Simultaneous Allocations of Multiple Tightly-Coupled Multi-Robot Tasks to Coalitions of Heterogeneous Robots

G.P. Das, Student Member, IEEE, T.M. McGinnity, Senior Member, IEEE and S.A. Coleman, Member, IEEE

Abstract—Most multi-robot task allocation algorithms are concerned with the allocation of individual tasks to single robots. However certain types of tasks require a team of robots for their execution, and for the allocation of such tasks non-conflicting robot teams have to be formed. Most of the existing allocation algorithms for such tasks mainly address the robot-team formation and the tasks are allocated sequentially. However, allocating multiple tasks simultaneously will result in a more balanced distribution of robots into teams. A market based algorithm for simultaneous allocation of multiple tightly coupled multi-robot tasks to coalitions of heterogeneous robots are proposed in this paper. The simultaneous allocations are deadlock-free and significant improvement in overall execution time is achieved as demonstrated by empirical evaluations.

1. INTRODUCTION

The major motivating factors for using a multi-robot system (MRS) instead of a single highly skilled robot are economic viability and redundancy [1]. In addition, by sharing tasks and executing them in parallel, the overall task completion time can be reduced, if the tasks are distributed in space and time or functionality [2]. Various multi-robot task allocation (MRTA) algorithms have been proposed to achieve these benefits [3], [4]. The taxonomy introduced in [4] for MRTA algorithms classifies robots into single task (ST) and multi-task (MT) robots, tasks into single robot (SR) and multi-robot (MR) tasks, and assignment into time extended allocation (TA) and instantaneous allocation (IA). As most of the real-world robots are ST robots, the focus in this paper is the allocation of closely coupled MR tasks (cooperative tasks) to coalitions of heterogeneous ST robots. An ST-MR MRTA algorithm tries to solve the problem “which coalition of robots should execute which task?” where a coalition is a temporary robot group formed for the execution of a task [5]. As the allocation of cooperative tasks (closely coupled MR tasks) is NP-hard [4], some trade-off between the solution optimality and computational complexity is required. Although centralised approach is not scalable to large MRSs due to the increasing communications, it provides a more efficient solution than decentralised or distributed approaches when tight cooperation between the robots are required [6].

Some ST-MR MRTA algorithms focus mainly on how coalitions are formed for a task. Lundh et al. [7] proposed an MRTA algorithm for robot cooperation by sharing functionalities. Similar to this, the Automated Synthesis of Multi-robot Task solutions through software Reconfiguration (ASyMTRe) [8] algorithm creates cooperative robot teams. The Mutual State Capability based Role Assignment (MuSCRA) model [9] assigns robots to specific roles in a team using concepts of synergy between robots. In the market based algorithm proposed in [10] coalitions are formed based on their balance and fault tolerance while sharing a workload.

Some other algorithms address both coalition formation and allocation of a set of tasks to different coalitions. An ST-MR MRTA algorithm using a genetic algorithm (GA) is proposed in [11]. The stochastic policies proposed in [12] avoid inter-robot communication. However, stochastic modelling of the robot behaviours can be a time consuming and inefficient process for a large heterogeneous MRS. In [13] the Double Round (DR) auction, where coalitions are selected based on the execution time estimates and interference predictions, is proposed. The MURDOCH algorithm [14] uses a hierarchical classification of tasks and a decentralised auction using the Contract Net Protocol (CNP) [15] for allocations. The Coupled-constraint Consensus Based Bundle Algorithm (CCBBA) [18] allocates duo-tasks (task with two subtasks). However the restriction on the number of subtasks in a task to two limits the applications of the algorithm.

Market based algorithms, which mimic the real-world market with robots bidding on tasks and allocations made based on these bids, have shown promising results [3]. In a real-world auction, money is exchanged to buy the items being auctioned and in most cases the items have a minimum base price. However, in the auction process commonly implemented in MRTA algorithms base prices are not set for tasks, there is no exchange of money and the robots do not raise the bids on a task; the attractive feature is the decentralisation of the computation for a robot’s suitability for a task. In real-world multi-item auctions sequential allocation can improve the profitability of the auctioneer [19]; however, can be complex and time-consuming for large number of items. Due to the difference mentioned earlier, when a set of tasks has to be allocated in an ST-MR allocation problem, simultaneous allocations of multiple tasks can improve the overall allocation efficiency and overall execution time; however, deadlock may occur where the robots bid on different tasks and no tasks have enough bids to start coalition formation [20].

The Centralised Co-operative Task Allocation (CeCoTA), a market based algorithm for allocation of cooperative MR tasks to coalitions of heterogeneous ST robots, is proposed.
in this paper. The contribution of this paper is that this algorithm addresses (i) heterogeneity of robots and tasks, (ii) coalition formation for each task and (iii) deadlock-free simultaneous near-optimal allocations of multiple tasks to coalitions without any conflicts. Most existing ST-MR MRTA algorithms use time-extended allocations, which are unsuitable for an environment with dynamic changes such as addition of new robots or tasks. In the CeCoTA algorithm presented herein, tasks are incrementally allocated, making it suitable for dynamic environments. To evaluate the performances of the proposed algorithm, they are compared with that of a GA based task allocation method.

The remainder of the paper is organised as follows. The MRTA problem addressed using the CeCoTA algorithm is defined in Section II. The details of the CeCoTA algorithm are explained in Section III. The experimental results are presented in Section V followed by the concluding remarks in Section VI.

II. DEFINITION OF THE PROBLEM

The heterogeneity of the robots is based on the set of skills available to each robot. A skill is the capability of a robot to execute certain actions (e.g. navigation, vision etc.). The robots may have different hardware and software modules to achieve these actions. Based on the available resources, the proficiency of these actions may vary from one robot to another. This proficiency is quantified as the robot’s expertise value associated with that skill. For example a robot with a 5-DoF (degrees of freedom) manipulator has limited object handling capabilities as compared to a 7-DoF manipulator, therefore has a lower expertise value. Although the expertise values are manually set at this stage, they can be defined as a function of the resources. The tasks are classified into different types (e.g. box-pushing, object-transportation etc.) and each task is divided into a set of subtasks. Each subtask requires a single robot for its execution and has a set of skills required from the robot assigned to it. As each task is tightly-coupled, all of its subtasks have to be executed in parallel. This task definition is similar to those in [14], but without the hierarchy. In the MRTA problem addressed in this work, coalitions have to be formed and each task has to be assigned to a coalition without any conflicts, and can be defined as follows.

Let \( R = \{ r_i, i = 1, \ldots, R \} \) be the given set of \( R \), ST robots, and a robot \( r_i \in R \) has a set \( S_i = \{ s_{i,q}, q = 1, \ldots, R_i \} \) of \( R_i \) skills and a set \( C_i = \{ c_{i,q}, q = 1, \ldots, R_i \} \) of corresponding expertise values, where \( c_{i,q} > 0 \) (0 \(<\ c_{i,q} \leq 1 \) is the expertise value of \( r_i \) on the skill \( s_{i,q} \). Let \( T = \{ t_j, j = 1, \ldots, R \} \) be the given set of \( R \), tasks, where a task \( t_j \in T \) is a set \( s_{t(j,k)}, k = 1, \ldots, R_{t(j,k)} \) of \( R_{t(j,k)} \) subtasks. In addition, a subtask \( s_{t(j,k)} \in t_j \) requires a set \( c_{t(j,k)} = \{ c_{q,k}, q = 1, \ldots, R_{t(j,k)} \} \) of \( R_{t(j,k)} \) skills. Let \( C_j = \{ c_{m,j}, m = 1, \ldots, R_j \} \) be the set of all possible coalitions in \( R \) for the task \( t_j \), then the MRTA algorithm has to find a unique winning coalition for each task in \( T \) so as to maximise

\[
\text{maximise} \sum_{t_j \in T} \sum_{(c_{m,j} \in \mathcal{C})} \alpha_{m,j} \beta_{m,j} CE_{m,j} \tag{1}
\]

subject to

\[
\sum_{(c_{m,j} \in \mathcal{C})} \alpha_{m,j} \leq 1, \forall t_j \in T \tag{2}
\]

\[
\alpha_{m,j} \in \{0,1\} \tag{3}
\]

\[
\beta_{m,j} \in \{0,1\} \tag{4}
\]

where \( \alpha_{m,j} \) is the allocation factor which defines whether the coalition \( c_{m,j} \) is assigned to the task \( t_j \), \( \beta_{m,j} \) is the coalition-suitability factor which defines whether the robots in the coalition \( c_{m,j} \) can be uniquely assigned to the subtasks of the task \( t_j \) and the utility factor \( CE_{m,j} \) is the coalition expertise of the coalition \( c_{m,j} \) on the task \( t_j \). The value of \( CE_{m,j} \) is equal to one if the task \( t_j \) is allocated to the coalition \( c_{m,j} \) and is zero otherwise. Allocation of a task is conflict free only if it is assigned to a single coalition, given by (2).

\[
S_{st}^{(i,k)} \subseteq S_r^i \tag{5}
\]

\[
p_{(m,n)}^i = \{(s_{t(j,k)}^i), r_i, \forall t_{j}, r_i \in c_{m,j} \} \tag{6}
\]

\[
soe_{t(j,k)}^i = \sum_{s_{t(j,k)}^i} \lambda_{q}^j \tag{7}
\]

\[
PE_{(m,n)}^i = \sum_{(s_{t(j,k)}^i) \in p_{(m,n)}^i} \left( soe_{t(j,k)}^i \right) \tag{8}
\]

\[
CE_{m,j}^j = \max_{p_{(m,n)}^i} \left( PE_{(m,n)}^i \right) \tag{9}
\]

A coalition \( c_{j,m} \in \mathcal{C}^j \) is a valid coalition for task \( t_j \) if and only if each subtask of \( t_j \) can be allocated to at least one robot in \( c_{m,j} \) and no robot is allocated to more than one subtask. This is a necessary condition, as each subtask has to be executed by a single robot and all subtasks of task have to be executed in parallel. For a robot \( r_i \) to be allocated to a subtask \( s_{t(j,k)} \in t_j \), the condition in (5) should be satisfied, where \( S_{st}^{(i,k)} \) is the set of skills required by \( s_{t(j,k)} \), and \( S_r^i \) is the set of skills of \( r_i \). This ensures that \( r_i \) has all the necessary skills to execute \( s_{t(j,k)} \). Let \( p_{(m,n)}^i \) be the unique permutation of the robots in \( c_{m,j} \) for the subtasks of task \( t_j \) such that there is exactly one subtask-robot pair for each subtask and no robot is paired with more than one subtask (Note that the index of subtask is changed to indicate this pairing). The value of the coalition-suitability factor \( \beta_{m,j} \) of coalition \( c_{j,m} \) for task \( t_j \) is equal to one, if there is at least one possible permutation for \( t_j \) in \( c_{m,j} \), and is equal to zero otherwise. Let \( r_i \) be the robot paired to \( s_{t(j,k)}^i \) in \( p_{(m,n)}^i \) and then the utility factor of this pairing is the sum of expertise \( soe_{t(j,k)}^i \) of \( r_i \) on \( s_{t(j,k)}^i \) given by (7). The permutation-expertise \( PE_{(m,n)}^i \) of permutation \( p_{(m,n)}^i \) on task \( t_j \) is given by (8). From all permutations in coalition \( c_{m,j} \) suitable for \( t_j \), the coalition-expertise \( CE_{m,j} \) is given by (9).
III. THE CeCoTA ALGORITHM

A. Parallel Allocation and Execution Framework

The CeCoTA algorithm proposed in this work is a market based algorithm in which the winning coalitions for the tasks are determined based on the bids placed on the subtasks by the robot. The parallel allocation and execution (PAE) framework introduced in [21] to address dynamic changes in the environment such as arrival of new tasks or new robots and to improve the overall allocation and execution time in ST-SR problems is modified for ST-MR problems and is used in the CeCoTA algorithm. In the original PAE for ST-SR problems, any robot will be allocated only one task at a time and will bid for the next task in parallel to the execution of its allocated task. In the modified PAE for ST-MR problems used in the CeCoTA algorithm presented in this paper, a robot can have two allocated subtasks (i) current-subtask and (ii) next-subtask, at any given point of time, and these may possibly be as part of two different coalitions.

Initially a robot is not part of any coalition. The robot will bid on subtasks and based on these bids coalitions are formed by an auctioneer. If the coalition, the robot is part of, wins a task, a subtask of that task will be set as the current-subtask. The robot will navigate to the task location and wait for the other robots in the coalition to join before the actual execution of the task is started. During the actual execution of this task, the robot will start bidding for its next-subtask. If a coalition, the robot is part of, wins a new task, the robot will stop further bidding and will set a subtask of the new task as its next-subtask. When the execution of the current-subtask is finished, the next-subtask will be made the current-subtask and the cycle is repeated for future allocations. By this modified PAE, the other free robots in the coalition of the main task of the robot’s next-subtask can navigate to the task location without waiting for the robot to finish its current-subtask. This can improve the overall execution time. The unallocated tasks will be iteratively allocated. Based on this PAE, the states of a robot are given in Table I and the transition between these states is given in Fig.1. A robot bids only when it is in NIAC and EXEC.

<table>
<thead>
<tr>
<th>State</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NIAC</td>
<td>The robot is not in any coalitions</td>
</tr>
<tr>
<td>NAVG</td>
<td>The robot has only current-subtask and is navigating to its location. The actual execution is not started</td>
</tr>
<tr>
<td>EXEC</td>
<td>The robot has only current-subtask and is executing it</td>
</tr>
<tr>
<td>EXEC&amp;N</td>
<td>The robot is executing its current-subtask and has a next-subtask</td>
</tr>
<tr>
<td>NEXT</td>
<td>Temporary state before the next-subtask is made the current-subtask after the completion of the previous current-subtask</td>
</tr>
</tbody>
</table>

B. Decomposing the MRTA problem to Sub-Problems

In the CeCoTA algorithm, bidding is initiated by the robots and the auctioneer identifies winning coalitions from the bids and allocates the tasks. As the problem given in Section II is NP-hard, it is divided into sub-problems, in which subsets of tasks are allocated simultaneously, in the CeCoTA algorithm to reduce the complexity.

\[
\bigcup \tau_a = T, a = 1, \ldots, k_a
\]

\[
ACE_a = \sum \sum \alpha_m^j \beta_m^j CE_m^j
\]

Let \(\tau_a\) be a subset of \(T\) being allocated in the \(a^{th}\) sub-problem, then it is possible to identify such subsets to satisfy (10), where \(k_a\) is the number of subsets. The objective of the \(a^{th}\) allocation sub-problem is maximisation of aggregate coalition expertise (\(ACE_a\)), given by (11). The values of \(\alpha_m^j, \beta_m^j\) and \(CE_m^j\) vary as defined in Section II. The set of tasks \(\tau_a\) are selected by the auctioneer based on the bidding by the robots. Any allocated task cannot be bid on later, and therefore will not be part of any future sub-problems. If a task is not allocated in a sub-problem, it will be considered in future allocations based on the received bids. Solving all sub-problems will result in the allocation of all tasks.

\[
\text{maximise} \sum A CE_a = \sum \sum \alpha_m^j \beta_m^j CE_m^j
\]

Based on this, the main objective given in (1) can be modified to (12). The processes involved in the CeCoTA algorithm to identify and solve the allocation sub-problems are explained in Sections III-C and III-D.

C. Bidding Process

Assume the CeCoTA algorithm is solving the \(a^{th}\) sub-problem for allocation of tasks in \(\tau_a\). The robots bid directly on the subtasks and based on these bids the auctioneer identifies the set of tasks \(\tau_a\) to be allocated in the current sub-problem.

\[
est_t^{(j,k)} = \begin{cases} 
    \frac{D_t^{j,k}}{v_t}, & \text{if } r_i \in \text{NIAC} \\
    \text{eff} + \frac{D_t^{j,k}}{v_t}, & \text{if } r_i \in \text{EXEC} \text{ of } t_g
\end{cases}
\]

A bid by a robot \(r_i\) on a subtask \(st_{(j,k)} \in t_j\) is based on the estimated execution start time \(est_t^{(j,k)}\) of \(st_{(j,k)}\) and the sum of expertise \(soc_t^{(j,k)}\) of \(r_i\) on the skills required for \(st_{(j,k)}\). The \(est_t^{(j,k)}\) is given by (13) where \(D_t^{j,k}\) is the distance from the current location of \(r_i\) to \(t_j\), \(D_t^{k}\) is the distance from task \(t_k\) to the task \(t_j\), \(\text{eff}\) is the estimated finishing time of the task \(t_k\) and \(v_t\) is the average velocity of \(r_i\). It is assumed that the execution time of a task is given and from this and the time
at which execution of $t_k$ was started, $est^S$ can be calculated. The sum of expertise $soe_{t_{(j,k)}}^r$ of the robot $r_i$ on the subtask $st_{(j,k)}$ is given by (7).

![Fig. 2. Message sequencing between the auctioneer and a robot in the bidding process](Image)

The messaging between a robot and the auctioneer during the two stage bidding process is shown in Fig. 2. Initially a robot calculates its est on subtasks, which satisfy (5), of all unallocated tasks in $T$. The robot identifies the main task of the subtask for which it has the minimum $est$, and bids on all subtasks, which satisfy (5), of this task. The auctioneer receives bids from different robots and communicates back which tasks are being bid on. When a robot receives this information, it will place bids on the subtasks of these tasks, thereby helping other robots to form coalitions. This step removes the possibility of deadlock. At this stage the auctioneer has the bids from all free robots on all tasks, thereby helping other robots to form coalitions. Allocation of this small set of tasks forms an allocation sub-problem of the CeCoTA algorithm.

### Allocation Process

The auctioneer solves the allocation sub-problem by finding non-conflicting coalitions to meet the objective given by (11) and allocate the tasks. This allocation process is given in Algorithm 1, which takes the bids, $\tau_a$ and the coalitions $\mathcal{C}$ for the tasks in $\tau_a$. In line 1 of Algorithm 1, $GetCoExpertise$ is the function that calculates the coalition expertise ($CE$) of all coalitions in $\mathcal{C}$. The coalition expertise $CE_{c_m}^i$ of a coalition $c_m^i$ for a task $t_j$ is given by (9). $GetMinStartTime$ in the line 2 is the function to calculate the estimated start time ($EST$) of the coalitions. The estimated start time $EST_{c_m}^j$ of a coalition $c_m^j$ on task $t_j$, is given by

$$EST_{c_m}^j = \max \left( est_{t_j}^{(j,k)} \right), \forall t_j \in T, \forall r_i \in c_m^j$$

(14)

For each task in $\tau_a$, a neighbourhood of near-best coalitions are identified using the function $GetENeighbour$ in line 3. For a task $t_j \in \tau_a$, let $CE_{max}$ be the maximum $CE$ corresponding to the best coalition. A neighbourhood of near-best coalitions for $t_j$ is defined as

$$N_{\varepsilon}^i = \left\{ c_m^i : \left(1 - \frac{CE_{c_m}^i}{CE_{max}}\right) \leq \varepsilon, \forall c_m^i \in \mathcal{C} \right\}$$

(15)

where $\varepsilon (0 < \varepsilon < 1)$ is the neighbourhood width and $\mathcal{C}$ is the set of all coalitions for $t_j$.

In line 4 of the Algorithm 1, $BuildTree$ is the function that builds the coalition tree, which is a binary-tree of coalitions from $N_{\varepsilon}$ for finding the winning coalitions for tasks in $\tau_a$. This tree has a dummy root node and each of the remaining levels corresponds to a task in $\tau_a$. Each node at a level corresponds to a coalition for the task at that level. Each path from the root to any of the leaves in the coalition tree is a sequence of disjoint coalitions; i.e., there are no common robots between any two coalitions on the same path ($c_1 \cap c_2 = \emptyset, \forall c_1$ and $c_2$ on the same path). An example of this coalition tree is shown in Fig. 3.

![Fig. 3. Coalition tree for selecting combinations of task-coalition pairs](Image)
Let $c^m_m$ be a coalition node for the task $t_j$ on tree. The set of all robots $R_j$ already used in the path up to and including this node is given by (16). Let $t_k$ be the task next to $t_j$, $c^m_m$ can have at least one child node and at most two child nodes. The permanent child node is given by (17). If $C^g$ given by (18) is the set of all coalitions in $N^g$ with no robots common with $R_j$, then the second child node is given by (19). If $child_2 = \{ \}$, the second child is not added and if $|child_2| > 1$, the coalition with the smallest index is selected as the second child node.

Example near-optimal coalitions ($\epsilon = 0.1$) with their $CE$ and $EST$ for a sub-problem are given in Table II and the corresponding coalition tree is given in Fig. 3. While populating this tree, the coalitions at level 1 corresponding to $t_3$ are $c^3_3$ and $c^3_{12}$, based on the $EST$. While adding the child nodes at level 2 to these nodes, the coalitions $c^6_3$ is selected as the child node of $c^3_3$ and $c^6_{12}$ is selected as the child node of $c^3_{12}$, along with $c^6_{34}$ based on (19). Although $est^6_{12} < est^6_{34}$, the coalition $c^6_{12}$ cannot be a child node of $c^3_{12}$ as the robots in this coalition ($r_1$ and $r_2$) are already used on the path up to $c^3_{12}$. Similarly, level 3 nodes are also added.

<table>
<thead>
<tr>
<th>Task</th>
<th>Coalitions in $N_e$</th>
<th>$CE$</th>
<th>$EST$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_3$</td>
<td>$c^3_{12}; {t_1, t_2}$</td>
<td>2.70</td>
<td>300s</td>
</tr>
<tr>
<td></td>
<td>$c^3_{12}; {t_3, t_4, t_5}$</td>
<td>2.90</td>
<td>350s</td>
</tr>
<tr>
<td>$t_6$</td>
<td>$c^6_{12}; {t_1, t_2}$</td>
<td>3.20</td>
<td>320s</td>
</tr>
<tr>
<td></td>
<td>$c^6_{12}; {t_3, t_4}$</td>
<td>3.00</td>
<td>400s</td>
</tr>
<tr>
<td>$t_7$</td>
<td>$c^6_{12}; {t_1, t_3}$</td>
<td>3.80</td>
<td>280s</td>
</tr>
<tr>
<td></td>
<td>$c^6_{12}; {t_2, t_5}$</td>
<td>3.60</td>
<td>300s</td>
</tr>
<tr>
<td></td>
<td>$c^6_{12}; {t_2, t_4}$</td>
<td>4.00</td>
<td>400s</td>
</tr>
</tbody>
</table>

In line 5 of Algorithm 1, the best path from the root node to any leaf node is found using the A* algorithm, so as to maximise $ACE_e$, given by (11). In the A* search, the past path coalition-expertise is the sum of $CE$s at the previous nodes on the path and the future path coalition-expertise is estimated using an admissible heuristic function. Let $c^m_m$ be a node in an intermediate level in the coalition tree for a task $t_k \in \mathcal{T}_e$ and $\mathcal{T}_e$ be the set of all tasks at levels above this node and $t_k$, then the heuristic function at this node is given by

$$h(c^m_m) = \sum_{t_j \in (\mathcal{T}_e \setminus \mathcal{T}_e)} CE_{max}$$

where $\mathcal{T}_e \setminus \mathcal{T}_e$ is the set of tasks at the lower levels of the node and $CE_{max}$ is the maximum $CE$ at these levels. This heuristic is admissible, as it does not underestimate the $CE$ from a node and therefore the solution of the A* algorithm is optimal for a given coalition tree.

For the coalition tree in Fig.3 the optimal path is $Root \rightarrow c^3_{12} \rightarrow c^6_{12} \rightarrow c^7_{12}$ with an $ACE$ value of 5.70. The corresponding allocations are $(c_1, c^3_{12})$ and $(c_6, c^6_{12})$, and $r_7$ is unallocated in this sub-problem. Once these tasks are allocated to these coalitions, the robots in these coalitions withdraw their other bids, here on the subtasks of $r_7$. The robots set subtasks from the allocated task as the current-subtask or next-subtask based on their execution status as explained earlier. When the robots start the actual execution of the tasks assigned in this sub-problem, they will start the bidding process for their next-subtasks. The robots will consider the task $r_7$ which was unallocated in this sub-problem along with the any other unallocated tasks in the future bidding processes. When all allocation sub-problems are solved, the objective in (12) is achieved.

**E. Performance Analysis**

**Feasibility:** The solution of an allocation sub-problem is feasible if the robots in the coalition can be assigned to the task, if a task is not allocated to more than one coalition, and if a robot (as part of same of different coalitions) is not assigned more than one task in the same sub-problem. As the robots bid truthfully, maximising $ACE$ identifies from the bids can execute the allocated task. Each level in the coalition tree corresponds to a single task, and in the optimal path has only one node from each level, as there are no looping edges in a binary tree. So only one coalition will be allocated to a task. If a task is already allocated, the robots will not bid on the task, and therefore will not be part of any future sub-problems and will not be reallocated to a different coalition. As there are no common robots between any two coalitions on the same path, a robot will not be assigned to more than one task as part of different coalitions. Therefore the nodes on the selected path correspond to allocation of disjoint coalitions to tasks in $\mathcal{T}_e$ and the allocations are feasible.

**Optimality:** As the robots bid truthfully, maximising $ACE$ maximises the overall task execution performance. As the coalition tree is built from $N_e$, the winning coalition of a task is near-optimal for the task. For a given tree, the A* algorithm finds the optimal path when the heuristic function is admissible. As the heuristic (20) used here is admissible, and the coalitions are near-optimal for their tasks, the selection of coalitions for the tasks based on the nodes in the selected path is near-optimal.

**Complexity:** The complexity of the CeCoTA is same as the complexity of A* algorithm. In the worst case, the complexity of A* algorithm is polynomial on the number of edges of the tree. Being a binary tree, the worst case of the coalition tree is when it is a perfect binary tree. Let there be $n$ levels ($n$ tasks) in a perfectly binary coalition tree. The maximum number of nodes is $2^{n+1} - 1$ and the maximum
number of leaf nodes is $2^n$. As the maximum number of tasks being bid by the robots to form a sub-problem is limited to the number of free robots, the value of $n \leq \aleph_r$. However, as only disjoint coalitions are allowed on a path, the number of nodes is always less than $2^{n+1} - 1$ and the number of leaf nodes is always less than $2^n$. This further reduces the complexity of the algorithm. The search tree used in [22] has similarities to the coalition tree used in the CeCoTA algorithm. However, in [22] ASyMTRe is used for coalition formation and as all coalitions are used to build the search tree can result in a large number of nodes.

**Dynamic environments:** The PAE framework helps to address changes in the environment without affecting the performance compared to time-extended allocation algorithms [21]. As the CeCoTA uses incremental allocations of previously unallocated tasks while the robots complete the allocated tasks as part of the PAE, when a change in the environment such as addition of new tasks or new robots takes place, the CeCoTA can easily handle it by considering the new tasks or robots for future allocation sub-problems. In comparison for the case of time-extended allocation algorithms, all unfinished tasks would be reallocated for the current environment state, resulting in redundant complex computations.

**IV. DISTRIBUTED APPROACH**

Centralised approaches have various drawbacks such as poor scalability due to the high computational complexity and communication bandwidth required for large number of robots, and reduced robustness as the central controller is a single point of failure, compared with decentralised or distributed approaches. In order to address these, the CeCoTA algorithm can be extended to a distributed approach by completely removing the auctioneer and implementing the computations of the auctioneers in all robots, and then using inter-robot communication to resolve any conflicts. In this distributed approach, the computational complexity remains the same and is distributed among the robots. The detailed discussion on this distributed approach will be published separately.

**V. EMPIRICAL EVALUATIONS**

The CeCoTA algorithm was implemented in Python and evaluated in a series of experiments. For comparison, a centralised MRTA algorithm using a GA was also implemented. The GA based MRTA algorithm sequentially allocates multiple tasks as in the proposed algorithm, by encoding each chromosome to represent possible conflict-free allocations. Both crossover and mutation were used as genetic operators. In crossover between two chromosomes, the coalitions for tasks were exchanged between chromosomes and in mutation a robot in a coalition was replaced by unused robots. The $ACE$ of the coalitions in a chromosome is calculated as the fitness value of the chromosome and the GA maximises it. During the selection process, elitism was used for 20% of the population and the remaining 80% was chosen using roulette wheel selection. The experiments were carried out in a simulated obstacle free environment of size $30m \times 30m$ to compare the performance of the algorithms. In these experiments, a set of tasks were allocated to a group of heterogeneous robots. All tasks were assumed to have a constant execution time.
The number of tasks and robots in the MRS was varied from 10 to 30. The number of subtasks in each task is randomly selected from [2, 4]. The performance measures used to compare the performances of these algorithms are the aggregate coalition expertise (ACE) for allocation of all tasks, the total distance (TD) travelled by the robots and the overall execution time (OET) based on the allocations. The OET value includes bidding and allocation time, time to navigate to each task location, waiting time for the other robots in the coalitions to join and time for actual execution. All experiments were repeated five times and the average performances were calculated for comparisons. In the CeCoTA algorithm, $\varepsilon = 0.1$ was used to find $N_e$.

<table>
<thead>
<tr>
<th>Table III</th>
<th>Average performance (in percentage) in the CeCoTA w.r.t the GA based allocation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$r$</td>
<td>ACE</td>
</tr>
<tr>
<td>10</td>
<td>99.939</td>
</tr>
<tr>
<td>15</td>
<td>98.772</td>
</tr>
<tr>
<td>20</td>
<td>98.964</td>
</tr>
<tr>
<td>25</td>
<td>100.122</td>
</tr>
<tr>
<td>30</td>
<td>100.084</td>
</tr>
</tbody>
</table>

Figs.4(a-c) show the average ACE, TD and OET values respectively for the allocations using the CeCoTA algorithm and Figs.4(d-f) show the average ACE, TD and OET values for the allocations using the GA. When ACE values from Fig.4(a) and Fig.4(d) are compared, the CeCoTA algorithm has resulted in nearly equal ACE values, which is the main objective of the problem addressed (12). In addition, the CeCoTA algorithm reduces the TD and OET values compared to the GA. When the TD values are compared from Fig.4(b) and Fig.4(e), it can be seen that the robots travel less distance when the allocations are carried out using the CeCoTA algorithm. This can in turn result in energy efficiency of the MRS. The OET values can be compared from Fig.4(c) and Fig.4(f). The performance of the CeCoTA algorithm for these allocations is compared to that of the GA based allocation, considering the performance of the GA based allocation as 100.0, in Table III.

VI. CONCLUSION

The CeCoTA algorithm proposed here address the allocation of tightly-coupled MR tasks in a heterogeneous MRS. It has been shown that the allocations using this algorithm are near-optimal (based on ACE) and also minimise the distance travelled by the robots and the overall execution time. The scalability of the CeCoTA algorithm to allocate a large number of tasks to a moderately large MRS is empirically shown. Results from the experiments show the feasibility of the CeCoTA algorithm, nearly equal ACE and improvements in the OET and the TD when compared to a GA based allocation algorithm. The decomposition of tasks into subtasks helps to form closely coupled coalitions and allocate the tasks based on the capabilities of the robots. While the existing MRTA algorithms for ST-MR problems mainly focus on the coalition formation for each task and allocate the tasks sequentially, the simultaneous allocation of more than one task in parallel using the coalition tree, in the CeCoTA algorithm results in more balanced allocations. The CeCoTA algorithm addresses only the allocation of tasks and the actual cooperative execution of a task is beyond its scope. In future, additional constraints such as inter-task dependencies and task prioritisation will be added to the definition of tasks.

REFERENCES