DMRR: Dynamic Multi-Robot Routing for Evolving Missions

Mihai-Ioan Popescu, Olivier Simonin, Gabriela Czibula, Anne Spalanzani, Fabrice Valois

DMRR: Dynamic Multi-Robot Routing for Evolving Missions

Mihai-Ioan Popescu, Olivier Simonin, Fabrice Valois
INSA Lyon, CITI-INRIA Laboratory
University of Lyon
F-69621 Villeurbanne, France
{mihai-ioan.popescu, olivier.simonin, fabrice.valois}@insa-lyon.fr

Gabriela Czibula
Department of Computer Science
Babeş-Bolyai University
Cluj-Napoca, Romania
gabis@cs.ubbcluj.ro

Anne Spalanzani
INRIA Rhône-Alpes
University of Grenoble
Grenoble, France
anne.spalanzani@inria.fr

Abstract

The paper proposes Dynamic Multi Robot-Routing (DMRR), as a continuous adaptation of the multi-robot target allocation process (MRTA) to new discovered targets. There are few works addressing dynamic target allocation. Existing methods are lacking the continuous integration of new targets, handling its progressive effects, but also lacking dynamicity support (e.g. parallel allocations, participation of new robots). The present paper proposes a framework for dynamically adapting the existing robot missions to new discovered targets. Missions accumulate targets continuously, so the case of a saturation bound for the mission costs is also considered. Dynamic saturation-based auctioning (DSAT) is proposed for allocating targets, providing lower time complexities (due to parallelism in allocation). Comparison is made with algorithms ranging from greedy to auction-based methods with provable sub-optimality. The algorithms are tested on exhaustive sets of inputs, with random configurations of targets (for DMRR with and without a mission saturation bound). The results for DSAT show that it outperforms state-of-the-art methods, like standard sequential single-item auctioning (SSI) or SSI with regret clearing.

1 Introduction

As robot missions have become reality, a growing number of applications requires teams of mobile robots to autonomously accomplish missions incorporating task groups. *Multi-robot Task Allocation* (MRTA) has been widely studied in collaborative multi-robot planning, and generally for multi-agent coordination. Key applications include search and rescue operations (Wei, Hindriks, and Jonker 2016; Beck et al. 2016), multi-robot exploration (Tovey et al. 2005) or patrolling (Pippin, Christensen, and Weiss 2013). As underlined by an extensive MRTA survey of 2015 (Khamis, Hussein, and Elmogy 2015), despite the large number of papers in the task allocation domain, there is only few work concerning subjects such as dynamic task allocation or allocation of complex tasks in multi-robot systems.

In practice, targets are found progressively and robots carry missions that evolve in time. Therefore, a recent challenging problem is how to dynamically and efficiently adapt the robots and their work to new targets, in order to let robots continue or extend their missions (while keeping the envisaged objectives).

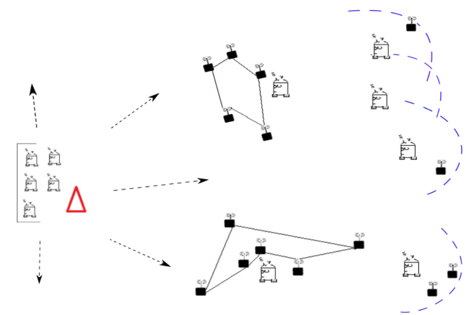

Figure 1: Robots deployed for executing missions on targets and for progressive target discovery. Targets are represented by the black dots and missions by the cyclic paths. Robots move on these paths between their allocated targets.

Multi-robot routing (MRR) (Lagoudakis et al. 2005) has often been the main testbed for MRTA scenarios, being at base part of the location routing problems (Toth and Vigo 2014). The MRR problem assumes a set of static targets and robots with known locations. The goal is then to assign the targets to the robots while optimizing an objective function. Under typically considered objectives, MRR is a difficult combinatorial optimization problem – NP-hard (Lagoudakis et al. 2005) – therefore heuristics have been studied in order to provide solutions, though not optimal, but in feasible time. For the rest of the paper, as in MRTA literature, target allocation and task allocation are used interchangeably.

Solving MRR problems has been addressed using market-based approaches like combinatorial, parallel or sequential auctions (Koenig et al. 2006; Koenig, Keskinocak, and Tovey 2010), but also through other optimization based techniques, mainly TSP-based optimizations (Mosteo, Montano, and Lagoudakis 2009; 2008). Lately, approaches were proposed for combining auctions with methods such as clustering (Heap and Pagnucco 2011; Elango, Nachiappan, and Tiwari 2011) or matroid theory techniques (Williams, Gasparri, and Ulivi 2017). Aspects of dynamic task allocation have been envisaged by few works. These include the consideration of dynamic clusters in auctions (Heap and Pagnucco 2012), the re-auctioning of uninitiated tasks (Nanjanath and Gini 2010) (both presuming delays, e.g. caused by communication loss or dynamic obstacles), or more recently, prediction methods for search and rescue operations (Wei, Hindriks, and Jonker 2016). However, these approaches are lacking support for dynamic target allocation (like the use of parallel allocations). The algorithms require numerous allocation rounds to finish, trading off time complexities for solution quality. This does not cope with dynamic allocation needs, like repeated algorithm execution. Also, existing methods do not handle effects of continuous growths of the sets of targets (e.g. a robot may have its missions saturated, due to resource constraints).

This paper proposes *Dynamic Multi-Robot Routing* (DMRR) as an incremental dynamic adaptation of the MRR process to new targets. DMRR assumes that targets are continuously discovered by a group of exploration robots, and then integrated in the missions of the working robots (see Figure 1). Hence, robot missions continuously grow in size. No prior information is considered regarding target locations, since the environment is presumed to be dynamic, with a high degree of uncertainty.

After presenting the background and related work (Section 2), a framework is proposed around DMRR, which formalizes the problem, contains a mathematical model and complexity proofs (Section 3). The MRR objectives are redefined for the dynamic case, while asymptotic objectives and target coverage maximization are envisaged as well (DMRR remains NP-hard, under the MRR objectives). Because of the continuous mission growth, missions may get saturated, so resource limitation is taken into account in a version of DMRR, called DMRR-Sat. This problem assumes a so-called mission saturation bound (i.e. a bound for the mission cost). Consequently, proofs for DMRR-Sat hardness are provided.

Finally, Section 4 proposes *dynamic saturation-based auctioning* (DSAT) for allocation of the new discovered targets at each time step. DSAT encapsulates an auctioning algorithm introduced here as *inverse-SSI*. It combines elements of parallel and sequential auctions, providing better time complexities than state-of-the-art solutions. The algorithm is described along with its complexity analysis and experiments compare the DSAT and inverse-SSI with state-of-the-art auction algorithms such as standard single-item auctions (SSI (Lagoudakis et al. 2005)) or SSI with regret clearing (Zheng et al. 2008). Computations are performed for an extensive set of scenarios (Section 5). We discuss empirical results, showing that DSAT and inverse-SSI perform better than existing state-of-the-art methods.

2 Background and Related Work

Multi-robot routing considers a set of mobile robots R and static targets T, whose locations are known in the two-dimensional plane. A cost is considered for traveling between two locations on the map and all robots can communicate between them without errors.

Definition 2.1 (MRR)**.** *The multi-robot routing problem (Lagoudakis et al. 2005) consists in finding allocations of targets T to robots R (and paths to visit these targets), such that a team objective function is optimized.*

MRR has been the standard testbed for multi-robot task allocation problems. Originally part of the vehicle routing problems, VRP (Pillac et al. 2013), MRR differs from VRP on at least two fundamental things. First, it does not rely on a predefined path graph (but considers movement between any two locations in the area). Second, for the dynamic case, targets might not be discarded after one visit, but remain part of the robot mission. Hence, the problem objective is dependent on the whole mission, not only on the robot's current position. Throughout this paper, like in usual MRTA terminology, task allocation is equally referred as target allocation (in literature, this often depends on the context). Task is used more to emphasize the action (which may have an ending time), and it can be seen as a treatment of a target.

Taxonomies regarding MRTA (Gerkey and Matarić 2004; Korsah, Stentz, and Dias 2013) have been widely used to classify task allocation problems. Thereby, (Gerkey and Matarić 2004) classifies MRTA problems into categories: single-task (ST) or multi-task (MT) robots, single-robot (SR) or multi-robot (MR) tasks, instantaneous assignment (IA), or time-extended assignment (TA) (w.r.t. the target allocation). MRR, as well as dynamic MRR, proposed in this paper, are part of the MT-SR taxonomy, since a task can be assigned to only one robot at a time. However, MRR has often been treated in IA scenarios, whereas DMRR is rather TA, since the process is dynamic and communication is limited, so the assignation of targets is done in time.

A more recent taxonomy (Korsah, Stentz, and Dias 2013) classifies MRTA problems also based on the dependencies between tasks or robots: no-dependencies (ND), inter-schedule dependencies (ID), cross dependencies (XD) and complex dependencies (CD). DMRR falls into the ID class, along with other well-known NP-hard problems like m-TSP[1] or MRR, since the cost of treating a target (e.g. time, travel distance) depends on the other targets treated before.

To our knowledge, so far no work considered target allocation for ongoing evolving missions, nor offered a framework which correlates the target discovery and the dynamic adaptation to these new targets. Paper (Tardioli et al. 2010) offers a framework treating three problems simultaneously: multi-task allocation, cooperative navigation while maintaining the multi-hop robot network connected, and ensuring link quality for real-time communication. However, this considers robot clusters may be assigned to one task (i.e. MR tasks), and the task allocation plan is performed w.r.t. the connectivity maintenance constraint.

In the last years, MRTA techniques have been used for online planning in scenarios such as warehouse commissioning problems (Claes et al. 2017) or search and rescue operations with uncertainty of tasks (introduced as UMRTA) (Beck et al. 2016), where the information about the tasks relies on probability distributions.

Solving MRTA problems has been addressed mainly using market-based techniques and optimization-based approaches. Market-based techniques use the concept of auctions (Koenig, Keskinocak, and Tovey 2010; Lagoudakis et

[1] Multiple Traveling Salesman Problem, NP-hard

al. 2004; Choi, Brunet, and How 2009), considered state-of-the-art decentralized solutions in MRTA. The auctions require that robots can communicate and use their own information to bid for a task they may probably execute. The negotiation process chooses a winner robot, which the task is assigned to. Optimization-based techniques include deterministic and stochastic methods such as clustering (Janati et al. 2017), graph search (Kartal et al. 2016), or trajectory-based optimizations (Mosteo, Montano, and Lagoudakis 2009; 2008). Recently, auctions were combined with techniques such as clustering (Heap and Pagnucco 2011; Elango, Nachiappan, and Tiwari 2011), or matroid theory techniques (Williams, Gasparri, and Ulivi 2017).

Auction-based methods for MRR solving include combinatorial, sequential or parallel auctions (Koenig, Keskinocak, and Tovey 2010). Combinatorial auctions require exponential amounts of time to compute solutions which minimize the objective. For this, robots bid on all possible combinations of targets. Parallel auctions allocate the targets in linear time single-round auctions, robots bidding for every target only. Sequential single-item auctions (SSI (Koenig et al. 2006)) are a trade-off between combinatorial and parallel auctions, providing in a reasonable time results with guaranteed sub-optimality. Lately, parallel simulations of SSI auctions have been studied (Kishimoto and Nagano 2016). Improvements of SSI auctions include SSI auctions with rollouts, bundle-bids or regret clearing (Koenig, Keskinocak, and Tovey 2010). In comparison with standard SSI, these auction methods trade-off between better running time and solution quality.

Dynamic task allocation gained attention in the last years, though few works emerged. For example, in search and rescue operations has been addressed with prediction methods (Wei, Hindriks, and Jonker 2016) executed by single-task robots. However, dynamicity consists in interleaving exploration with target retrieval, and the problem is part of the ST-SR-TA taxonomy (one target per robot).

Late results for MRR include the work of (Heap and Pagnucco 2011), which lets robots sequentially bid on task clusters rather than one task at a time. Clusters are formed before the auction, using k-means clustering, and the empirical results consist in lower team costs than SSI auctions (Koenig et al. 2006) (in a similar runtime). As improvement for dynamic allocation, (Heap and Pagnucco 2012) performs cluster-based reallocation once a robot finished one of its allocated tasks. It extends the previous work of sequential single-cluster auctions (SCC) (Heap and Pagnucco 2011), and combines it with ideas of repeated auctioning for post-initial allocation (Nanjanath and Gini 2010). The latter algorithm allows robots to exchange tasks if this improves the overall team objective. The re-plan is performed after every treatment of a task and the empirical results show the final allocation is close to optimal. However, dynamicity is related only to possible delays, caused by communication loss or dynamic obstacles.

K-means clustering with auctions has been proposed in (Elango, Nachiappan, and Tiwari 2011), for balancing the task allocation among all robots. This technique searches to evenly distribute work loads between robots, and however diverges from typical goals of MRR auction methods. None of the previous approaches considers the existence of targets in ongoing missions when beginning the actual allocation, nor any dynamic adaptation of robot missions to new discovered targets. Algorithms require many iterations for allocation (usually one for every target), not always appropriate for dynamic repetitive allocations.

3 Dynamic Multi-Robot Routing (DMRR)

The following definition extends the multi-robot routing (MRR) problem of (Lagoudakis et al. 2005), to evolving missions (growing groups of targets). It introduces *Dynamic Multi-Robot Routing* (DMRR) as a continuous adaptation of MRR to new targets discovered in time. For this, time steps are denoted by the *superscript* t throughout all the paper.

3.1 DMRR Definition

Let $F{=}R{\cup}E$ be a finite fleet composed of robots $R{=}\{r_1, r_2, \dots, r_n\}$ executing missions on their own set of static targets (T_{r_i}) and robots E exploring the environment for target discovery. Let $T{=}\{t_1, t_2, \dots, t_m\}{=}\cup_i T_{r_i}$ be the set of total targets under missions and $T_E{=}\{t_{m+1}, t_{m+2}, \dots, t_{m+p}\}$ be the set of new discovered targets. The locations of robots R, targets T_E and T are known. Finally, let c_{ij} be the cost of moving between two locations i and j in both directions (can be related to measures like energy, distance, travel time, etc.).

Definition 3.1 (DMRR). *The objective of DMRR (at every time new targets T_E are discovered) is to find an allocation of targets $T{\cup}T_E$ to the robots in $R{\cup}S$ (with $S{\subseteq}E$) and paths to treat these targets s.t. an objective function is optimized.*

After each allocation, targets from T_E become part of T, since they are allocated to robot missions. Robots from E can pass to R, in time: at a given time step, allocations can be made for $R^{t+1}{=}R^t{\cup}S$, where $S{\subseteq}E^t$, while exploration robots become $E^{t+1}{=}E^t{\setminus}S$. Until becoming part of R, robots that explore are simply used for updating the set of discovered targets (T_E). Hence, these robots' positions are not priori necessary, if not being subject to allocation. Throughout the paper, the equivalent time dependent notations are $R^t, E^t, T^t_{r_i}, T^t$ and T^t_E; for readability, these are used only when emphasizing time is necessary.

Using the notations of MRR(Lagoudakis et al. 2005), the team objective is written as follows: at any moment in time,

$$\min_{A} f(g(r_1, A_1), g(r_2, A_2), \dots, g(r_n, A_n)); \qquad (1)$$

consequently, let the asymptotic objective be defined as:

$$\lim_{t\to\infty} \min_{A^t} f(g(r_1, A^t_1), g(r_2, A^t_2), \dots, g(r_n, A^t_n)) = opt, \qquad (2)$$

where f and g measure the performances of the whole team and of each robot, respectively. $A^t{=}\{A^t_1, A^t_2, \dots, A^t_n\}$ is a partition of the targets $T^t{\cup}T^t_E$, where A^t_i is allocated to robot r_i, at moment t. Because the allocation process is repetitive in time, the team objective can be optimized on the short term as well as on the long term (when $t \to \infty$). So the limit opt represents an optimum which f may converge to.

The role of exploration robots is to provide positions of new discovered targets, and to participate to missions when necessary (for example, when the missions of robots R are saturated, e.g. cannot include more targets). Unless explicitly stated otherwise, any robot subject to target allocation refers to a mission robot (not exploration one).

The robot missions might not have an ending time, and targets may be repeatedly visited. Therefore, when the robot position is required, one may use the average of robot's target positions (i.e. centroid of its target cluster). Nevertheless, one may consider the robot is at the position of its last allocated target (MRR assumption).

At moment $t{=}0$, there may be no targets under missions ($T^0{=}\emptyset$, which in fact corresponds to the actual MRR problem), or even no new targets ($T^0{=}T^0_E{=}\emptyset$), if robots did not start the exploration. Then, at any time step t, the set of new targets T_E can change, because of discovered targets, or because of dynamically allocated ones. Like this, targets from T_E always move to T, in time, after their allocation.

In literature, when addressing the MRR problem, several objectives are usually considered, including:

MINSUM: Minimizing the sum of path costs over all robots.
MINMAX: Minimizing the maximum path cost over all robots.

The path cost of a robot r is the cost of visiting all the targets of a cluster S, and let it be denoted by $c(r, S)$ (may be related to energy, distance, time, etc.). Mission costs may also include the time spent on treating targets. Therefore, our discussion makes abstraction of the target treatment cost. Typically for MRR problems, the above objectives concern just the targets in the T_E set only (so the costs $c(r_i, A_i)$, with $A_i{\subseteq}T_E$). For DMRR, A^t is a partition of the $T^t{\cup}T^t_E$, so let the same objectives be defined as follows: at any moment t,

$$\text{MINSUM}_D : \min_{A^t} \sum_i c(r_i, A^t_i), \tag{3}$$

$$\text{MINMAX}_D : \min_{A^t} \max_i c(r_i, A^t_i), \tag{4}$$

and the asymptotic objectives be defined as:

$$\text{MINSUM}_\infty : \lim_{t\to\infty} \min_{A^t} \sum_i c(r_i, A^t_i) = opt_{\text{MINSUM}}, \tag{5}$$

$$\text{MINMAX}_\infty : \lim_{t\to\infty} \min_{A^t} \max_i c(r_i, A^t_i) = opt_{\text{MINMAX}}. \tag{6}$$

The $c(r_i, A_i)$'s represent the costs of executing robot mission r_i on the targets $A_i \subseteq T{\cup}T_E$. The asymptotic objectives consider what happens in time with these costs (after a certain amount of steps or an undetermined period of time). In particular, the sum or the average of the path costs may converge towards an optimal value. Under the above objectives, MRR is NP-hard (Lagoudakis et al. 2005), so DMRR is hard to solve as well, being at least as complex as MRR.

Theorem 3.1. *1. There is no polynomial algorithm solving the DMRR, under any of the* MINSUM_D, MINMAX_D, MINSUM_∞ *or* MINMAX_∞ *objectives (unless P=NP).*

Proof. Consider that $T^0{=}\emptyset$ and T^0_E contains the new discovered targets. Solving DMRR for time step $t{=}0$ equals to solving the MRR problem. So MRR reduces to DMRR. □

3.2 DMRR with Saturation (DMRR-Sat)

In the context of dynamic adaptation, since the robot mission evolves, the tasks may not have an ending time (e.g. in patrolling, robot execution does not finish). That is why, as robot mission grows in size (e.g. costs of handling new targets), the mission can get saturated. The saturation can be due to robot or target constraints (e.g. robot energy, target visit frequency). The problem, namely DMRR-Sat, considers a saturation bound for each robot mission (in particular, a bound for the path costs $c(r_i, T_{r_i})$). As a consequence, the robots that explore (E) may become part of the robots that treat the targets (R), e.g. when the already ongoing missions get saturated.

Considering the reallocation of all existing targets every time new targets are discovered is extremely time consuming in practice, since the allocation input would be $|T|$, whereas $|T|{\gg}|T_E|$. In addition, it can break the ongoing missions and their related constraints (e.g. for targets which depend on continuous robot execution, like in patrolling), requesting resource consumption for reestablishing all missions. Hence, in DMRR-Sat, only the targets T^t_E are allocated at time t.

Definition 3.2 (DMRR-Sat)**.** *Consider the setting of the DMRR problem. Let Sat be a saturation bound for any robot mission, of the same type as the cost function c. Consider that exploration robots (E) can execute missions on targets from T^t_E at any time t. The DMRR-Sat problem consists in finding allocations of T^t_E to R and robot paths that optimize the team objective f, while respecting constraint $c(r_i, T^t_{r_i}) \leq Sat$, for every robot r_i.*

Since in DMRR-Sat the already allocated targets are not subject to new reallocations, in the objectives (1) and (2) function f becomes

$$f(g(r_1, T_{r_1}{\cup}B_1), g(r_2, T_{r_2}{\cup}B_2), \ldots, g(r_n, T_{r_n}{\cup}B_n))$$

and for objectives (3), (4), (5) and (6), the costs become $c(r_i, T^t_{r_i}{\cup}B^t_i)$, where $\{B_i\}_{i:1,n}$ is a partition of T_E only. Let these objectives be denoted by MINSUM_{D-Sat}, MINMAX_{D-Sat}, $\text{MINSUM}_{\infty-Sat}$ and $\text{MINMAX}_{\infty-Sat}$. For example, (3) becomes

$$\text{MINSUM}_{D-Sat} : \min_{B^t} \sum_i c(r_i, T^t_{r_i}{\cup}B^t_i).$$

Because the missions can get saturated and the fleet F is finite, an intuitive objective to consider is that robots in R cover as many targets as possible (the MAXTAR objective). The following objectives are considered for DMRR-Sat:
MINSUM_{D-Sat}, MINMAX_{D-Sat},
$\text{MINSUM}_{\infty-Sat}$, $\text{MINMAX}_{\infty-Sat}$ (after a certain amount of steps),
MAXTAR: $\max_{B^t} \sum_t \sum_i |T^t_{r_i}{\cup}B^t_i|$.

These objectives are subject to the mission saturation constraint $c(r_i, T^t_{r_i}) \leq Sat$.

Theorem 3.2. *There is no polynomial algorithm solving the DMRR-Sat under the above objectives, unless P=NP.*

Proof. The proof shows that DMRR reduces to DMRR-Sat. Let T and T_E be the sets of targets of DMRR problem which needs to be solved. Consider a large enough saturation bound, (e.g. greater than any $c(r_i, T \cup T_E)$ or even $Sat = \infty$). At every time step of DMRR we solve DMRR-Sat for $T' = \emptyset$ and $T'_E = T \cup T_E$, which gives the solution for DMRR. Hence, reduction time is constant. Using Th.3.1, the proof is complete (for the first four objectives).

For the MAXTAR objective, we show that the NP-complete TSP reduces to DMRR-Sat. Let S be the set of targets for the TSP problem. Let all the finite precision real values be scaled to integers. Consider the interval $[Left, Right]=[0, TSP_{sol}]$, where TSP_{sol} is any solution of TSP computed ad-hoc in $O(|S|)$ time. Fix $Sat = \lfloor (Left + Right)/2 \rfloor$ and solve DMRR-Sat for $T=\emptyset$ and $T_E=S$; if MAXTAR $=|S|$ then $Left \leftarrow Sat$, otherwise $Right \leftarrow Sat$. Continue the binary search until $Left=Right$, which is the actual TSP optimal solution. This takes $\log(TSP_{sol})$ steps, so the reduction is linear in the size of the input (and DMRR-Sat is NP-hard under MAXTAR). □

MAXTAR considers that missions must integrate as many targets as possible (intuitively speaking, robots should treat the maximum number of tasks they can). This also minimizes the number of mission robots ($|R|$), resulting in a maximization of the number of exploring robots ($|E|$), which again motivates the objective.

4 Target Allocation Heuristics

Like stated in the related work section, auction-based methods represent the state-of-the-art decentralized solutions for MRR problems. However, existing solutions do not handle allocations of targets for missions that continuously grow and get saturated. Before proposing the dynamic saturation-based auction (DSAT), this section describes sequential and parallel single-item auction methods, like SSI or PSI auctioning. DSAT method combines advantages of sequential and parallel single-item auctions, that are presented in what follows.

4.1 Sequential and Parallel Auctions

Sequential Single-item Auction (SSI) In standard SSI auctions (Koenig et al. 2006), every robot submits the lowest bid among all targets it can bid on. Then, the lowest bid is chosen among all robots, and the corresponding target is acquired by that robot. Once a robot wins a target, it updates its location to the one of the target just got. The auction continues until all the targets have been allocated.

Sequential Single-item Auction with Regret Clearing (SSI-rc) SSI auctions with regret clearing (Zheng et al. 2008) use same mechanism as standard SSI, with one exception: the winner determination phase allocates the target with the largest regret to the robot that made the smallest bid on it. The regret of a target is the difference between its two lowest bids. This time, the robots send the bids for all the targets, in the bidding phase.

Ordered Single-item Auction (OSI) The auction-based greedy approach introduced in (Schneider et al. 2015) assumes the targets are placed in an ordered list. Each target is offered to all robots, one in every auction round. The robot with the smallest bid receives the target and in addition, it updates its own location with the one of the target it just won.

Parallel Single-item Auction (PSI) The method called parallel single-item auction (Koenig et al. 2006) allocates all the targets in one round. The robots submit all their bids, and each target goes to the robot that made the smallest bid on it. This is the fastest approach in terms of running times, but the solution quality is usually the worst, since no path cost is considered in the bidding.

4.2 Dynamic Saturation-based Auction (DSAT)

The coordination system that this paper proposes (DSAT auctioning) combines benefits of sequential and parallel single-item auctions. It allows both single or multiple target allocation in one auction round. DSAT allocation can handle scenarios in which mission can continuously grow in size, and targets can lose candidate robots due to high robot mission costs. While trying to maximize the target coverage, DSAT copes with ideas of parallel auctions (for achieving better running times) and SSI auctions (for objectives such as MINMAX or MINSUM).

At the beginning, all new targets are initially unallocated. The locations of the new targets T_E are known. For any mission robot r, its location is known, and it may be approximated based on the positions of its targets T_r. The location of an exploration robot is required only if the auctioning decides to form a new mission in the end, because of already saturated ones. Robots bid on the targets with an estimated allocation cost. This cost may usually consider the path cost of robot's already allocated targets. The cost can be an approximative or an optimal path cost. Usually, computing the cost of the optimal path can be expensive (e.g. TSP solution). So the robot computes final path minimization when the allocation is finished. Thereby, when bidding for a target t, an approximative path cost is computed by robot r : $c(r, T_r \cup \{t\})$ (usually computed as the sum of distances between allocated targets).

The auction computations can be run by each robot individually (decentralized), or centralized, by an auctioneer robot. The auctioneer would then deliver the results after every auction round. In the decentralized case, a robot listens to the others and memorizes all the other robot bids. Therefore, it can perform the rest of the computations and determine the winners by itself. Like this, all robots execute the algorithm simultaneously and know which target has been allocated to which robot. Because of the decentralized behavior, more messages are passed between robots, but this avoids robots being dependent on an auctioneer.

DSAT Algorithm Description (Alg. 1) The algorithm starts with robots submitting their bids for every target. A robot submits a "not_a_candidate" bid if it estimates that integrating a target would result in its mission cost exceeding the saturation bound. Upon the arrival of all robot bids, the bid lists $B_i=\{b_i^1, b_i^2, \dots, b_i^{k_i}\}$ are formed for every tar-

```
Algorithm 1: DSAT-Auction(R, E, T_E)
1  Input: R: the set of mission robots
2         E: the set of exploration robots
3         T_E: the set of new discovered targets
4  Output: T_E=∅: All targets are allocated
5  for each robot r ∈ R do
6      for each target t ∈ T_E do
7          submit bid(r, t) if c(r, T_r∪{t}) ≤ Sat_r;
8          else submit "not_a_candidate";
9  for each target t_i ∈ T_E do
10     form the bid list B_i;
11     sort(B_i);
12 group the t_i's by their |B_i| (nb. of candidate robots);
13 let k := max_i |B_i| and G_k := {t_i ∈ T_E | k = |B_i|}
14 while T_E ≠ ∅ and k ≥ 1 do
15     // one round of inverse SSI
16     T' := InverseSSI(G_k) (for targets with |B_i|=k);
17     T_E := T_E \ T';
18     update robot bids in every B_i (and update |B_i|);
19     regroup t_i's by their |B_i| and recompute k;
20 if k = 0 then
21     // form new missions for targets with |B_i| = 0
22     choose robots E' ⊆ E for missions on T_E;
23     run DSAT-Auction(E', E\E', T_E);
```

```
Algorithm 2: InverseSSI(G_k)
1  // sorted lists B_i={b_i^1, b_i^2, ..., b_i^k} are already formed
2  // G_k is the target group with |B_i| = k, ∀t_i ∈ G_k
3  G'_k := G_k; Alloc := ∅;
4  while G'_k ≠ ∅ do
5      let t_i ∈ G'_k;
6      Q := {t_j ∈ G'_k | r(b_j^1) = r(b_i^1)};
7      t := arg min_{t_j ∈ Q} b_j^1;
8      Alloc ∪= {t};  R' ∪= {r(t)};
9      T_{r(t)} ∪= {t};  G'_k \= Q;
10 for each robot r ∈ R' do
11     for each target t ∈ T_E do
12         submit bid(r, t) if c(r, T_r∪{t}) ≤ Sat_r;
13         else submit "not_a_candidate";
14 return Alloc;
```

get $t_i \in T_E$, and then they are sorted. This helps for faster identification of the smallest bid of a target (b_i^1) when robots update their bids, at the end of an auction round.

DSAT auctioning continues with a clustering-like phase, where targets are grouped based on the size of the B_i lists (i.e. the number of their candidate robots; Lines 12 and 13). Then, the targets with the biggest number of candidate robots are auctioned first (Line 13).

The winer determination and the new bidding phase are performed in what is defined as inverse-SSI auction (and explained below, in Algorithm 2). After one call of the InverseSSI function (i.e. one round of inverse-SSI), the new bids are received by all robots. Each robot then updates the bids in the B_i lists (Line 18). Robots can submit "not_a_candidate" bids, so targets can lose or gain candidate robots. Thereby, the sizes of the candidate lists are updated, so targets get regrouped while the maximum $|B_i|$ is recomputed on the way (Line 19).

When the auction finishes, targets with zero candidates are re-auctioned to form new missions. A set S of exploration robots is selected from E (e.g. the closest robots to T_E centroid). These robots become the input set R in the re-auctioning process. At the end, they might all be mission robots ($R^{t+1}{=}R^t{\cup}S$ and $E^{t+1}{=}E^t{\setminus}S$), unless there exists a robot $r \in S$, with $T_r{=}\emptyset$.

Inverse-SSI Auction (Alg. 2) The technique defined by this paper as *inverse-SSI auction* is used as part of the dynamic saturation-based auction, but also as a standalone algorithm. It consists of the robot bidding and winner determination phases. The mechanism is somehow an inverse of the standard SSI auction reasoning. The bids are evaluated from the target perspective and a conflict resolution is performed for the targets which prefer the same robot. However, it differs from standard SSI by its parallelism, since inverse-SSI permits allocating multiple targets in one round. For example, it may be the case that all targets are allocated in one auction round, or to the contrary, taking $|T|$ rounds to complete the auction.

Bidding Phase: The auctioning starts with all robots bidding on all targets. In each of the next auction rounds, only the robots which won in the previous round are submitting their new bids (Line 10). The new bids take into account the new cost of the robot missions, estimated upon the last target allocation.

Winner Determination: Targets are grouped according to the robot of their smallest bid: $r(b_i^1)$, where $t_i \in T_E$. This is the robot that a target prefers, i.e. its first candidate. For targets which prefer the same robot (set Q, Line 6), the target with the smallest bid is selected and allocated for that robot (Line 7). After an allocation, all the targets in Q (Line 9) are discarded from the current auction round. This is because the robot they prefer changed its mission saturation value, after the allocation. Its new bids are used to update the target bid lists used in the next auction rounds.

As standalone algorithm, inverse-SSI is run on the set of targets T_E (main input parameter). It is then executed repeatedly (like standard SSI or SSI-rc), until no more allocations can be made.

DSAT Auction Complexity If a number of $|T|$ targets is subject to allocation, then the auctioning process of DSAT can finish in 1 up to $|T|$ rounds, so the best and worst case running times differ by a large amount of computations. This means DSAT best case time complexity reduces to one complete auction round $O(|T||R| \log |R|)$, as explained below, which outdoes best case complexities of SSI and SSI-rc ($O(|T||R| + |T|^2)$ and $O(|T|^2 \log |R|)$, respectively).

In the beginning, all robots bid on all targets, hence $|R|^2$ messages are transmitted between robots. Then, at most $|T||R|$ messages are sent for all the other rounds, since only the winning robots bid again. This gives a maximum of $|R|^2 + |T||R|$ messages for the whole DSAT auction (like for standard SSI).

If an auctioneer is used (to centralize messages and computations), the number of messages is decreased by a factor of $|R|$ (since no broadcast is needed among robots). Note that if all robots submit their bids at the end of an inverse-SSI round, then a total of $|T||R|^2$ messages are transmitted (the equivalent of SSI with regret clearing).

In what follows, a denotes the number of rounds performed by DSAT auction. The bidding lists B_i are formed and sorted in $|T||R| \log |R|$ steps (using MergeSort algorithm, for example). Grouping the targets by their B_i size takes $|T|$ steps, so a total of $a|T|$ for all DSAT rounds. To complete the winner determination, one round of inverse-SSI takes $2|T|$ time steps: selecting the targets which prefer the same robot and obtaining the minimum of their bids. In total, this requires $a|T|$ steps until DSAT finishes.

Updating a bid of a robot in the B_i lists can be done in time $O(|T| \log |R|)$, since the lists are sorted. For example, binary search can be used for removing the old bid and inserting the new bid in the B_i's. This is performed at most $|T|$ times during the whole auctioning, so the total complexity is $O(|T|^2 \log |R|)$. If required in the end of DSAT auctioning, exploration robots can be chosen randomly in constant time. However, depending on the objective of the DMRR-Sat, a polynomial-time heuristic can be used to compute an approximative cost C of a mission on T_E. Then, a number of $\lceil C/Sat \rceil$ robots from E can be chosen (e.g. robots closest to the T_E centroid).

In conclusion, the worst case time complexity of the whole DSAT algorithm is $O(|T|^2 \log |R|)$ and a number of $|T||R| + |R|^2$ messages are transmitted. This complexity equals both SSI and SSI-rc auctions in terms of running time, and the one of standard SSI for the number of transmitted messages. One may note that the complexity of the winner determination phase is linear in the number of bids submitted: $O(|T||R|)$.

5 Experimental Results

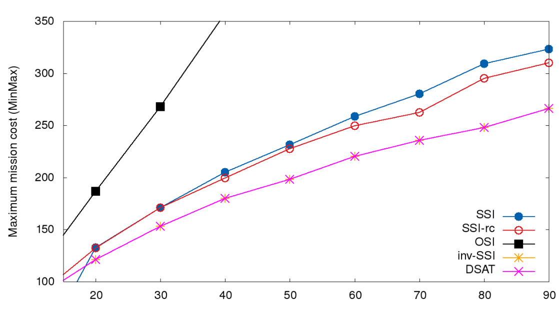

Figure 2: The maximum mission cost among all robots, over the number of targets discovered. No saturation bound is considered ($Sat = \infty$).

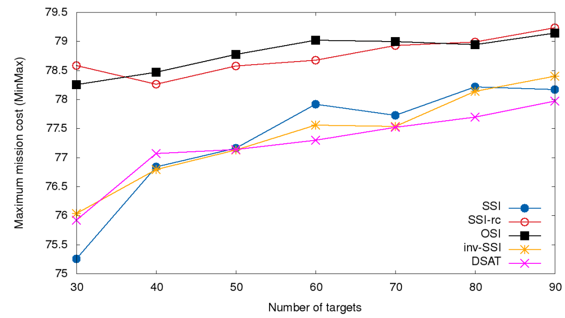

Figure 3: The maximum mission cost among all robots, for a mission saturation bound of 80.

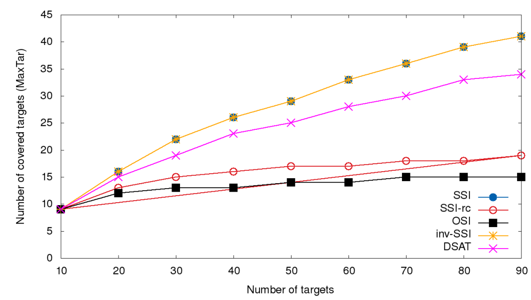

Figure 4: The number of covered targets in the area, for a mission saturation bound of 80. Evolution of costs w.r.t. the number of discovered targets.

Experiments are performed considering euclidean distances in the 2D plane. The targets are randomly generated using the uniform distribution, in an area of 100×100 units. The robots bid for the targets, taking always into account the cost of their mission. For simplicity, these costs are estimated using the sum of the traveling paths (euclidean distances), among the targets they own. The distances are computed in the order in which targets get allocated to the robot. A robot is allowed to bid for a target only if the approximated cost of its integration does not exceed the mission saturation bound. All the computations are performed for an extensive set of 100 random configurations of the targets in the area. All the metrics in this section are mean values obtained in the computations. At the beginning, 4 robots are deployed in the area. Targets are randomly generated, and robots start the auction algorithm for target allocation. Every time new targets are generated, the algorithms are run again. Due to the saturation bound, not all targets can be covered. Therefore, another 3 robots are added in the area, in time. This happens once the targets can not be all covered by the allocation (see Figure 4).

In the experiments, the performance of parallel single-item auction (PSI) is extremely bad. Hence, because of the big differences in the graphics, we leave it out and compare the other five algorithms.

Figure 2 shows the results obtained for the MINMAX$_D$ objective, when no saturation bound is considered. DSAT and standalone inverse-SSI, which perform equally here,

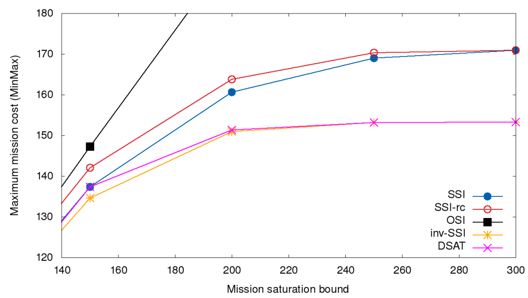

Figure 5: The behavior of the maximum mission cost w.r.t. different saturation values. The number of fixed targets: 30.

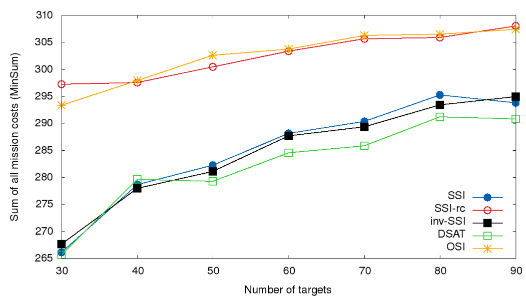

Figure 6: The sum of all the mission costs of all robots, w.r.t. number of discovered targets. Mission saturation bound: 80.

show lower mission costs than all the other algorithms. In this case, all targets are covered by the robots. The mean of MinMax_D for the 100 random target configurations shows a difference of $\approx$ 50 units between SSI and DSAT. This represents around 16% of the mission cost, DSAT outperforming all the other algorithms. Mission costs translate into robot resource usage, hence resource consumption is optimized among all robots. This result applies to DMRR and consequently to MRR, since no saturation bound was considered. Figure 3 shows the resulting MinMax_{D-Sat} for the same configurations, when a mission saturation bound of 80 is considered. Because the costs are now bounded, the values are more compact, and SSI, DSAT and inverse-SSI have almost the same performance (with a slight advantage for DSAT). The costs of OSI do not increase like before (Figure 2), because of the saturation bound; however less targets are covered by the algorithm (Figure 4). Mission saturations reduce the target coverage percentage. In Figure 4, SSI, inverse-SSI and DSAT outperform the other three algorithms with a difference of 50% of covered targets. Here, SSI and inverse-SSI cover the same number of targets.

When changing the saturation bound for the same configurations of targets, the allocations can oscillate, since new bids are allowed to participate. Results in Figure 5 show that DSAT and standalone inverse-SSI continue to outperform standard SSI and SSI with regret clearing, with a difference of $\approx$ 12% in the mission cost. This again results in lower resource usage for the robot missions.

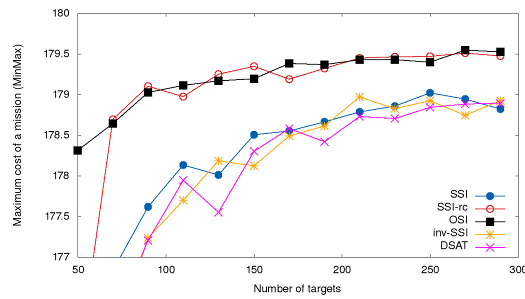

Figure 7: Convergence of the maximum mission cost ($\text{MinMax}_{\infty-Sat}$) towards the optimum ($Sat$=180). An exhaustive generation of targets is performed.

The sum of the mission costs obtained in the experiments can be observed in Figure 6. DSAT has a slight advantage in front of SSI and inverse-SSI (which perform equally), but all three algorithms outperform SSI-rc by a percentage of 5. For the $\text{MinMax}_{\infty-Sat}$ objective, Figure 7 shows the convergence of the algorithms towards the optimum $opt_{\text{MinMax}}=Sat=180$, as new targets are discovered in time. Experiments show that the maximum mission cost tends to the considered optimum, as new targets appear in the area.

6 Conclusion

The present paper exposed the challenges of dynamic target allocation and the drawbacks of the existing literature. It then proposed a framework which defines DMRR (Dynamic Multi-Robot Routing) and contains a mathematical model, complexity results and proofs, but also a collateral problem generated by the evolving missions which dynamically grow (DMRR-Sat). Asymptotic objectives for DMRR were also analyzed, even though assuming an undetermined (or infinite) time period makes them impractical for experiments. DSAT (Dynamic Saturation-based Auction) is proposed as solution for dynamic target allocation and robot coordination. It relies on a technique defined in the paper as inverse-SSI, which stands for both sequential and parallel allocations. Complexity analysis showed that DSAT and inverse-SSI provide running times that overdo the existing state-of-the-art, yet providing better solution qualities when tested (for both DMRR and DMRR-Sat). The experimental results performed on exhaustive sets of inputs show that DSAT and inverse-SSI outperform methods such as SSI or SSI-rc, especially for objectives such as MinMax or MinSum. These results apply also to classic MRR, since experiments were performed as well for DMRR without a saturation bound. Further work emerging from DMRR could consider problems such as dynamic reallocation of targets over time, dynamic reallocations with saturation bounds, or dynamic clustering in target allocation.

Acknowledgments

Funding for this project was provided by a grant from la Région Rhône-Alpes (France) and by the bilateral Franco-Romanian PHC Project "DRONEM".

References

Beck, Z.; Teacy, L.; Rogers, A.; and Jennings, N. R. 2016. Online planning for collaborative search and rescue by heterogeneous robot teams. In *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*, 1024–1033. International Foundation for Autonomous Agents and Multiagent Systems.

Choi, H.-L.; Brunet, L.; and How, J. P. 2009. Consensus-based decentralized auctions for robust task allocation. *IEEE transactions on robotics* 25(4):912–926.

Claes, D.; Oliehoek, F.; Baier, H.; and Tuyls, K. 2017. Decentralised online planning for multi-robot warehouse commissioning. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*, 492–500. International Foundation for Autonomous Agents and Multiagent Systems.

Elango, M.; Nachiappan, S.; and Tiwari, M. K. 2011. Balancing task allocation in multi-robot systems using k-means clustering and auction based mechanisms. *Expert Systems with Applications* 38(6):6486–6491.

Gerkey, B. P., and 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.

Heap, B., and Pagnucco, M. 2011. Sequential single-cluster auctions for robot task allocation. *AI 2011: Advances in Artificial Intelligence* 412–421.

Heap, B. G. J., and Pagnucco, M. 2012. Repeated sequential auctions with dynamic task clusters. In *AAAI*.

Janati, F.; Abdollahi, F.; Ghidary, S. S.; Jannatifar, M.; Baltes, J.; and Sadeghnejad, S. 2017. Multi-robot task allocation using clustering method. In *Robot Intelligence Technology and Applications 4*. Springer. 233–247.

Kartal, B.; Nunes, E.; Godoy, J.; and Gini, M. 2016. Monte carlo tree search with branch and bound for multi-robot task allocation. In *The IJCAI-16 Workshop on Autonomous Mobile Service Robots*.

Khamis, A.; Hussein, A.; and Elmogy, A. 2015. Multi-robot task allocation: A review of the state-of-the-art. In *Cooperative Robots and Sensor Networks 2015*. Springer. 31–51.

Kishimoto, A., and Nagano, K. 2016. Evaluation of auction-based multi-robot routing by parallel simulation. In *ICAPS*, 495–503.

Koenig, S.; Tovey, C.; Lagoudakis, M.; Markakis, V.; Kempe, D.; Keskinocak, P.; Kleywegt, A.; Meyerson, A.; and Jain, S. 2006. The power of sequential single-item auctions for agent coordination. In *Proceedings of the national conference on artificial intelligence*, volume 21, 1625. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999.

Koenig, S.; Keskinocak, P.; and Tovey, C. A. 2010. Progress on agent coordination with cooperative auctions. In *AAAI*, volume 10, 1713–1717.

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

Lagoudakis, M. G.; Berhault, M.; Koenig, S.; Keskinocak, P.; and Kleywegt, A. J. 2004. Simple auctions with performance guarantees for multi-robot task allocation. In *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, volume 1, 698–705. IEEE.

Lagoudakis, M. G.; Markakis, E.; Kempe, D.; Keskinocak, P.; Kleywegt, A. J.; Koenig, S.; Tovey, C. A.; Meyerson, A.; and Jain, S. 2005. Auction-based multi-robot routing. In *Robotics: Science and Systems*, volume 5, 343C350. Rome, Italy.

Mosteo, A. R.; Montano, L.; and Lagoudakis, M. 2008. Multi-robot routing under limited communication range. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, 1531–1536. IEEE.

Mosteo, A. R.; Montano, L.; and Lagoudakis, M. G. 2009. Guaranteed-performance multi-robot routing under limited communication range. *Distributed Autonomous Robotic Systems* 8:491–502.

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

Pillac, V.; Gendreau, M.; Guéret, C.; and Medaglia, A. L. 2013. A review of dynamic vehicle routing problems. *European Journal of Operational Research* 225(1):1–11.

Pippin, C.; Christensen, H.; and Weiss, L. 2013. Performance based task assignment in multi-robot patrolling. In *Proceedings of the 28th annual ACM symposium on applied computing*, 70–76. ACM.

Schneider, E.; Sklar, E. I.; Parsons, S.; and Özgelen, A. T. 2015. Auction-based task allocation for multi-robot teams in dynamic environments. In *Conference Towards Autonomous Robotic Systems*, 246–257. Springer.

Tardioli, D.; Mosteo, A. R.; Riazuelo, L.; Villarroel, J. L.; and Montano, L. 2010. Enforcing network connectivity in robot team missions. *The International Journal of Robotics Research* 29(4):460–480.

Toth, P., and Vigo, D. 2014. *Vehicle routing: problems, methods, and applications*. SIAM.

Tovey, C.; Lagoudakis, M.; Jain, S.; and Koenig, S. 2005. The generation of bidding rules for auction-based robot coordination. *Multi-Robot Systems. From Swarms to Intelligent Automata Volume III* 3–14.

Wei, C.; Hindriks, K. V.; and Jonker, C. M. 2016. Dynamic task allocation for multi-robot search and retrieval tasks. *Applied Intelligence* 45(2):383–401.

Williams, R. K.; Gasparri, A.; and Ulivi, G. 2017. Decentralized matroid optimization for topology constraints in multi-robot allocation problems. In *Robotics and Automation (ICRA), 2017*, 293–300. IEEE.

Zheng, S. K. X.; Tovey, C.; Borie, R.; Kilby, P.; Markakis, V.; and Keskinocak, P. 2008. Agent coordination with regret clearing. In *Proceedings of the Twenty-third (AAAI-08) Conference on Artificial Intelligence, 13-17 July 2008, Chicago, Illinois, USA: Machine Learning*, 101. AAAI Press.