Multiple Traveling Robot Problem: A Solution Based on Dynamic Task Selection and Robust Execution

Sanem Sariel-Talay, *Member, IEEE*, Tucker R. Balch, and Nadia Erdogan

***Abstract*—The multiple traveling robot problem (MTRP), the real-world version of the well-known NP-hard multiple traveling salesman problem (MTSP), asks for finding routes of robots to visit a set of targets. Various objectives may be defined for this problem (e.g., minimization of total path length, time, etc.). The overall solution quality is dependent on both the quality of the solution constructed by the paths of robots and the efficient allocation of the targets to robots. Unpredictability of the exact processing times of tasks, unstable cost values during execution, and inconsistencies due to uncertain information further complicate MTRP. This paper presents a multirobot cooperation framework employing a dynamic task selection scheme to solve MTRP. The proposed framework carries out an incremental task allocation method that dynamically adapts to current conditions of the environment, thus handling diverse contingencies. Globally efficient solutions are obtained through mechanisms that result in the allocation of the most suitable tasks from dynamically generated priority-based rough schedules. Since the presented approach is for real-world task execution, computational requirements are kept at a minimum, and the framework is designed to be applicable on real robots even with limited capabilities. The efficiency and the robustness of the proposed scheme is evaluated through experiments both in simulations and on real robots.**

***Index Terms*—Distributed multirobot task allocation, incremental task selection, multiple traveling robot problem (MTRP), robustness.**

I. Introduction

THE USE OF a multirobot team is usually beneficial for efficiency in search and rescue (SR), space, and reconnaissance/surveillance operations. These application domains have a high resemblance in problem formulation although their overall optimization objectives may differ. In all these domains, basically the targets, either determined previously or in runtime, are visited by the robot team. The problem can be reduced to the multirobot multi-target exploration problem in which a certain objective is optimized. In SR operations, time is of the essence, while in space exploration operations, it is better to reduce energy consumption, which is proportional to the total path length traversed by all robots.

Manuscript received July 21, 2008; revised October 24, 2008. First published March 27, 2009; current version published April 15, 2009. Recommended by Guest Editor M.-Y. Chow. This work was supported in part by the U. S. Navy, in part by the National Science Foundation, in part by SIEMENS Turkey, and in part by Tincel Kultur Vakfi.

S. Sariel-Talay and N. Erdogan are with the Department of Computer Engineering, Istanbul Technical University, 34469 Istanbul, Turkey (e-mail: sariel@itu.edu.tr; nerdogan@itu.edu.tr).

T. Balch is with the College of Computing, Georgia Institute of Technology, Atlanta, GA 30308 USA (e-mail: tucker.balch@cc.gatech.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMECH.2009.2014157

This paper investigates the multirobot exploration problem and proposes a new framework, **D**istributed and **E**fficient **M**ulti**R**obot-**C**ooperation **F**ramework (DEMiR-CF) [1] as a solution. DEMiR-CF, designed for multirobot teams cooperating to achieve a global mission, can be used to solve problems on a wide variety of application domains. Team members cooperate to fulfill a mission by dividing the labor of task execution through individual decisions that coordinate their actions in a distributed manner.

The single-robot multi-target exploration problem is NP-hard [2]. In the multirobot case, besides route generation, allocation of targets to robots also has a significant impact on the solution quality. Although the problem domain considered in this paper consists of a single type of task that can be executed by a homogeneous team of robots, still it is NP-hard due to the combinatorial structure of the problem. The problem area is well studied in the field of operations research, and optimal solutions can be obtained by integer programming (IP) formulations. However, these approaches may become impractical when the size of the mission is even moderate or the cost values change frequently due to uncertain knowledge, changes in the environment (including failures), or changing structure of the mission (e.g., online generated tasks). Furthermore, robots have continuous path planning burdens for target sets in dynamic environments. Expensive computational efforts for initial allocations may become redundant as conditions change. DEMiR-CF eliminates these redundant efforts by means of incremental assignments based on the current state of the environment. It can also handle contingencies by the precaution routines embedded in its integrated structure. Communication failures may sometimes prevent allocations from being optimal. DEMiR-CF can also detect these situations and maintains high solution quality through a dynamic task selection/exchange scheme.

The contributions of this paper are twofold. First, it deeply investigates MTRP, and second, it introduces a robust, distributed framework, DEMiR-CF, as a solution to this problem. Simulation experiments and the real-world scenarios reveal both the task allocation efficiency and the contingency handling performance of DEMiR-CF.

II. Multiple Traveling Robot Problem: Problem Statement

The single-robot exploration problem, a variation of the well-known NP-hard Traveling Salesman Problem (TSP), is to find the minimum cost traversal of a given number of targets (T) without considering the return cost from the last target to the initial location for a single robot. The problem can be stated as

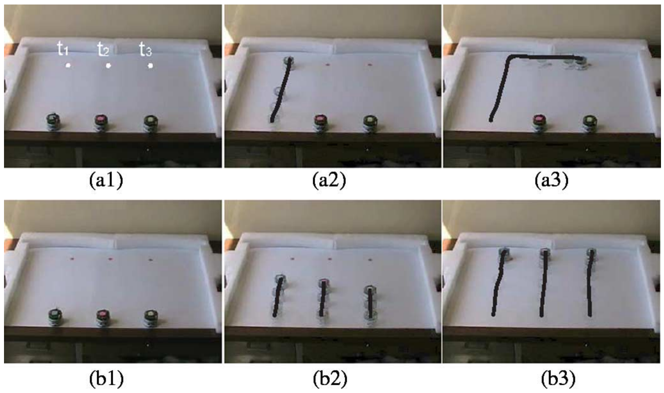

Fig. 1. Two different optimization objectives for MTRP with three targets. The first row illustrates achievement of the total path length minimization objective. In the second row, robots minimize time in achieving the mission.

finding the minimal Hamiltonian path on a given fully connected graph with all nodes to be visited [2].

Multirobot multi-target exploration problem further extends TSP. This problem is called the Multiple Traveling Robot Problem (MTRP) and involves a team of robots (R) to visit targets ($t_i \in M_{\mathrm{MTRP}}$) at least once (ideally at most once). The overall solution quality is dependent on both the quality of the solution constructed by the paths of robots and the efficient allocation of the targets to robots.

A. Cooperation Objectives

Different objectives complementary to the main goal may be selected to optimize the performance of the system, as in scheduling problems. Examples of such objectives are total path length minimization, time minimization, average energy minimization, makespan minimization, etc. Based on the selected objective function, cost evaluation may need to be designed differently. Fig. 1 illustrates the paths that robots traverse in order to optimize target allocations for two different objectives. In the initial configuration, the robots are located at the bottom of the figure, facing the three targets ($t_1, \ldots, t_3$) to be visited. The first row of the figures [(a1)–(a3)] illustrates the achievement of the total path length minimization objective, whereas in the second row of the figures [(b1)–(b3)], the objective is to minimize the total time to complete the mission. When the total path length is to be minimized, a single robot visits all targets. However, for the time minimization objective, all robots are involved in the target visiting process [1]. Related videos of these runs are available online [3].

B. Application Domains for MTRP

MTRP forms an important basis for several domains such as SR, Space exploration, object construction, pick-up/delivery, etc. All these domains contain MTRP ingredients in their problem representations even though their implementations and the complementary objectives may be different. In SR operations, candidate locations that should be visited can be modeled as MTRP targets. Recovery operations after SR operations can also be modeled as MTRP. In space explorations, observation locations can be modeled as MTRP targets. Instead of optimizing time, battery/fuel life of robots may be optimized in this domain. Multirobot systems are also used in nanoassembly planning, where the main objective is to construct paths for robots working to coordinate assembly tasks [4]. In this domain, path waypoints can also be treated as MTRP targets for which efficient generation of paths is needed. The frontier-cell-based exploration to cover unknown environments and creating maps [5] is one of the related application domains to MTRP. In this case, the coverage problem reduces to the assignment of frontier cells to robots efficiently. In naval mine countermeasure mission, the coverage problem is represented as visiting waypoints, which can also be formulated as MTRP [6]. Therefore, although the coverage problem and MTRP seem to be different, they share common structures when appropriately represented.

C. Treating MTRP From the Real-World Perspective

Even in the presence of attentively written orchestra scores or playbooks, the real dynamics of physical task performance enforce some unplanned actions to be taken. Since the world is beyond the control of the robots and changes take place continuously in real-world applications, the difficulty of the multirobot task execution problem goes beyond the task allocation problem. In particular, multirobot systems deal with difficulties arising from noisy sensor information, unexpected outcomes of actions, environmental limitations (especially in communication), and presence of hardware failures. All these factors may influence the overall solution. The evolving circumstances that may change the solution can be listed as follows.

1) *Failure detection:* Robots detect either their own failure or the failure of another robot.
2) *Change in the estimated cost:* Environmental dynamics, uncertain knowledge, or hardware problems may cause delays in task execution or result in early achievements of tasks. Uncertain sensor and/or localization information may also result in incorrect estimations.
3) *Change in the task definitions:* Task dependencies, priorities, or the overall objective (goal) may change. Some tasks may become invalid during runtime.
4) New online tasks may be introduced by human operators or discovered by the robots themselves.
5) New robots may be released, or some failed robots may be repaired or may recover from trap-like threats.
6) Intervention and manual changes by external agents.

III. Solution Methods for MTRP

A. Operations Research Methods for MTRP

MTRP is studied as the multiple TSP problem or the Multi Depot Vehicle Routing Problem (MDVRP) problem in the field of operations research [7]. There are several methods to obtain solutions with either exact optimal value or bounded optimality. Optimal results can be obtained using IP formulations. However, these approaches may become impractical when the size of the mission is even moderate or the cost values change frequently due to uncertain knowledge, changes in the environment (including failures), or changing structure of the mission (e.g., online tasks) [8]. Furthermore, robots have continuous path planning burdens for target sets in dynamic environments. Expensive computational efforts for initial allocations may become redundant. MTRP can be solved by branch and bound algorithms, the performance of which is dependent on the branching and bounding algorithms [7]. This approach also suffers from the solution time guarantees perspective. There are several heuristic approaches to find approximate solutions. Classical heuristic approaches perform limited exploration of the search space and typically produce good results within modest computing times. On the other hand, in metaheuristics approach, the quality of the solution is much higher. However, time complexity worsens dramatically and the applied procedures are usually context-dependent and require finely tuned parameters [7].

B. Robotic Research Methods for MTRP

Operations research methods are applied and integrated into robot systems in earlier work. Both combinatorial [9], [10] and single-auction methods are studied for MTRP. In a combinatorial auction method, different combinations of tasks are offered and allocated to robots by considering all tasks in these combinations. Thus, this method may become intractable for large instances or for dynamic situations in which calculations should be made frequently as in the case of IP method. Computational requirements for combinatorial auctions increase drastically for dynamic environments.

An extensive analysis of the multirobot exploration problem is presented in [11] from the point of view of solution guarantees. The Prim allocation method [12], based on the Prim's algorithm [13], [14], generates a minimum spanning forest (MSF) of the targets and robots. An MSF consists of separate robot trees constructed by adding each unallocated target to the closest robot path containing the node with the minimum distance to the target, until all targets are allocated. In other words, a new target is added by considering the distances between the target and the nodes of the robot tree instead of considering the last position of the robot. Each robot offers an auction for a target and one of the targets is allocated at each round. Before robots run and visit the targets, all targets are allocated. Whenever the world knowledge changes, the remaining unvisited targets are reallocated using the same algorithm. Since the Prim allocation method is discussed with details necessary to implement, it has been possible to compare the performance of DEMiR-CF to that of the Prim allocation method and the results are presented in Section V-B. Depth-first traversal solution of an MST is bounded by 2×OPT, and the traversal and subtree selection does not affect the solution quality in solving TSP. However, for the open-loop version of TSP, as in MTRP, selection of the subtree that is traversed has an impact on the solution quality. To improve the solutions generated by the Prim allocation method, the shortest depth subtree may be selected for the next traversal. This improvement has been incorporated into the Prim allocation method during the experiments presented in this paper.

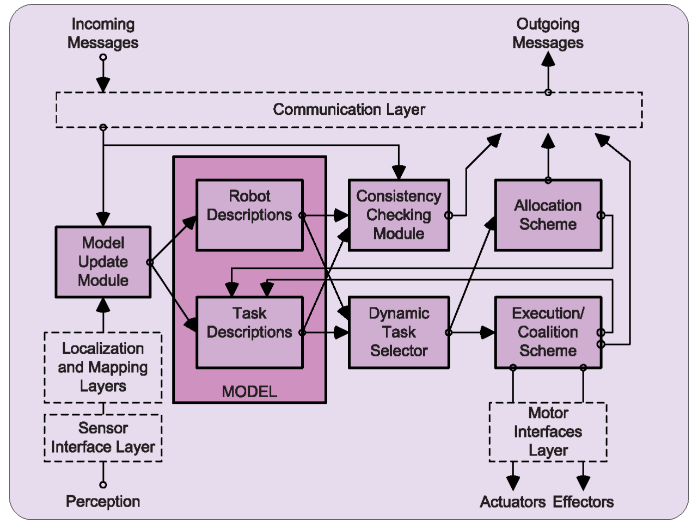

Fig. 2. Overall architecture of DEMiR-CF.

IV. DEMiR-CF

In practical applications, computing the true optimal solutions is not always required due to several reasons [15] such as the incorrect modeling of the underlying problem (targets) or lack of sufficient time to find the optimal solution. These are common cases in robot applications along with the real-time issues presented in Section II. As a solution method to MTRP that meets these limitations, a dynamic and distributed task allocation scheme, DEMiR-CF [1], is proposed to coordinate robots that cooperate to fulfill different parts of a mission. DEMiR-CF combines *The Dynamic Priority-Based Task Selection Scheme* (*DPTSS*), *Distributed Task Allocation Procedures*, and *Coalition Formation Schemes* as cooperation components and *The Precaution Routines*, some of which are implemented by *The Coalition Maintenance/Dynamic Task Selection Scheme*. These components are integrated into a single framework to provide an overall system that finds efficient solutions for real-time task execution. The modules that embody the framework and information flow among them are given in Fig. 2. Each robot keeps a model, up-to-date state information, of the other robots and the mission tasks. *The Model Update Module*, *The (System) Consistency Checking Module*, and *The Dynamic Task Selector Module* perform *precaution routines* by either updating the model maintained by the robot or activating warning mechanisms. Model updates are initiated by either incoming information from the other robots or information perceived by the robot itself. If a system inconsistency exists, *The Consistency Checking Module* is responsible for initiating warning mechanisms and informing the corresponding robots. *The Dynamic Task Selector Module* employs *DPTSS* to select the most suitable task by considering the model of the robot. *The Distributed Task Allocation Scheme* ensures distributed task allocation by executing the required negotiation procedures for the selected task. *The Execution/Coalition Scheme* implements synchronized task execution and coalition maintenance procedures. Task models are updated according to the selected task and the task currently in execution.

A. Dynamic Task Selection and Distributed Allocation

There is a tight connection between route generation and allocations for MTRP. In the proposed heuristic approach, each robot (r_j) initially generates its rough route (rough schedule), and then, selects the most suitable target for itself among the targets in its rough schedule (T_{Rj}). Next, each robot proceeds to announce its intention to execute the selected task. A Contract Net Protocol (CNP) based allocation [16] is implemented to determine the most appropriate robot among the team of robots to perform the task (to visit the target). CNP is modeled on the contracting mechanism used by businesses to govern the exchange of goods and services. The contract net provides a solution to find the most appropriate agent to work on a given task. In the CNP, an agent that has a task to be solved announces the task to the network, along with specifications for the assignment. A recipient of the announcement decides whether it is eligible, and if so, it formulates a bid. The manager collects bids, and awards the task to the contractors with the best bid. DEMiR-CF uses this mechanism to ensure distributed, robust, and scalable allocations.

T_{Rj} is constructed by selecting targets close to the robot r_j, among unvisited targets (T_U) according to (1), where dist function returns the Euclidean distance between two points. Targets in T_{Rj} are considered as the candidate targets for robot r_j. Therefore, before selecting the most suitable target, each robot constructs these rough route sets. This heuristic does not compel an actual commitment, and the targets in the rough routes are not necessarily assigned to the corresponding robots in future auctions. Instead, it provides a global view to the problem from a local perspective.

$$\begin{aligned}\text{reldist}(r_j, t_i) &= \text{dist}(r_j, t_i) - \min(\text{dist}(r_k, t_i))\\ &\quad \{\forall k \neq j, r_k \text{ is active}\}\\ T_{Rj} &= \cup t_i, \quad \text{reldist}(r_j, t_i) < 0 \qquad \forall t_i \in T_U. \qquad (1)\end{aligned}$$

Each robot executes Algorithm 1 to generate its rough schedule. The robot then selects the most suitable candidate task (t_s, the top most suitable target among the rough schedule targets) to perform. Algorithm 2 forms the main loop of incremental task allocation procedure, and it is called in the beginning of mission execution and whenever the world knowledge of the robot changes. Each robot executes the same algorithm concurrently until the end of the mission, when all traversable targets are visited. The given algorithm may be used to allocate all targets from scratch. However, an incremental assignment approach eliminates both the complexity of the decision on allocating all targets to all robots at a time and the redundant allocations for dynamic environments. The cost function design can be determined based on the capabilities of the robot. Two heuristic cost functions, details of which are explained in Section IV-B, are proposed to be employed for the task selection strategy in DEMiR-CF. After selecting the most suitable target for itself, each robot announces its intention by a single-item auction. The best robot–task match is determined using CNP. When robots receive messages that reveal intention to execute a task (as an auction), they either send their cost values as bids for the announced target or warning messages due to inconsistent situations. These warning messages are sent if the auction is for an invalid target, or for a task that has already been achieved or is being currently executed by another robot. Although CNP presents the formalism on the relationships between managers and contractors, some decisions are left to the designer. In most auction-based multirobot task allocation schemes, allocations of one/subset of tasks of the overall mission are detailed. However, some information regarding when task announcements and reassignments are made is usually not reported. DEMiR-CF allows for multiple auctioneers and winners for different tasks, depending on the optimization objective. In the case of the total path length minimization objective, ending one auction at a time results in better performance since the decision is made step by step. On the other hand, simultaneous auctions/executions are canceled only if there exist relations between the targets in consideration for the time minimization objective.

Algorithm 1 MTRP-FormRoughSchedule for r_j

```
input: T_U
output: T_Rj and t_s
T_Rj = φ (a heap with task cost as the key)
t_s = φ
while T_U is not empty do
  if t_i is in the rough schedule region - Equation (1) then
    c_ji = evaluateCost(t_i ∈ T_U) (*)
    insert t_i into T_Rj
  end if
end while
if ||T_Rj|| > 0 then
  t_s = top(T_Rj)
end if
* c_ji can be evaluated by using either Equation (2) or (3)
```

Algorithm 2 MTRP-DPTSS algorithm for robot r_j

```
input: T_U
output: An action to be performed
[T_Rj, t_s] = MTRP-FormRoughSchedule(T_U)
if t_s ≠ φ then
  if t_s is the current task then
    continue with current execution
  else
    if t_s is an available task then
      offer an auction to announce intention on execution
    else
      begin executing the awarded task
    end if
  end if
else
  stay idle
end if
```

B. Cost Estimation and Evaluation

Although TSP problem is NP-hard, there are many efficient heuristic methods in literature generating k-OPT solutions [2].

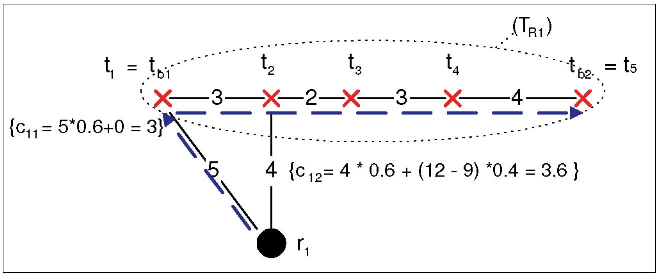

Fig. 3. Target selection strategy by the FAC heuristic function. The dashed arrows present the route generated for r_1 by employing FAC. This heuristic forwards r_1 to t_1 although t_2 is closer to itself, resulting in a better route.

Based on the analysis of earlier methods, two heuristic cost functions are designed [17] to be evaluated for the targets in T_R for each robot. These heuristic cost functions, namely, closest cost (CC) and farthest addition cost (FAC) are integrated into the proposed framework. The CC heuristic cost value for robot r_j and target t_i is evaluated by (2). This heuristic cost function considers only the distance between targets in T_{Rj} and the robot r_j

$$c_{ji} = \text{dist}(r_j, t_i)\ t_i \in T_{Rj}. \tag{2}$$

The FAC heuristic function is designed to consider costs by applying a penalty for not visiting the boundary targets in T_{Rj} as in (3). Boundary targets, t_{b1} and t_{b2}, are the targets in T_{Rj} with the maximum distance value. The FAC heuristic forwards robots to these targets in T_{Rj} to some degree. The main motivation behind this approach is that the open-loop traversal should contain both t_{b1} and t_{b2}. If the robot heads toward one of these targets, if profitable (α), this maximum distance can be traversed by traversing other targets on the path. A sample illustration of this cost function is given in Fig. 3. In this figure, although t_2 is closer to r_1 than t_1, with the FAC heuristic applied, t_1's cost value is smaller than that of t_2 ($3 < 3.6$), hence forcing the robot to t_1, the farther target, yet resulting in a better route shown by the dashed arrows. After an empirical analysis of the various values for α, the best results have been observed for a value of 0.6

$$\begin{aligned} c_{ji} = {} & \alpha * \text{dist}(r_j, t_i) + (1-\alpha) * [\text{dist}(t_{b1}, t_{b2}) \\ & - \max(\text{dist}(t_i, t_{b1}), \text{dist}(t_i, t_{b2})] \\ & \{\text{dist}(t_{b1}, t_{b2}) = \max(\text{dist}(t_k, t_l))\ t_{i,k,l} \in T_{Rj}\}. \end{aligned} \tag{3}$$

C. System-Wide Contingency Handling Mechanism

In DEMiR-CF, information is not assumed to be complete. However, the framework can take advantage of communication when it is reliable. The consistency related with task states is ensured by the precautions taken in a completely distributed manner. *The precaution routines* are embedded into the framework to enable the system to react dynamically to various failure modes and to recover from them. The current implementation uses explicit communication to detect conflicts and contingencies. However, failures in communication can also be handled by the precaution routines. Each robot keeps the models of the system tasks and robots in its knowledge base. Models of different robots may become inconsistent because of uncertainties, incomplete knowledge, faulty assumptions, etc. It is not always possible to share common world knowledge in decentralized systems as in the case presented here. Related to the contingent situations, appropriate precaution routines are activated to either correct the models, or to initiate recoveries. If robots can observe each other implicitly, model updates can be implemented in a similar manner. Recovery operations may include warning other robots about the problem or changing the model accordingly. Inconsistencies usually arise in real-world operations when robots are not informed about tasks that are achieved, and that are still under execution or under auction. To maintain system consistency, robots use explicit communication and broadcast the following information:

1) tasks known to have been achieved in predefined time periods to prevent redundant executions; this feature provides a bucket-brigade type of information sharing, which handles communication range limitations;
2) newly discovered online tasks that are not yet achieved;
3) task execution message containing an updated cost value and estimated task achievement deadline information acting as clues for the executer robot to be still alive and the task being under execution;
4) task achievement message when a task is over;
5) cancellation message if task execution is canceled;
6) task invalidation message when an invalid situation is detected.

Details of the designed precautions are given in [1]. Most of the contingencies are detected by checking models, and then, corresponding model updates are carried out. One standard way of detecting robot failures is sending heart-beat signals. However, in DEMiR-CF, messages from other robots are taken as clues for their being in a running state. Some misleading actions such as setting the state of a robot as *failed* although it is running properly may result in parallel executions. This is a desired feature for the mission completion point of view. Designed precautions resolve these kinds of inconsistencies if communication resources permit in later steps.

D. Computational Analysis of the Approach

DEMiR-CF for MTRP offers a polynomial time solution. Sorting the distances to find the boundary targets takes $O(n\log(n))$ for all n number of tasks. Cost and queue initialization is implemented each in $O(n)$. Top element selection and deletion is performed in $O(\log n)$. Therefore, the total complexity is bounded by $O(n\log(n))$. In the worst case, where the environment is dynamic and cost values change frequently in the order of $O(l)$, the total complexity becomes $O(ln\ \log(n))$ for each robot.

V. Experiments

MTRP experiments are designed in three sets. In the first set, the performance of the proposed heuristic cost function for MTRP is analyzed. In the second set of experiments, both

(a) (b)

Fig. 4. (a) Khepera II robot base. (b) Khepera II radio module.

DEMiR-CF and the Prim allocation method are evaluated to measure their solution quality. The optimal solutions are generated on CPLEX by using the IP formulation given in [12]. In simulation experiments, the environment is represented as a grid with 100 $\times$ 100 nodes; the number of robots change in the range 1–50 and the number of targets in 10–50. Task allocation procedures of the approaches are applied on distance matrices. The third set of experiments is performed on real robots, namely Khepera II. Khepera II [Fig. 4(a)] is a differential wheel robot equipped with a 25 MHz MC68331 micro-controller, 512 K Flash and 512 K RAM memories and eight IR proximity sensors with limited obstacle detection range [18].

The multirobot platform involves a team of Khepera II robots. Self-controlled Khepera II robots coordinate their actions through the algorithms offered by DEMiR-CF. The BIOS of Khepera II is designed to provide both low- and high-level tasks running in parallel, eventually communicating together and using shared resources. Each robot executes multithreaded controller software that performs task selection/allocation algorithms, along with contingency handling mechanisms to achieve the functionality of DEMiR-CF. Different modules on the task allocation layer are integrated with the low-level sensory interface, motor interface, motion model, and mapping modules in the multithreading structure.

Communication among robots is achieved through wireless links by using radio modems [Fig. 4(b)] mounted on Khepera II robots. These modules have their own local processors (M68331) for the management of the emission and reception procedures at 418 MHz radio frequency and 9600 b/s transmission speed. The radio channel is a half-duplex channel, and the messages transmitted through the channel are encapsulated along with the information on the type of the message, the sender ID, the destination ID, the length of data, and a checksum for error correction.

A. Experiment 1: Evaluation of the FAC Function

In the first set of experiments, the performance of the FAC heuristic is analyzed for the known TSP instances [19]. The results are given in Table I as cost of visiting all nodes by a single robot. These results reveal the near-optimal performance of FAC heuristic function with at most 15.24% deviation from the optimum (for a large TSP instance). Note that, these results are for the open-loop TSP and present solutions without any additional improvements applied on them. Open-loop routes generated by the FAC heuristic function compared to the optimum are given in [1]. Although it is arguable that better heuristic cost functions would be designed, the proposed heuristic cost functions together with the incremental assignment procedures of DEMiR-CF ensure close to optimal results efficiently in terms of both computation and memory requirements. This also ensures DEMiR-CF to perfectly fit in robots with limited computational capabilities.

TABLE I
FAC PERFORMANCE RESULTS FOR THE TSP INSTANCES

TSP Instance	FAC	Optimum	Deviation from the Optimum(%)
ATT48 - 48 nodes	33537.83	31470.4	6.5
EIL51 - 51 nodes	444.01	413.51	7.37
BERLIN52 - 52 nodes	8104.99	7305.38	10.94
EIL101 - 101 nodes	725.31	629.38	15.24

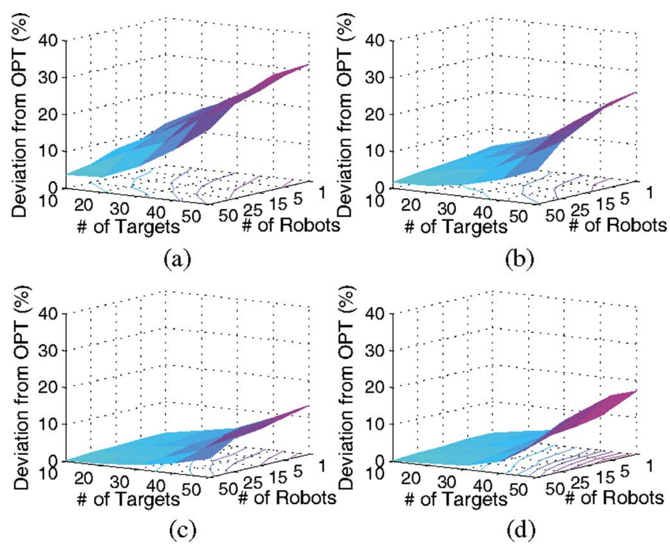

Fig. 5. Performance results of the heuristic approaches are illustrated compared to the optimal results. (a) Prim allocation method with in its original form (PRIM-ORG). (b) Prim allocation method with the additional improvements (PRIM-SD). (c) DEMiR-CF using FAC (DEMiR-CF + FAC). (d) DEMiR-CF using CC. (DEMiR-CF + CC).

B. Experiment 2: Overall Performance of DEMiR-CF

The objective of the second set of experiments is to compare task allocation performance of DEMiR-CF with that of the Prim allocation method and the IP approach. Two instances of DEMiR-CF are considered; one implementing the CC heuristic and the other implementing the FAC heuristic. Experiments are conducted in simulations for randomly generated test sets with different numbers of robots and targets. The overall performance results are given in Fig. 5 as deviations from the optima with standard deviation, averaged over 100 runs. Results are presented in terms of the target allocation strategies of the two heuristic methods compared to the optima. PRIM-ORG values represent results of the Prim allocation method in its original form ignoring subtree sizes on the traversal, while PRIM-SD values represent the results with shortest subtree selection improvement. Results of DEMiR-CF with the FAC heuristic are promising even for single-robot instances. The target allocation strategy has significant impact on the solution quality for increasing number of robots. Therefore, results of DEMiR-CF with both the CC and the FAC heuristics become closer to the optima with a very small value of deviation. The decrease in

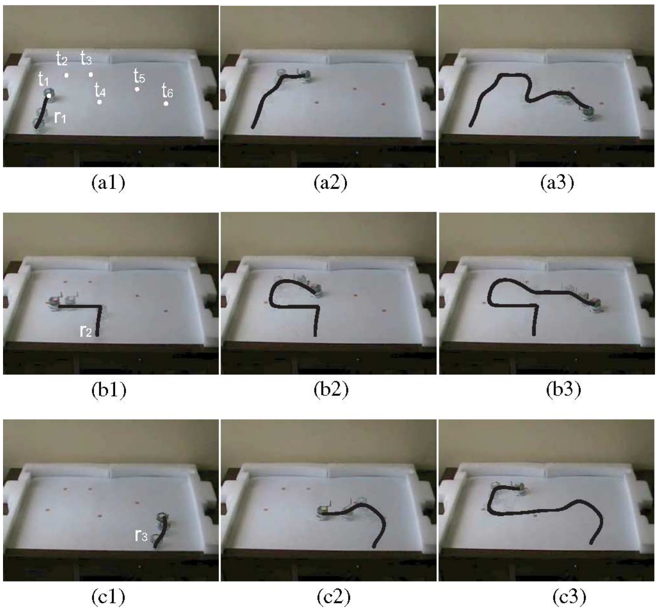

Fig. 6. Routes by DEMiR-CF corresponding to different initial deployment locations are illustrated for a single-robot six-target scenario.

target/robot proportion results in a decrease in the deviations of the results from the optima. However, results obtained for the Prim allocation method still deviate from the optima because of the allocation method. This is prevented in DEMiR-CF by the dynamic selection of T_Rs and integrating target allocation into route construction.

C. Experiment 3: Real-World Experiments

In the last set of experiments, the system is ported to Khepera II robots, and the contingency handling performance is analyzed. Each robot is initially informed about the target locations to be visited before they are deployed in the environment to run autonomously and to complete the mission cooperatively.

Depending on the initial location of the robot, the path constructed to traverse the targets differs for the single-robot case. This is illustrated in Fig. 6 for a six-target case. In this figure, each row [(a)–(c)] represents the illustration of an independent run with a different initial deployment location for robots r_1, r_2, and r_3. The continuous video frames [3] are divided into episodes in which the images of the videos are overlapped and the final overlapped image is illustrated (e.g., [(a1)–(a3)] in the figure.

Fig. 7 illustrates a successful scenario (scenario 1) with three robots for the same setting of the target set, where the objective is time minimization. The target pairs t_1–t_2, t_3–t_4, and t_5–t_6 are visited by robots r_1, r_2, and r_3, respectively.

The failure of r_3 and the rest of the run are illustrated in Fig. 8 (scenario 2). In this scenario, r_3 fails after completing its assigned task (t_6). The failure of the robot is enforced by the human agent isolating the related robot. At the time of the failure, the other robots are busy with their own target visiting tasks. The failure of r_3 does not block the execution. Since the allocations are performed incrementally, t_5 that is assigned to r_3 in scenario 1 can no longer be allocated to this robot because of its failure

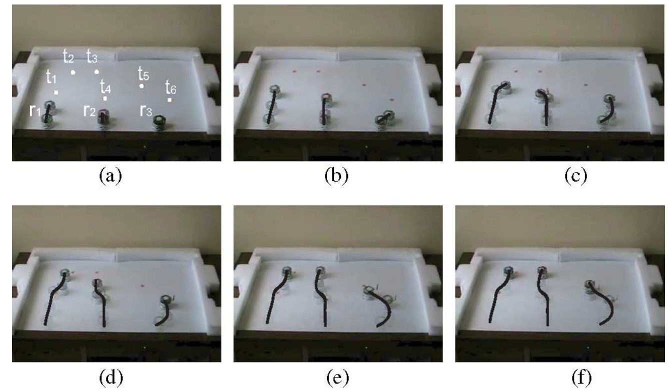

Fig. 7. Scenario 1: routes by DEMiR-CF in a three-robot six-target scenario.

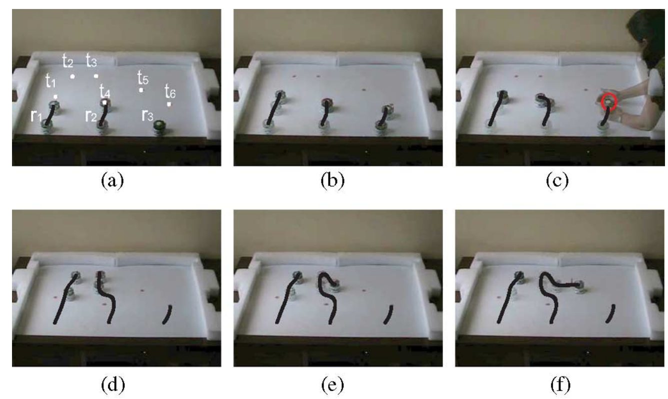

Fig. 8. Scenario 2: routes by DEMiR-CF in a three-robot six-target scenario including the failure of r_3 after completing its task.

immediately after achieving its first task. After the failure, r_2 selects t_5 as an available target and completes the mission. There is no failure detection and recovery in this scenario, but it reveals how redundant reallocation procedures are eliminated by the incremental task selection strategy of DEMiR-CF.

If the failure of r_3 occurs before completing its assigned task (t_6), then a recovery is needed (Fig. 9, scenario 3). By the *precaution routines*, DEMiR-CF can handle these types of contingency cases. After detecting that the state of the task (t_6) assigned to r_3 is not updated in the estimated time, the *Model Update Modules* of the other robots set the state of the task as *uncertain*, which, in this case, is treated as an available task. Eventually, r_2 contributes to the efficient completion of the mission by visiting targets t_5 and t_6 as well. The other failure scenarios are given in [1]. The videos of the presented scenarios are available online [3].

VI. Related Work and Discussion

Simplified or adopted versions of the MTRP problem are studied in earlier work, some of which investigate the problem focusing only on the task allocation dimension. The Prim allocation method [12] ensures a mechanism to allocate targets to robots based on Prim's algorithm [13], [14]. An overall allocation/reallocation scheme is applied as opposed to assigning tasks to robots incrementally as in the case of DEMiR-CF. This results

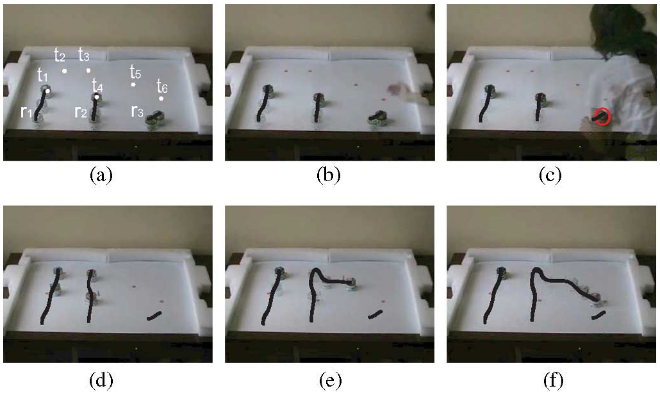

Fig. 9. Scenario 3: routes by DEMiR-CF in a three-robot six-target scenario including the failure of r_3 before completing its task. The failure is handled by the *precaution routines*. r_2 completes the mission by visiting t_5 and t_6, although the latter is assigned earlier to r_3.

in less efficient solutions from the point of view of task allocation performance. The Prim allocation method does not offer a contingency handling mechanism. Lemaire *et al.* propose a task allocation scheme for multi-unmanned air vehicle (UAV) cooperation with balanced workloads of robots [20]. Allocations are performed whenever the world knowledge of robots changes. A watch-out task is proposed to detect failures in communication. In GRAMMPS [21], one of the earliest works to solve MTRP, a mission planner works centrally either on one of the robots or as an operator to select a robot for each target. The system can regenerate plans when the environment changes. Although initial allocations may be suboptimal, the system can find close to optimal solutions in later steps by using simulated-annealing approach. In their latest work, Dias and Stentz propose a market-based scheme introducing the leader approach for combinatorial task exchanges [9]. These leaders are responsible for multiparty multitask optimizations for obtaining optimal results. A market mechanism with a combinatorial-auction- based allocation scheme [9], [10] may become intractable for real-world scenarios either with large instances or in dynamic settings in which calculations should be made frequently.

DEMiR-CF is different from earlier work as it ensures both instantaneous assignment procedures incrementally and forming rough schedules to consider the problem as a whole from global perspectives. Combinatorial task exchange mechanisms as in market-based approaches are not used in DEMiR-CF and the rough schedule generation process uses polynomial time. Auctions are used by robots to announce intentions about task execution and select appropriate task executers to deal with world information incompleteness. Contingency handling mechanisms are directly integrated into the dynamic task selection mechanisms, which, in turn, facilitate recovering from failures dynamically and efficiently, reconfiguring robots during runtime, and maintaining system consistency. These utilities are ensured by autonomous robots using DEMiR-CF in a completely distributed manner without central authorities and/or complete knowledge injected manually.

The proposed FAC heuristic function is evaluated for single-robot route construction in the first set of experiments. The performance of the FAC function is compared to the optimal results generated by using an Integer Programming formulation. It has been observed that the DEMiR-CF results generated with the use of the designed heuristic cost function deviate from the optimal solutions by at most 15.24% for a large TSP instance. Although it is arguable that better heuristic cost functions would be designed, the proposed heuristic cost functions together with the incremental assignment procedures of DEMiR-CF ensure close to optimal results efficiently in terms of both computation and memory requirements. This also ensures DEMiR-CF to perfectly fit in robots with limited computational capabilities. In the second set of experiments, the task allocation approach of DEMiR-CF is compared with both the Prim allocation method and the Integer Programming approach. As expected, both DEMiR-CF and the Prim allocation methods have tractable computational complexities compared to the Integer Programming approach. However, as the results reveal, DEMiR-CF integrated with the proposed heuristic cost functions produces results that are close to the optima for the multirobot case of the problem. The third set of the experiments validates robustness of DEMiR-CF in real-world scenarios which are prone to failures.

VII. Conclusion

The proposed solution for MTRP integrates dynamic and distributed task selection and allocation, simultaneous route construction, real-time contingency handling, and low-level hardware procedures to make robots contribute to the overall team objective. The results of the experiments support the claim that in real-world environments, an incremental task selection approach eliminates redundant efforts that are introduced by allocating all tasks from scratch if there is an unexpected change. The rough schedule generation scheme forms loose commitments, which, if needed, can be canceled in the future. Thus, it offers a way to reconsider the problem globally when it is appropriate. The approach is efficient with its polynomial computational complexity. Precaution routines ensure that the mission is successfully ended. If real resources permit, failures are handled to maintain system consistency. Empirical evaluations of the system are performed on real robots as well. Experiments reveal the success of the approach as a whole with its integrated components and its applicability on even very simple and small robots like Khepera II.

Acknowledgment

The authors would also like to thank M. Lagoudakis, S. Koenig, and P. Keskinocak for helpful discussions.

References

[1] S. Sariel, "An integrated planning, scheduling and execution framework for multi-robot cooperation and coordination," Ph.D. dissertation, Istanbul Technical Univ., Istanbul, Turkey, 2007.

[2] E. L. Lawler, J. K. Lenstra, R. Kan, and D. B. Shmoys, *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization*. New York: Wiley, 1985.

[3] (2008). *Khepera II Real-World MTRP Videos* [Online]. Available: http://www2.itu.edu.tr/~sariel/videos/KheperaII-Movies.html

[4] X. Yuan and S. X. Yang, "Multirobot-based nanoassembly planning with automated path generation," *IEEE/ASME Trans. Mechatronics*, vol. 12, no. 3, pp. 352–356, Jun. 2007.

[5] W. Burgard, M. Moors, C. Stachniss, and F. E. Schneider, "Coordinated multi-robot exploration," *IEEE Trans. Robot. Autom.*, vol. 21, no. 3, pp. 376–386, Jun. 2005.

[6] S. Sariel, T. Balch, and N. Erdogan, "Naval mine countermeasure missions: A distributed, incremental multirobot task selection scheme," *IEEE Robot. Autom. Mag.*, vol. 15, no. 1, pp. 45–52, Mar. 2008.

[7] P. Toth and D. Vigo, *The Vehicle Routing Problem*. Philadelphia, PA: SIAM, 2001.

[8] S. Sariel and T. Balch, "Efficient bids on task allocation for multi-robot exploration," in *Proc. 19th Int. Florida Artif. Intell. Res. Soc. (FLAIRS) Conf.*, 2006, pp. 116–121.

[9] M. B. Dias and A. Stentz, "Opportunistic optimization for market-based multirobot control," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2002, pp. 2714–2720.

[10] M. Berhault, H. Huang, P. Keskinocak, W. Elmaghraby, P. Griffin, and A. Kleywegt, "Robot exploration with combinatorial auctions," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2003, pp. 1957–1962.

[11] M. G. Lagoudakis, E. Markakis, D. Kempe, P. Keskinocak, A. Kleywegt, S. Koenig, C. Tovey, A. Meyerson, and S. Jain, "Auction-based multi-robot routing," in *Proc. Int. Conf. Robot. Sci. Syst. (RSS)*, 2005, pp. 343–350.

[12] M. G. Lagoudakis, M. Berhault, S. Koenig, P. Keskinocak, and A. Kleywegt, "Simple auctions with performance guarantees for multi-robot task allocation," in *Proc. IEEE/RSJ Intl. Conf. Intell. Robots Syst. (IROS)*, 2004, pp. 698–705.

[13] V. Jarnik, "O jistem problemu minimalnim (about a certain minimal problem)," *Prace Moravske Prirodovedecke Spolecnosti*, vol. 6, pp. 57–63, 1930.

[14] R. C. Prim, "Shortest connection networks and some generalisations," *Bell Syst. Tech. J.*, vol. 36, pp. 1389–1401, 1957.

[15] G. Reinelt, *The Traveling Salesman: Computational Solutions for TSP Applications*. New York: Springer-Verlag, 1994.

[16] R. G. Smith, "The contract net protocol: High level communication and control in a distributed problem solver," *IEEE Trans. Comput.*, vol. C-29, no. 12, pp. 1104–1113, Dec. 1980.

[17] S. Sariel and T. Balch, "Real time auction based allocation of tasks for multi-robot exploration problem in dynamic environments," in *Proc. Integrating Planning Scheduling: Papers 2005 AAAI Workshop*, pp. 27–33, Paper WS-05-06.

[18] *Khepera II Programming Manual*, K-Team, 2008.

[19] G. Reinelt, "Tsplib—A traveling salesman problem library," *ORSA J. Comput. 3*, vol. 3, pp. 376–384, 1991.

[20] T. Lemaire, R. Alami, and S. Lacroix, "A distributed task allocation scheme in multi-UAV context," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2004, pp. 3622–3627.

[21] B. L. Brumitt and A. Stentz, "Grammps: A generalized mission planner for multiple robots in unstructured environments," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 1998, pp. 1564–1571.

Sanem Sariel-Talay (M'06) received the B.S., M.Sc., and Ph.D. degrees in computer engineering from Istanbul Technical University (ITU), Istanbul, Turkey, in 1999, 2002, and 2007, respectively.

During her Ph.D. studies, she worked as a Researcher at Georgia Institute of Technology, Atlanta from 2004 to 2006. She is currently an Assistant Professor at ITU. Her research interests include distributed problem solving, multirobot systems, and intelligent agents.

Tucker R. Balch received the B.S. and Ph.D. degrees in computer science from Georgia Institute of Technology (Georgia Tech), Atlanta, in 1984 and 1998, respectively.

He is currently an Associate Professor in the School of Interactive Computing, Georgia Tech, where he is also the Director of the Institute for Personal Robots in Education (IPRE), where he is engaged in understanding how robots can be used to help students learn more effectively. His current robotics research interests include machine learning for robot navigation and large-scale multirobot systems. He has authored or coauthored more than 80 technical articles and two books.

Nadia Erdogan received the B.S. degree in electrical engineering and the M.Sc. degree in computer science from Bosphorus University, Istanbul, Turkey, and the Ph.D. degree from Istanbul Technical University, Istanbul, in 1978, 1980, and 1987, respectively.

She is currently a Professor in the Computer Engineering Department, Istanbul Technical University. Her current research interests include distributed computing and execution environments, mobile agent systems, multiagent systems, and parallel programming.