Task Allocation and Coordination for Limited Capability Mobile Robots

Praneel Chand
Praneel.Chand@vuw.ac.nz
Dale A. Carnegie
Dale.Carnegie@vuw.ac.nz
Victoria University of Wellington
Wellington, New Zealand

Abstract

In multi-robot systems, predefined task allocation and coordination may not always work as desired. This is due to the inability to entirely model all aspects of the robot's interactions with the environment before task execution. Robots with limited capabilities also present the challenge of how their limited resources should be utilised to achieve the objectives of the global task. This paper presents a task allocation and coordination strategy for mobile robots with limited capabilities. To determine task-robot combinations, task allocation utilises a VOTS that represent a robot's suitability for a task. Task allocation yields a team of robots comprising task managers and task executers. After task allocation, task executer robots execute the global task, while task manager robots monitor the performance of the global task. If the global task's performance is not optimal, the task executer robots' resource utilisation is varied using a feedback coordination mechanism. The task allocation and coordination strategy is applied to a multi-robot map building task and preliminary results are presented.

1 Introduction

Multi-robot systems have been employed in a range of task domains including foraging, coverage, multi-target observation, box pushing, object transportation, exploration, flocking, and soccer [Farinelli et al., 2004]. Central to the success of many multi-robot systems is the ability of the individual robots to cooperate and coordinate their activities. This can result in advantages such as increased efficiency in performing tasks and robustness to failure of individual robots. Coordinating a team of mobile robots usually involves implementing task allocation and coordination mechanisms. Task allocation mechanisms address the question: "which robot should execute which task? [Gerkey and Mataric, 2003]" Coordination mechanisms enable the actions performed by each robot to take into consideration the actions of the other robots in the team resulting in coherent team operation [Farinelli et al., 2004]. Recently research in multi-robot systems has also addressed coalition formation, the organising of multiple robots into temporary subgroups to accomplish an assigned task that would otherwise be impossible to complete [Parker and Tang, 2006; Vig and Adams, 2006b].

Our research group is currently developing a multi-robot system for urban search and rescue (USAR) that comprise grandmother, mother, and daughter robots [Williamson and Carnegie, 2006]. The robots in this system are heterogeneous, having varying computational and physical capabilities. The grandmothers have the most powerful computers and are physically the largest. On the other hand, the daughters have the least powerful computers and are the smallest in physical size. The robots also possess a variety of exteroceptive sensors such as infrared and ultrasonic in addition to wireless communication devices. Thus, each robot in the multi-robot system has the capability to process, sense, act, and communicate, and these capabilities are indicative of the resources that the robot possesses. Generally, the mothers and daughters have limited capabilities, particularly for processing, sensing, and acting.

With the varying capabilities of the robots in the multi-robot system described above, each robot's ability to execute a task will differ and can be expressed as a function of their capabilities. It is possible to devise a task allocation and coordination mechanism based on the resources that the robots possess and how efficiently they utilise their resources to perform tasks. It is hypothesised that such a mechanism may allow a global task to be carried out with increased efficiency.

2 Related Work

Various methods for coordination and task allocation in multi-robot systems have been discussed in [Farinelli et al., 2004; Gerkey and Mataric, 2003; 2004]. Whereas [Farinelli et al., 2004] focuses on coordination, [Gerkey and Mataric, 2003; 2004] address task allocation.

Of the classifications based on coordination identified in [Farinelli et al., 2004], the weakly centralized systems [Dias and Stentz, 2002; Noreils, 1993; Simmons et al., 2000] are of particular interest. In these systems, a leader robot is selected dynamically during task execution based on the situation of the team and the environment. The selection can be based on the physical abilities of the robots as proposed in [Noreils, 1993; Simmons et al., 2000]. However, these proposed systems have not been fully implemented and they do not attempt to optimize the resource utilisation of robots during task execution.

In [Gerkey and Mataric, 2004] a taxonomy has been developed for the multi-robot task allocation problem, differentiating robots as either single-task (ST) or multi-task (MT), tasks as either single-robot (SR) or multi-robot (MR), and assignment types as either instantaneous (IA) or time-extended (TA). Representative approaches to multi-robot task allocation [Botelho and Alami, 1999; Chaimowicz et al., 2004; Dias and Stentz, 2001; Gerkey and Mataric, 2002; Parker, 1998; Werger and Mataric, 2000; Zlot et al., 2002] are analysed. These approaches typically divide a task into indivisible subtasks and assign single robots to each subtask (ST-SR).

ALLIANCE [Parker, 1998] and BLE [Werger and Mataric, 2000] are examples of behaviour-based approaches to multi-robot task allocation. ALLIANCE uses motivational behaviours to monitor and dynamically reallocate tasks thus achieving fault tolerance and adaptive behaviour. In the BLE system, each robot has a corresponding behaviour that is capable of executing each task. The robots select a task to execute by continuously broadcasting locally computed eligibilities followed by determining the most eligible task using a greedy algorithm. A behaviour-based approach to multi-robot task allocation that uses the concept of vacancy chains is presented in [Dahl et al., 2003]. This approach is demonstrated in groups of homogeneous robots where vacancy chains emerge through reinforcement learning.

Market based task allocation methods [Botelho and Alami, 1999; Dias and Stentz, 2001; 2002; Gerkey and Mataric, 2002; Mataric et al., 2004; Zlot et al., 2002] have also been widely utilised in multi-robot systems. These approaches can divide a task into subtasks for the robots to bid and negotiate. An auctioning mechanism utilises a task to revenue/cost mapping function to greedily assign subtasks to the highest bidders. TraderBots [Dias and Stentz, 2003] is a market-based approach for resource, role, and task allocation in multi-robot coordination. In this system, a RoboTrader module on each robot coordinates the activities of the agent and its interactions with other agents. Building on the success of market-based multi-robot coordination techniques, an approach to complex task allocation is presented in [Zlot and Stentz, 2005] where a complex task is represented using task trees. The distributed task allocation algorithm allows robots to simultaneously and continuously allocate and decompose complex tasks.

Dynamic role assignment [Chaimowicz et al., 2004] assigns roles to each robot in the team. During the execution of a cooperative task, the robots dynamically exchange roles in a synchronised manner adapting to changes in the environment. Specialised dynamic role assignment methods have been used for robotic soccer [Pagello et al., 2006; Stone and Veloso, 1999] where the robots dynamically switch between roles such as attacker and defender or master and supporter.

Burgard et al. [2005] address task allocation and coordination in multi-robot exploration. For each robot, they trade-off the utility and cost of potential target points for exploration. In this manner, each robot is assigned a target point for exploring. A more recent example using a similar technique for coordination is [Fox et al., 2006]. The coordination strategies of [Burgard et al., 2005; Fox et al., 2006] are not explicitly based on the computational and physical resources that each robot possesses and assumes that each robot is capable of solving the exploration problem.

Teamwork models have been developed that provide mechanisms for agents to form teams to accomplish a common goal [Jennings, 1995; Tambe, 1997]. A general model of teamwork, STEAM (Shell for TEAMwork) is presented in [Tambe, 1997]. STEAM is based on joint intentions theory [Cohen and Levesque, 1991] and SharedPlans theory [Grosz, 1996]. It facilitates monitoring of team performance and allows team reorganisation. Another general model based on joint intentions theory is the joint responsibility GRATE (Generic Rules and Agent model Testbed Environment) system [Jennings, 1995]. This system involves satisfying defined preconditions before collaboration can begin in addition to generating plans for agent behaviour during collaboration.

Parker and Tang [2006] consider the problem of single-task robots performing multi-robot tasks in the development of heterogeneous robot coalitions that solve single multi-robot tasks. They use an approach called ASyMTRe (Automated Synthesis of Multirobot Task solutions through software Reconfiguration) to generate multi-robot coalitions using complete information. The approach is demonstrated on tasks that require multiple robots to share sensor and effector capabilities. Robot capabilities are represented using schemas rather than resources and the performance of the coalition is not monitored during task execution.

Vig and Adams [2006b] identify issues that arise while attempting to use multi-agent coalition formation algorithms for multi-robot systems. Their work also addresses the problem of single-task robots performing multi-robot tasks. They develop a multi-robot coalition formation algorithm using an adaptation of Shehory and Kraus' [1998] algorithm. The first stage of the algorithm involves distributively calculating initial coalition values for all possible coalitions while a second stage involves robots agreeing on coalitions and forming them. Iteratively, the algorithm is able to form multiple coalitions, hence assigning multiple robots to multiple tasks. The performance of the formed coalitions are not monitored during task execution. Vig and Adams [2006a] also developed RACHNA (Robot Allocation through Coalitions using Heterogeneous Non-Cooperative Agents), a market-based multi-robot task allocation scheme. It is based on a reversed auction scheme that allows tasks to bid on robot services and is applied to the multi-robot coalition formation problem.

3 Contribution

The work presented in this paper varies significantly from those reported in the literature. While it may be possible to use an adapted version of some of the task allocation and coordination methods reported, there is a key difference. We require a mechanism that can monitor and optimize an individual robot's resource utilisation based on both local and global performance information. Resource utilisation is determined by the robot's processor usage towards each physical resource type for a particular task.

Robot capabilities are expressed by numerical vectors of merit (VOM) that represent processing, communication, sensing, and actuation resources.

Similarly, tasks are expressed by numerical vectors of task requirements (VOTR) representing their resource requirements. Our initial robot selection is based on a greedy assignment that utilises a vector of task suitability (VOTS) representing each robot's suitability for a particular task. This initial selection is a two stage process selecting managers and workers for the task. Since it is often impossible to entirely model all aspects of the robot's interactions with the environment before task execution, task reallocation may be required. This task reallocation process will be controlled by optimising the robots' resource utilisation based on their performance.

Our method can readily be used for single-task robots performing single-robot tasks. We intend to demonstrate it on a multi-robot map-building and exploration task expressed as multi-task robots performing single-robot tasks. This task allocation and coordination mechanism can also be applied to single-task robots performing multi-robot tasks.

4 System Overview

An overview of the proposed system is shown in Figure 1. Initially, a global task is specified in terms of the resources required. Next, based on the available resources (robots) a team is formed that comprises two levels of control – task management and task execution. The task managers will generally comprise the most computationally powerful robots (the grandmothers in our USAR team) while the task executers will usually be the less powerful robots (the mothers and daughters in our USAR team).

The task manager robots (managers) identify the task executer robots (workers) and initially allocate tasks to them. After initial task allocation, a hierarchy exists amongst the task executer robots depending on the tasks they are assigned. Following this, the task executer robots carry out their tasks while the task manager robots monitor their performance. Monitoring the performance of the task executer robots involves determining how well they use their resources and coordinating their resource utilisation using a feedback coordination mechanism (section 8). If the global performance is not optimal, the task executer robots' resource utilisation is varied. Thus, the position of task executer robots in the hierarchy can change.

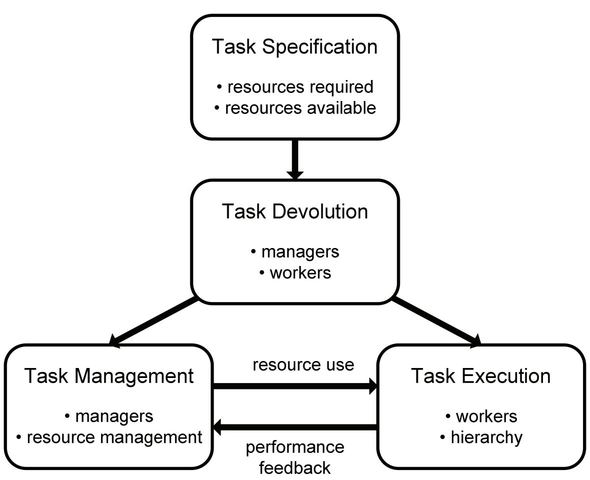

Figure 1: Overview of the task allocation and coordination mechanism.

5 Task Specification

A global task is specified manually by a human user via a remote base station (computer). The global task consists of set of n tasks that are specified by a number of criteria representing the resources required and conditions for that particular task. Figure 2 illustrates these criteria. The tasks are divided into two categories: n_1 management tasks and n_2 worker tasks. Management and worker tasks are assigned to appropriate task manager and task executer robots respectively. The task type can either be one-off or continuous. There are four types of resources: processing, communication, sensing, and actuation. Each resource type is represented by a minimum capability requirement and a set of weights that are used in the robot selection process. Task executer robot tasks employ additional criteria marked with an asterisk in figure 2. A robot quantity criteria specifies an initial, minimum, and maximum number of task executer robots required for the task. For example, in a map-building task the quantities can represent the number of planner and explorer robots required. Similarly, for an object pushing task they can represent the number of robots needed to move an object. Tasks that can be simultaneously executed with the current task are specified in the concurrent tasks field. Since there can be multiple identical tasks, there are m_1 management and m_2 worker tasks that are unique.

All tasks require control algorithms that are executed by the robot's processor. Thus, resource utilisation is estimated by the time it takes to execute these control algorithms on the robot's processor. There are four categories of resource utilisation: planning, communication, sensing, and actuation. An estimated initial resource utilisation is specified for each task-robot combination. A task to resource utilisation mapping function is employed to encode each task-robot combination. It specifies what a change in resource utilisation means for the task-robot combination and is used by the feedback coordination mechanism.

A set of rules RL used by the feedback coordination mechanism is also specified for the global task.

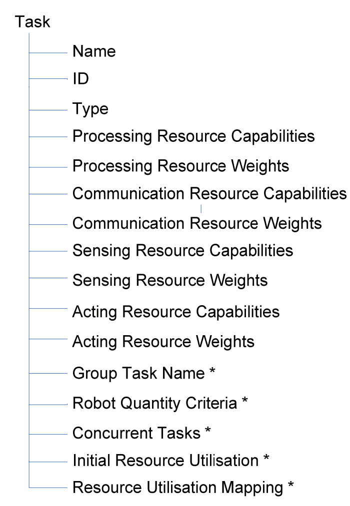

Figure 2: Summary of task specification criteria.

Similarly, the p robots available for the global task are specified with a number of criteria that represent the resources they possess as shown in figure 3.

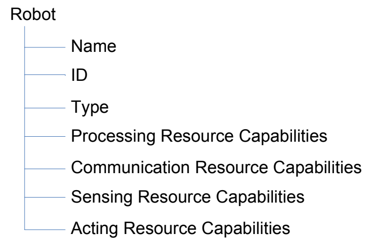

Figure 3: Brief description of robot specification criteria.

Numerical VOTR and VOM represent the resource requirements for tasks and capabilities of robots respectively. For each task t_i and robot r_j the capability of resource *type* RC_{type} is specified as:

$$RC_{type} = \left\{ rc_{type1}, rc_{type2}, \ldots, rc_{typek}, \ldots, rc_{typeq} \right\} \tag{1}$$

where:

$type \in \left\{ proc, comm, sense, act \right\}$,

rc_{typek} is the k^{th} sub-resource type, and

q is the number of sub-resources for resource RC_{type}.

The sub-resources for each resource type are:

- Processing: benchmark and memory.
- Communication: bandwidth and range.
- Sensing: a score representing each type of sensor present on the robot.
- Actuation: battery capacity, base size, base performance, and manipulator.

Sensing and acting sub-resource numerical values rc_{sensek} and rc_{actk} are determined from a weighted combination of additional normalised sub-parameters rcs_{sensei} and rcs_{acti}. The weights w_{si} and w_{ai} represent default preferences for the sub-parameters and can be task specific if required. A final sensing sub-resource value is obtained by multiplying the weighted linear combination by the quantity of sensors qty_k present on the robot.

$$rc_{sensek} = qty_k \times \sum_{i=1}^{S_k} w_{si} \times rcs_{sensei} \tag{2}$$

$$rc_{actk} = \sum_{i=1}^{S_k} w_{ai} \times rcs_{acti} \tag{3}$$

A set of weights KC_{type}, comprising sub-resource weights kc_{typek}, is also specified for each task. These weights are used in conjunction with RC_{type} to bias the sub-resources rc_{typek}.

$$KC_{type} = \left\{ kc_{type1}, kc_{type2}, \ldots, kc_{typek}, \ldots, kc_{typeq} \right\} \tag{4}$$

The resource utilisation of each category for each task-robot combination RU_{ijcat} is specified as:

$$RU_{ijcat} = \left\{ ru_{ijcat1}, ru_{ijcat2}, \ldots, ru_{ijcats}, \ldots, ru_{ijcatz} \right\} \tag{5}$$

where:

$cat \in \left\{ plan, comm, sense, act \right\}$,

ru_{ijcats} is the s^{th} task dependent resource utilisation sub-category, and

z is the number of sub-categories for resource utilisation RU_{ijcat}.

6 Task Devolution

Task devolution performs two primary functions. Firstly, it transfers the control of the global task from the base station to the robots. Secondly, an initial allocation of tasks to the robots is performed.

A key element of the task devolution process is the VOTS. $VOTS_{ij}$ is the VOTS for a task-robot pair and represents the j^{th} robot's suitability for the i^{th} task. The VOTS entry for each resource *type* $VOTS_{typeij}$ is a function of the sub-resources rc_{typek} available on the robot $r_j rc_{typek}$ and the sub-resources required for task the $t_i rc_{typek}$. Robot r_i is considered capable of performing task t_i if $VOTS_{typeij} \geq 1$ for all resource types.

$$VOTS_{typeij} = \sum kc_{typek} \frac{r_i rc_{typek}}{t_i rc_{typek}} \tag{6}$$

There are two stages in the task devolution process. At the first stage, the base station performs an initial task devolution to identify the task manager robots and assign tasks to them. The task specification and robot capability details are then transferred to the task manager robots. In the second stage, the task manager robots carry out a secondary task devolution, identifying and assigning tasks to the task executer robots. The initial and secondary task devolution stages are both greedy assignment processes.

The main steps of the initial task devolution are as follows:

1. Identify a subset of all the robots that are capable of performing at least one management task.
2. Rank the capable robots in descending order based on the sum of their VOM entries for all tasks that they are capable of performing.
3. Consider the highest ranked robot. Determine the capability of this robot to perform multiple combinations of management tasks that have not been assigned.
4. Assign a multiple task combination to the highest ranked robot. The goal is to maximise the robot's resource utilisation such that all VOM entries for the multiple task combination approach unity.
5. Remove the highest ranked robot from the selection process.
6. If all management tasks have not been assigned and all ranked robots have not been considered go to 3.

When steps 3-6 are executed, all management may not be assigned. The greedy nature of the algorithm tends to favour combinations of smaller tasks over individual larger tasks. To account for this imbalance, an additional iterative step is included. The sum of VOM entries for each multiple task combination is overrated by a factor CO. Weight w_{CO} is incremented in steps of 0.1 from 0 to 1 when all management tasks are not assigned successfully. The number of tasks in the combination is n_{1c} and the maximum number of tasks possible in a combination is n_{1cp}.

$$CO = 1 + w_{CO} \times \frac{n_{1c}}{n_{1cp}} \tag{7}$$

Stage 2 of the task devolution process, secondary task devolution is outlined below:

1. Determine the robots that are capable of performing each worker robot task.
2. For each robot that is capable of performing a worker robot task, determine the sum of the VOM entries for that particular task-robot combination. Let this sum be the robot's load handling capacity *lhc* for that task.
3. Determine the number of other tasks that each robot is capable of executing. Let this number be the robot's task diversity capability *tdc*.
4. Using the *lhc* value from step 2 and the *tdc* value from step 3 determine the each robot's utility U to perform each task using (8). The weights w_{lhc} and w_{tdc} control the balance between the two variables.
5. Use the utilities from step 4 to sort the capable robots for each task in descending order of utility.
6. Sort the tasks in ascending order of the number of capable robots for each task.
7. Starting with the highest ranked task, assign the number of robots required for each type of task as specified in the worker task requirements.
8. Enable initial resource utilisation for each robot-task combination.

$$U = w_{lhc} \times lhc + w_{tdc} \times tdc \tag{8}$$

7 Task Execution

In the task execution phase, the task executer robots carry out their assigned tasks using the resource utilisation details provided by the task manager robots. In this paper, a multi-robot map-building task is presented as an example of the task execution phase.

Many of the task executer robots have limited capabilities making it inefficient for them to be both planners and explorers simultaneously. Thus, the map-building task is expressed as multi-task robots performing single-robot tasks, where the task executer robots can be planners and/or explorers. The global environment *ge* is divided into local environments *le* based on the robot's processing and sensing capabilities. Both global and local environments are represented using rectangular occupancy grids [Thrun, 2003]. The individual robots use a navigation system that employs both reactive and deliberative control [Chand and Carnegie, 2005; Lee-Johnson et al., 2007]. Deliberative control is based on adaptations of the A* path planning algorithm [Pearl, 1984]. The vector field histogram method [Ulrich and Borenstein, 1998] and dynamic window approach [Fox et al., 1997] are tailored to suit the reactive control mechanism.

Planners negotiate and allocate local environments for the explorers to explore. They also plan global paths for the explorers to reach their allocated local environments. Initially, all local environments are assigned a utility U_{el} value of 1 for each explorer. Explorers are allocated unexplored local environments using a greedy algorithm that trades off U_{el} with a cost C_{el} associated with reaching each unexplored local environment. The trade-off value is the difference between U_{el} and C_{el}. Cost C_{el} consists of the robot's distance to the local environment d_{rel} and the mean distance of other explorers to the local environment $\overline{d_{oel}}$ (9). The weights control the balance between the two inputs. The utility of local environments in the immediate vicinity of an allocated local environment is multiplied by a reduction factor *urf* for other explorers. The reduction factor enables the explorers to spread out in the global environment. Additional issues for explorer coordination, such as speed, are addressed by the feedback coordination mechanism (section 8). Planners utilise a two-tiered A* global path planning algorithm to direct explorers to their allocated environments [Chand and Carnegie, 2007]. The two-tiered algorithm sequentially searches for paths through local environments to generate a global path.

$$C_{el} = w_{rel} \times d_{rel} + w_{oel} \times \overline{d_{oel}} \tag{9}$$

Path planning within a local environment employs a modified A* algorithm. Nodes whose occupancy probabilities exceed a threshold T_o are eliminated from the cost calculation. The cost $g(x)$ of all other nodes is linearly dependant on a cost multiplier c_m:

$$g(x) = \begin{cases} g(x_{par}) + d_n(x, x_{par}).c_m(x) & \text{if } p_o(x) < T \\ \infty & \text{otherwise} \end{cases} \tag{10}$$

The unit interval cost multiplier c_m takes into account both occupancy probability $p_o(x)$ of node x and the mean probability $\overline{p_o(y)}$ of nodes y representing the robot clearance:

$$c_m(x) = 1 + w_1.p_o(x) + w_2.\overline{p_o(y)} \tag{11}$$

Weights w_1 and w_2 control the balance between the two inputs.

Once the explorer robots are inside their allocated local environments they begin constructing a map of the local environment. Inspired by the trapezoidal decomposition coverage method [Choset, 2001], a line scanning method that accounts for the sensing range of the robot is employed to direct map-building and exploration. The method involves generating a set of waypoints for the robot to navigate through in order to achieve coverage of the local environment. Paths are planned between adjacent waypoints in the set such that unexplored space is favoured. During exploration, the occupancy probabilities of the map are updated using Baye's rule [Moravec, 1988].

Planners and explorers coordinate their activities with the help of the task manager robots. A single queue of explorer requests for new local environments and paths is maintained by the task manager robots. These requests are forwarded to planners to process. After processing the requests, the planners forward new directions and instructions to the explorer robots.

8 Task Management

Task management is controlled by the task manager robots. Inspired by recent work on multi-agent coalition formation for grid computing [Lee and Chen, 2006], we will investigate the development a case-based reasoning (CBR) [Aamodt and Plaza, 1994] system with reinforcement learning [Sutton, 1988] for the feedback coordination mechanism.

The CBR system uses the set of initial global task rules to intelligently predict new resource utilisation values for the task executer robots. Rules *RL* and their preferences *PR* will be modified using the reinforcement learning component depending on the success of using them. An overview of the feedback coordination mechanism is presented in figure 4. The intelligent resource utilisation prediction (IRUP) mechanism consists of the CBR and reinforcement learning components.

Inputs to the feedback coordination mechanism include the current resource utilisation *RU* and its achievement *RA* for each task executer robot. A battery charge level input is added to allow robots to relinquish sensing and actuation tasks before breakdown. The *RU* of each robot is compared with the target resource utilisation *RU'* set by the task manager robot to determine robot load success *RLS* values. Similarly, the *RA* of each robot is compared with an estimated target achievement *RA'* set by the task manager robot to obtain task execution success *TES* values. The *RLS* and *TES* values are then input to the IRUP mechanism. A new set of target resource utilisation values and corresponding target achievements for the task executer robots is determined by the IRUP mechanism. If the performance of the task executer robots is satisfactory then the targets are unchanged. On the other hand, if the performance is not satisfactory then the targets are modified based on the global task requirements.

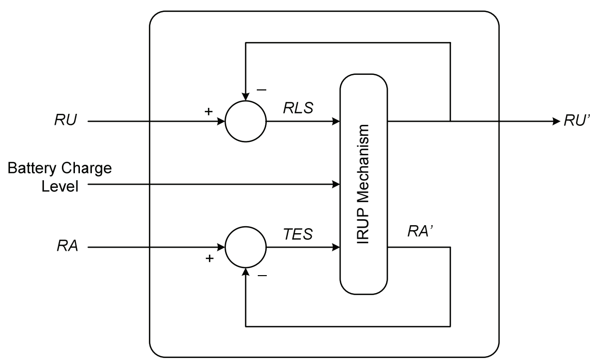

Figure 4: Overview of the feedback coordination mechanism.

The resource utilisation achievement of each resource category for each task-robot combination RA_{ijcat} is specified as:

$$RA_{ijcat} = \left\{ ra_{ijcat1}, ra_{ijcat2}, \ldots, ra_{ijcats'}, \ldots, ra_{ijcatz'} \right\} \qquad (12)$$

where:

$cat \in \{ plan, comm, sense, act \}$,

$ra_{ijcats'}$ is the s'^{th} task dependent resource utilisation achievement sub-category, and

z' is the number of sub-categories for resource utilisation achievement RA_{ijcat}.

Resource utilisation achievement values are determined from the performance of the physical resources that a robot possesses. For instance, planning achievement generally includes the number of local and/or global plans made. The volume of messages successfully transmitted and received indicates the achievement of communication. Sensing achievement can incorporate factors such as the accuracy of areas explored, or targets identified. The achievement of actuation can include distance travelled, average travelling speed, or objects successfully manipulated.

9 Experiments

The task allocation and coordination mechanism has been tested for a multi-robot map building task. A preliminary version of the task allocation and coordination mechanism has been implemented using MATLAB® 2007a. The multi-robot map building task has been simulated using the same software.

9.1 Task Devolution

The initial and secondary task devolution algorithms have been tested on a multi-robot map building task. The worker and management task requirements are represented in tables 1 and 2 respectively. Processing capability data represents the processor benchmark and the available memory. Communication capability data comprises bandwidth and range. The sensing capability data represents infrared sensing ability and odometer accuracy. Actuation capabilities include battery capacity, base size, and base performance value. Note that capabilities with entry values of negative one (-1) are ignored the selection process because they are unimportant for the task.

In the worker task specification (table 1), the robot quantity criteria specifies the initial, minimum, and maximum number of robots required for the task. The initial and maximum number of robots is determined from four parameters. These parameters include an estimated environment area, an initial robot density, an initial ratio of tasks, and the minimum allocations of each task required for the multi-robot map-building task. The initial resource utilisation and resource utilisation mapping parameters have been omitted in table 1 due to space constraints.

Similarly, the capabilities of eight candidate robots for the multi-robot map building task are shown in table 3.

Following the execution of the initial and secondary task devolution algorithms, a team of robots is selected for the global task. The resulting team and task allocations are shown in table 4.

9.2 Task Execution and Management

Using the task-robot combinations obtained in section 9.1 the multi-robot map building task has been tested in a simulated environment with randomly distributed obstacles. Figure 5 shows the task executer robots exploring and building maps of their allocated local environments. The green dashed-lines represent the boundaries of the local environments. Task manager robots have been excluded from the simulation display.

Three trial configurations have been tested to illustrate how a robot's processor usage can be varied. In trial 1 all robots have a maximum travel speed of 0.5 m/sec and sensing and actuation frequencies of 10 Hz.

The maximum travel speed is reduced to 0.25 m/sec in trial 2 while keeping the same sensing and actuation frequencies of trial 1. In trial 3, the sensing and actuation frequencies are reduced to 5 Hz with a maximum travel speed of 0.25 m/sec. Table 5 shows the resource utilisation for a planner robot (robot4) and an explorer robot (robot7). Each robot is assumed to have a 10% fixed processor overhead.

Table 1: Worker task requirements.

Task ID	Criteria	Value
W1	Name	planner
	Type	continuous
	Processing Capabilities	{6,8}
	Processing Weights	{0.5,0.5}
	Communication Capabilities	{11,100}
	Communication Weights	{0.5,0.5}
	Sensing Capabilities	{30,3.9}
	Sensing Weights	{0.5,0.5}
	Acting Capabilities	{1,-1,2}
	Acting Weights	{0.6,0,0.4}
	Group Task Name	multi-robot map-building
	Robot Quantity Criteria	{1,1,4}
	Concurrent Tasks	{W2}
W2	Name	explorer
	Type	continuous
	Processing Capabilities	{3,2}
	Processing Weights	{0.5,0.5}
	Communication Capabilities	{11,100}
	Communication Weights	{0.5,0.5}
	Sensing Capabilities	{30,3.9}
	Sensing Weights	{0.5,0.5}
	Acting Capabilities	{1,-1,2.5}
	Acting Weights	{0.6,0,0.4}
	Group Task Name	multi-robot map-building
	Robot Quantity Criteria	{4,1,4}
	Concurrent Tasks	{W1}

Table 2: Management task requirements.

Task ID	Criteria	Value
M1	Name	maintain global info
	Type	continuous
	Processing Capabilities	{30,100}
	Processing Weights	{0.8,0.2}
	Communication Capabilities	{11,100}
	Communication Weights	{0.5,0.5}
	Sensing Capabilities	{-1,-1}
	Sensing Weights	{1,1}
	Acting Capabilities	{-1,-1,-1}
	Acting Weights	{1,1,1}
M2	Name	secondary task devolution & feedback system
	Type	continuous
	Processing Capabilities	{50,100}
	Processing Weights	{0.8,0.2}
	Communication Capabilities	{22,100}
	Communication Weights	{0.5,0.5}
	Sensing Capabilities	{-1,-1}
	Sensing Weights	{1,1}
	Acting Capabilities	{-1,-1,-1}
	Acting Weights	{1,1,1}

The proportion of time spent on planning is reduced when the robots' maximum travel speed is decreased from 0.5 m/sec to 0.25 m/sec. This is indicated by the planning resource utilisation of table 5 for trials 2 and 3. In addition to this, a robot's sensing and actuation resource utilisation can be decreased by lowering the respective control algorithm execution frequency. The lower utilisations of trial 3 indicate this. Other explorer robots (robot2 and robot5) exhibited similar trends in resource utilisation for the three test configurations.

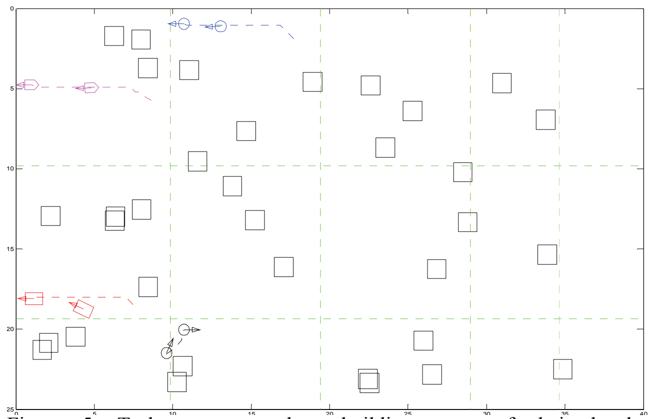

Figure 5: Task executer robots building maps of their local environments.

Table 3: Capability data of eight heterogeneous robots.

Robot ID	Criteria	Value
1	Name	robot1
	Type	tricycle-pentagon
	Processing Capabilities	{60,256}
	Communication Capabilities	{54,100}
	Sensing Capabilities	{37.79,3.96}
	Acting Capabilities	{2.04,0.50,3.00}
2	Name	robot2
	Type	tricycle-pentagon
	Processing Capabilities	{6,10}
	Communication Capabilities	{11,100}
	Sensing Capabilities	{37.79,3.96}
	Acting Capabilities	{2.14,0.50,3.00}
3	Name	robot3
	Type	differential-circular
	Processing Capabilities	{40,256}
	Communication Capabilities	{54,100}
	Sensing Capabilities	{34.35,3.96}
	Acting Capabilities	{2.15,0.40,2.75}
4	Name	robot4
	Type	differential-rectangular
	Processing Capabilities	{10,10}
	Communication Capabilities	{11,100}
	Sensing Capabilities	{48.10,3.96}
	Acting Capabilities	{2.53,0.60,2.90}
5	Name	robot5
	Type	differential-circular
	Processing Capabilities	{6,4}
	Communication Capabilities	{11,100}
	Sensing Capabilities	{43.55,5.80}
	Acting Capabilities	{1.25,0.40,2.75}
6	Name	robot6
	Type	differential-circular
	Processing Capabilities	{6,2}
	Communication Capabilities	{11,100}
	Sensing Capabilities	{43.55,5.80}
	Acting Capabilities	{1.40,0.40,2.75}
7	Name	robot7
	Type	differential-circular
	Processing Capabilities	{6,4}
	Communication Capabilities	{11,100}
	Sensing Capabilities	{43.55,5.80}
	Acting Capabilities	{1.19,0.40,3.00}
8	Name	robot8
	Type	differential-rectangular
	Processing Capabilities	{6,2}
	Communication Capabilities	{11,100}
	Sensing Capabilities	{60.98,5.80}
	Acting Capabilities	{0.95,0.60,2.90}

Table 4: Resulting team and task allocations

Task ID	Robot ID
M1	3
M2	1
W1	4
W2	4
W2	2
W2	7
W2	5

Table 6 illustrates the achievement data for robot4 and robot7. Reducing the maximum travelling speed produces a significant increase in the global task completion time (elapsed time). Map accuracy increases slightly when maximum travelling speed is reduced while maintaining the same sensing and actuation frequencies (trials 1 and 2). The area explored does not vary significantly since all task executer robots operated at the same speeds for the three trial configurations.

Table 5: Resource utilisation details of two task executer robots.

Robot ID	Resource Utilisation		Value (%)		
	Resource	Sub-Category	Trial 1	Trial 2	Trial 3
4	Planning	Global	0.81	0.42	0.39
		Local	0.43	0.32	0.20
	Communication		0.03	0.02	0.02
	Sensing	Obstacle & Pose Detection	18.01	18.21	9.17
		Local Map Building	2.40	2.42	1.27
	Actuation	Motion Control	42.19	40.82	20.60
		Motor Commands	2.43	2.45	1.22
	Free		23.70	25.34	57.13
7	Planning	Global	0.00	0.00	0.00
		Local	0.27	0.14	0.14
	Communication		< 0.01	< 0.01	< 0.01
	Sensing	Obstacle & Pose Detection	29.76	30.07	15.09
		Local Map Building	3.71	3.69	1.88
	Actuation	Motion Control	32.80	31.81	16.23
		Motor Commands	4.37	4.40	2.21
	Free		19.09	19.89	54.45

Utilising the presented task allocation and coordination strategy, the global goal of the multi-robot map building task is to produce a good quality map while reducing the overall execution time and distance travelled. This will be achieved by optimising each individual robot's resource utilisation. A robot's resource utilisation can be modified by varying its travel speed and sensing and actuation frequencies. These variations can be good or bad depending on the performance of the team. For example, explorers can perform global planning tasks if their sensing and actuation frequencies are reduced. This may be desirable if more planners are required due to poor performance or failure. Alternatively, explorer robots may suffer sensing and/or actuation failures which would render them more suitable for planning activities.

Table 6: Achievement details of two task executer robots.

Robot ID	Achievement Category		Value		
	Resource	Sub-Category	Trial 1	Trial 2	Trial 3
4	Planning (n Plans)	Global	17	16	17
		Local	58	65	82
	Communication (bytes)		739737	716002	891444
	Sensing	Area Explored (m^2)	220.83	231.62	313.44
		Map Accuracy (%)	96.58	97.41	96.48
	Actuation	Distance Travelled (m)	381.76	438.07	542.24
		Average Speed (m/sec)	0.24	0.15	0.15
	Elapsed Time (sec)		1599.9	2909.7	3581.8
7	Planning (n Plans)	Global	0	0	0
		Local	84	84	73
	Communication (bytes)		134945	161057	147582
	Sensing	Area Explored (m^2)	376.55	382.64	320.94
		Map Accuracy (%)	95.74	96.92	95.41
	Actuation	Distance Travelled (m)	471.70	476.63	425.96
		Average Speed (m/sec)	0.28	0.15	0.15
	Elapsed Time (sec)		1703.7	3248.9	2891.6

10 Conclusion

An approach to task allocation and coordination for limited capability mobile robots has been presented. During initial task allocation, robots are selected for two categories of control – task management and task execution. The initial task allocation uses a VOTS to represent a robot's suitability for a task. After initial task allocation, task executer robots execute the global task, while task manager robots monitor the performance of the global task. If the global task's performance is not optimal, the task executer robots' resource utilisation is varied using a feedback coordination mechanism.

The task allocation and coordination mechanism has been demonstrated for a multi-robot map building task. Preliminary results thus far are encouraging. The feedback coordination mechanism for task management is currently under development. More extensive results are anticipated for presentation at the conference. We are constructing a number of test environments and robot team configurations to represent scenarios where our task allocation and coordination strategy is likely to be beneficial.

Future work includes comparing our strategy to existing schemes outlined in the literature and determining scalability to real robots.

References

[Aamodt and Plaza, 1994] Agnar Aamodt and Enric Plaza. Case-based reasoning: foundational issues, methodological variations and system approaches. *AI Communications*, 7(1):39-59, March 1994.

[Botelho and Alami, 1999] Silvia C. Botelho and Rachid Alami. M+: a scheme for multi-robot cooperation through negotiated task allocation and achievement.

Proceedings of the IEEE International Conference on Robotics and Automation, pages 1234-1239, Detroit, Michigan, 10-15 May 1999. IEEE.

[Burgard et al., 2005] Wolfram Burgard, Mark Moors, Cyrill Stachniss and Frank Schneider. Coordinated multi-robot exploration. *IEEE Transactions on Robotics*, 21(3):376-386, June 2005.

[Chaimowicz et al., 2004] Luiz Chaimowicz, Vijay Kumar and Mario F. M. Campose. A paradigm for dynamic coordination of multiple robots. *Autonomous Robots*, 17(1):7-21, July 2004.

[Chand and Carnegie, 2005] Praneel Chand and Dale A. Carnegie. Reactive control of a tricycle mobile robot. *Proceedings of the Twelfth Electronics New Zealand Conference*, pages 129-134, Manukau, New Zealand, 14-15 November 2005. Manukau Institute of Technology.

[Chand and Carnegie, 2007] Praneel Chand and Dale A. Carnegie. Memory-time tradeoffs in a path planning approach utilising limited memory robots. *Proceedings of the International Conference on Computational Intelligence, Robotics and Autonomous Systems*, Palmerston North, New Zealand, 28-30 November 2007. Massey University (to be published).

[Choset, 2001] Howie Choset. Coverage for robotics - a survey of recent results. *Annals of Mathematics and Artificial Intelligence*, 31(1-4):113-126, October 2001.

[Cohen and Levesque, 1991] Phillip R. Cohen and Hector Levesque. Teamwork. *Nous*, 25(4):487-512, October-December 1991.

[Dahl et al., 2003] Torbjorn S. Dahl, Maja J. Mataric and Gaurav S. Sukhatme. Multi-robot task allocation through vacancy chains. *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2293-2298, Taipei, Taiwan, 14-19 September 2003. IEEE.

[Dias and Stentz, 2001] M. Bernardine Dias and Anthony Stentz. A market approach to multi-robot coordination. Technical Report CMU-RI -TR-01-26, Carnegie Mellon University, August 2001.

[Dias and Stentz, 2002] M. Bernardine Dias and Anthony Stentz. Opportunistic optimization for market-based multirobot control. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2714-2720, Lausanne, Switzerland, October 2002. IEEE.

[Dias and Stentz, 2003] M. Bernardine Dias and Anthony Stentz. Traderbots: a market-based approach for resource, role, and task allocation in multirobot coordination. Technical Report CMU-RI-TR-03-19, Carnegie Mellon University, August 2003.

[Farinelli et al., 2004] Alessandro Farinelli, Luca Iocchi and Daniele Nardi. Multirobot systems: a classification focused on coordination. *IEEE Transactions on Systems, Man, and Cybernetics*, 34(5):2015-2028, October 2004.

[Fox et al., 1997] Dieter Fox, Wolfram Burgard and Sebastian Thrun. The dynamic window approach to collision avoidance. *IEEE Robotics and Automation Magazine*, 4(1):23-33, March 1997.

[Fox et al., 2006] Dieter Fox, Jonathan Ko, Kurt Konolige, Benson Limketkai, Dirk Schulz and Benjamin Stewart. Distributed multirobot exploration and mapping. *Proceedings of the IEEE*, 94(7):1325-1339, July 2006.

[Gerkey and Mataric, 2002] Brian P. Gerkey and Maja J. Mataric. Sold! auction methods for multi-robot coordination. *IEEE Transactions on Robotics and Automation*, 18(5):758-768, October 2002.

[Gerkey and Mataric, 2003] Brian P. Gerkey and Maja J. Mataric. Multi-robot task allocation: analyzing the complexity and optimality of key architectures. *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3862-3868, Taipei, Taiwan, R.O.C., 14-19 September 2003. IEEE.

[Gerkey and Mataric, 2004] Brian P. Gerkey and Maja J. Mataric. A formal analysis and taxonomy of task allocation in multi-robot systems. *International Journal of Robotics Research*, 23(9):939-954, September 2004.

[Grosz, 1996] Barbara J. Grosz. Collaborative systems. *AI Magazine*, 17(2):67-85, June-August 1996.

[Jennings, 1995] Nick R. Jennings. Controlling cooperative problem solving in industrial multi-agent systems. *Artificial Intelligence*, 75(2):195-240, June 1995.

[Lee-Johnson et al., 2007] Christopher P. Lee-Johnson, Praneel Chand and Dale A. Carnegie. Applications of a Adaptive Hierarchical Mobile Robot Navigation System. *Proceedings of the Australasian Conference on Robotics and Automation*, Brisbane, Australia, 10-12 December 2007. ARAA (submitted for publication).

[Lee and Chen, 2006] Hsiu-Hui Lee and Chung-Hsien Chen. Multi-agent coalition formation for long-term task or mobile network. *Proceedings of the International Conference on Computational Intelligence for Modelling, Control and Automation, and International Conference on Intelligent Agents, Web Technologies and Internet Commerce*, pages 52-58, Sydney, Australia, 29 November-1 December 2006. IEEE.

[Mataric et al., 2003] Maja J. Mataric, Gaurav S. Sukhatme and Esben H. Ostergaard. Multi-robot task allocation in uncertain environments. *Autonomous Robots*, 14(2-3):255-263, March-May 2003.

[Moravec, 1988] Hans Moravec. Sensor fusion in certainty grids for mobile robots. *AI Magazine*, 9(2):61-74, July-August 1988.

[Noreils, 1993] Fabrice R. Noreils. Toward a robot architecture integrating cooperation between mobile robots: application to indoor environment. *International Journal of Robotics Research*, 12(1):79-98, February 1993.

[Pagello et al., 2006] Enrico Pagello, Antonio D'Angelo and Emanuele Menegatti. Cooperation issues and distributed sensing for multirobot systems. *Proceedings of the IEEE*, 94(7):1370-1383, July 2006.

[Parker, 1998] Lynne E. Parker. ALLIANCE: an architecture for fault tolerant multirobot cooperation. *IEEE Transactions on Robotics*, 14(2):220-240, April 1998.

[Parker and Tang, 2006] Lynne E. Parker and Fang Tang. Building multirobot coalitions through automated task solution synthesis. *Proceedings of the IEEE*, 94(7):1289-1305, July 2006.

[Pearl, 1984] Judea Pearl. *Heuristics: Intelligent Search Strategies for Computer Problem Solving*, Addison-Wesley, Reading Massachusetts, 1984.

[Shehory and Kraus, 1998] Onn Shehory and Sarit Kraus. Methods for task allocation via agent coalition formation. *Artificial Intelligence*, 101(1-2):165-200, May 1998.

[Simmons et al., 2000] Reid Simmons, Sanjiv Singh, David Hershberger, Josue Ramos and Trey Smith. *First results in the coordination of heterogeneous robots for large-scale assembly*. In *Experimental Robotics VII*. vol. 271. *Lecture Notes in Control and Information Sciences*. Springer-Verlag, London, UK, pages 323-332, 2000.

[Stone and Veloso, 1999] Peter Stone and Manuela Veloso. Task decomposition, dynamic role assignment, and low-bandwidth communication for real-time strategic teamwork. *Artificial Intelligence*, 110(2):241-273, June 1999.

[Sutton, 1988] Richard S. Sutton. Learning to predict by the methods of temporal difference. *Machine Learning*, 3(1):9-44, August 1988.

[Tambe, 1997] Milind Tambe. Towards flexible teamwork. *Journal of Artificial Intelligence Research*, 7(1):83-124, September 1997.

[Thrun, 2003] Sebastian Thrun. *Robotic mapping: a survey*. In Gerhard Lakemeyer and Bernhard Nebel, editors. *Exploring artificial intelligence in the new millennium*. Morgan Kaufmann Publishers Inc., San Francisco, California, pages 1-35, 2003.

[Ulrich and Borenstein, 1998] Iwan Ulrich and Johann Borenstein. VFH+: Reliable obstacle avoidance for fast mobile robots. *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1572-1577, Leuven, Belgium, 16-21 May 1998. IEEE.

[Vig and Adams, 2006a] Lovekesh Vig and Julie A. Adams. Market-based multi-robot coalition formation. *Proceedings of the International Symposium on Distributed Autonomous Robotic Systems*, pages 227-236, Minneapolis/St. Paul, Minnesota, 12-14 July 2006. Springer-Verlag.

[Vig and Adams, 2006b] Lovekesh Vig and Julie A. Adams. Multi-robot coalition formation. *IEEE Transactions on Robotics*, 22(4):637-649, August 2006.

[Werger and Mataric, 2000] Barry B. Werger and Maja J. Mataric. Broadcast of local eligibility for multi-target observation. *Proceedings of the International Symposium on Distributed Autonomous Robotic Systems*, pages 347-356, Knoxville, Tennessee, 4-6 October 2000. Springer-Verlag.

[Williamson and Carnegie, 2006] David A. Williamson and Dale A. Carnegie. Embedded platform for search and rescue applications. *Proceedings of the International Conference on Autonomous Robots and Agents*, pages 373-378, Palmerston North, New Zealand, 12-14 December 2006.

[Zlot and Stentz, 2005] Robert Zlot and Anthony Stentz. Complex task allocation for multiple robots. *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1515-1522, Barcelona, Spain, 18-22 April 2005. IEEE.

[Zlot et al., 2002] Robert Zlot, Anthony Stentz, M. Bernardine Dias and Scott Thayer. Multi-robot exploration controlled by a market economy. *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3016-3023, Washington, DC, 11-15 May 2002. IEEE.