A Distributed Multi-Robot Cooperation Framework for Real Time Task Achievement

Sanem Sariel[1*] and Tucker Balch[2]

[1] Istanbul Technical University, Department of Computer Engineering, Istanbul, 34496, TURKEY sariel@cs.itu.edu.tr
[2] Georgia Institute of Technology, College of Computing Department, Atlanta, GA, 30332, USA tucker.balch@cc.gatech.edu

Summary. In this paper, we propose a general framework, DEMiR-CF, for a multi-robot team to achieve a complex mission including inter-related tasks that require diverse capabilities and/or simultaneous executions. Our framework integrates a distributed task allocation scheme, cooperation mechanisms and precaution routines for multi-robot team execution. Its performance has been demonstrated in Naval Mine Countermeasures, Multi-robot Multi-Target Exploration and Object Construction domains. The framework not only ensures near-optimal solutions for task achievement but also efficiently responds to real time contingencies.

1 Introduction

In this paper, we present a generic framework, **D**istributed and **E**fficient **M**ult**i** **R**obot - **C**ooperation **F**ramework (DEMiR-CF), designed for efficient mission achievement of a multi-robot team for inter-related tasks that require diverse capabilities and simultaneous executions. Since real world applications present additional challenges than software platforms, robustness is a key issue of a multi-robot cooperation/coordination framework. DEMiR-CF with its integrated structure can respond to several real time contingencies efficiently while dynamically maintaining high solution quality in a fully distributed fashion. In this paper, we present the generic architecture of our framework even suitable for complex mission execution in environments with failure potentialities and limited communication features (such as bandwidth limitations, limited ranges, or unexpected delays). We report the experimental results and several scenarios in the context of Naval Mine Countermeasures mission for multi-AUV coordination in [12]; an extended technical report for the framework and evaluations is provided in [13].

* Sanem Sariel is also affiliated with Georgia Institute of Technology

2 Background and Related Work

Multi-robot coordination has been an active and attractive field during the last decade because of the demand for multiple robot, UAV, UGV or rover missions, especially in military and space applications. Among different approaches, the centralized approach is not robust especially when communication is limited between operator and individual robots, and failures are highly probable. Therefore our focus is on distributed coordination frameworks in this work, and we will review literature on this subject. Parker presents one of the earlier works for distributed multi-robot task allocation, ALLIANCE, with a behavior based framework [9] for instantaneous task assignment. M+ [1] is a distributed task allocation and achievement scheme for multi-robot cooperation addressing many real time issues including plan merging paradigms. MURDOCH [4] is a framework achieving publisher/subscriber type allocation for instantaneous assignment. Dias et al. proposes a combinatorial auction based task allocation scheme: TraderBots [2]. Lemarie et al. proposes a task allocation scheme for multi-UAV cooperation with balanced workloads of robots [8]. According to [3], existing market mechanisms are not fully capable of re-planning task distributions, re-decomposing tasks, re-scheduling commitments, and re-planning coordination during execution. We would like to fill these gaps by our integrated cooperation framework. Our primary contribution in this work is the presentation of an integrated cooperation framework for a multi-robot team and the extensive design of precaution routines and solution quality maintenance schemes for single-item auctions in real time task execution. DEMiR-CF can address different types of domains, and can generate near optimal solutions even for NP-Hard problems [11] with efficient bid evaluation methods. From our point of view, task allocation, execution and contingency handling should be integrated into the cooperation framework without assuming they are achieved separately, if globally optimal solutions are desired. This is the main rationale behind our framework.

3 Distributed and Efficient Multi Robot - Cooperation Framework

DEMiR-CF is designed for complex missions including inter-related tasks that require diverse (heterogeneous) capabilities and simultaneous execution. The framework combines *distributed task allocation and coalition formation schemes* and *dynamic task selection scheme* as cooperation components and *Plan B precaution routines* some of which are implemented by *coalition maintenance/dynamic task switching scheme.* These components are integrated into one framework to provide an overall system that finds near-optimal solutions for real time task execution.

The overall objective of the robot team ($r_j \in R$, $0 < j \leq ||R||$) equipped with our framework is to achieve a mission (M) consisting of interrelated tasks

T_i $(0 < i \leq ||M||)$, by incremental assignment of all $T_i \in M$ to $r_j \in R$ while optimizing the specified objective function. Coalitions (C_i) [6] are formed to meet requirements of simultaneous executions of tasks (T_i) synchronously by a group of robots. Sizes of coalitions vary according to the required minimum number of robots ($reqno_i$) to execute the tasks.

Definition 1. *(executable task): T_i is an executable task, if at least $reqno_i$ number of robots can be assigned for execution.*

An example of such a task may be pushing a heavy object requiring more than one robot. Tasks are preemptive: the activity of task execution can be split during runtime if another advantageous situation arises or environmental conditions impel.

Definition 2. *(candidate task and suitable robot) T_i is a candidate task for the robot r_j if the $reqcap_i$ is a subset of the cap_j and the precedence constraints of the task are satisfied; r_j is a suitable robot for the task T_i.*

Coalitions are formed by *suitable robots*. For the robots to be prepared for the contingencies, models of the system tasks and other robots are kept in each robot's world knowledge as corresponding FSMs. Task states are: *free*, *auctioned*, *being executed*, *achieved*, *uncertain* (interpreted as state *free*) and *invalid*. Robot states are: *idle*, *executing*, *failed* and *auctioneer*. The state transitions of FSMs are activated by either own motivations or incoming information from other robots. *Model Update Module* is responsible for checking and updating robot's own models. All modules in the framework and information flow among them are given in Figure 1. *Model Update*, *(System) Consistency Checking and Dynamic Task Selector* modules perform *Plan B precaution routines*. *Allocation scheme* ensures distributed task allocation. *Coalition scheme* implements synchronized task execution and *coalition maintenance* procedures. A sample flow of the operations in the framework is summarized as:

1. Mission task definitions are given to the robots (time-extended representation of tasks with precedence constrains to achieve overall mission).
2. Each robot selects the most suitable candidate task to execute by global cost consideration among mission tasks (dynamic task selection/switching).
3. Corresponding robots offer auctions for the selected tasks. In auctions, inconsistencies and conflicts are resolved.
4. Coalitions are formed for the announced tasks making sure that each robot is in the most suitable coalition from global solution quality point of view.
5. During task execution, simultaneously, dynamic task selecting/switching mechanism ensures to switch between tasks, if it is profitable; real time contingencies are handled. Then corresponding auction and coalition formation procedures (2-4) are applied continually.

Real time situations in which task switching is necessary are given in the next section.

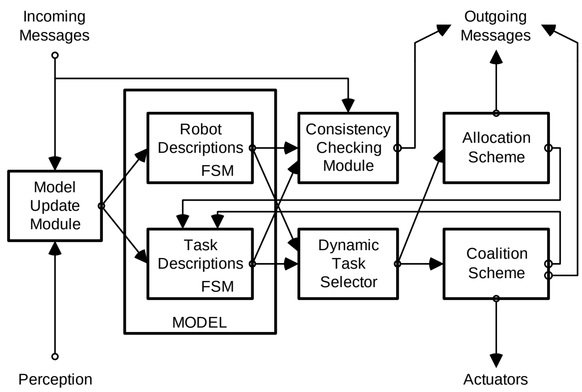

Fig. 1. DEMiR-CF Modules

3.1 Real Time Issues and Requirements

Since the world is beyond the control of the robots and change continuously in real world applications, the difficulty of multi-robot task execution problem goes beyond the task allocation problem. In particular, multi-robot systems deal with difficulties arising from noisy sensor information, unexpected outcomes of actions, environmental limitations (especially in communication) and presence of failures of hardware. All these factors may affect the overall solution. We list evolving circumstances that may change the solution as:

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

Some of these situations may arise after either internal or external events. Given these contingencies, even solution of an approach capable of finding optimal solutions may become sub-optimal under uncertainties of real world applications. Verification of the solution optimality is a difficult issue for real world applications. Therefore in the last decade researchers proposed effective approaches, opportunistic methods without giving boundaries on the overall solution quality except [7]. However, their work assumes perfect communication and contingencies are not considered in the boundaries. For now, these boundaries are given for the complete information cases.

DEMiR-CF is designed as being capable of dealing with the situations presented above. The framework can efficiently respond to these events and solution quality is maintained simultaneously with real time task execution.

3.2 Task Representation

Tasks are represented by a data structure containing information regarding the task execution requirements and the task status. Tasks are represented as septuples $< id, type, reqcap, deplist, reqno, relinfo, precinfo >$. System generated task ids are generated initially before mission execution and common for all robots. However online task ids may be different for each robot. Robots have initial knowledge about task types (*type*) and corresponding execution methods before mission execution. Requirements (*reqcap*) define special sensors and capabilities required to execute the task. Dependencies (*deplist*) are represented with hard and soft dependent task ids. We define two types of dependencies for representing precedence relations. Hard dependency implies sequential execution while soft dependency allows parallel execution [10]. Minimum number of robots to execute the task (*reqno*) is determined. Related information ($relinfo$) represents information regarding the type such as latest location, target location, etc. Precaution information ($precinfo$) is used for contingency handling: task state, estimated task achievement time and current execution cost.

The mission is defined as an acyclic graph (not necessarily a connected graph) of inter-related tasks connected by arcs representing dependencies. An example graph representation for the object construction mission can be found in [10]. Task definitions can be changed during execution. In particular, $relinfo$, $precinfo$ and *reqno* are subject to change during execution.

3.3 Distributed Task Allocation Scheme

Task allocation and initial assignments may be carried out by using operations research methods. However, our research addresses issues of real time execution when managing the overall team by a central authority is not possible due to several real world limitations. Auction based task allocation approach is suitable to provide a scalable and efficient way of distributing tasks. Contract Net Protocol (CNP) [14] is used to select task executers. Although CNP presents the formalism on relationships between managers and contractors, it does not present details for the following questions: When should task announcements be made? How should bid values be evaluated to get globally optimal solutions? Which subset (or all) of the already allocated tasks should be re-auctioned to maintain solution quality? When should reallocations be implemented and who decides on them? Most auction based task allocation schemes offer solutions for allocating one/subset of tasks of the overall mission. However there is usually little information about when task announcements and reassignments are made. In our framework, any robot becomes an auctioneer when it intends to execute a task. Each robot selects best suitable candidate task among mission tasks by the *dynamic task selection scheme.* Basically, auction announcements are ways to illustrate intentions to execute tasks for which $reqno = 1$ or to select members of coalitions to execute tasks for which $reqno > 1$. Therefore, if more than one robot intends to execute the same task, more suitable one(s) is selected in the auction by considering cost values. Single items are auctioned and allocated in auctions. Auction negotiation implemented in the framework consists of standard steps to clear an auction. Robots can get the necessary task details from the auction offers, and then check the validity of the auction. If the auction is invalid, related precaution routines are activated. Otherwise, the candidate robot sends its cost value as a bid. The other candidate robots behave simultaneously as well. If the auctioneer cannot get the required number ($reqno$) of bids (also counting in own bid) from the other robots until the predefined deadline, it cancels the auction. Otherwise it ranks all bids and assigns the best suitable robot with the lowest cost value to the executable task (if $reqno = 1$), or suitable coalition members (if $reqno > 1$). The framework allows multiple auctions and winners for different tasks at the same time.

3.4 Dynamic Task Selection Scheme and Online Scheduling

Dynamic task selection is implemented by forming a priority queue of unachieved candidate tasks and selecting the task with the lowest cost. Priority queue is formed either by costs for executing these tasks or by considering rough schedules depending on the selected domain. If costs are the same, the priorities are considered in the given order: Robot's current task (if any), tasks already being executed, tasks awarded in auctions, and free tasks. Dynamic task switching mechanism is used by robots to switch between tasks if updates in the world knowledge compel. Therefore issues related to both online scheduling and scheduling under uncertainty are addressed.

3.4.1 Coalition Maintenance/Dynamic Task Switching Scheme

In the framework, instead of using complicated re-allocation procedures, we propose incremental selection and task switching schemes for behaving myopically while thinking globally using bid evaluation heuristics. Provided with an efficient bid evaluation heuristic, dynamic task selection scheme ensures task switching whenever it is profitable. Each robot, independent from executing a task or not, can offer another auction or select to execute a task already being executed by another robot with a worse cost value than that it will cost for itself. If task switching occurs with a coalition member, the corresponding coalition member is released from the coalition becoming a suitable robot for other tasks. When robots participate in coalitions, they are only allowed to select other tasks, when they are released from these coalitions controlled by a robot in the coalition.

3.5 Bid/Cost Evaluation

The impact of bid (cost) evaluation on the solution quality is inevitable for auction based systems, and research in this area desires more investigation. According to the taxonomy given in [5], multi-robot task allocation problems are divided into two classes based on the mission description: instantaneous vs. time-extended. Most multi-robot architectures offer solutions for instantaneous assignments. DEMiR-CF can address both types of classes by implementing incremental allocation of tasks with efficient bidding strategies. Therefore global solution quality is maintained for the mission tasks from time-extended view of the problem by means of the bid considerations. However, the approach is capable itself to offer solutions for instantaneous changes on the task description. Therefore we classify our framework capable of addressing both types of problem classes. Unless efficient bid evaluation strategies are designed, it is not possible to observe globally optimal solutions for NP-Hard problems, and additional adjustments are required to change allocations with an additional cost of communication as in combinatorial auctions. In our earlier work, we have shown that by efficient bid evaluation approach, globally near optimal solutions can be observed for auction based approach. In that work, we analyze performance of different heuristic cost functions combined with our framework for multi-robot multi-target exploration domain [11]. Incremental assignments eliminate redundant considerations for environments in which the best solution is highly probable to change, and efficient bidding strategies ensure solutions to be close to optimal with a time-extended view of the problem. Although we have shown that our approach can find near-optimal solutions for multi-robot multi-target exploration problem, we still need further investigation on bidding strategies for different domains.

3.6 Models for Contingency Handling and Plan B Precautions

In DEMiR-CF, information is not assumed to be complete. Therefore Plan B Precaution routines are embedded in the framework to enable the system to dynamically respond to various failure modes and recover from them. These precautions are taken by each robot in a distributed fashion. Current implementation use explicit communication to detect conflicts and contingencies. However failures in communication can also be handled by precaution routines. (If robots can observe each other implicitly, model updates can be implemented in a similar manner.) Related to the contingent situations, appropriate precaution routines are activated to either correct the models, or initiate a recovery. Recovery operations may include warning other robots about the problem or changing the model accordingly. These inconsistencies usually arise when robots are not informed about tasks that are achieved, under execution or under auction. To keep system consistency, robots broadcast:

- known achieved tasks in predefined time periods to prevent redundant executions. (This feature provides a bucket-brigade type of information sharing handling communication range limitations.)
- new discovered online tasks which are unachieved yet.
- task execution messages containing the updated cost value and estimated task achievement deadline information in predefined time periods as clues for the executer robot is still alive and the task is under execution.
- task achievement message when the task is achieved.
- cancellation message if task execution is cancelled.
- task invalidation message when invalidities are detected.

Precaution routines are given in Table 1. Most of the contingencies are detected by checking models, and model updates are implemented (Table 2-3). One standard way of detection of robot failures is sending heart-beat signals. However in our framework, incoming messages from other robots are taken as clues for running properly. More complicated prediction models may be used for more accurate failure prediction. Some misleading beliefs such as setting state of a robot as *failed* although it is running properly may cause 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. In the design of precautions, it is assumed that robots are trusted and benevolent.

Table 1. Precautions for Contingencies and Conflicts

Contingency or Conflict by inconsistencies	**Precaution**
Any message from an unrecognized system robot is received.	Robot model is created with the corresponding state derived from the message.
Any message related to an unrecognized task is received.	Task is added to the task list with the corresponding state.
An already achieved task is announced as a new task/being executed/cancelled/auctioned.	Warning message is sent to the sender.
A task being executed/auctioned is announced as being executed/auctioned.	Only the robot with the minimum cost continues to the operation.
Cancellation message is received for a task already being executed by own.	Robot state is set “idle”.
A cancellation is message is received for a task being executed by the sender robot.	Task and robot states are set as “free” and “idle”, respectively.

Table 2. Model Checking for Tasks and System Robots

Status	**Action**
The time duration from the latest communication with a robot is longer than the threshold.	Robot state is set as "failed". Related task state is set as "uncertain".
Task in execution is not achieved although the estimated deadline is reached.	Task state is set as "uncertain".
Task state is "auctioned" for longer than predefined time period.	The task state is set as "uncertain".

Table 3. Model Updates Related to The Messages

Message Type	**Action**
Any type	Current time is registered as the latest comm. time with the robot and for the task.
"achieved" - valid	The robot and task states are set as "idle" and "achieved", respectively. If the task is in consideration (in schedule or in execution), it is cancelled.
"execution" - valid	If there are other tasks with state "being executed by this robot", states are changed as "uncertain".

4 Evaluation of DEMiR-CF

Our framework is evaluated in three different domains: Object construction [10], Multi-robot multi-target exploration [11] and Multi-AUV Naval Mine Countermeasures domains [12]. First two evaluations are implemented on an abstract simulator, while the third one is on the realistic US NAVY simulator. Readers are referred to the corresponding papers and to the extended technical report [13] for the performance evaluations of the framework. From general point of view, there is a tradeoff between maintaining high solution quality and the increasing communication and continuous bid evaluation requirements. According to the metric defined in [5], computational requirements per task is $O(1)/bidder$ and $O(m)/auctioneer$, when the number of robots is m. Before selecting the best bidder, each robot selects the best suitable task for itself and offers an auction for the task from time-extended point of view of the mission. Therefore among n tasks, assuming bid evaluation complexity is $O(l)$ for each task, the selection is implemented in $O(nl)$. If $nl \gg m$, this bound is given as $O(nl)/task$. To ensure system solution quality, robots continuously evaluate bids for the unachieved tasks and dynamically switch among tasks, if it is profitable, one of the differences of our framework from others, in each time step (the worst case). For standard task allocation, communication complexity is $O(nm)$ under normal circumstances. To maintain high solution quality, precaution messages are sent. However, still the complexity is given with the same bound (multiplied by an additional scalar). The performance of DEMiR-CF is shown to be bounded by 2*OPT for the Multi-robot multi-target exploration domain [11].

5 Conclusions

In this work, we present our generic cooperation framework, DEMiR-CF, for multi-robot teams. The framework combines a distributed auction based allocation method and several precaution routines to handle contingencies and communication limitations of real world domains and to maintain high solution quality with available resources. DEMiR-CF is evaluated for different domains. Near future work include further evaluations of the framework for different complex domains and specifying design issues for these domains.

References

1. Botelho SC and Alami R (1999) M+: a scheme for multi-robot cooperation through negotiated task allocation and achievement. IEEE Intl. Conf. on Robotics and Automation
2. Dias MB, and Stentz A (2002) Opportunistic Optimization for Market-Based Multirobot Control. IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems
3. Dias MB, Zlot RM, Kalra N, and Stentz A (2005) Market-Based Multirobot Coordination: A Survey and Analysis, Robotics Institute. Carnegie Mellon University, Tech Report, CMU-RI-TR-05-13
4. Gerkey G and Matric MJ (2002) Sold!: Auction Methods for Multirobot Coordination. IEEE Trans. on Robotics and Automation, vol. 18 no.5, pp. 758-768
5. Gerkey B and Mataric MJ (2004) A Formal Analysis and Taxonomy of Task Allocation. Intl. Journal of Robotic Research, 23(9): 939-954
6. Horling B and Lesser V (2005) A Survey of Multi-Agent Organizational Paradigms. The Knowledge Engineering Review, 19(4):281-316
7. Lagoudakis MG, Berhault M, Koenig S, Keskinocak P and Kleywegt AJ (2004) Simple Auctions with Performance Guarantees for Multi-Robot Task Allocation. IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems
8. Lemarie T, Alami R, and Lacroix S (2004) A Distributed Task Allocation Scheme in Multi-UAV Context. IEEE Intl. Conf. on Robotics and Automation
9. Parker LE (1998) ALLIANCE: An Architecture for Fault Tolerant Multi-Robot Cooperation. IEEE Trans. on Robotics and Automation, 14(2): 220-240
10. Sariel S and Balch T (2005) Robust Multi-Robot Coordination in Noisy and Dangerous Environments. Georgia Institute of Technology, College of Computing, GVU, Tech Report GIT-GVU-05-17
11. Sariel S and Balch T (2006) Efficient Bids on Task Allocation for Multi-Robot Exploration Problem. The Nineteenth International FLAIRS Conference
12. Sariel S, Balch T and Stack JR (2006) Empirical Evaluation of Auction-Based Coordination of AUVs in a Realistic Simulated Mine Countermeasure Task. The 8th Intl. Symposium on Distributed Autonomous Robotic Systems (DARS)
13. Sariel S, Balch T and Stack JR (2006) Distributed Multi-AUV Coordination in Naval Mine Countermeasure Missions. Georgia Institute of Technology, College of Computing, GVU, Tech Report GIT-GVU-06-04
14. Smith RG (1980) The Contract Net Protocol: High level communication and control in a distributed problem solver. IEEE Trans. on Computers, C-29(12): 1104-1113