Defence Science Journal, Vol. 68, No. 2, March 2018, pp. 175-182, DOI : 10.14429/dsj.68.11187

Multi-objective Optimisation of Multi-robot Task Allocation with Precedence Constraints

K. Padmanabhan Panchu[#,*], M. Rajmohan[#], R. Sundar[@], and R. Baskaran[#]

[#]*Department of Industrial Engineering, College of Engineering Guindy, Anna University, Chennai - 600 025, India*
[@]*Department of Production Engineering, National Institute of Technology, Tiruchirappalli - 620 015, India*
[*]*E-mail: panchu81@gmail.com*

ABSTRACT

Efficacy of the multi-robot systems depends on proper sequencing and optimal allocation of robots to the tasks. Focuses on deciding the optimal allocation of set-of-robots to a set-of-tasks with precedence constraints considering multiple objectives. Taguchi's design of experiments based parameter tuned genetic algorithm (GA) is developed for generalised task allocation of single-task robots to multi-robot tasks. The developed methodology is tested for 16 scenarios by varying the number of robots and number of tasks. The scenarios were tested in a simulated environment with a maximum of 20 robots and 40 multi-robot foraging tasks. The tradeoff between performance measures for the allocations obtained through GA for different task levels was used to decide the optimal number of robots. It is evident that the tradeoffs occur at 20 per cent of performance measures and the optimal number of robot varies between 10 and 15 for almost all the task levels. This method shows good convergence and found that the precedence constraints affect the optimal number of robots required for a particular task level.

Keywords: Multi-robot task allocation; Multi-robot task sequencing; Foraging tasks; Multi-objective optimisation, Genetic algorithm; Taguchi DOE

NOMENCLATURE

T_i^p	Task '*i*' with priority *p*, *i=1 to n* and *p= 1 to q*
$\mathcal{R}_j$	Robot '*j*' *j=1 to m*
$\mathcal{D}_k$	Destination location *k=1 to s*
$n_\mathcal{T}$	Number of tasks
$n_\mathcal{R}$	Number of robots
$n_\mathcal{P}$	Population size
n_G	Number of generations
$\mathcal{P}_\mathcal{M}$	Mutation probability
$\mathcal{P}_\mathcal{C}$	Crossover probability
f_t	Fitness function
$\mathcal{C}_t^{T_i^p}$	Task completion time for each task 'i' with priority 'p'.
$T_{T_i^p}^{\mathcal{R}_j}$	Travel time of robot $\mathcal{R}_j$ from its current location to source location of the task T_i^p
$T_{\mathcal{R}_j^{\mathcal{D}_k}}^{T_i^p}$	Travel time of robot $\mathcal{R}_j$ from the source location of task T_i^p to destination location $\mathcal{D}_k$

1. INTRODUCTION

Multi-Robot Systems (MRS) finds real-life applications in automated material handling at warehouses, war front assistance, office support system, environmental cleansing and the robotic scientists are keen in exploring the usage in many other areas. Physical foraging tasks like cooperative object transportation reasonably demand the need for multiple robotic units for task execution. MRS exhibits scalable, fault tolerant and expedited task completion capabilities when compared to the single standalone robot system. However, a carefully planned task allocation among the individual robotic units plays a significant role in realising the efficiency of the MRS. Multi-robot task allocation (MRTA) problems are to determine the best possible assignment of robots to each task and to find the sequence of the tasks for each of the robot to minimise the total task completion time. An improper allocation of the robots to such tasks would undoubtedly lead to extended completion time and excess resource utilisation. At times it may end in a deadlock situation, when two jobs may be waiting invariably for two different robots waiting for each other. Apart from allocating the robots to a given set-of-tasks, finding the optimal $n_\mathcal{R}$ to be deployed would help the system to operate at minimal cost and with high efficiency.

2. RELATED WORK

The detailed classification[1] of MRTA problems available in the literature are based on

(i) Type of robots (single task (ST) *vs.* multiple task (MT) robots)
(ii) Type of tasks (single-robot (SR) task and multi-robot (MR) tasks), and
(iii) Type of assignments (instantaneous assignment (IA) and time-extended assignment (TA).

Further classification[2] on time-extended assignments includes time windows, synchronisation and precedence constraints. Having such a broad classification, researchers attempt to solve numerous problems in the MRTA domain. The

Received : 15 February 2017, Revised : 23 October 2017
Accepted : 22 November 2017, Online published : 13 March 2018

problem complexity also varies with the application for which the robots are deployed and also based on the robot capabilities. Ultimately, task allocation is done to execute the task on hand with available robots. However, the task allocation depends significantly on the objectives that the user tries to optimise. Different objectives were considered such as minimisation of energy expenditure[3], maximisation of utility value[4,5] of robots, minimisation of task completion time[6,7], minimisation of distance travelled by the robots[8,9] or at times if the robots speed are variable, robot travel time considered. Some problems were solved are in single objective in nature some are multi-objective types. Some multi-objective optimisation approaches considered rewards, path cost and damage to the vehicle as objectives[10] and used a weighted technique to obtain a solution.

A review of MRTA algorithms and comparison of market-based approaches along with the simulated annealing and Genetic algorithm[11] concludes that the latter performs well. Algorithm time complexities between Hungarian Algorithm and GA were compared[12] and found that GA performs better while scaling up robotic units. GA with the different mutation parameters were analysed[13] and found the inversion mutation perform better when compared to the swap mutation. Other related works for MRTA includes Ant colony optimisation based task allocation[14] and comparison between Tabu search, Simulated Annealing and random search methods[15] for different tasks. Social welfare based task allocation method[16] minimises the resource consumption and maximises the completion rate of tasks. Probabilistic task allocation method[17] under uncertain costs conditions and with risk preference concludes that risk has no effect on optimal assignment but, uncertainty plays a significant role in deciding the optimality. The two-level distributed method[18] with the centralised method for production planning and task allocation using multiple robots shows that the decentralised approach invites much production cost due to information scarcity. But the centralised method involves higher robot distance cost but still proved to be superior. Decentralised sub-planning and centralised optimisation of task allocation procedure[19] reduces the burden on the task manager, but the computation complexity of this method increases as the robots are scaled up. A general classification of solution methodologies for MRTA problems includes centralised and decentralised approaches. Decentralised approaches outperform centralised methods in finding the solutions to the problem in a short time. Whereas, centralised approaches perform well in the global optimisation grounds.

The [ST-MR] configuration is addressed by very few researchers[20,21] and the use of conventional optimisation methods to solve problems of [ST-MR] setup involves many variables, and hence it is computationally intensive to find a solution. Moreover, the effect of precedence constraints in [ST-MR] problems and objectives related to robots such as a number of robots utilised, waiting time and idle times of robots were given least importance when compared to task-related goals. This paper proposes an evolutionary optimisation technique to find the solution in a centralised approach to MRTA problem considering, precedence relationship including both robot centric and task-centric objectives.

The following are the notable contributions of this work:

(i) GA based methodology for MRTA problem with new combination of multiple objectives.

(ii) Statistical method to identify the best levels for GA parameters for solving MRTA problems.

(iii) Explored a way to identify the optimal number of robots for a given set-of-tasks with precedence constraints.

3. PROBLEM STATEMENT

At any given instant of time, set-of-tasks with precedence constraints are represented as $\mathbb{U}_T = \{\mathcal{T}_1^1,\ \mathcal{T}_2^1, \ldots.. \mathcal{T}_a^1,\ \ \mathcal{T}_1^2,\ \mathcal{T}_2^2, \ldots.. \mathcal{T}_b^2, \ldots\ldots \mathcal{T}_1^q,\ \ \mathcal{T}_2^q, \ldots\ldots\ldots \mathcal{T}_y^q\}$ where *a, b* and *y* are $n_{\mathcal{T}}$ in each precedence level *P=1 to q*. Homogeneous robots are to be allocated to identical foraging tasks at various locations. Each prioritised foraging task $\mathcal{T}_i^P$ requires *m* robots for its completion. Let the set of homogeneous robots available for completing the tasks are $\mathbb{U}_R = \{\mathcal{R}_1, \mathcal{R}_2, \ldots\ldots.., \mathcal{R}_n\}$. The source location of tasks $\left(x_{T_i}^S, y_{T_i}^S\right)$ and the $n_{\mathcal{R}}$ required to complete the task are known in advance. Each allocated task will have a group of robots that get assembled at the site and transport the material to the nearest destination point $\left(x_{T_i}^D, y_{T_i}^D\right)$. Once the task is completed, the robot is free to take the next allocated task. It is assumed that the robot moves at a speed of 2 unit distance per time unit, at no load condition and one unit distance per time unit, at the loaded state.

This paper considers the following four minimisation objectives for optimisation.

(i) Total task completion time $\mathcal{C}_t^{\mathcal{T}}$ – time-taken for completion of all the task in all the priorities

(ii) Aggregated robot travel time $T_t^{\mathcal{R}}$ - total travel-time taken by all the robots to complete all the tasks

(iii) Aggregated robot waiting time $\mathcal{W}_t^{\mathcal{R}}$ - if a task is a multi-robot task, then the robots that arrive at task site have to wait until all the other robots required to perform a task arrives at the site. Addition of all such waiting time, of all the robots and for all the instances, is Aggregated Robot Waiting Time.

(iv) Aggregated robot idle time $\mathcal{I}_t^{\mathcal{R}}$ - an individual robot may go unallocated between tasks due to precedence constraints. Time for which the robot stays unallocated is considered as robot idle time. Aggregated idle time of all the robotic units is considered as aggregated robot idle time.

The simulated environment of size 100 m x 100 m is considered for this study. A maximum of 20 robots was deployed to complete 40 multi-robot foraging tasks with precedence constraints. Figure 1 shows robots, tasks with a number of robots required for completion and the destination locations.

4. PROPOSED METHODOLOGY

MRTA problem is one with significantly large solution space and hence, it is computationally intensive for finding the optimal solution. Non-conventional optimisation methods like genetic algorithm (GA) is a better choice for solving problems

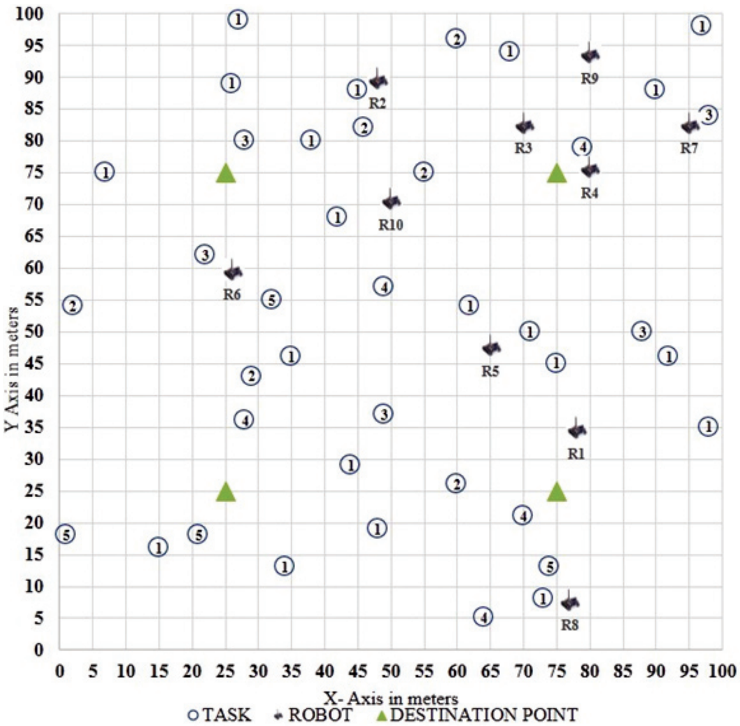

Figure 1. Environment: a sample view.

with a large solution space. GA works better in avoiding local optima and searches for global optima by a random search strategy. Application of elitism concept in GA preserves the best individuals.

4.1 Fitness Function

GA requires a fitness function for selection of the best solution among the various randomly generated ones. The fitness function for a multi-objective optimisation is represented as a single weighted objective function. The multi-objective fitness function is given as Eqn. (1).

$$f_t = \lambda_1 \mathcal{C}_t^{\mathcal{T}} + \lambda_2 T_t^{\mathcal{R}} + \lambda_3 \mathcal{W}_t^{\mathcal{R}} + \lambda_4 \mathcal{I}_t^{\mathcal{R}} \tag{1}$$

where $\lambda_1, \lambda_2, \lambda_3 \, and \, \lambda_4$ are the weights of the respective functions. Weighted sum approach to solve multi-objective optimisation is one of the computationally efficient procedures in generating non-dominated solutions[22, 23]. Some studies[24, 25] obtained best results by using equal weights to solve the multi-objective problems. This work considers a generalised task allocation problem with multiple objectives but not oriented to any specific application. Hence, same weight is considered for multiple objectives. However, when the problem is solved to a specific application appropriate weights may be assigned to multiple objectives.

The following steps were performed to evaluate $\mathcal{C}_t^{\mathcal{T}}$

Step 1: Arrange tasks based on precedence constraints.

Step 2: Divide the tasks within the same precedence constraints into two sets $\mathbb{C}_T$ and $\mathbb{D}_T$. Where set $\mathbb{C}_T$ consists of tasks that do not contain any resource (robot) constraint and can be performed in parallel. All the remaining tasks were taken into the set $\mathbb{D}_T$. The robots required for completing the task in the set $\mathbb{C}_T$ were kept in the set $\mathbb{E}_R$ and the remaining robots are kept in $\mathbb{F}_R$.

Step 3. Arrange the tasks in set $\mathbb{C}_T$ in ascending order based on the required time to complete the task. Task completion time for each task 'i' with priority 'p' denoted by $\mathcal{C}_t^{\mathcal{T}_i^p}$ and calculated based on travel time. The travel time in Eqn (2) consists of two components (i) no-load travel time of robot $\mathcal{R}_j$ from its current location to source location of the task $\mathcal{T}_i^p$ denoted by $T_{\mathcal{T}_i^p}^{\mathcal{R}_j}$ (ii) loaded travel time of robot $\mathcal{R}_j$ from the source location of task $\mathcal{T}_i^p$ to destination location $\mathcal{D}_k$ indicated by $T_{\mathcal{R}_j^{\mathcal{D}_k}}^{\mathcal{T}_i^p}$

$$\mathcal{C}_t^{\mathcal{T}_i^p} = argmax\left(T_{\mathcal{T}_i^p}^{\mathcal{R}_j} * \mathcal{A}_j^{\mathcal{T}_i^p}\right) \forall j + T_{\mathcal{D}_k}^{\mathcal{T}_i^p} \tag{2}$$

where $\mathcal{A}_j^{\mathcal{T}_i^p} = \begin{cases} 1 \text{ if task } \mathcal{T}_i^p \text{ is allocated to Robot } j \\ 0 \text{ otherwise} \end{cases}$

Step 4. Delete all the completed task from the set $\mathbb{C}_T$ and add the particular robots used for the task back to set $\mathbb{F}_R$

Step 5. Verify the availability of robots in the set $\mathbb{F}_R$ to perform tasks in set the $\mathbb{D}_T$ and if it is so, it was inserted in the ascending order based on $\mathcal{C}_t^{\mathcal{T}_i^p}$ in the set $\mathbb{C}_T$ accordingly robots from the set $\mathbb{F}_R$ is added to the set $\mathbb{E}_R$.

Step 6. Repeat steps 1 to 3 until all tasks in a particular priority are completed.

Step 7. The completion time for the last job was the completion time for the first priority tasks.

Step 8. Repeat steps 1 to 5 until completion of all the tasks.

Step 9. The completion time of all the tasks was calculated as follows

After performing the above steps C_t^T, T_t^R, $W_t^R \, and \, I_t^R$ is calculated by Eqns. (3), (4), (5), and (6) respectively

$$\mathcal{C}_t^{\mathcal{T}} = \sum_{p=1}^{q}\sum_{i=1}^{m} \mathcal{C}_t^{\mathcal{T}_i^p} \tag{3}$$

$$T_t^{\mathcal{R}} = \sum_{p=1}^{q}\sum_{i=1}^{n}\sum_{j=1}^{m} T_{\mathcal{T}_i^p}^{\mathcal{R}_j} * \mathcal{A}_j^{\mathcal{T}_i^p} + \sum_{p=1}^{q}\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{s} T_{\mathcal{R}_j^{\mathcal{D}_k}}^{\mathcal{T}_i^p} * \mathcal{B}_k^{\mathcal{T}_i^p} \tag{4}$$

where $\mathcal{B}_k^{\mathcal{T}_i^p} = \begin{cases} 1 \; if \; task \; \mathcal{T}_i^p \; is \; allocated \; to \; Destination \; \mathcal{D}_k \\ 0 \; otherwise \end{cases}$

$$\mathcal{W}_t^{\mathcal{R}} = \sum_{p=1}^{q}\sum_{i=1}^{n}\sum_{j=1}^{m} \mathcal{W}_{\mathcal{R}_j}^{\mathcal{T}_i^p} \tag{5}$$

where $\mathcal{W}_{\mathcal{R}_j}^{\mathcal{T}_i^p}$ is the waiting time of robot $\mathcal{R}_j$ at task $\mathcal{T}_i^p$

$$\mathcal{I}_t^{\mathcal{R}} = \sum_{j=1}^{m} \mathcal{J}_{\mathcal{R}_j} \tag{6}$$

where $\mathcal{J}_{\mathcal{R}_j}$ is the total idle time of robot $\mathcal{R}_j$ between various tasks.

4.2 Chromosome Design

In GA, the chromosome represents the actual solution to the problem. So each chromosome provides the allocation of a set of robots to all the tasks. The chromosome design is of a matrix form which contains, *n* rows and *m* columns. The first row denotes the tasks arranged according to the priorities. When there are *m* tasks with *p* priorities and *a, b* and *y* number of tasks in each priority respectively, the total number of tasks equals the sum of the number of tasks in each priority. The maximum number of rows in the matrix is equal to the maximum number of robots required for any task. So the first row of each column is allocated to a task, and its following rows are allocated to the robots necessary for the task. A sample chromosome is as shown in Fig. 2. A randomly generated chromosome may offer an infeasible solution (i.e., the same robot is allocated more than once for a particular task). The random permutation generation method used in this work ensures the generation of a feasible chromosome (i.e., without assigning the same robot for any specific task).

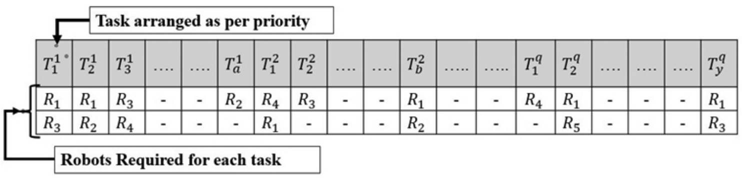

Figure 2. Chromosome design.

4.3 Crossover

In GA, crossover operator is used for recombining the best solutions to form two new solutions. Crossover is performed with two parent solutions to form two new child solutions. Crossover is done within priorities. Multi-point crossover is applied at random sites for each priority as shown in Fig. 3. Child 1 is formed by joining the first part of parent 1 and second part of parent 2. Whereas for child two the procedure is done vice versa.

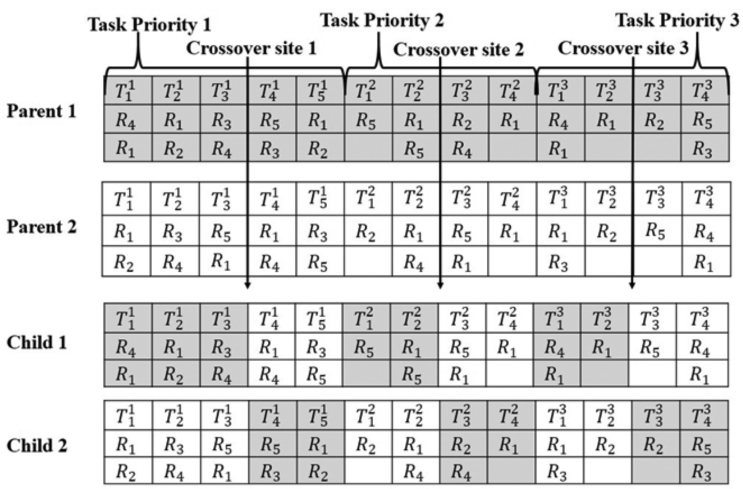

Figure 3. Crossover procedure.

4.4 Mutation

Crossover operator enables the algorithm to choose better children by swapping of a set of genes at the crossover point, but it cannot change the individual gene of a chromosome, Whereas Mutation alters the specific genes. The mutation procedure isolates the robots assigned to the task in set A, and the remaining robots to another set B. Each element in set A is swapped with a randomly selected element in Set B. Swaps are subjected to mutation probability. If the task requires all the robots, then Set B will be a null set, and hence mutation has no meaning.

4.5 Optimal Level Selection for GA Parameters

The solution quality of the GA depends on operating parameters $n_{\mathcal{P}}$, $\mathcal{P}_{\mathcal{C}}$, $\mathcal{P}_{\mathcal{M}}$, and n_G. To study the effect of parameters on the solution and to identify the best levels for the parameter, the following experimental design has been constructed. The parameters are at three levels each as given in Table 1. L_9 Orthogonal Array (OA) is a standard array suitable for a analysing the effect of 4 factors with three levels. The maximum number of columns available in standard L_9 OA is 4. The columns in L_9 OA are sufficient enough to accommodate only four factors and interaction between factors cannot be accommodated. In order to analyse the interaction effects, additional columns are required. Hence, L_{27} OA is chosen to accommodate all the four main factors and two interactions effects.

Table 1. GA parameters and their levels

Levels	Parameters			
	$n_{\mathcal{P}}$	$\mathcal{P}_{\mathcal{C}}$	$\mathcal{P}_{\mathcal{M}}$	n_G
Level 1	25	0.6	0.01	50
Level 2	50	0.7	0.03	100
Level 3	75	0.8	0.05	150

A pilot run of experiments was conducted using the above levels, and the effect of GA parameters is studied. Table 2 shows the sample ANOVA for performing 20 tasks with 15 robots. Statistical analysis reveals that $\mathcal{P}_{\mathcal{M}}$ and n_G are the most significant factors with a contribution percentage of 66.57 per cent and 20.85 per cent, respectively. All the other factors are insignificant.

The OA based experiments were performed with the different robot to task ratios to check whether the change affects the levels of GA parameters. Table 3 summarises the optimal levels for GA parameters for the various robot to task ratios. Also, it shows that the level 1 (0.01) for $\mathcal{P}_{\mathcal{M}}$ and level 3 (150) for n_G being chosen as the best level for the different robot to task ratios. The optimal levels obtained matches with the work done by Khuntia[26], *et al.* for multi-robot task allocation.

4.6 Genetic Algorithm-Pseudo Code

The pseudo code for the GA is presented below, and Fig. 4 shows the sample convergence plot based on the

Table 2. Analysis of variance (ANOVA) table

Factor	*SS*	DOF	μ_{SS}	F_{Calc}	F_{tab}	Contribution (per cent)	Significance
$n_{\mathcal{P}}$	0.04	2.00	0.02	3.29	4.10	3.50	No
$\mathcal{P}_{\mathcal{C}}$	0.01	2.00	0.01	0.97	4.10	1.03	No
$\mathcal{P}_{\mathcal{M}}$	0.83	2.00	0.42	62.59	4.10	66.57	Yes
n_G	0.26	2.00	0.13	19.61	4.10	20.85	Yes
$n_{\mathcal{P}}$ X n_G	0.02	4.00	0.00	0.68	3.47	1.45	No
$\mathcal{P}_{\mathcal{M}}$ X n_G	0.02	4.00	0.00	0.61	3.47	1.29	No
Error	0.07	10.00	0.01			5.31	
Total	1.25	26.00				100.00	

fitness value.

// Input
$n_{\mathcal{R}}$, $n_{\mathcal{T}}$, $\mathcal{D}_k$, $n_{\mathcal{P}}$, n_G, $\mathcal{P}_{\mathcal{M}}$, $\mathcal{P}_{\mathcal{C}}$
//Generate Initial Population
Set *i*=1
Repeat
  Generate a Random Chromosome *C (i)*
  i=i+1
Until (*i*! $= n_{\mathcal{P}}$)
// Evaluation of Initial Population
Set *j=1*
Repeat
  Calculate f_n *(j)* for *C (j)*
  j=j+1
Until (*j*! $= n_{\mathcal{P}}$)
// Mating Pool Formation
  Sort *C (j)* descending based on f_n *(j)*
  Save *C (1)*as the Best
  For m=1 to 10
    M (m) =C (m)
  End For
// Generation Loop
  Set *x=0*
Repeat
  //Crossover
    Set *k=0*
    For *P*= 1 to 10
      For *Q*= *P*+1 to 10
      Child C (k) & C (k+1) = C(P)
X C(Q)

      *k=k+*2
      End For
    End For
    Calculate *C (k)*
    Sort *C (k)* descending based on f_n *(k)*
    Save *C (1)*as the Best
  //Mutation
    Perform gene wise Mutation *C (k)*
    Calculate $f_n(k)$ *C (k)*
    Sort *C (k)* descending based on $f_n(k)$
    Save *C (1)*as the Best
  //Replacing the Mating pool population
    For *k, P*=1 to 10
    C (P) =C (k)
    End For
Until (x= n_G)
//Display Results

Table 3. Percentage contribution (% C) and optimal levels (L) for different task sizes

$n_{\mathcal{R}}$	$n_{\mathcal{T}}$	$n_{\mathcal{P}}$		$\mathcal{P}_{\mathcal{C}}$		$\mathcal{P}_{\mathcal{M}}$		n_G	
		% C	**L**	**% C**	**L**	**% C**	**L**	**% C**	**L**
20	10	5.81	3	6.82	2	23.26	1	29.8	3
20	20	1.13	2	0.33	1	65.23	1	23.72	3
20	30	0.2	3	1.1	3	81.99	1	7.91	3
20	40	0.99	3	0.37	1	85.02	1	4.55	3

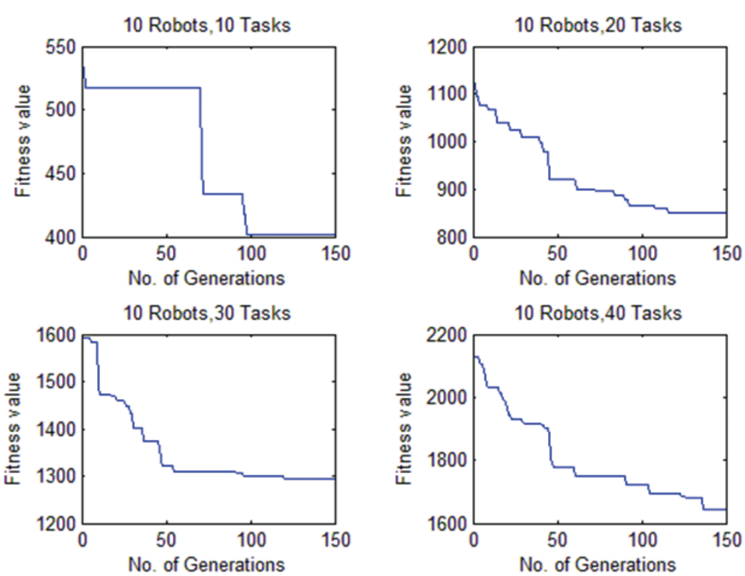

Figure 4. Convergence plot for the different $n_{\mathcal{R}}$ and $n_{\mathcal{T}}$.

5. RESULTS AND DISCUSSION

The change in the $n_{\mathcal{R}}$ allocated to a set-of-tasks affects the performance measures of the entire system. When the number of robots is more than the required level, an increase in average idle time per robot (μI_t^R) is seen, which is undesirable. A smaller $n_{\mathcal{R}}$ than the necessary level increases the Average task completion time (μC_t^T), Average waiting time per robot $(\mu \mathcal{W}_t^{\mathcal{R}})$, and average travel per robot (μT_t^R). So, for a given task level it is essential to identify the optimal number of robots required to complete it. A tradeoff between $\mu \mathcal{I}_t^{\mathcal{R}}$ and $\mu \mathcal{C}_t^{\mathcal{T}}$, $\mu \mathcal{W}_t^{\mathcal{R}}$, $\mu \mathcal{T}_t^{\mathcal{R}}$ is done to determine the optimal $n_{\mathcal{R}}$ required to complete the tasks. Table 4 presents the results obtained by varying $n_{\mathcal{R}}$ for a given set-of-tasks at different levels.

Refer Fig. 5(a). On plotting the average waiting time per robot against the varying $n_{\mathcal{R}}$ and $n_{\mathcal{T}}$, the following was observed. $\mu \mathcal{W}_t^{\mathcal{R}}$ increases for the increase in $n_{\mathcal{T}}$ keeping the $n_{\mathcal{R}}$ constant and decreases for the increase in the $n_{\mathcal{R}}$ for a given $n_{\mathcal{T}}$. Similarly (refer Fig. 5(b).) $\mu \mathcal{I}_t^{\mathcal{R}}$ increases with increase in $n_{\mathcal{R}}$ keeping the $n_{\mathcal{T}}$ constant.

The tradeoff chart for performance measures corresponding to different task levels is as shown in Fig. 6. The optimal $n_{\mathcal{R}}$ for the most of the cases are found to be varying between 10 and 15 and occurs at 20 per cent of performance measures. There should be a proportionate increase in the $n_{\mathcal{R}}$ for the corresponding increase in the $n_{\mathcal{T}}$. However, here there is not much variation in the $n_{\mathcal{T}}$ required for 10 and 40 tasks. The reason lies in the inability in the allocation of the robots to the waiting task due to the prioritisation of tasks before completing the preceding ones. In fact, removing the precedence constraints would increase the optimal $n_{\mathcal{R}}$ for the corresponding $n_{\mathcal{T}}$.

Table 4. Performance measures

$n_{\mathcal{R}} - n_{\mathcal{T}}$	f_t	Performance measures in (time units)			
		$\mathcal{C}_t^{\mathcal{T}}$	$\mathcal{T}_t^{\mathcal{R}}$	$\mathcal{W}_t^{\mathcal{R}}$	$\mathcal{I}_t^{\mathcal{R}}$
5R-10T	465.11	320.22	1201.28	338.96	0
10R-10T	401.46	169.24	1085.23	261.95	313.03
15R- 10 T	490.79	151.78	1152.56	345.79	1003.67
20R-10T	586.83	147.14	1050.44	146.08	0.00
5R-20T	806.23	613.11	2201.51	410.29	324.24
10R-20T	851.71	360.71	2246.83	475.07	1318.30
15R-20T	1027.26	307.66	2064.47	418.60	2305.85
20R-20T	1281.93	265.37	2141.75	414.75	0.00
5R-30T	1146.43	839.84	3194.11	551.78	583.04
10R-30T	1294.48	583.18	3244.31	767.38	1491.25
15R-30T	1551.93	483.55	3253.46	979.44	2825.72
20R-30T	1871.88	404.07	3084.87	1172.87	0.00
5R-40T	1587.40	1143.06	4361.25	845.28	324.24
10R-40T	1642.69	712.85	4271.55	1262.14	1630.55
15R-40T	1968.02	622.87	4423.62	1195.03	3700.18
20R-40T	2465.96	560.99	4145.37	1457.30	89.44

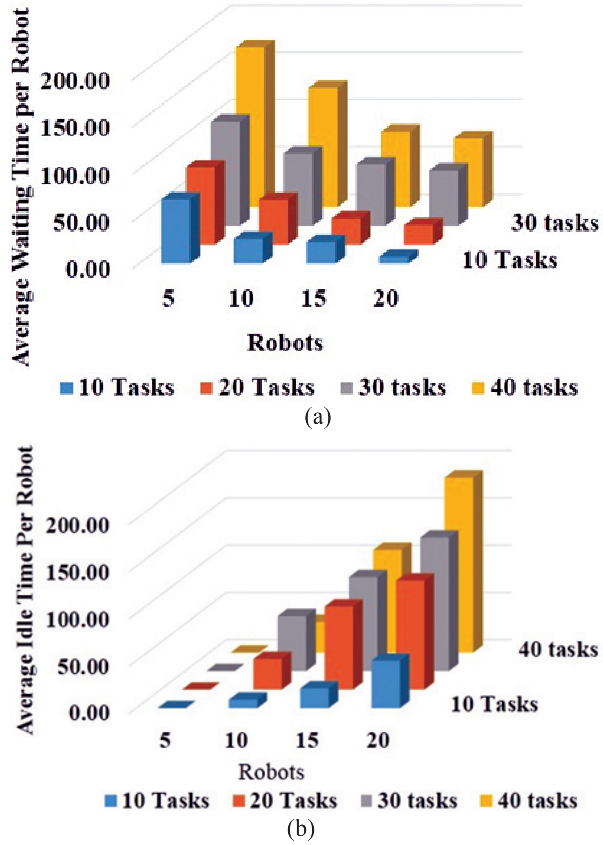

Figure 5. (a) Average waiting time per robot and (b) Average idle time per robot.

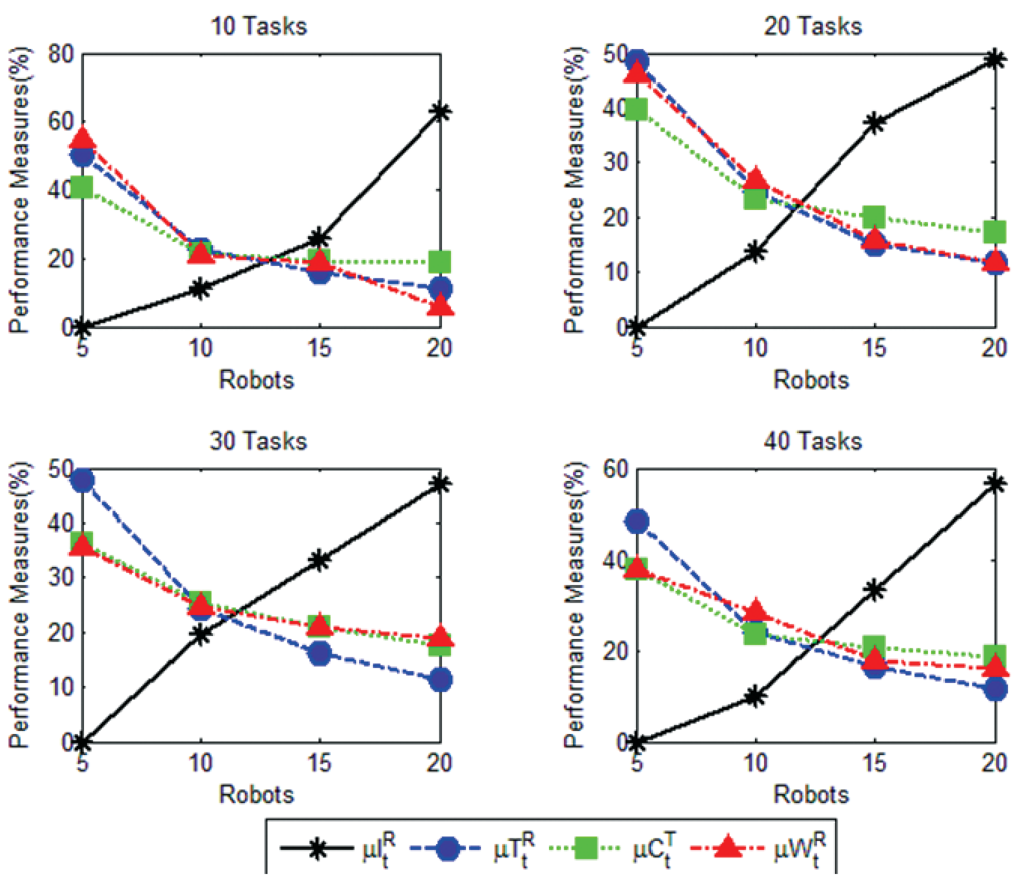

Figure 6. Performance measures.

6. CONCLUSION AND FUTURE SCOPE

The augmented use of autonomous robots in various applications have given rise to the considerable attention of MRTA problem in recent years. This work addresses the allocation of multiple robots to multiple tasks considering four objectives with precedence constraints. The work is aimed to find the best allocation of robots to task and to find the optimal for a given set-of-tasks at any instant of time. GA based methodology was developed to obtain the best allocation of single robots to the multi-robot foraging tasks. Changes in the values of GA parameters have more significant effects on the results. Hence, Taguchi's L_{27} orthogonal array experimental design is used to identify the best values for the GA parameters. It was observed from ANOVA that the Mutation probability and Number of generations are the most significant factors that influence the performance of GA.

The optimal value for the mutation probability (0.01) and the number of generations (150) is unchanged for the different robot to task ratio. The developed methodology was tested by varying the $n_{\mathcal{R}}$ and $n_{\mathcal{T}}$ values, and the results were compared. Though all the objectives are of minimisation type, it is observed that for a fixed number of tasks, $\mu\mathcal{I}_t^{\mathcal{R}}$ increases with a decrease in $\mu\mathcal{C}_t^{\mathcal{T}}$, $\mu\mathcal{W}_t^{\mathcal{R}}$, and $\mu T_t^{\mathcal{R}}$ for an increase in $n_{\mathcal{R}}$ from 5 to 20. On plotting these performance measures, the tradeoff occurs at almost 20% for all the cases, and the optimal $n_{\mathcal{R}}$ varies from 10 to 15. Also, observed that the precedence constraints limit the use of robots beyond a certain level and hence, the idle time increases with the addition of robots more than the required number.

This work can be extended to different applications by fixing appropriate weights for each of the objectives as per the application requirements and using different methods. Also, the performance of the algorithm may be tested by scaling up the $n_{\mathcal{R}}$ and $n_{\mathcal{T}}$ to a greater extent and comparing it with existing market-based approaches.

REFERENCES

1. Gerkey, B.P. & Mataric, M.J. A formal analysis and taxonomy of task allocation in multi-robot systems. *Int. J. Robotics Res.*, 2004, **23**(9), 939-954.
doi: 10.1177/0278364904045564
2. Nunes, E.; Manner, M.; Mitiche, H. & Gini, M. A taxonomy for task allocation problems with temporal and ordering constraints. *Robotics Autonomous Sys.*, 2017, **90**, 55-70.
doi: 10.1016/j.robot.2016.10.008
3. Tolmidis, A.T. & Petrou, L. Multi-objective optimization for dynamic task allocation in a multi-robot system. *Eng. Appl. Artificial Intell.*, 2013, **26**(5), 1458-1468.
doi: 10.1016/j.engappai.2013.03.001
4. Chen, J. & Sun, D. Coalition-based approach to task allocation of multiple robots with resource constraints. *IEEE Trans. Automation Sci. Eng.*, 2012, **9**(3), 516-528.
doi: 10.1109/TASE.2012.2201470
5. Han, X.; Haili, Q.; Xun, L. & Hongxu, M. Swarm intelligence based WSN-mediated distributed multi-robot task allocation. *In* 27th Chinese Control Conference, 2008. CCC 2008. (pp. 451-456). IEEE.
doi: 10.1109/CHICC.2008.4605682
6. Elango, M. & Nachiappan, S.P. Balancing multi-robot prioritized task allocation: A simulation approach. *In* International Conference on Industrial Engineering and Engineering Management (IEEM 2011), 1725-1729, IEEE. 2011
doi: 10.1109/IEEM.2011.6118211
7. Mitiche, H.; Boughaci, D. & Gini, M. Efficient heuristics for a time-extended multi-robot task allocation problem. In First International Conference on New Technologies of Information and Communication (NTIC 2015), 1-6, IEEE 2015
doi: 10.1109/NTIC.2015.7368756`
8. Coltin, B. & Veloso, M.M. Scheduling for transfers in pickup and delivery problems with very large neighborhood search. *In* AAAI, 2014, 2250-2256, 2014.
9. Kartal, B.; Nunes, E.; Godoy, J. & Gini, M.L. Monte Carlo tree search for multi-robot task allocation. *In* AAAI 2016, 4222-4223, 2016.
10. Zuo, Y.; Peng, Z. & Liu, X. Task allocation of multiple UAVs and targets using improved genetic algorithm. *In* 2nd IEEE International Conference on Intelligent Control and Information Processing (ICICIP2011), 2011, 2, 1030-1034.
doi: 10.1109/ICICIP.2011.6008408
11. Khamis, A.; Hussein, A. & Elmogy, A. Multi-robot task allocation: A review of the state-of-the-art. *In* Cooperative Robots and Sensor Networks, 2015, 31-51.
doi: 10.1007/978-3-319-18299-5_2
12. Jianping, C.; Yimin, Y. & Yunbiao, W. Multi-robot task allocation based on robotic utility value and genetic algorithm. *In* IEEE International Conference on Intelligent Computing and Intelligent Systems, (ICIS 2009), 2009, **2**, 256-260.
doi: 10.1109/ICICISYS.2009.5357957
13. Liu, C. & Kroll, A. On the performance of different mutation operators of a subpopulation-based genetic algorithm for multi-robot task allocation problems. 2016, arXiv preprint arXiv:1606.00601.
14. Xu, Z.; Xia, F. & Zhang, X. Multi-robot dynamic task allocation using modified ant colony system. *Artificial Intelligence Computational Intelligence*, 2009, 288-297.
15. Kmiecik, W.; Wojcikowski, M.;Koszalka, L. & Kasprzak, A. Task allocation in mesh connected processors with local search meta-heuristic algorithms. *Intel. Info. Database Sys.*, 2010, 215-224.
doi: 10.1007/978-3-642-12101-2_23
16. Kim, M.H.; Kim, S.P. & Lee, S. Social-welfare based task allocation for multi-robot systems with resource constraints. *Comput. Industrial Eng.*, 2012, **63**(4), 994-1002.
doi: 10.1016/j.cie.2012.06.011
17. Nam, C. & Shell, D.A. Analyzing the sensitivity of the optimal assignment in probabilistic multi-robot task allocation. *IEEE Robotics Automation Lett.*, 2017, **2**(1), 193-200.
doi: 10.1109/LRA.2016.2588138
18. Giordani, S.; Lujak, M. & Martinelli, F. A distributed multi-

agent production planning and scheduling framework for mobile robots. *Comput. Industrial Eng.*, 2013, **64**(1), 19-30.
doi: 10.1016/j.cie.2012.09.004
19. Liu, F.; Liang, S. & Xian, X. Multi-robot task allocation based on utility and distributed computing and centralized determination. *In* 27th Chinese Control and Decision Conference (CCDC 2015), 2015, 3259-3264.
doi: 10.1109/CCDC.2015.7162482
20. Su, X.; Zhang, M. & Bai, Q. Coordination for dynamic weighted task allocation in disaster environments with time, space and communication constraints. *J. Parallel Distributed Comput.*, 2017, **97**, 47-56.
doi: 10.1016/j.jpdc.2016.06.010
21. Jones, E.G.; Dias, M.B. & Stentz, A. Time-extended multi-robot coordination for domains with intra-path constraints. *Autonomous Robots*, **30**(1), 41-56.
doi: 10.1007/s10514-010-9202-3
22. Coello, C.A. An updated survey of GA-based multiobjective optimization techniques. *ACM Computing Surveys* (CSUR), 2011, **32**(2), 109-143.
doi: 10.1145/358923.358929
23. Cho, J.H.; Wang, Y.; Chen, R.; Chan, K.S. & Swami, A. A survey on modeling and optimizing multi-objective systems. *IEEE Commun. Surveys Tutorials*, 2017.
doi: 10.1109/COMST.2017.2698366
24. Das, D.B. & Patvardhan, C. New multi-objective stochastic search technique for economic load dispatch. *In* IEE Proceedings-Generation, Transmission and Distribution, 2017, **145**(6), 747-752.
doi: 10.1049/ip-gtd:19982367
25. Saramago, S.F.P. & Steffen, V. Optimization of the trajectory planning of robot manipulators taking into account the dynamics of the system. *Mechanism Machine Theory*, 1998, **33**(7), 883-894.
doi: 10.1016/S0094-114X(97)00110-9
26. Khuntia, A.K.; Choudhury, B.B.; Biswal, B.B. & Dash, K.K. A heuristics based multi-robot task allocation. *In* Recent Advances in Intelligent Computational Systems (RAICS2011), IEEE 2011, 407-410.
doi: 10.1109/RAICS.2011.6069344

CONTRIBUTORS

Mr Padmanabhan Panchu K. has completed ME (Industrial Engineering) and currently working as Assistant professor in the Department of Industrial Engineering, Anna University Chennai. His areas of interest include : Path planning and scheduling of operations.
His contribution to the current study includes formulation of the problem, identification of solution methodology, computer programming of GA, analysis of results and manuscript preparation.

Dr Rajmohan M. has completed PhD (Industrial Engineering) and currently working as an Associate professor in the Department of Industrial Engineering, Anna University Chennai. He has served as the Honourable Secretary of ORSI Chennai Chapter His areas of research includes : Vehicle route optimisation problem, design of experiments and supplier selection methods in supply chain management.
His contributions to the current study include performing ANOVA and identifying the best levels for GA parameters, manuscript preparation, and overall guidance.

Mr Sundar R. has completed ME (Industrial Engineering) and currently pursuing his PhD in the Department of Production Engineering, National Institute of Technology, Tiruchirappalli. He is working in the area of simplification of optimisation algorithms for nonlinear optimisation problems.
His contribution to the current study includes literature survey, programming of GA and manuscript preparation.

Dr Baskaran R. has completed PhD (Industrial Engineering) and currently working as an Associate professor in the Department of Industrial Engineering, Anna University Chennai. His research interest includes : Route scheduling, route evaluation, optimisation, modelling of manufacturing systems and systems engineering.
His contribution to the current study includes mathematical formulation and identification of performance measures.