporal Constraints

Temporal constraints can be characterized as hard or soft constraints. Hard temporal constraints cannot be violated [Borning et al., 1992]. They are used in MRTA/TOC and related areas for tasks like surveillance, routing for blood supply, and order fulfillment by warehouse robots. Soft temporal constraints allow some temporal constraints to be violated or some tasks to be skipped entirely, but the robot incurs a penalty for doing so [Bistarelli et al., 2007, Domshlak et al., 2006, Gerevini and Long, 2006]. The penalty incurred may differ depending on which constraint was violated; for example, finishing tasks late may be penalized more severely than doing tasks early.

Fig. 3 illustrates the difference between a soft deadline utility function (left) and a hard deadline utility function (right).

Common types of soft temporal constraints include:

1. tasks can be started early and/or finish late with some penalty (called *soft* constraints in real time system terminology);

2. deadlines need to be satisfied only with some probability [Zheng and Woodside, 2003];

3. a number of consecutive tasks or some percentage of the tasks can be skipped [Bernat et al., 2001] without penalty (called *weakly hard* constraints in the real time systems terminology);

4. some tasks can be done late without reward, or skipped without penalty (called *firm* tasks in [Bernat et al., 2001]);

5. positive and negative preferences can be used as soft constraints [Bistarelli et al., 2007, Hoogendoorn and Gini, 2008].

Penalizing infeasible solutions obtained by relaxing some of the temporal constraints has been shown empirically to be useful to speed up and improve the quality of heuristic search for VRP Vidal et al. [2014], producing some new best known solutions for benchmark problems.

The *weakly hard constraints* from real time systems are not widely used in MRTA, even though they transfer quite sensibly into MRTA problems. In a weakly hard system with periodic task release, the distribution of met and missed deadlines in a time period is precisely bounded [Bernat et al., 2001]. Other approaches to skipping some deadlines for periodic tasks include degradation policies in overloaded systems [Beccari et al., 1999] or exploiting skips to improve response time for aperiodic tasks [Caccamo and Buttazzo, 1997].

There are four types of weakly hard constraints to consider: making any n in m deadlines, making n in a row in m deadlines, missing any n in m deadlines, and missing n in a row in m deadlines. In this way, any regularly scheduled sequence of tasks can allow some missed deadlines without penalty, while still allowing the agent responsible for those tasks to schedule and make most of its deadlines. Agricultural drones, for example, may have regularly scheduled sampling, such as fertilization, weed picking, or soil testing responsibilities that allow to skip a few deadlines.

Real world robot-task assignment problems might demand periodic tasks. A Mars rover, for example, has a regularly scheduled self-maintenance period, as well as periodic deadlines to finish uploading data or downloading instructions. These deadlines are usually hard deadlines, so the robot can shut down overnight and clear memory caches; other regularly scheduled robotic activities are not so sensitive to the time of execution.

Having discussed temporal models and constraints, and the nature of ordering constraints, we switch focus to optimization objectives. Determining these objectives is another important aspect to consider when building models for MRTA/TOC problems.

5. Optimization Objectives

Applications of MRTA/TOC problems require the robots to achieve a given optimization objective. In the rest of this paper we will refer to $f(\cdot)$ as a generic function representing one of these objectives. There can be a single or multiple objectives [Jozefowiez et al., 2008]. Depending on the deterministic or stochastic nature of the problem, objectives will either be over actual or expected values.

Optimization objectives might require a quantity to be minimized, usually a cost [Nunes and Gini, 2015, Gombolay et al., 2013, Chopra and Egerstedt, 2012] or regret [Heap and Pagnucco, 2014, Wu and Jennings, 2014], or to be maximized, usually a score [Mercker et al., 2010, Ponda et al., 2010] or a reward [Korsah et al., 2012, Melvin et al., 2007, Koes et al., 2005]. Single optimization objectives may be of spatial nature (e.g. minimize total distance traveled) or of temporal nature (e.g. minimize makespan).

Common optimization objectives for MRTA/TOC problems include:

- MiniSUM, i.e. minimize the sum of the robot path costs over all the robots [Lagoudakis et al., 2005]. Minimizing the distance traveled is common [e.g. Coltin and Veloso, 2014b, Chopra and Egerstedt, 2012, MacKenzie, 2003]) but some instead minimize a time measure over robot paths [e.g. Heap and Pagnucco, 2014, Barbulescu et al., 2010]).

- MiniMAX, i.e. minimize the maximum path cost of a robot over all the robots [Lagoudakis et al., 2005]. Instead of minimizing the maximum path cost, a similar objective function is to minimize the makespan, i.e. the time difference between the start of the first and the end of the last task [Graham et al., 1979]. In [Nunes and Gini, 2015] the makespan is minimized in a decentralized manner while in [Gombolay et al., 2013] the makespan, along with other objectives, is minimized using a near-optimal centralized MILP-based planner.

- MiniAVE: i.e. minimize the average per task cost of the path over all the tasks. The per task cost is the cost of the path from the robot initial location to the task location [Lagoudakis et al., 2005]. This is known as the Traveling Repairman Problem [Fakcharoenphol et al., 2007], where the objective is to minimize the wait time of the customers (or tasks) for a repairman (or robot).

- Minimize lateness or tardiness, which is the difference between the earliest start time of a task and the actual arrival time of the robot [Ponda et al., 2010, Rubinstein et al., 2012, Beck and Refalo, 2003]. A similar objective is to minimize the idle time of the robots [Hasgül et al., 2009].

- Maximize the number of tasks completed [Lau et al., 2003, Colorni and Righini, 2001] or minimize the number of tasks missed [Hasgül et al., 2009].

- Minimize the number of robots used. This is common in vehicle routing problems, where the number of vehicles available is unlimited [Luo and Schonfeld, 2007, Bräysy and Gendreau, 2005a, Desrochers et al., 1988].

- Maximize profit, measured as the difference between the reward of tasks and their respective costs [Korsah et al., 2012, Melvin et al., 2007], or as the team utility [Amador et al., 2014, Ponda et al., 2010, Koes et al., 2005].

While not extensively covered here, multi-objective problems are common [Jozefowiez et al., 2008], especially when objectives are combined through linear aggregation. For example, makespan and distance are minimized in [Ponda et al., 2010, Nunes and Gini, 2015], while [Gombolay et al., 2013] also minimizes workspace overlap. In [Alighanbari et al., 2003] a multi-objective function minimizes the maximum and average task completion times, as well as total idle times.

The problems in the subcategories of our taxonomy can be formalized using MIP formulations. In cases where tasks' locations or durations, or travel times are probabilistic, stochastic models (e.g. Markov Decision Processes) are more commonly used. We give specific examples of different objectives within the subcategories, using $f(\cdot)$ as a generic objective function. The constraints include coverage constraints that dictate the number of robots required to complete a task as well as the number of tasks a robot is allowed to complete at a time; ordering constraints, for example precedence and synchronization constraints; and side constraints, such as resource constraints.

6. Taxonomy

We are now ready to introduce our extensions to the taxonomy of Gerkey and Matarić [2004] and focus on time-extended assignments, in which robots build schedules for the tasks. We categorize the literature according to time window and precedence constraints. We add the following new axes in our taxonomy:

- *Time Window (TW) vs. Synchronization and Precedence (SP) constraints.* Within each subcategory, when appropriate, we further distinguish works that consider
  - (a) hard temporal constraints vs. soft temporal constraints. Hard temporal constraints require that no temporal constraint is violated, while soft temporal constraints allow some violations with a penalty.
  - (b) deterministic vs. stochastic models. In deterministic models the output of the model is completely determined by the initial conditions, while stochastic models assume a model of the uncertainty is available. Despite the importance of uncertainty in robotics, most MRTA models are deterministic and limit dealing with uncertainty at execution time.

We now illustrate our taxonomy in terms of single- vs. multi-task robots (SR vs. MR), single- vs. multi-robot tasks (ST vs. MT), and time windows vs. synchronization and precedence constraints (TW vs. SP). We begin with the least complex problem settings, in which single-task robots get allocated single-robot tasks.

6.1. ST-SR-TA:TW – Single-Task robots, Single-Robot tasks, Time-extended Assignments: Time Windows

6.1.1. Hard Temporal Constraints

Deterministic allocations. Deterministic ST-SR-TA:TW problems typically assume that there are more tasks than robots, and that all tasks are known in advance; they require time-extended assignments, and that tasks have to be performed within their time windows. These problems are comprised of three intertwined subproblems: (1) an assignment subproblem, to find the assignment of tasks to robots that optimizes the given objective function $f(\cdot)$; (2) a task sequencing subproblem, to find feasible orderings of tasks that result in optimal assignments, and (3) a scheduling subproblem, to assign times to tasks in a way that optimizes $f(\cdot)$.

We summarize the notation we use in Table 1.

Robots	
A	set of robots
a	robot in set A
q_a	capacity, or maximum workload, of robot a
Tasks	
K	set of tasks
k	task in set K
es_k	earliest start time of task k
ls_k	latest start time of task k
ef_k	earliest finish time of task k
lf_k	latest finish time of task k
st_k	actual start time of task k
ft_k	actual finish time of task k
du_k	duration of task k
$tt_{kk'}$	travel time between tasks k and k'
w_k^a	workload for task k when performed by robot a
Optimization	
$f(\cdot)$	generic optimization function
x_k^a	indicator of assignment of task k to robot a
$o_{kk'}^a$	indicator that robot a performs task k' directly after k
v_k^a	indicator that robot a performs task k first
z_k^a	indicator that robot a performs task k last
U_k^a	reward robot a collects for performing task k

Table 1: Notation used in the paper for tasks with time window constraints

Tasks have to be scheduled so that no constraint is violated. Temporal constraints are violated when robots do tasks at times that are not consistent with the temporal constraints on the tasks. Assignment violations occur when two or more robots are assigned to the same task, or two or more tasks are scheduled to be done at the same time by the same robot.

In the mixed integer linear programming formulation in Fig. 4, q_a is the capacity of robot a, st_k and ft_k are respectively the actual start and finish times for task k, $tt_{kk'}$ is the travel time between tasks k and k', w_k^a is the amount of work robot a has to perform when assigned task k. x_k^a is an indicator variable that takes the value 1 if robot a is assigned task k and 0 otherwise, $o_{kk'}^a$ is an indicator variable that takes the value 1 if robot a performs task k followed directly by task k', and 0 otherwise, v_k^a is an indicator variable that is 1 if task k is the first task in robot a's schedule and 0 otherwise, and z_k^a is an indicator variable that is 1 if task k is the last task in robot a's schedule and 0 otherwise. We assume all times are strictly positive.

minimize or maximize $f(\cdot)$
subject to

(a)	$\sum_{a \in A} x_k^a = 1$	$\forall k \in K_a^+$
(b)	$\sum_{k \in K_a^+} v_k^a = 1$	$\forall a \in A$
(c)	$\sum_{k \in K_a^+} z_k^a = 1$	$\forall a \in A$
(d)	$\sum_{k \in K_a^+} w_k^a x_k^a \leq q_a$	$\forall a \in A$
(e)	$\sum_{k \in K_a^+} o_{kk'}^a + v_{k'}^a = x_{k'}^a$	$\forall a \in A, k' \in K_a^+$
(f)	$\sum_{k' \in K_a^+} o_{kk'}^a + z_k^a = x_k^a$	$\forall a \in A, k \in K_a^+$
(g)	$es_k \leq st_k \leq ls_k$	$\forall k \in K_a^+$
(h)	$ef_k \leq ft_k \leq lf_k$	$\forall k \in K_a^+$
(i)	$ft_k - st_k \geq du_k$	$\forall k \in K_a^+$
(j)	$ft_k + tt_{kk'} - M * (1 - o_{kk'}^a) \leq st_{k'}$	$\forall a \in A, k \in K_a^+, k' \in K_a^+$
(k)	$x_k^a \in \{0, 1\}$	$\forall a \in A, k \in K_a^+$
(l)	$o_{kk'}^a \in \{0, 1\}$	$\forall a \in A, k \in K_a^+, k' \in K_a^+$
(m)	$v_k^a \in \{0, 1\}$	$\forall a \in A, k \in K_a^+$
(n)	$z_k^a \in \{0, 1\}$	$\forall a \in A, k \in K_a^+$

Figure 4: Mixed integer linear programming formulation of assignment of tasks with time windows

Since each robot starts at its initial location, we create an empty task for each robot a at its initial location. The empty task starts at time 0, and has a duration of ϵ. We indicate the set of all the tasks plus the empty task at the start location of robot a as $K_a^+ = K \cup \{\text{start location of } a\}$. ϵ should be smaller than the early start time of any task.

The objective function $f(\cdot)$ in the optimization formulation can be a cost function to be minimized [e.g. Gombolay et al., 2013]), or a value function to be maximized [e.g. Koes et al., 2005]). It can also be single or multi-objective.

For instance, to minimize the makespan the optimization function would be

$$\text{minimize}_{x_k^a, o_{kk'}^a, z_k^a, st_k, ft_k} \max_{a \in A} \max_{k \in K_a^+} ft_k \tag{1}$$

where ft_k is the actual finish time of task k.

For ST-SR-TA:TW problems the coverage constraints in Fig. 4 enforce that (a) each task gets at most one robot, each robot has a first (b) and last (c) task and that (d) each robot has no more tasks than its capacity allows.

The sequencing constraints require that (e) every task k assigned to robot a except the first has a predecessor, and that (f) every task except the last has a successor. Temporal constraints (g)–(j) are constraints on the service times of tasks. Constraint (j) ensures that the interval between two consecutive tasks is large enough for the robot to travel to it. The constraint includes a sufficiently large constant M to make the formulation a mixed-integer linear program. Constraints (k)–(n) bound the values for the indicator variables.

Advances in MILP formulations for VRPTW [Barnhart et al., 1998, Feillet, 2010] and more recently for MRTA problems [Korsah et al., 2012] have proposed formulations based on set covering and set partitioning. These formulations assign routes, instead of tasks, to robots. The allocation problem is decomposed into what is known as the master problem, and a pricing subproblem. One of the advantages of such formulations is that the master problem can be restricted to evaluating subsets of tasks at a time, instead of the entire set of tasks. Pricing subproblems solve temporally constrained shortest path problems rooted at robot locations, in which routes can be computed via heuristic methods, such as D* lite as in [Korsah et al., 2012]. Such formulations benefit from the insight that for very large problems many routes are not part of any optimal solution. Thus, selectively incrementing candidate routes decreases computational and memory costs. Feillet [2010] provides a technically rigorous overview of such formulations and their advantages for VRPTW problems.

Y_a	set of possible routes y for agent a
s_y^a	indicator that route y is assigned to agent a
b_{yk}^a	binary constant that task k is in route y of agent a
C_y^a	cost to robot a of route y

Table 2: Notation used for formulation based on set partitioning

As defined in Table 2, let Y_a be a set of routes for robot a computed using the shortest path algorithm with resource constraints; s_y^a is an indicator variable that assumes a value of 1 if robot a is assigned route $y \in Y_a$ and 0 otherwise; C_y^a is the expected cost robot a incurs for performing route y; finally, b_{yk}^a is a binary constant that is 1 if task k is performed in route $y \in Y_a$ of robot a and 0 otherwise. An example of a simple set partitioning formulation of ST-SR problems is shown in Fig. 5. A more complex formulation with cross-scheduling temporal and location dependencies, time windows, precedence and synchronization constraints can be found in Korsah [2011].

Stochastic allocations. In stochastic problems, it is assumed that a model of uncertainty is available. Stochastic ST-SR-TA:TW problems, like other stochastic problems in our taxonomy, are usually modeled as pure or mixed stochastic

minimize $\sum_{a\in A}\sum_{y\in Y_a} C_y^a s_y^a$		
subject to		
(a) $\sum_{y\in Y_a} s_y^a \le 1$	$\forall a \in A$	Every robot gets at most 1 route
(b) $\sum_{a\in A}\sum_{y\in Y_a} s_y^a b_{yk}^a = 1$	$\forall k \in K$	Each task is on 1 route
(c) $s_y^a \in \{0,1\}$	$\forall a \in A, y \in Y_a$	Indicator of route to robot

Figure 5: Set partitioning formulation for MRTA/TOC problems with no time windows and no capacity limits.

integer programs, or as Markov Decision Processes (MDPs) [Gendreau et al., 1996]. When modeled as stochastic integer programs [Ponda et al., 2012b] they assume the form in Eq. 2 with the constraints shown in Fig. 4 or stochastic constraints [Shen et al., 2009]. In Eq. 2 the objective function is the expected reward, $\theta \in \Theta$ is the uncertainty model that is available to the robots, and U_k^a is the reward that agent a gets for doing task k.

$$\text{maximize } \mathbb{E}_\theta(\sum_{a\in A}\sum_{k\in K} U_k^a x_k^a) \tag{2}$$

Examples of uncertainty models include probability distributions for task arrival, robot travel time, task availability, and more [Miao et al., 1991].

Other stochastic formulations are used in the dynamic and stochastic VRPTW literature. For instance, [Bopardikar et al., 2014] studied a dynamic VRP problem in which demands (or tasks) with deterministic time constraints arrive randomly, and the goal is to maximize the fraction of demand met. In [Pavone et al., 2009] demand is stochastic and time window constraints are considered. Both [Bopardikar et al., 2014] and [Pavone et al., 2009] analyze a different number of requirements, such as bounds on the number of vehicles used and maximum number of tasks that can be missed. In both, temporal constraints cannot be violated. However, in order to prove properties about their solutions, some strong assumptions are made, such as all time windows have the same length [Pavone et al., 2009].

An alternative way of modeling uncertainty uses MDPs. In [Dean et al., 1993, Beynier and Mouaddib, 2007] MDP states are locations in a map with obstacles, tasks and robots. In [Beynier and Mouaddib, 2007] a state is a triplet representing the previously visited state, the amount of resources left, and the time window. The goal is to search for policies that maximize a value function for the augmented states. Dolgov et al. [2007] poses the problem as a combinatorial resource scheduling problem with uncertainty, which can be easily extended to include locations, forming a MRTA problem.

Uncertainty models for ST-SR-TA:TW problems are, to the best of our knowledge, rarely explored in the MRTA literature, although stochastic planning could lead to practical gains in terms of producing sound and robust allocation policies for robots. Instead, it is more common to address stochasticity by model-free methods, such as reinforcement learning, or to deal with uncertainty by replanning during task execution.

6.1.2. Soft Temporal Constraints

Deterministic allocations. Deterministic ST-SR-TA:TW with hard and soft constraints differ only in the hardness of the time constraints. Classic problems include vehicle routing problems with soft scheduling constraints. We will also look at problems and solutions in the real time systems and artificial intelligence literature to assist in modeling temporal and ordering constraints in MRTA.

In soft time window constraints for vehicle routing, the goal is to find the best agent-task assignments that minimize the cost function $f(\cdot)$ of servicing some number of clients. The total cost of assigning a set of agents/vehicles and departure times is equal to the fixed cost of operating the agents, plus the cost of operating the agents on the specific routes, plus the penalty cost for arriving early or late to the clients on the routes [Taillard et al., 1997]. Penalty costs for arriving early may be different than for arriving late, (e.g. early arrival is a small penalty, late arrival is a larger penalty) and these may vary by domain.

Stochastic allocations. Stochastic versions of ST-SR-TA:TW problems with soft constraints have an uncertainty model available like in the hard constraint case. However, they use soft windows and allow agents to gain value even when doing tasks outside their original time window. The objective is still to minimize a cost function (e.g distance or energy) or maximize a utility function, with the inclusion of some probability model; often these are probabilities of travel delay between tasks and therefore travel times, but could be specific to the task and affect other costs. For instance, Hsu et al. [2007] models the process of delivering perishable food, which affects inventory costs. The cost function now includes the cost of using vehicles and the cost of arriving to a task outside the proper time window. Work in [Taş et al., 2013] models travel time delays with several distribution types, which change the service cost of operating a vehicle. The travel time probability directly impacts whether the agent arrives early or late, which is why we frequently see stochastic formulations in soft time windows but no other kinds of preferred constraints.

6.2. ST-SR-TA:SP – Single-Task robots, Single-Robot tasks, Time-Extended Assignments: Synchronization and Precedence

Synchronization and precedence constraints can be formulated as in [Bredström and Rönnqvist, 2008] (Eq. 3 and Eq. 4). Let $k, k' \in P$ where P is a set of task pairs with precedence constraints, and $P^{sync} \subseteq P$ is the subset of tasks that have to start at the same time. Eq. 3 states that regardless of which robot(s) is assigned to tasks k and k', task k' should start ϵ time units after the finishing time of task k. If $\epsilon > 0$ $k, k' \in P$ (Eq. 3), and if $\epsilon = 0$ then $k, k' \in P^{sync}$ (Eq. 4).

$$\sum_{a \in A} st_{k'} x^a_{k'} - \sum_{a \in A} ft_k x^a_k > \epsilon + M(1 - o^a_{kk'}) \quad \forall a \in A, k, k' \in P, \epsilon > 0 \tag{3}$$

$$\sum_{a \in A} st_{k'} x^a_{k'} - \sum_{a \in A} st_k x^a_k = 0 \qquad k, k' \in P^{sync} \tag{4}$$

ST-SR-TA:SP problems have received some attention in the MRTA literature [McIntire et al., 2016, Luo, 2014, Chopra and Egerstedt, 2012, Korsah et al., 2012, Barbulescu et al., 2010]. Luo [2014] present a model for tasks with set precedence constraints that divides tasks into disjoint sets with strict ordering between the sets, assuming that each robot can do at most one task per set. The model heavily constrains the type of allowable precedence graphs, but the algorithm proposed is proved to be sound and complete. A general model for allocation of tasks with any type of precedence constraint is presented in [McIntire et al., 2016].

Sometimes executing a task precludes the execution of another. This type of problem is typically addressed at the planning stage, enforcing precedence constraints between the tasks [e.g. Olawsky and Gini, 1990].

These different works highlight that synchronization and precedence constraints can be used to model different types of temporal relationships between tasks. The applications range from a music wall [Chopra and Egerstedt, 2012] to structure assembly by a team of robots [Heger et al., 2005]. The common thread among the referenced works is that precedence constraints are in the form that the start time of a task cannot occur earlier than the end time of any of its predecessors. Other, lesser used, precedence models include start to start, start to end and end to end constraints [Lombardi and Milano, 2012]. Start to start constraints require that a task does not start until its precedents have started, the remaining types of precedence constraints follow a similar interpretation.

6.3. ST-MR-TA:TW – Single-Task robots, Multi-Robot tasks, Time-Extended Assignments: Time Windows

6.3.1. Hard Temporal Constraints

In ST-MR-TA:TW allocation problems agents are scheduled to work simultaneously on tasks as coalitions while respecting the time window constraints. Coalition-based task allocation occurs when tasks cannot be executed by a single agent, or when task execution is more efficient when done by more agents [Vig and Adams, 2006, Shehory and Kraus, 1998]. In disaster rescue, for instance, fire fighters working in coalitions may extinguish the same number of fires earlier than if these rescuers had to work individually on each fire [Parker et al., 2015]. Moreover, in scenarios where the number of agents is limited, coalition-based allocations may enable a higher task completion rate [Ramchurn et al., 2010b, Su et al., 2016].

Coalition formation, in general, requires dealing with two subproblems: coalition value computation and coalition structure generation [Sandholm et al., 1999]. The former is concerned with computing the expected utilities (or costs) of forming all possible coalitions, whereas the latter is concerned with partitioning the set of agents into exhaustive and disjoint groups that maximize the total utility. In MRTA, the coalition value is typically a combination of the utility gained and the coordination cost necessary to perform a task. Coalition size may be restricted by the physical constraints which limit the number of agents that can work simultaneously on the same task.

Let 2^A be the set of agent coalitions that may be formed with the agents in A (i.e., all subsets of A) and x_k^c be an indicator variable that takes the value of 1 if coalition $c \in 2^A$ is assigned to task k and 0 otherwise. For simplicity, we assume that all agents start their tours from an initial node 0 and finish at node $m+1$. Let $o_{kk'}^a = 1$ when agent a visits task k' directly after k and 0 otherwise. $o_{0k}^a = 1$ denotes the fact that a visits k at the very beginning of the route. $|c|$ indicates the coalition size. Similarly, $o_{k(m+1)}^a = 1$ when agent a visits task k at the end of its route, and 0 otherwise. ST-MR-TA:TW allocation problems can generally be formalized by the MILP in Fig. 6.

minimize or maximize $f(.)$
subject to

(a)	$\sum_{c \in 2^A} x_k^c \leq 1$	$\forall k \in K$
(b)	$\sum_{a \in c} x_k^a = \|c\| x_k^c$	$\forall c \in 2^A, k \in K$
(c)	$\sum_{k \in K} o_{0k}^a = 1$	$\forall a \in A$
(d)	$\sum_{k \in K} o_{k(m+1)}^a = 1$	$\forall a \in A$
(e)	$\sum_{k \in K, k \neq k'} o_{kk'}^a - \sum_{k'' \in K, k' \neq k''} o_{k'k''}^a = 0$	$\forall a \in A, k' \in K$
(f)	$st_k + du_k + tt_{kk'} - M * (1 - o_{kk'}^a) \leq st_{k'}$	$\forall a \in A, k, k' \in K$
(g)	$es_k \leq st_k \leq ls_k$	$\forall k \in K$
(h)	$ef_k \leq ft_k \leq lf_k$	$\forall k \in K$
(i)	$ft_k - st_k \geq du_k$	$\forall k \in K$
(j)	$x_k^a \in \{0, 1\}$	$\forall a \in A, k \in K$
(k)	$x_k^c \in \{0, 1\}$	$\forall c \in 2^A, k \in K$
(l)	$o_{kk'}^a \in \{0, 1\}$	$\forall a \in A, k \in K, k' \in K$

Figure 6: Standard mixed integer linear formulation of the task allocation problem with single-task robots, multiple-robot tasks, and hard temporal constraints

Constraint (a) guarantees allocations of no more than one coalition per task. As the problem maybe over-constrained, not all tasks may be allocated. Constraint (b) guarantees if a coalition is assigned to a task then all the agents in the coalition are assigned to that task too. Constraint (c) guarantees that all the agents start from the initial location, and constraint (d) ensures that they finish their routes at the final location. Constraint (e) guarantees the connectivity of the routes, so that a robot reaches all its assigned tasks in sequence. Constraints (f)–(i) ensure that the visit time-line is feasible and the time windows are respected (as in MILP for ST-SR-TA:TW in Fig. 4). An extra waiting time may be imposed after the task's earliest start time to form the coalition. When coalition work affects the task execution efficiency, the task duration (du_i)

should be computed accordingly, [Ramchurn et al., 2010b]. Constraints (j)–(l) bound the values for the indicator variables.

An alternative model for ST-MR-TA:TW problems is the set partitioning model shown earlier in Fig. 5. When cast as a set partitioning problem, a set of coalitions $C = \{c_1, \dots, c_{|C|}\}$ corresponds to a set partition of A if and only if the coalitions are exhaustive, i.e. $\bigcup_{c_i \in C} c_i = A$, and the elements of C are pairwise disjoint, i.e., $\forall c_i, c_j \in C$ s.t. $i \neq j : c_i \cap c_j = \emptyset$. The solution to the set partition problem is a partition of A that maximizes the utility $u : S \to \mathbb{R}^+$. The NP-hard nature of problems in this class require approximate solutions for practical coalition-based MRTA problems.

Certain side constraints, such as capability and resource constraints, are very important in this class of MRTA/TOC problems. Agents may have limited resources, especially in the case of small robots. For instance, in disaster rescue scenarios, fire trucks may need a certain amount of fuel to travel to a fire and a certain amount of water to extinguish it. Tasks might require coalitions of agents with certain capabilities. For instance, a fire might require a coalition of fire fighters, while police and ambulances can collaborate to dig out and carry survivors to refuge centers [Kitano and Satoshi, 2001, Parker et al., 2015].

MRTA researchers have proposed coalition-based frameworks for heterogeneous robotics. Examples include the ASyMTRe [Parker and Tang, 2006] architecture. ASyMTRe is a reasoning system for heterogeneous robots to form coalitions to do tasks that require tight robot coordination. The architecture uses a collection of schemas for perception and motor control, which are connected at run time, enabling the robots to share information as needed to complete the tasks. The architecture has been extended [Zhang and Parker, 2013a] to ensure that only feasible coalitions are formed. Efficient scheduling heuristics for coalitions are proposed in [Zhang and Parker, 2013b].

Stochastic allocations. To the best of our knowledge, no literature addresses stochastic ST-MR problems with either hard or soft constraints. We will now examine tasks with soft constraints.

6.3.2. Soft Temporal Constraints

ST-MR-TA:TW with soft constraints assumes that multiple agents can work simultaneously on the same task and are allowed to violate some temporal constraints, with a penalty for the violation. ST-MR-TA:TW problems with soft and hard constraints are similar, except that the objective function for the soft constraint case includes a temporal violation penalty (Eq. 5).

$$\underset{S \in 2^A}{argmax} \sum_{c \in S} \sum_{k \in K} x_k^c U_k^c \pi_k^{st_k} \tag{5}$$

In Eq. 5 S is a coalition structure, U_k^c is coalition c's utility for performing task k and $\pi_k^{st_k} \in [0, 1]$ is the utility decay coefficient function for task k. This coefficient is set as $\pi_k^{st_k} = 1$ when task k is started and/or finished within the time window (i.e., $es_k \leq t \leq ls_k$ and $st_k + du_k \leq lf_k$). Early and/or late task executions are penalized by setting $\pi_k^{st_k}$ to values in the range $[0, 1]$. In particular, $\pi_k^{st_k} = 0$, $\forall k \in K$ when $st_k + du_k > ls_k$ [Koes et al., 2005, Amador et al., 2014].

Most research on ST-MR-TA:TW problems with soft constraints is motivated by application areas such as urban search and rescue [Koes et al., 2005, Scerri et al., 2005], or law enforcement where police officers are assigned to crime events in a city [Amador et al., 2014]. In [Scerri et al., 2005] an expected utility model is used to allocate interdependent tasks. Late task executions are penalized by subtracting the delay cost from the total utility. The work subdivides large tasks into smaller subtasks that are linked with simultaneous execution interdependency, and coalitions of agents execute the smaller subtasks. The coalition formation problem is simplified by fixing the coalition size and reducing the number of allowed coalitions.

In [Koes et al., 2005] the task utility decays over time from the beginning of the mission and becomes zero by the mission deadline. Likewise, in [Amador et al., 2014] the utility of tasks delayed beyond the soft deadline decays exponentially over time. The coalition value depends on the number of agents and is a function of the agents' fitness in performing a task.

6.4. ST-MR-TA:SP – Single-Task robots, Multiple-Robot tasks, Time-Extended Assignments: Synchronization and Precedence

ST-MR-TA:SP is comprised of the same subproblems as ST-MR-TA:TW, the only exception comes from the inclusion of precedence constraints. ST-MR-TA:SP problems have received more limited attention than the ST-SR-TA:SP counterpart. Part of the reason might be the fact that current robotics applications do not use coalitions as a way to achieve tasks more efficiently.

In [Shehory and Kraus, 1998] tasks that require a set of capabilities are allocated to a set of robots with different types of capabilities. Robots form coalitions to perform tasks with precedence constraints. The work proposes greedy distributed set partitioning and set covering algorithms to give an approximate answer to the problem. Propose a framework in which coalition formation and task allocation are solved in turn. The coalition formation uses ASyMTRe and auctions are used for allocation of tasks with precedence constraints to coalitions. Coalitions bid through a coalition leader. In [Tang and Parker, 2007] the coalition formation problem is solved using ASyMTRe, and auctions are used for allocation of tasks with precedence constraints to coalitions. Coalitions bid through a coalition leader. The common theme among these works is that they do not handle synchronization constraints.

In [Sariel and Balch, 2006] precedence and synchronization constraints are considered. The work handles dynamic allocation of tasks with precedence and synchronization constraints. In addition to precedence tasks also require that a certain number of robots work in them. The goal is to minimize the makespan. It combines auctions with coalition maintenance, and employs recovery routines to deal with exogenous events. In addition, [Jones et al., 2011, Parker et al., 2015] handle both temporal and precedence constraints. The precedence constraints are in the form of deadline constraints. The latter two model search and rescue domains.

Works that address the Robocup Search and Rescue (e.g [Parker et al., 2015, Scerri et al., 2005]) problem often span both ST-MR-TA:TW and ST-MR-TW:SP, because robots can only collect utilities on tasks before tasks expire, and some tasks can only be performed after others are completed (e.g blockades need to be removed before fires are extinguished).

6.5. *MT-SR-TA:TW – Multi-task robots, Single-robot tasks, Time-extended Assignments: Window Windows*

Deterministic allocations. The MT-SR problem with hard temporal constraints is no more common now than it was in [Gerkey and Matarić, 2004], but we can provide some additional context for multi-task robots regardless of hard or soft time constraints. Gerkey's work likens the MT-SR problem to the ST-MR problem, using the same mathematical formulation for both problems but switching the role of tasks and agents in the formula. Multi-task robots do exist in real life, however, such as the mission-driven Mars rover Curiosity, so the multi-task robot problem merits deeper discussion before we consider it in terms of hard or soft time windows.

A multi-tasking robot can perform located tasks, such as grasping and manipulating objects, and non-located tasks, such as taking pictures of nearby objects. Furthermore, located tasks can be near to or far away from the robot; for example, an object two feet in front of the robot is close, but an object ten feet away is probably considered far from the robot. Nearby objects should be relatively easy to grasp, but farther-away objects will require larger or longer actuators and thus more complex kinematic calculations to properly manipulate them. Consider unmanned aerial vehicles; reconnaissance drones may track objects and take pictures (a relatively easy task) or may need to track objects on the ground and drop packages (a more difficult task that includes more intense object manipulation).

Lastly, a multi-tasking robot can either preempt tasks or not; preemptable tasks require priority knowledge and may require task rescheduling, whereas a simpler system with non-preemptive tasks may miss important tasks that arrive during execution. In preemptive cases where the robot was physically manipulating the environment, additional overhead time is required to restore the robot's pose and to continue grasping or other movement [Groth and Henrich, 2014]. In Amador et al. [2014] tasks can be interrupted by higher priority tasks and resumed later with a penalty that decreases over time. Additionally, the robot must deal with failures; not only must the robot prioritize tasks, but it must decide (or have a plan for) what to do when the preempting task fails. Does the robot retry the failed task, move directly back to the preempted task, or drop into some kind of re-calibration or maintenance mode? Consider the Mars rover – if it runs into a rock or becomes stuck while navigating to a site where it has to perform chemical analysis, it should stop and get unstuck (or consult Earth-based humans for assistance), then return to navigation towards its earlier goal. If instead a piece of the rover's chemical analysis fails due to hardware problems, it should probably stop all analyses until it can relay its problems and receive solutions from Earth.

Very limited literature exists on multi-tasking robots; much of the work focuses instead on robots that have many tasks to do very close together temporally, for example a UAV that searches for objects, targets an object, and releases a bomb. Groth and Henrich [2014] discusses a multi-tasking robot with nearby tasks (taking pictures of people and finding objects, or taking pictures of walls and greeting humans) with preemption; the robot stops taking pictures of walls if a human is in the way, for example.

In general, the MT-SR problem, regardless of the type of time window, can be approached heuristically as a bin packing problem, where each robot is a bin and each task is assigned to a robot that has available capacity and resources to perform that task. Other approaches to scheduling tasks are inspired by operating systems, such as shortest job first and priority scheduling; however, because context switching is more time-demanding for robots than it is for processors, these methods may be highly suboptimal.

Stochastic allocations. As far as we know, no literature has explored the addition of stochastic formulations with temporal constraints on the multi-task robot problem. Probabilistic events, such as a delivery robot's travel times due to traffic, would affect the agent's ability to perform single tasks, just as stochastic formulations do in the ST-SR problem.

6.6. MT-SR-TA:SP: Multiple-Task robots, Single-Robot tasks, Time-Extended Assignments: Synchronization and Precedence

Like its MT-SR-TA:TW counterpart, these types of problems have not received much attention in the MRTA literature. An exception is the work by Landén et al. [2012], which provides a distributed solution to a problem with precedence constraints. Complex tasks are modeled by a *task specification tree*, which specifies precedence and dependencies between tasks. A distributed constraint satisfaction solver is used to check constraint consistency. Task allocation is performed by a recursive search over feasible allocations.

6.7. MT-MR-TA:TW – Multi-Task robots, Multi-Robot tasks, Time-Extended Assignments: Time Windows

Multi-task robots and multi-robot task problems remain sparsely explored [Korsah et al., 2013, Gerkey and Matarić, 2004], even when additional temporal constraints are not considered. This class of problems can be modeled as an overlapping coalition formation problem [Chalkiadakis et al., 2010] combined with a routing and scheduling problem. Standard coalition formation methods produce either a super-coalition (with all the robots) or a set of non-overlapping subsets of robots.

MT-MR-TA:TW problems are comprised of the following subproblems: (1) assigning coalitions to tasks, (2) assigning different coalitions to the same robot as long as no resource constraints are violated, and (3) assigning values to the start and finishing times of tasks. Each of these subproblems is NP-hard.

Multi-task robots and multi-robot task problems can also be modeled as cooperative games with overlapping coalitions. In cooperative games with overlapping coalitions, agents can do more than one task at a time. This may lead robots to commit to a task assigned to more than one coalition. Overlapping coalitions have been used to model collaborative smartphone sensing in [Di et al., 2013]. In that work, smartphone users form overlapping networks, and an incentive function rewards users' contributions to different tasks. Unfortunately, finding the optimal overlapping coalition is NP-complete.

MT-MR-TA:TW problems with soft constraints also lack coverage in the MRTA literature. The models used for MT-MR-TA:TW with hard constraints can be extended to this class of problems, the only difference being that temporal constraints are allowed to be violated. This requires that one of the optimization objectives minimizes penalties from violating the constraints.

6.8. MT-MR-TA:SP – Multi-Task robots, Multi-Robot tasks, Time-Extended Assignments: Syncronization and Precedence

We could not find works in this group.

6.9. Summarizing the Taxonomy

We summarize the literature (see Table 3) by classifying papers from the MRTA and related literatures from 2003-2016. In Table 3 *Det* and *Sto* stand for deterministic and stochastic.

7. Dynamic Task Release and Execution

Execution of tasks in MRTA/TOC problems vary according to the dynamics considered. Dynamics may be due to faulty robots, changes in estimated cost of tasks due to uncertainties, changes in task definitions, online arrival of tasks, addition of robots to the team, and other changes made by external agents [Sariel-Talay et al., 2009]. While the execution aspect is outside of the task allocation scope, the planning-execution-replanning of tasks forms a planning loop that is usually addressed at once in dynamic domains. Here we consider dynamics caused by task arrival and during task execution separately.

Some dynamics are caused by the arrival of tasks over time without further knowledge of future tasks. Usually when a new task arrives there is already an existing allocation for previously scheduled tasks that have not yet been performed. Thus, replanning occurs at task arrivals, while robots are executing previously assigned tasks [Cordeau and Laporte, 2007]. In [Nunes and Gini, 2015] both deterministic and dynamic task arrivals are considered, assuming the robots have perfect knowledge of the map where tasks appear. In contrast, problems usually defined as online pickup and delivery problems or dial-a-ride include not only online arrival of tasks but other uncertain events, such as vehicle breakdowns and delays [Cordeau and Laporte, 2007]. Recent examples of online pickup and delivery consider transfers, in addition to the arrival of tasks with hard temporal constraints [Coltin and Veloso, 2014b,a, Bouros et al., 2011].

Reference	ST	MT	SR	MR	TW	SP	HC	SC	Det	Sto
[McIntire et al., 2016]	✓		✓			✓	✓		✓	
[Luo et al., 2015]	✓		✓		✓		✓		✓	
[Nunes and Gini, 2015]	✓		✓		✓		✓		✓	
[Coltin and Veloso, 2014c]	✓		✓		✓		✓	✓	✓	
[Luo, 2014]	✓		✓			✓	✓		✓	
[Gombolay et al., 2013]	✓		✓		✓	✓	✓		✓	
[Taş et al., 2013]*	✓		✓		✓			✓		✓
[Chopra and Egerstedt, 2012]	✓		✓		✓		✓		✓	
[Korsah et al., 2012]	✓		✓		✓	✓	✓		✓	
[Barbulescu et al., 2010]	✓		✓		✓	✓	✓		✓	
[Ponda et al., 2010]	✓		✓		✓		✓		✓	
[Pavone et al., 2009]*	✓		✓		✓		✓			✓
[Shah et al., 2009]	✓		✓		✓	✓	✓		✓	
[Bredström and Rönnqvist, 2008]*	✓		✓		✓	✓	✓		✓	
[Beynier and Mouaddib, 2007]	✓		✓		✓		✓			✓
[Melvin et al., 2007]	✓		✓		✓		✓		✓	
[Ando and Taniguchi, 2006]*	✓		✓		✓		✓			✓
[Heger et al., 2005]	✓		✓			✓	✓		✓	
[Alighanbari et al., 2003]	✓		✓			✓	✓		✓	
[Su et al., 2016]	✓			✓	✓		✓		✓	
[Pujol-Gonzalez et al., 2015]	✓			✓	✓	✓	✓		✓	
[Parker et al., 2015]	✓			✓	✓	✓		✓	✓	
[Amador et al., 2014]	✓			✓	✓			✓	✓	
[Jones et al., 2011]	✓			✓		✓	✓		✓	
[Ramchurn et al., 2010b]	✓			✓	✓		✓		✓	
[Tang and Parker, 2007]	✓			✓		✓	✓		✓	
[Sariel and Balch, 2006]	✓			✓		✓	✓		✓	
[Scerri et al., 2005]	✓			✓	✓	✓		✓	✓	
[Koes et al., 2005]	✓			✓	✓	✓		✓	✓	
[Landén et al., 2012]		✓	✓			✓	✓		✓	

Table 3: Select papers from each category. Papers with a * symbol are not MRTA papers, but are included for completeness.

The dynamics that occur during plan execution [Block et al., 2006, Sariel-Talay et al., 2009, Shah et al., 2009, Barbulescu et al., 2010, Ponda et al., 2010] are very important for the practical use of robots, because execution can fail due to many reasons and replanning is essential to maintain some level of efficiency. In [Barbulescu et al., 2010] dynamics during execution are created by unexpected events and changes in costs and constraints; in [Ponda et al., 2010] dynamics are caused by breaks in communication links, which may cause conflicting assignments, as more than one robot could be assigned the same task. In [Usug and Sariel-Talay, 2011] temporary failures are considered, such as obstacles, which can be overcome by replanning.

8. Typical Solution Approaches

So far, we have proposed a taxonomy for MRTA/TOC problems; now we discuss the most popular solutions and how to map these to our taxonomy. Here we simply divide the methods into centralized vs. decentralized. Centralized methods are further separated into exact and approximate methods, while decentralized methods are grouped into distributed constraint-based and market-based according to nature of the proposed solutions.

8.1. Centralized Solutions

Centralized methods rely on a central controller that allocates tasks to robots. The autonomy of the robots in pure centralized methods is limited or non-existent, as they solely execute the dispatched orders and do not make decisions on what tasks to do.

MRTA/TOC is intractable for a non-trivial number of robots and tasks. Optimal centralized solutions are intractable because they need to evaluate a large number of candidate solutions in order to guarantee optimality. Thus, the focus of MRTA/TOC solutions is largely on approximation and heuristic solution methods. We discuss some of the common centralized exact and heuristic methods next.

8.1.1. Exact Solutions

Exact solutions are optimal, but their computation time is impractical for realistic robotics applications. The most naive way to search for such solutions is to exhaustively search for all possible allocations that do not violate the temporal constraints. This is, however, intractable, because an exhaustive search leads to worst-case $O(|K|!)^{|A|}$ complexity for $|K|$ tasks and $|A|$ robots. We have to search through all the possible sequences of tasks and all possible allocations of tasks to robots, and in addition to all feasible assignments of times to tasks.

Optimal solutions can be more efficiently computed using Branch-and-Bound (B&B) [Clausen, 1997] and its variants: Branch-and-Cut [Ropke et al., 2007, Bard et al., 2002], Branch-and-Price [Korsah et al., 2012, Feillet, 2010, Dohn et al., 2009, Barnhart et al., 1998], and Branch-Price-and-Cut [Barnhart et al., 2000]. B&B searches the state space of candidate solutions represented as a tree and uses upper and lower bounds of the optimal solution to prune the branches of the search tree that have costs higher than the computed lower bounds. Among the variants, Branch-Price-and-Cut is becoming popular in VRPTW [Bettinelli et al., 2011, Archetti et al., 2011, Desaulniers, 2010, Ropke and Cordeau, 2009]. However, as far as we know it has not been used in MRTA/TOC problems.

Exact methods often use tools such as CPLEX [ILOG, 2006], Gurobi [Gurobi Optimization, 2014], ABACUS [Jünger and Thienel, 2000], lp_solve [Berkelaar et al., 2004] or other tools to build and solve the underlying MILP formulations. MILP-based formulations and solutions have been predominantly used in ST-SR-TA:TW problems (e.g., [Korsah et al., 2012, Alighanbari et al., 2003], but these models have also been used in other parts of the taxonomy (e.g. ST-MR-TA:TW [Ramchurn et al., 2010b, Koes et al., 2005]).

8.1.2. Approximate and Heuristic Solutions

To reduce computation time, MILP-based heuristics are used to find approximate partial allocations while searching the state-space tree. Such approaches have been used to address problems in our taxonomy (e.g. ST-SR-TA:TW [Gombolay et al., 2013, Korsah et al., 2012]). As far as we know, these methods do not provide any theoretical guarantees, but in some cases (e.g. [Gombolay et al., 2013]) they experimentally achieve results that are only 10% away from the optimal value (makespan).

Another way to gain computational efficiency is to use metaheuristic approaches. Metaheuristics are algorithmic templates that approximately solve hard combinatorial optimization problems. Unlike other combinatorial optimization algorithms, metaheuristics may allow lower quality solutions in the search process to escape local optima, and often embed off-the-shelf heuristics to solve the problem [Bräysy and Gendreau, 2005b, Vidal et al., 2013].

Metaheuristic approaches to VRPTW, TOPTW and related routing and scheduling problems have been shown to outperform many other methods (e.g. construction heuristics and local search) for standard benchmarks [Bräysy and Gendreau, 2005b, Hu and Lim, 2014]. Recent trends in the metaheuristic literature seek to reduce the computation time and improve the solution quality by using parallelization and hybridization of different heuristics and exact techniques. However, metaheuristic parameters remain hard to tune [Birattari, 2009, Bräysy and Gendreau, 2005b].

8.2. Decentralized Solutions

Decentralized approaches vary widely; a detailed categorization is outside the scope of this paper and we refer the reader to [Mosteo and Montano, 2010, Koenig et al., 2010]for more thorough taxonomies on MRTA methods. Here we focus on distributed constraint optimization and market- and negotiation-based algorithms since these have received a great deal of attention in the MRTA community.

8.2.1. Distributed Constraint (DCOP)-Based Methods

MRTA/TOC problems can be modeled as a Distributed Constraint Optimization Problem (DCOP) [Maheswaran et al., 2004] and solved using DCOP methods. Solving DCOP exactly is NP-hard and impractical even for unconstrained MRTA problems [Junges and Bazzan, 2008]. Thus, approximate methods such as Max-Sum have been used for task allocation in sensor networks [Farinelli et al., 2014] and in RoboCup Rescue [Ramchurn et al., 2010a, Pujol-Gonzalez et al., 2015].

Ramchurn et al. [2010a] proposed the Fast Max-Sum algorithm, which was shown to be robust in situations where the number of tasks is dynamic; the approach reduced the computation time, number and size of messages sent compared to Max-Sum, but it is still exponential. The computation overhead in dynamic environments is reduced in [Macarthur et al., 2011] by using online domain pruning and branch-and-bound. When the constraints are Tractable

Higher Order Potentials the computation time can be reduced to polynomial [Pujol-Gonzalez et al., 2015].

Another approximation method is LA-DCOP [Scerri et al., 2005, Farinelli et al., 2006], which uses token passing [Xu et al., 2005] as follows: when an agent perceives a task, it creates a token for it. It can decide to do the task or pass the token to a randomly chosen agent. This tends to guide the search quickly towards a greedy solution, which is reasonable for ST-SR-TA:TW problems with hard constraints.

In [Ferreira et al., 2008] LA-DCOP and Swarm-GAP are compared in RoboCup settings. In Swarm-GAP an agent chooses a task according to a probability that depends on the stimulus generated by the task and the agent's threshold. Results show that both DCOP approaches behave similarly, and both perform better than a greedy task allocation. Their approach works for ST-MR-SC problems, where agents are allowed to arrive late to tasks. Recently, to facilitate comparing the performance of DCOP algorithms, RMASBench, a system that provides a library of state-of-the-art solvers for DCOP and for comparing them, has been created in [Kleiner et al., 2013].

8.2.2. Market and Negotiation-Based Methods

Among the decentralized algorithms, sequential auction- and negotiation-based algorithms [e.g. Sariel-Talay et al., 2009, Nanjanath and Gini, 2010, Ponda et al., 2010, Nunes and Gini, 2015] are more prevalent than other methods. Sequential auction algorithms produce solutions that are two away from optimal in the worst-case in both single-item [Lagoudakis et al., 2004] and multi-item auctions [Choi et al., 2009]. This, together with the ease of implementation and extension to dynamic scenarios and robust execution [Nanjanath and Gini, 2010] makes sequential auctions an attractive solution. However, the greedy nature of sequential auctions and the complex structure of most MRTA/TOC problems cause the addition of temporal constraints to auction algorithms to produce suboptimal solutions [Nunes et al., 2012]. Temporal modeling and balancing between temporal- and distance-based objectives can help auctions perform better [Nunes and Gini, 2015, Ponda et al., 2010]. In [Amador et al., 2014] Fisher markets are used for dynamic ST-MR-SC problems, where tasks can be interrupted for a penalty.

Auctions distribute the computation to individual agents but require communication to share bids and results. To reduce the need for communication, several approaches use consensus algorithms [Zavlanos et al., 2008, Choi et al., 2009, Ponda et al., 2010], where each agent determines independently which tasks it should do. An equilibrium is reached by iteratively sharing information with neighbors and re-allocating tasks if needed. [Godoy and Gini, 2012] extended the Consensus Based Bundle Algorithm (CBBA) [Choi et al., 2009] to optimize the number of completed tasks for tasks with temporal constraints in ST-SR-TA:TW problem with hard constraints. Another method, called emergent task allocation [Atay and Bayazit, 2003], distributes the computation of task allocation for a surveillance task to individual robots, by sharing intentions and directives with 1-hop away neighbors. The method has been shown to converge to the optimal solution as the number of iterations of information sharing increases.

Despite the development of many decentralized methods for MRTA/TOC problems, very limited work offers theoretical analysis of the quality of these solutions. There is a need for theoretical performance bounds for both centralized and decentralized heuristics for the MRTA/TOC problem.

There are other decentralized approaches to task allocation that are not market- or DCOP-based. For instance, [Chapman et al., 2010] formulated ST-MR MRTA as a stochastic game and used overlapping potential games to approximate an optimal solution. Their approach is robust to restricted agent communication and observation range.

Swarm-based approaches have been proposed for various tasks, such as foraging, where robots need to find food and bring it to the nest [Lerman et al., 2006, Brutschy et al., 2014] or where swarms of robots are allocated different monitoring tasks without any communication among the robots [Berman et al., 2009]. Swarm methods often work well but do not have theoretical guarantees.

9. Summary, Open Issues, and Directions for Future Research

Problems that consider temporal and ordering constraints relate to many well studied problems, such as vehicle routing, job-shop scheduling, and multi-robot task allocation. A large portion of the literature in MRTA/TOC focuses on ST-SR-TA:TW problems with hard constraints, some address the soft constraint version of this class of problem; however, the literature is sparser for other classes of problems that consider multi-task robots and multi-robot tasks.

9.1. Summary

We surveyed the multi-robot task allocation literature related to problems where tasks have constraints on where, when, and possibly the order in which they have to be performed. We built on a previous taxonomy and added a classification axis that separates the literature according to the hard versus soft nature of the constraints. Where appropriate, we gave a generic mathematical formulation of problems both in deterministic and stochastic cases, and offered an account of some common execution dynamics. We briefly discussed the methods applied to the problems in our taxonomy, and split solutions into centralized and decentralized approaches. In addition, our work drew parallels between multi-robot task allocation with temporal and ordering constraints with other areas of research, and throughout the paper we discussed models and solutions coming from these areas. Lastly, our work discussed areas that are still sparsely covered.

9.2. Open Issues and Future Research

There are several open issues that need to be addressed, which we did not exhaustively address here. Progress in the following topics would greatly advance research in MRTA/TOC: (1) study of theoretical guarantees of approximate solutions, (2) richer and more complex temporal models with provably good and efficient algorithms, (3) models and algorithms for stochastic MRTA/TOC problems, (4) models and algorithms for task allocation to multi-task robots, (5) studies on the effects of time scales and time sensitivity in MRTA/TOC problems and (6) the development of a research platform to make software and data available to researchers.

Research in stochastic MRTA/TOC problems is still very sparse. The development of MRTA methods that take advantage of simulation and stochastic models to better plan under uncertainty is an endeavor worth pursuing because robots often operate in uncertain environments. Important research questions can be asked here; for example, in an uncertain environment is it more beneficial to build a complex model that incorporates uncertainty, or is it enough to build less well-informed plans and replan as often as needed to quickly react to unexpected events?

There is also a need for work on theoretical guarantees for many heuristic schedulers developed for MRTA/TOC problems. The NP-complete nature of the problem and the need for relatively fast planners has generated many heuristics. However, such heuristics typically lack performance guarantees, which can be crucial for safety critical systems, to ensure that robots work effectively even in the worst possible scenarios.

More work needs to be done to address more complex temporal constraint types, such as disjunctive temporal models. The literature could also benefit from work that combines soft and hard time windows, and precedence with synchronization constraints. A mix of these constraints might produce more expressive models for a larger set of real-world problems.

The challenge of allocating tasks to multi-task robots, which are robots that can perform more than one task at a time, remains open. As we can imagine this is difficult for many existing robots because they lack the necessary actuators. The lack of literature might also be due to the lack of practical applications of multi-task robots. We are not aware of any practical problems that strictly requires robots to perform multiple tasks concurrently; one example of such problems could be in military domains where a drone robot could be required to strike a target while at the same tracking other targets in nearby areas.

Another interesting, yet not so explored topic, regards time scales and sensitivity. Robots that move and navigate in an environment can be on a short time scale, as in exploring a building in hours, or a large time scale of exploring a planet for years. Even for a single robot, tasks can have varied time sensitivities; some tasks may have short hard time constraints, whereas others may have long time horizons with soft time windows. For instance, the Mars rover Curiosity has periodic tasks that occur every day for years (data upload), constantly running tasks (temperature regulation), and sporadic tasks (chemical analysis of collected material and drilling). Each of these tasks have different time sensitivities; for example, data upload needs to occur when the receiving orbiters are within view of the rover. Temperature regulation requires constant vigilance, and drilling can be postponed, but needs to occur when the rover is within reach of the material. Chemical analysis in the rover's internal chambers can occur regardless of location. This single robot has tasks with hard time windows, soft time windows, varied scheduling horizons, and varied sensitivities. Considering tasks in terms of time scales and task sensitivities in the same robotic system thus holds value for any researcher interested in real world problems.

Lastly, we are concerned with the public availability of research data and methods. We advocate for a computational infrastructure for MRTA problems (in general, or in particular problems with temporal and ordering constraints). A tool identical to the Computational Infrastructure for Operations Research COIN-OR [2015]) could greatly benefit MRTA researchers. COIN-OR is an open source software project in which many operations research algorithms are implemented and maintained by scholars in the area. That in combination with datasets would help researchers verify their results on publicly available data and methods, allowing for richer comparisons among methods.

Acknowledgment: Partial support is gratefully acknowledged from NSF grant CNF-1531330 and IIS-1208413.

Alighanbari, M., Kuwata, Y., How, J. P., June 2003. Coordination and control of multiple UAVs with timing constraints and loitering. In: American Control Conf. pp. 5311–5316.

Allahverdi, A., Ng, C., Cheng, T., Kovalyov, M. Y., 2008. A survey of scheduling problems with setup times or costs. European Journal of Operational Research 187 (3), 985 – 1032.

Allen, J. F., Nov. 1983. Maintaining knowledge about temporal intervals. Comm. of the ACM 26 (11), 832–843.

Amador, S., Okamoto, S., Zivan, R., 2014. Dynamic multi-agent task allocation with spatial and temporal constraints. In: Proc. AAAI Conf. on Artificial Intelligence. pp. 1384–1390.

Ando, N., Taniguchi, E., 2006. Travel time reliability in vehicle routing and scheduling with time windows. Networks and Spatial Economics 6 (3-4), 293–311.

Archetti, C., Bouchard, M., Desaulniers, G., 2011. Enhanced branch and price and cut for vehicle routing with split deliveries and time windows. Transportation Science 45 (3), 285–298.

Archetti, C., Speranza, M. G., Vigo, D., 2014. Vehicle routing problems with profits. Vehicle Routing: Problems, Methods, and Applications 18, 273–297.

Atay, N., Bayazit, B., 2003. Emergent task allocation for mobile robots. In: Robotics: Science and Systems (RSS).

Balas, E., Simonetti, N., Vazacopoulos, A., 2008. Job shop scheduling with setup times, deadlines and precedence constraints. Journal of Scheduling 11 (4), 253–262.

Barbulescu, L., Rubinstein, Z. B., Smith, S. F., Zimmerman, T. L., 2010. Distributed coordination of mobile agent teams: the advantage of planning ahead. In: Int'l Conf. on Autonomous Agents and Multi-Agent Systems. pp. 1331–1338.

Bard, J. F., Kontoravdis, G., Yu, G., 2002. A branch-and-cut procedure for the vehicle routing problem with time windows. Transportation Science 36 (2), 250–269.

Barnhart, C., Hane, C. A., Vance, P. H., 2000. Using branch-and-price-and-cut to solve origin-destination integer multicommodity flow problems. Operations Research 48 (2), 318–326.

Barnhart, C., Johnson, E. L., Nemhauser, G. L., Savelsbergh, M. W. P., Vance, P. H., 1998. Branch-and-price: Column generation for solving huge integer programs. Operations Research 46 (3), 316–329.

Beccari, G., Caselli, S., Reggiani, M., Zanichelli, F., 1999. Rate modulation of soft real-time tasks in autonomous robot control systems. In: Proc. 11th Euromicro Conference on Real-Time Systems. IEEE, pp. 21–28.

Beck, L. C., Refalo, P., 2003. A hybrid approach to scheduling with earliness and tardiness costs. Annals of Operations Research 118 (1–4), 49–71.

Berkelaar, M., Eikland, K., Notebaert, P., 2004. lp_solve 5.5, open source (mixed-integer) linear programming system.
URL `http://lpsolve.sourceforge.net/5.5/`

Berman, S., Halasz, A., Hsieh, M., Kumar, V., 2009. Optimized stochastic policies for task allocation in swarms of robots. IEEE Trans. on Robotics 25, 927–937.

Bernat, G., Burns, A., Liamosi, A., 2001. Weakly hard real-time systems. IEEE Trans. on Computers 50 (4), 308–321.

Bettinelli, A., Ceselli, A., Righini, G., 2011. A branch-and-cut-and-price algorithm for the multi-depot heterogeneous vehicle routing problem with time windows. Transportation Research Part C: Emerging Technologies 19 (5), 723–740.

Beynier, A., Mouaddib, A.-I., 2007. Decentralized Markov Decision Processes for handling temporal and resource constraints in a multiple robot system. In: Alami, R., Chatila, R., Asama, H. (Eds.), Distributed Autonomous Robotic Systems 6. Springer Japan, pp. 191–200.

Birattari, M., 2009. Tuning Metaheuristics: A Machine Learning Perspective. Springer.

Bistarelli, S., Pini, M., Rossi, F., Venable, K., 2007. Bipolar preference problems: Framework, properties and solving techniques. In: Azevedo, F., Barahona, P., Fages, F., Rossi, F. (Eds.), Recent Advances in Constraints. Vol. 4651 of Lecture Notes in Computer Science. Springer, pp. 78–92.

Block, S. A., Wehowsky, A. F., Williams, B. C., 2006. Robust execution on contingent, temporally flexible plans. In: Proc. AAAI Conf. on Artificial Intelligence. pp. 802–808.

Boerkoel, J. C., Durfee, E. H., 2012. A distributed approach to summarizing spaces of multiagent schedules. In: Proc. AAAI Conf. on Artificial Intelligence. pp. 1742–1748.

Boerkoel, J. C., Durfee, E. H., 2013. Decoupling the multiagent disjunctive temporal problem. In: Int'l Conf. on Autonomous Agents and Multi-Agent Systems. pp. 1145–1146.

Boerkoel, J. C., Planken, L. R., 2012. Distributed algorithms for incrementally maintaining multiagent simple temporal networks. In: Int'l Conf. on Autonomous Agents and Multi-Agent Systems. pp. 216–235.

Bopardikar, S., Smith, S., Bullo, F., 2014. On dynamic vehicle routing with time constraints. IEEE Trans. on Robotics 30 (6), 1524–1532.

Borning, A., Freeman-Benson, B., Wilson, M., 1992. Constraint hierarchies. Lisp Symb. Comput. 5 (3), 223–270.

Bouros, P., Sacharidis, D., Dalamagas, T., Sellis, T., 2011. Dynamic pickup and delivery with transfers. In: Proc. 12th Int'l Conf. on Advances in Spatial and Temporal Databases (SSTD'11). pp. 112–129.

Bräysy, O., Gendreau, M., Feb. 2005a. Vehicle routing problem with time windows, part I: Route construction and local search algorithms. Transportation Science 39 (1), 104–118.

Bräysy, O., Gendreau, M., Feb. 2005b. Vehicle routing problem with time windows, part II: Metaheuristics. Transportation Science 39 (1), 119–139.

Bredström, D., Rönnqvist, M., 2008. Combined vehicle routing and scheduling with temporal precedence and synchronization constraints. European Journal of Operational Research 191 (1), 19–31.

Brutschy, A., Pini, G., Pinciroli, C., anf Marco Dorigo, M. B., 2014. Self-organized task allocation to sequentially interdependent tasks in swarm robotics. Journal of Autonomous Agents and Multi-agent Systems 28 (1), 101–125.

Caccamo, M., Buttazzo, G., 1997. Exploiting skips in periodic tasks for enhancing aperiodic responsiveness. In: Proc. IEEE Real-Time Systems Symposium. pp. 330–339.

Carrabs F., Cordeau J.-F., L. G., 2007. Variable neighborhood search for the pickup and delivery traveling salesman problem with LIFO loading. INFORMS Journal on Computing 19 (4), 618–632.

Cesta, A., Oddi, A., May 1996. Gaining efficiency and flexibility in the simple temporal problem. In: 3rd Int'l Wprkshop on Temporal Representation and Reasoning. pp. 45–50.

Cesta, A., Oddi, A., Smith, S., 1999. An iterative sampling procedure for resource constrained project scheduling with time windows. In: Proc. Int'l Joint Conf. on Artificial intelligence. pp. 1022–1029.

Cesta, A., Oddi, A., Smith, S., 2000. A constraint-based method for project scheduling with time windows. Journal of Heuristics 8 (1), 109–136.

Chalkiadakis, G., Elkind, E., Markakis, E., Polukarov, M., Jennings, N. R., 2010. Cooperative games with overlapping coalitions. Journal of Artificial Intelligence Research 39 (1), 179–216.

Chao, I.-M., Golden, B. L., Wasil, E. A., 1996. The team orienteering problem. European Journal of Operational Research 88 (3), 464–474.

Chapman, A. C., Micillo, R. A., Kota, R., Jennings, N. R., 2010. Decentralized dynamic task allocation using overlapping potential games. The Computer Journal 53 (9), 1462–1477.

Cheng, P., Keller, J., Kumar, V., Sept 2008. Time-optimal UAV trajectory planning for 3D urban structure coverage. In: Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems. pp. 2750–2757.

Choi, H.-L., Brunet, L., How, J., Aug. 2009. Consensus-based decentralized auctions for robust task allocation. IEEE Trans. on Robotics 25 (4), 912–926.

Chopra, S., Egerstedt, M. B., 2012. Multi-robot routing for servicing spatio-temporal requests: A musically inspired problem. In: IFAC Conf. on Analysis and Design of Hybrid Systems. pp. 319–324.

Clausen, J., 1997. Branch and bound algorithms-principles and examples. Parallel Computing in Optimization, 239–267.

COIN-OR, 2015. COmputational INfrastructure for Operations Research. URL http://www.coin-or.org

Coles, A., Coles, A., Fox, M., Long, D., 2009. Incremental constraint-posting algorithms in interleaved planning and scheduling. In: Proc. Workshop on Constraint Satisfaction Techniques for Planning and Scheduling (COPLAS) at ICAPS. pp. 1–8.

Collins, J., Bilot, C., Gini, M., Mobasher, B., Jun. 2000. Mixed-initiative decision support in agent-based automated contracting. In: Proc. Int'l Conf, on Autonomous Agents. pp. 247–254.

Collins, J., Gini, M., 2006. Magnet: A multi-agent system using auctions with temporal and precedence constraints. In: Chaib-draa, B., Müller, J. (Eds.), Multiagent based Supply Chain Management. Vol. 28. Springer, pp. 273–314.

Colorni, A., Righini, G., 2001. Modeling and optimizing dynamic dial-a-ride problems. International Transactions in Operational Research 8 (2), 155–166.

Coltin, B., Veloso, M., 2014a. Online pickup and delivery planning with transfers for mobile robots. In: Proc. IEEE Int'l Conf. on Robotics and Automation. pp. 5786–5791.

Coltin, B., Veloso, M., 2014b. Optimizing for transfers in a multi-vehicle collection and delivery problem. In: Distributed Autonomous Robotic Systems. Vol. 104. pp. 91–103.

Coltin, B., Veloso, M. M., 2014c. Scheduling for transfers in pickup and delivery problems with very large neighborhood search. In: AAAI. pp. 2250–2256.

Cordeau, J.-F., Laporte, G., 2007. The dial-a-ride problem: models and algorithms. Annals of Operations Research 153 (1), 29–46.

Dahl, T. S., Matarić, M., Sukhatme, G. S., 2009. Multi-robot task allocation through vacancy chain scheduling. Robotics and Autonomous Systems 57 (6-7), 674–687.

Dantzig, G. B., Ramser, J. H., 1959. The truck dispatching problem. Management Science 6 (1), 80–91.

Davis, R. I., Burns, A., Oct. 2011. A survey of hard real-time scheduling for multiprocessor systems. ACM Comput. Surv. 43 (4), 35:1–35:44.

Dean, T., Kaelbling, L. P., Kirman, J., Nicholson, A., 1993. Planning with deadlines in stochastic domains. In: Proc. AAAI Conf. on Artificial Intelligence. pp. 574–579.

Dechter, R., Meiri, I., Pearl, J., 1991. Temporal constraint networks. Artificial Intelligence 49 (1-3), 61–95.

Desaulniers, G., 2010. Branch-and-price-and-cut for the split-delivery vehicle routing problem with time windows. Operations research 58 (1), 179–192.

Desrochers, M., Lenstra, J. K., Savelsbergh, M. W., Soumis, F., 1988. Vehicle routing with time windows: Optimization and approximation. Vehicle Routing: Methods and Studies 16, 65–84.

Di, B., Wang, T., Song, L., Han, Z., 2013. Incentive mechanism for collaborative smartphone sensing using overlapping coalition formation games. In: Porc. IEEE Global Communications Conf. (GLOBECOM). pp. 1705–1710.

Dohn, A., Kolind, E., Clausen, J., 2009. The manpower allocation problem with time windows and job-teaming constraints: A branch-and-price approach. Computers and Operations Research 36, 1145–1157.

Dolgov, D. A., James, M. R., Samples, M. E., May 2007. Combinatorial resource scheduling for multiagent MDPs. In: Int'l Conf. on Autonomous Agents and Multi-Agent Systems. pp. 657–664.

Domshlak, C., Prestwich, S., Rossi, F., Venable, K., Walsh, T., 2006. Hard and soft constraints for reasoning about qualitative conditional preferences. Journal of Heuristics 12 (4-5), 263–285.

Dondo, R., Cerdá, J., 2007. A cluster-based optimization approach for the multi-depot heterogeneous fleet vehicle routing problem with time windows. European Journal of Operational Research 176 (3), 1478–1507.

Fakcharoenphol, J., Harrelson, C., Rao, S., 2007. The K-traveling repairmen problem. ACM Trans. Algorithms 3 (4), Article No. 40.

Fang, C., Yu, P., Williams, B. C., 2014. Chance-constrained probabilistic simple temporal problems. In: Proc. AAAI Conf. on Artificial Intelligence. pp. 2264–2270.

Farinelli, A., Iocchi, L., Nardi, D., Ziparo, V., 2006. Assignment of dynamically perceived tasks by token passing in multirobot systems. Proceedings of the IEEE 94 (7), 1271–1288.

Farinelli, A., Rogers, A., Jennings, N., 2014. Agent-based decentralised coordination for sensor networks using the max-sum algorithm. Journal of Autonomous Agents and Multi-agent Systems 28 (3), 337–380.

Feillet, D., 2010. A tutorial on column generation and branch-and-price for vehicle routing problems. 4OR 8 (4), 407–424.

Ferreira, P., Boffo, F., Bazzan, A., 2008. Using Swarm-GAP for distributed task allocation in complex scenarios. In: Jamali, N., Scerri, P., Sugawara, T. (Eds.), Massively Multi-Agent Technology. Vol. 5043 of Lecture Notes in Computer Science. Springer Berlin / Heidelberg, pp. 107–121.

Gendreau, M., Laporte, G., Séguin, R., 1996. Stochastic vehicle routing. European Journal of Operational Research 88 (1), 3–12.

Gerevini, A., Long, D., 2006. Plan constraints and preferences in PDDL3 - the language of the deterministic part of the fifth international planning competition. In: ICAPS. pp. 11–17.

Gerkey, B. P., Matarić, M. J., 2004. A formal analysis and taxonomy of task allocation in multi-robot systems. The International Journal of Robotics Research 23 (9), 939–954.

Godoy, J., Gini, M., 2012. Task allocation for spatially and temporally distributed tasks. In: Proc. of the Int'l Conf. on Intelligent Autonomous Systems. pp. 603–612.

Gombolay, M., Wilcox, R., Shah, J., 2013. Fast scheduling of multi-robot teams with temporospatial constraints. In: Robotics: Science and Systems (RSS). Berlin, Germany, pp. 49–56.

Graham, R. L., Lawler, E. L., Lenstra, J. K., Kan, A. R., 1979. Optimization and approximation in deterministic sequencing and scheduling: a survey. Annals of Discrete Mathematics 5, 287–326.

Groth, C., Henrich, D., 2014. Single-shot learning and scheduled execution of behaviors for a robotic manipulator. In: Proc. 41st Int'l Symposium on Robotics (ISR). pp. 1–6.

Gurobi Optimization, I., 2014. Gurobi optimizer reference manual.
URL `http://www.gurobi.com`

Hasgül, S., Saricicek, I., Ozkan, M., Parlaktuna, O., 2009. Project-oriented task scheduling for mobile robot team. Journal of Intelligent Manufacturing 20 (2), 151–158.

Heap, B., Pagnucco, M., 2014. Minimising undesired task costs in multi-robot task allocation problems with in-schedule dependencies. In: Proc. AAAI Conf. on Artificial Intelligence. pp. 2542–2548.

Heger, F. W., Hiatt, L. M., Sellner, B., Simmons, R., Singh, S., 2005. Results in sliding autonomy for multi-robot spatial assembly. In: Proc. i-SAIRAS. pp. 448–455.

Hoogendoorn, M., Gini, M., 2008. Agents preferences in decentralized task allocation. In: European Conference on Artificial Intelligence. pp. 398–402.

Hsu, C.-I., Hung, S.-F., Li, H.-C., 2007. Vehicle routing problem with time-windows for perishable food delivery. Journal of Food Engineering 80 (2), 465–475.

Hu, Q., Lim, A., 2014. An iterative three-component heuristic for the team orienteering problem with time windows. European Journal of Operational Research 232 (2), 276–286.

Hunsberger, L., 2002. Algorithms for a temporal decoupling problem in multi-agent planning. In: Proc. AAAI Conf. on Artificial Intelligence. pp. 468–475.

ILOG, I., 2006. ILOG CPLEX: High-performance software for mathematical programming and optimization.
URL `http://www.ilog.com/products/cplex`

Jackson, J., Faied, M., Kabamba, P., Girard, A., 2013. Distributed constrained minimum-time schedules in networks of arbitrary topology. IEEE Trans. on Robotics 29 (2), 554–563.

Jones, E. G., Dias, M. B., Stentz, A., 2011. Time-extended multi-robot coordination for domains with intra-path constraints. Autonomous Robots 30 (1), 41–56.

Jozefowiez, N., Semet, F., Talbi, E.-G., 2008. Multi-objective vehicle routing problems. European Journal of Operational Research 189 (2), 293–309.

Jünger, M., Thienel, S., 2000. The ABACUS system for branch-and-cut-and-price algorithms in integer programming and combinatorial optimization. Software: Practice and Experience 30 (11), 1325–1352.

Junges, R., Bazzan, A. L. C., 2008. Evaluating the performance of DCOP algorithms in a real world, dynamic problem. In: Int'l Conf. on Autonomous Agents and Multi-Agent Systems. pp. 599–606.

Kang, K. H., Lee, Y. H., Lee, B. K., 2005. An exact algorithm for multi depot and multi period vehicle scheduling problem. In: Computational Science and Its Applications–ICCSA 2005. Springer, pp. 350–359.

Kitano, H., Satoshi, T., 2001. Robocup rescue : A grand challenge for multiagent and intelligent systems. AI Magazine 22 (1), 39–52.

Kleiner, A., Farinelli, A., Ramchurn, S., Shi, B., Maffioletti, F., Reffato, R., 2013. RMASBench: Benchmarking dynamic multi-agent coordination in urban search and rescue. In: Int'l Conf. on Autonomous Agents and Multi-Agent Systems. pp. 1195–1196.

Koenig, S., Keskinocak, P., Tovey, C. A., 2010. Progress on agent coordination with cooperative auctions. In: Proc. AAAI Conf. on Artificial Intelligence. pp. 1713–1717.

Koes, M., Nourbakhsh, I. R., Sycara, K. P., 2005. Heterogeneous multirobot coordination with spatial and temporal constraints. In: Proc. AAAI Conf. on Artificial Intelligence. pp. 1292–1297.

Kolen, A. W. J., Kan, A. H. G. R., Trienekens, H. W. J. M., 1987. Vehicle routing with time windows. Operations Research 35 (2), 266–273.

Korsah, G. A., 2011. Exploring bounded optimal coordination for heterogeneous teams with cross-schedule dependencies. Ph.D. thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA.

Korsah, G. A., Kannan, B., Browning, B., Dias, M. B., 2012. xBots: An approach to generating and executing optimal multi-robot plans with constraints. In: Proc. IEEE Int'l Conf. on Robotics and Automation. pp. 115–122.

Korsah, G. A., Stentz, A., Dias, M. B., 2013. A comprehensive taxonomy for multi-robot task allocation. The International Journal of Robotics Research 32 (12), 1495–1512.

Labadie, N., Mansini, R., Melechovskỳ, J., Calvo, R. W., 2012. The team orienteering problem with time windows: An LP-based granular variable neighborhood search. European Journal of Operational Research 220 (1), 15–27.

Lagoudakis, M., Keskinocak, P., Kleywegt, A., Koenig, S., 2004. Auctions with performance guarantees for multi-robot task allocation. In: Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems. pp. 1957–1962.

Lagoudakis, M. G., Markakis, E., Kempe, D., Keskinocak, P., Kleywegt, A., Koenig, S., Tovey, C., Meyerson, A., Jain, S., 2005. Auction-based multirobot routing. In: Robotics: Science and Systems (RSS). pp. 343–350.

Landén, D., Heintz, F., Doherty, P., 2012. Complex task allocation in mixed-initiative delegation: A UAV case study. In: Desai, N., Liu, A., Winikoff, M. (Eds.), Principles and Practice of Multi-Agent Systems: 13th Int'l Conf., PRIMA 2010. Lecture Notes in Computer Science 7057. Springer, pp. 288–303.

Laporte, G., Louveaux, F., Mercure, H., 1992. The vehicle routing problem with stochastic travel times. Transportation Science 26 (3), 161–170.

Lau, H. C., Sim, M., Teo, K. M., 2003. Vehicle routing problem with time windows and a limited number of vehicles. European Journal of Operational Research 148 (3), 559–569.

Lee, K., Choi, B.-C., Leung, J. Y. T., Pinedo, M. L., Jul. 2009. Approximation algorithms for multi-agent scheduling to minimize total weighted completion time. Inf. Process. Lett. 109 (16), 913–917.

Lerman, K., Jones, C., Galstyan, A., Matari, M. J., 2006. Analysis of dynamic task allocation in multi-robot systems. The International Journal of Robotics Research 25 (3), 225–241.

Lombardi, M., Milano, M., 2012. Optimal methods for resource allocation and scheduling: a cross-disciplinary survey. Constraints 17 (1), 51–85.

Luo, L., August 2014. Distributed algorithm design for constrained multi-robot task assignment. Ph.D. thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA.

Luo, L., Chakraborty, N., Sycara, K., July 2015. Distributed algorithms for multirobot task assignment with task deadline constraints. Automation Science and Engineering, IEEE Transactions on 12 (3), 876–888.

Luo, Y., Schonfeld, P., 2007. A rejected-reinsertion heuristic for the static dial-a-ride problem. Transportation Research Part B: Methodological 41 (7), 736–755.

Macarthur, K. S., Stranders, R., Ramchurn, S. D., Jennings, N. R., 2011. A distributed anytime algorithm for dynamic task allocation in multi-agent systems. In: Proc. AAAI Conf. on Artificial Intelligence. pp. 701–706.

MacKenzie, D. C., 2003. Collaborative tasking of tightly constrained multi-robot missions. In: Multi-Robot Systems: From Swarms to Intelligent Automata, Volume II. Vol. 2. pp. 39–50.

Maheswaran, R. T., Tambe, M., Bowring, E., Pearce, J. P., Varakantham, P., 2004. Taking DCOP to the real world: Efficient complete solutions for distributed multi-event scheduling. In: Int'l Conf. on Autonomous Agents and Multi-Agent Systems. pp. 310–317.

Manne, A. S., 1960. On the job-shop scheduling problem. Operations Research 8 (2), 219–223.

McIntire, M., Nunes, E., Gini, M., 2016. Iterated multi-robot auctions for precedence-constrained task scheduling. In: Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems. AAMAS '16. pp. 1078–1086.

Melvin, J., Keskinocak, P., Koenig, S., Tovey, C. A., Ozkaya, B. Y., 2007. Multi-robot routing with rewards and disjoint time windows. In: Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems. pp. 2332–2337.

Mercker, T., Casbeer, D., Millet, P., Akella, M., 2010. An extension of consensus-based auction algorithms for decentralized, time-constrained task assignment. In: American Control Conf. pp. 6324 –6329.

Miao, X., Luh, P., Kleimnman, D., Castanon, D., 1991. Distributed stochastic resource allocation in teams. IEEE Trans. on Systems, Man, and Cybernetics 21 (1), 61–70.

Mosteo, A. R., Montano, L., 2010. A survey of multi-robot task allocation. Instituto de Investigacin en Ingenierła de Aragn (I3A), Tech. Rep.

Nanjanath, M., Gini, M., 2010. Repeated auctions for robust task execution by a robot team. Robotics and Autonomous Systems 58 (7), 900–909.

Nunes, E., Gini, M., 2015. Multi-robot auctions for allocation of tasks with temporal constraints. In: Proc. AAAI Conf. on Artificial Intelligence. pp. 2110–2116.

Nunes, E., Nanjanath, M., Gini, M., 2012. Auctioning robotic tasks with overlapping time windows. In: Int'l Conf. on Autonomous Agents and Multi-Agent Systems. pp. 1211–1212.

Oddi, A., Rasconi, R., Cesta, A., Smith, S. F., 2011. Solving job shop scheduling with setup times through constraint-based iterative sampling: an experimental analysis. Annals of Mathematics and Artificial Intelligence 62 (3-4), 371–402.

Olawsky, D., Gini, M., 1990. Deferred planning and sensor use. In: Innovative Approaches to Planning, Scheduling and Control: Proc. 1990 DARPA Workshop. M. Kaufmann, San Mateo, Ca, pp. 166–174.

Parker, J., Nunes, E., Godoy, J., Gini, M., 2015. Exploiting spatial locality and heterogeneity of agents for search and +rescue teamwork. Journal of Field Robotics, n/a–n/a.
URL http://dx.doi.org/10.1002/rob.21601

Parker, L. E., Tang, F., 2006. Building multi-robot coalitions through automated task solution synthesis. Proceedings of the IEEE 94 (7), 1289–1305.

Pavone, M., Bisnik, N., Frazzoli, E., Isler, V., 2009. A stochastic and dynamic vehicle routing problem with time windows and customer impatience. Mobile Networks and Applications 14 (3), 350–364.

Pavone, M., Frazzoli, E., Bullo, F., 2011. Adaptive and distributed algorithms for vehicle routing in a stochastic and dynamic environment. IEEE Trans. on Automatic Control 56 (6), 1259–1274.

Pecora, F., Cirillo, M., 2012. A constraint-based approach for multiple nonholonomic vehicle coordination in industrial scenarios. In: ICAPS 2012 Workshop on Combining Task and Motion Planning for Real-World Applications. pp. 45–52.

Planken, L. R., de Weerdt, M. M., van der Krogt, R. P., Sep 2008. P3C: A new algorithm for the simple temporal problem. In: Proc. Int'l Conf. on Automated Planning and Scheduling. pp. 256–263.

Polacek, M., Hartl, R. F., Doerner, K., Reimann, M., 2004. A variable neighborhood search for the multi depot vehicle routing problem with time windows. Journal of Heuristics 10 (6), 613–627.

Ponda, S., Johnson, L., Kopeikin, A., Choi, H.-L., How, J., Jun. 2012a. Distributed planning strategies to ensure network connectivity for dynamic heterogeneous teams. IEEE Journal on Selected Areas in Communications 30 (5), 861–869.

Ponda, S. S., Johnson, L. B., How, J. P., June 2012b. Distributed chance-constrained task allocation for autonomous multi-agent teams. In: American Control Conf. pp. 4528–4533.

Ponda, S. S., Redding, J., Choi, H.-L., How, J., Vavrina, M., Vian, J., Jul. 2010. Decentralized planning for complex missions with dynamic communication constraints. In: American Control Conf. pp. 3998–4003.

Pujol-Gonzalez, M., Cerquides, J., Farinelli, A., Meseguer, P., Rodriguez-Aguilar, J. A., 2015. Efficient inter-team task allocation in RoboCup Rescue. In: Int'l Conf. on Autonomous Agents and Multi-Agent Systems. pp. 413–421.

Ramchurn, S., Farinelli, A., Macarthur, K., Polukarov, M., Jennings, N., 2010a. Decentralised coordination in RoboCup Rescue. The Computer Journal 53 (9), 1–15.

Ramchurn, S. D., Polukarov, M., Farinelli, A., Jennings, N., Trong, C., March 2010b. Coalition formation with spatial and temporal constraints. In: Int'l Conf. on Autonomous Agents and Multi-Agent Systems. pp. 1181–1188.

Ropke, S., Cordeau, J.-F., 2009. Branch and cut and price for the pickup and delivery problem with time windows. Transportation Science 43 (3), 267–286.

Ropke, S., Cordeau, J.-F., Laporte, G., 2007. Models and branch-and-cut algorithms for pickup and delivery problems with time windows. Networks 49 (4), 258–272.

Rubinstein, Z., Smith, S., Barbulescu, L., 2012. Incremental management of oversubscribed vehicle schedules in dynamic dial-a-ride problems. In: Proc. AAAI Conf. on Artificial Intelligence. pp. 1809–1815.

Sandholm, T., Larson, K., Andersson, M., Shehory, O., Tohm, F., 1999. Coalition structure generation with worst case guarantees. Artificial Intelligence 111 (12), 209–238.

Sariel, S., Balch, T., 2006. Dynamic and distributed allocation of resource constrained project tasks to robots. In: Multi-Agent Robotic Systems Workshop at 3rd Int'l Conf. on Informatics in Control, Automation and Robotic. pp. 34–43.

Sariel-Talay, S., Balch, T., Erdogan, N., 2009. Multiple traveling robot problem: A solution based on dynamic task selection and robust execution. IEEE/ASME Trans. on Mechatronics 14 (2), 198–206.

Savelsbergh, M. W., Sol, M., 1995. The general pickup and delivery problem. Transportation science 29 (1), 17–29.

Scerri, P., Farinelli, A., Okamoto, S., Tambe, M., 2005. Allocating tasks in extreme teams. In: Int'l Conf. on Autonomous Agents and Multi-Agent Systems. pp. 727–734.

Schneider, J., Apfelbaum, D., Bagnell, D., Simmons, R., 2005. Learning opportunity costs in multi-robot market based planners. In: Proc. IEEE Int'l Conf. on Robotics and Automation. pp. 1151–1156.

Segui-Gasco, P., Shin, H.-S., Tsourdos, A., Seguí, V., 2015. Decentralised submodular multi-robot task allocation. In: Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on. pp. 2829–2834.

Shah, J. A., Conrad, P. R., Williams, B. C., 2009. Fast distributed multi-agent plan execution with dynamic task assignment and scheduling. In: Proc. Int'l Conf. on Automated Planning and Scheduling. pp. 289–296.

Shehory, O., Kraus, S., 1998. Methods for task allocation via agent coalition formation. Artificial Intelligence 101 (1–2), 165–200.

Shen, Z., Ordónez, F., Dessouky, M. M., 2009. The stochastic vehicle routing problem for minimum unmet demand. In: Optimization and logistics challenges in the enterprise. Springer, pp. 349–371.

Smith, S. L., Bullo, F., 2007. Target assignment for robotic networks: Asymptotic performance under limited communication. In: American Control Conf. IEEE, pp. 1155–1160.

Solomon, M. M., Desrosiers, J., 1988. Survey paper – time window constrained routing and scheduling problems. Transportation Science 22 (1), 1–13.

Stergiou, K., Koubarakis, M., 2000. Backtracking algorithms for disjunctions of temporal constraints. Artificial Intelligence 120 (1), 81–117.

Su, X., Zhang, M., Bai, Q., Nov. 2016. Coordination for dynamic weighted task allocation in disaster environments with time, space and communication constraints. Journal of Parallel and Distributed Computing 97, 47–56.

Taş, D., Dellaert, N., Van Woensel, T., De Kok, T., 2013. Vehicle routing problem with stochastic travel times including soft time windows and service costs. Comput. Oper. Res. 40 (1), 214–224.

Taillard, E., Badeau, P., Gendreau, M., Guertin, F., Potvin, J. Y., 1997. A tabu search heuristic for the vehicle routing problem with soft time windows. Transportation Science 31 (2), 170–186.

Tang, F., Parker, L. E., 2007. A complete methodology for generating multi-robot task solutions using ASyMTRe-D and market-based task allocation. In: Proc. IEEE Int'l Conf. on Robotics and Automation. pp. 3351–3358.

Toth, P., Vigo, D. (Eds.), 2002. The vehicle routing problem. SIAM Monographs on Discrete Mathematics and Applications, Philadelphia, PA.

Tsamardinos, I., 2002. A probabilistic approach to robust execution of temporal plans with uncertainty. In: Methods and Applications of Artificial Intelligence: Proc. 2nd Hellenic Conference on AI (SETN '02). Lecture Notes in Computer Science. Springer-Verlag, pp. 97–108.

Usug, U. C., Sariel-Talay, S., 2011. Dynamic temporal planning for multirobot systems. In: Workshop on Automated Action Planning for Autonomous Mobile Robots at AAAI. pp. 64–69.

Vidal, T., 1999. Handling contingency in temporal constraint networks: from consistency to controllabilities. Journal of Experimental & Theoretical Artificial Intelligence 11 (1), 23–45.

Vidal, T., Crainic, T., Gendreau, M., Prins, C., 2013. Heuristics for multi-attribute vehicle routing problems: A survey and synthesis. European Journal of Operational Research 231 (1), 1–21.

Vidal, T., Crainic, T. G., Gendreau, M., Prins, C., 2014. Time-window relaxations in vehicle routing heuristics. Journal of Heuristics 21 (3), 329–358.

Vig, L., Adams, J. A., 2006. Multi-robot coalition formation. IEEE Trans. on Robotics 22 (4), 637–649.

Wu, F., Jennings, N., 2014. Regret-based multi-agent coordination with uncertain task rewards. In: Proc. AAAI Conf. on Artificial Intelligence. pp. 1492–1499.

Xu, L., Choueiry, B. Y., 2003. A new efficient algorithm for solving the simple temporal problem. In: Proc. of Int'l Symposium on Temporal Representation and Reasoning. pp. 212–222.

Xu, Y., Scerri, P., Yu, B., Okamoto, S., Lewis, M., Sycara, K., 2005. An integrated token-based algorithm for scalable coordination. In: Int'l Conf. on Autonomous Agents and Multi-Agent Systems. pp. 407–414.

Zavlanos, M. M., Spesivtsev, L., Pappas, G. J., Dec. 2008. A distributed auction algorithm for the assignment problem. In: IEEE Proc. Decision and Control Conf. pp. 1212–1217.

Zhang, Y., Parker, L. E., 2013a. IQ-ASyMTRe: Forming executable coalitions for tightly coupled multirobot tasks. IEEE Trans. on Robotics 29, 1–17.

Zhang, Y., Parker, L. E., 2013b. Multi-robot task scheduling. In: Proc. IEEE Int'l Conf. on Robotics and Automation. pp. 2992–2998.

Zheng, T., Woodside, M., 2003. Heuristic optimization of scheduling and allocation for distributed systems with soft deadlines. In: Computer Performance Evaluation. Modelling Techniques and Tools. Springer, pp. 169–181.