A Framework for Multi-Robot Coordination

Sanem Sariel

Istanbul Technical University, Department of Computer Engineering, Istanbul, TURKEY TR34496

sariel@cs.itu.edu.tr

Advisors: Tucker Balch (Georgia Institute of Technology), Nadia Erdogan (Istanbul Technical University)

Abstract

In this thesis work, a complete framework for multi-robot coordination in which robots collectively execute inter-dependent tasks of an overall complex mission requiring diverse capabilities is proposed. Given a heterogeneous team of robots and task dependencies, the proposed framework provides a distributed, robust mechanism for assigning robots to tasks in an order that efficiently completes the mission. The approach is robust to unreliable communication and robot failures. The framework is based on the market-based approach, and therefore scalable. In order to obtain optimum allocations in noisy environments, a coalition maintenance scheme ensuring dynamic reconfiguration is introduced. Additional routines, called precautions are added in the framework for addressing different types of failures common in robot systems and solving conflicts in cases of these failures. The final solutions are close to optimal with the available resources at hand by using appropriate cost functions. The framework has been tested in simulations that include variable message loss rates and robot failures. The experiments illustrate the effectiveness of the proposed system in realistic scenarios.

Introduction

In this thesis work, a coordination framework for multi-robot teams implementing complex missions of tasks having ordering constraints, requiring diverse capabilities and collective work is proposed. The main objective of this framework is dynamically allocating inter-dependent tasks to multi robots in a cost optimal manner without violating the constraints under uncertainties on dynamic environments. The approach used in this framework is a market based scheme supporting scalability. However the optimality of the generated schedules is not addressed in this scheme for changing situations. Robot failures or unreliable communication are such situations common in real world domains. The proposed framework presents different kind of recovery solutions to obtain optimum solutions for dynamic environments.

Recently proposed works somehow address the dynamic task allocation issue against robot failures (malfunction or death) or environmental changes. In these works, the selected test domain missions have usually independent sub-tasks. Closest works to the proposed framework, Zlot's (Zlot and Stentz 2005), Lemarie's (Lemarie, Alami and Lacroix 2004) and Kalra and Stentz's (Kalra, Ferguson, and Stentz 2005) works address task dependency issue for tightly coupled missions. In Kalra and Stentz's work, the implemented system was tested for the collective perimeter sweeping mission. The task dependencies are considered in keeping a formation while obeying some rules. The tasks do not require different capabilities. In Lemarie's work, the main goal is keeping the architecture distributed among the robots so as to gain scalability, robustness and reactivity. However the solution's optimality is not guaranteed. They also inspire from market based approach. Using a token-ring network approach, only one auctioneer is allowed to initiate an auction at a time step. Zlot's recently proposed task allocation scheme (Zlot and Stentz 2005) deals with task tree auctions to obtain globally optimum solutions. Complexity of the task tree auctions increases drastically for more complex task trees. This scheme generates a reconfiguration on the allocations from each robot's point of view. However when there are inter-dependencies among tasks, additional considerations should be taken into. In these approaches, performance for combined tests of failures is not measured.

In the proposed framework, a mechanism for reconfiguration by considering dependencies is provided. The framework relies on a set of task dependencies that are compiled *a priori* by a mission commander, or a planner before the mission. The task dependencies, specifications of the mission, and the robot teams' capabilities are distributed (reliably) to the robots before mission execution begins. At that point, negotiation of task assignments and execution of the mission begins. The proposed framework strives to provide optimal solutions while responding effectively to communication and robot failures. This is the first coordination scheme to address such a broad range of failures for heterogeneous teams executing tightly coupled tasks requiring collective work (Sariel and Balch 2005).

Proposed Framework

The proposed framework is for a multi-robot team ($r_j \in R$, $0 \leq j < ||R||$) that must coordinate to complete a complex mission (M) including tasks T_i s ($0 \leq i < ||M||$) that have ordering constraints and require diverse capabilities and collective work. The overall objective is completing *M* in a cost optimal manner. The framework combines *auctions*, *coalition maintenance* and recovery routines called *precautions* to provide an overall system that finds near optimal solutions in the face of noisy communication and robot failures. The *precaution* routines enable the system to dynamically respond to these failures at run time and complete the mission with valid plans at minimum cost.

Task Representation

A simple but effective task representation is used to generate valid plans. The generated plans without a general planner are always valid in the framework by means of the selected task representation. Information on the task definition, required capabilities, hard and soft ordering constraints, and required number of robots is embedded in the representation of each task. After the task dependencies are given initially, the framework generates optimal and valid allocations.

Roles

After being informed about the tasks and dependencies, robots are allowed to be in different roles during runtime to execute the tasks in coordination as required. The framework is designed for missions consisting of sub-tasks having dependencies and requiring either one or more than one robot to execute. Therefore the tasks may be executed by a group of robots. Coalition organizational paradigm (Bryan and Lesser 2004) is selected for teams of robots executing a task. The coalitions (C_i) can contain one or more robots numbers of which are defined in the task representation to execute a task T_i of overall mission *M*. The capabilities (cap_j) of robots r_j in a coalition should be a superset of the required capability set for T_i ($reqcap_i$). A robot (r_j) may be in different roles for task T_i such as auctioneer, bidder (B_{ij}), coalition leader (CL_i) and coalition member (CM_i) in different time steps.

- An *Auctioneer* robot manages auction negotiation steps and selects $reqno_i$ suitable members of a coalition.
- A *Bidder* robot is a candidate robot to become a member of a coalition executing a task.
- A *Coalition Leader* maintains the coalition and provides synchronization. It executes a portion of the task.
- A *Coalition Member* is one of the members of the coalition, and it executes a portion of the task.

A is the auctioneer set, B_{ij} is the bidder robot r_j for task T_i. A robot r_j may be in more than one B_{ij} roles for different tasks. However it is not allowed that a robot r_j is in more than one of roles A_i, CM_i, or CL_i. The coalition members are selected by auctions.

Precautions

For dealing uncertainties because of the message losses, each robot keeps track of the models of known system tasks and other robots in their world knowledge. When there is reliable communication, the robots update their world knowledge accordingly. Whenever there are message losses in the system, the world knowledge of each robot may be inconsistent. Such inconsistencies occur when robots are not informed about the tasks that are completed, under execution or under auction. *Precaution* routines discover conflicts and resolve them. When inconsistent messages are received, both corrections are made and warnings are released.

```
Routine Duties
if $r_j \in A$
    AuctionNegotitationProcess

if $r_j$ is a $CM_i$
    if the "synchronization message" has been arrived
        Set coalition count down
    else if the coalition count down is 0
        Cancel execution of the task
        Send leave message to the $CL_i$ ; break
    else
        if the last communication with $CL_i$ is more than $\tau$ (threshold)
            Send "coalition leader query" message
            Decrease coalition count down
    Send updated cost/bid value to the $CL_i$

if $r_j$ is a $CL_i$ and $\|C\|>1$
    if each $CM_i$ has already sent the execution/updated cost message
        Send "synchronization message" to all $CM_i$s
        Set coalition count down
    else if the coalition count down is 0
        Terminate the coalition; break
    else
        Decrease coalition count down
    Broadcast max cost ($CM_i$)
    Begin CoalitionMaintenanceProcess

if the robot is not executing a task || the task under execution is released
    Begin SelectAction
Send query messages for the tasks which are free
If there is a task selected, execute the own portion
```

Figure 1. Routine Duties of a Robot

Action Selection

Each robot initially updates its world knowledge based on the incoming messages. Then considering the new updated world knowledge and the current states of the processes for different roles, it selects the action. The action may be joining a coalition executing a task or becoming an auctioneer auctioning for a task. Before selection of the action, the robot should perform its routine duties for different roles. These duties are given in Figure 1. If it is leading an auction, it performs the negotiation process. Duties of roles CM_i and CL_i are performed for ensuring synchronization while coordinating in a coalition. If the robot is a CL_i, it checks the current coalition members' situations and can decide to reconfigure the coalition when new robots join or some members are not reachable.

While selecting the action, the robot considers the tasks awarded by an auctioneer, tasks under execution but with a higher value of cost than the robot has, the soft dependencies of the current task, and the free tasks which are not executed or under auction. The robot should be capable of executing all the tasks under consideration.

That means the hard dependencies of them should be completed, and the robot should be able to execute these tasks with its capabilities. Before the robot decides on one of these tasks, it ranks these tasks based on the costs for them and then selects the minimum cost task. This ranking procedure ensures optimum cost selection of the tasks for the given situation. The tasks awarded by an auctioneer have higher priority than free tasks, and the tasks of coalitions with higher maximum cost value than the robot's cost have the highest priority of all.

The selection of the task is performed by considering both the costs and priorities. The task with the minimum cost is selected. If the costs are the same for different tasks, the priorities are used for selection. Another ranking mechanism is used for the selection of a free task, if there are more than one free task having the minimum cost of all, the robot selects the one with the lowest number of soft dependencies and lowest number of robot requirements for executing. Therefore if the costs of the tasks are the same, initially the tasks requiring low coordination is selected. The algorithmic description of action selection process is given in Figure 2.

```
SelectAction for $r_j$
For each known task $T_i$
    if the $cap_j \subseteq reqcap_i \wedge$ hard dep. of $T_i$ are completed
        if the task is not in one of the following types skip
            The tasks for which the robot is accepted for joining (with higher max cost value
            than the robots)
            Awarded tasks by the auctioneers
            Free tasks which are not under consideration
     else add the task in a priority queue ordered by cost with the priorities in given order
Select the minimum cost task in the priority queue

if $r_j$ is in a coalition
    cancel executing the task, send "leave" message to the $CL_i$

if the selected task is a free task
    begin AuctionNegotitationProcess
else
    set the selected task as the current task
    if it is an awarded task
        send "accept become a member" message
```

Figure 2. Action Selection

Auction Negotiation Process

Each robot offers an auction for a free task if it is selected in the action selection step. When a robot becomes an auctioneer, it manages the auction negotiation process in which the coalition members and the leader are selected for the task to be executed.

Initially the auctioneer offers the auction. The robots can get the necessary task details from the auctions. After receiving an offer, the validity of the auction is checked. If the auction is invalid, a warning message is sent to the auctioneer. This invalidity may occur if the auctioneer has incomplete knowledge about the mission status. Possible situations may be that the task is completed or it has already been executing. If the auction passes the validity check, the candidate robot becomes a bidder of the task (B_{ij}), calculates the cost and sends the cost value as a bid. The other candidate robots behave as so. The auctioneer robot is also a bidder and generates a bid for the task at hand. It waits until the end of the deadline. If the auctioneer cannot get the necessary number of bids from the other robots until the deadline, it cancels the auction. Otherwise it ranks all the bids. It selects the robot with the minimum cost as the CL_i of the coalition. The remaining robots are selected among the other bidders in the ranking. If the necessary number of robots to execute the task is one, the selected leader is the only member of the coalition. In this ranking process, the auctioneer may also select itself either as a CL_i or a CM_i. A bidder robot may be awarded by different auctioneers. However in the action selection step, it selects the optimum cost task for it. Finally it sends a message to become a CM_i to only one of these auctioneers. In the current implementation, each robot involves in only one coalition executing one task.

Coalition Maintenance Process

In the proposed framework, coalition reconfiguration is ensured for obtaining optimal results for changing situations. The coalition leader is responsible to broadcast the maximum cost value of the coalition members in each execution step. If a robot out of coalition has a lower cost value than the maximum cost value for the corresponding task and selects this task in action selection step, it sends a *join request* message to the coalition leader. The leader getting *join request* message, directly adds the robot to the coalition. If the coalition leader detects that the size of the coalition is more than required, it can *release* redundant number of coalition members having the maximum cost value. A releasing and locking mechanism is added to prevent the coalition members leave the coalition until a new more suitable robot joined to the coalition. If a robot gets *released* message, it can select another more suitable task after then. When the coalition leader considers the size of the current coalition, it also checks the failures. Since each robot in the coalition broadcasts *under execution* and updated cost messages, their failure can be detected by the coalition leader. The failed robots are also released if there is enough number of members. If there is not enough number of members to execute the task for a period of time, the coalition leader terminates the execution of the task.

Experimental Design

The proposed framework is tested for collective building construction domain. In the designed experiment, there are different types of objects. The overall mission for the robots contains tasks of finding the necessary objects and making the construction while obeying the specified ordering restrictions. Nine robots with different capabilities are used in the experiments. The pushing requirements of these objects are also different. This mission both contains dependent tasks and requires cooperative work of the robots. In this domain, experiments are conducted in terms of time to complete the overall mission and active execution time of the robots. The performance is measured against message losses and robot failures. The conducted experiments are run on a simulator simulating the message exchange and execution of the missions. The results are from 100 independent runs with random seeds.

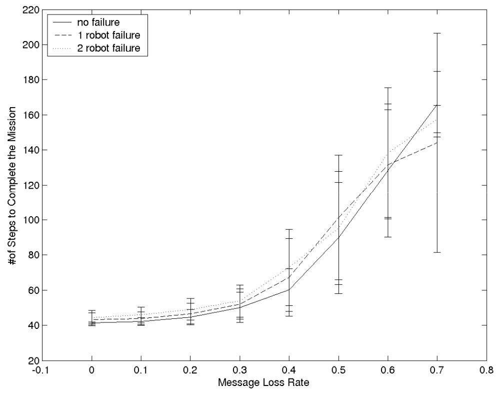

Figure 3. Number of Steps to Complete the Mission Analysis

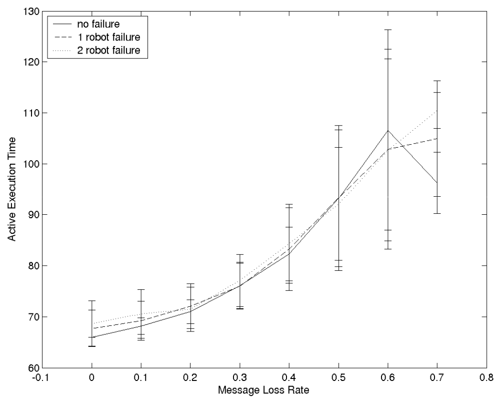

Figure 4. Total Active Execution Time Analysis

Experimental Results

Experiments illustrate that even for 50% message loss rate the robots can complete the mission for the given time period. Since the synchronization is highly required for some of the tasks, the message losses cause the synchronization be lost and also the auction negotiation steps not be completed reliably. If there is enough number of robots having required capabilities after failures, the system can also recover itself and complete the mission with an additional cost of recovery against robot failures. The mission completion time increases for increasing message loss rates logarithmically as in Figure 3. It can also be seen that the framework can easily handle robot failures. In Figure 4, the total execution time analysis can be seen. Combined experiments for robot failures and message losses illustrate that although there is a decrease in performance for the failure cases, the system can recover itself to a great extent while always generating valid plans.

Conclusion and Discussion

The proposed framework is a generic framework for multi robot teams. The generated plans are always valid by means of the selected task representation. The recovery solutions provided by *precaution* routines for different kind of failures ensure the approach is complete. In this framework, close to optimal solutions are generated with available resources at hand. Experiments to validate the approach were conducted in a construction domain.

Analysis of the effects of the cost function selection and proofs of optimality for different domains are part of the current research ongoing. Appropriate cost functions should be applied to obtain optimal results for different domains. This cost function may be defined based on some constraints. Currently the performance is being tested for different domains with different cost functions. One of the possible domains is NP-hard multi robot exploration problem also known as MTSP (Multi TSP). Performance of the proposed framework as a distributed approach for allocating targets to multi robots is being tested for this problem.

The framework is planned to handle online tasks and deadline constraints on the tasks. After observing performance results in simulations for different domains, it is expected to propose a general framework for a multi robot team capable of executing complex missions containing inter-dependent tasks requiring heterogeneity and coordination under uncertainties on dynamic environments.

The final evaluations are going to be implemented on real robots for a complex mission having tasks with ordering constraints.

References

Lemarie T., Alami R. and Lacroix S. 2004. A Distributed Task Allocation Scheme in Multi-UAV Context. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA).

Kalra N., Ferguson D. and Stentz A. 2005. Hoplites: A market-based framework for planned tight coordination in multi-robot teams. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA).

Sariel S. and Balch T. 2005. Robust Multi-Robot Coordination in Noisy and Dangerous Environments. GVU Technical Report: GIT-GVU-05-17, Georgia Institute of Technology..

Zlot R. and Stentz A. 2005. Complex Task Allocation for Multiple Robots. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA).

Bryan H. and Lesser V. 2004. A Survey of Multi-Agent Organizational Paradigms. UMass Computer Science Technical Report. 04-45.