Dynamic Task Allocation for Mobile Robot Teams based on Linear Integer Programming

Seenu.N
Center for Automation and Robotics Hindustan Institute of Technology and Science
Chennai, India

Gian Carlo
B. Tech Mechatronics Engineering Hindustan Institute of Technology and Science
Chennai, India

P. Saishashank
B.Tech Mechatronics Engineering
Hindustan Institute of Technology and Science
Chennai, India

R M. Kuppan Chetty
Center for Automation and Robotics Hindustan Institute of Technology and Science
Chennai, India

Guruprashanth
B.Tech Mechatronics Engineering Hindustan Institute of Technology and Science
Chennai, India

Senthil Kumar S
B.Tech Mechatronics Engineering Hindustan Institute of Technology and Science
Chennai, India

***Abstract*—Dynamic task allocation for multiple robot network is the fundamental criterion for successful completion of the conquest. Primarily the target task is partitioned into number of sub-tasks thereafter executed by the robots in a distributed manner. Multi-robot task allocation is a combinatorial optimization problem. Most of the prevailing optimization schemes of task allocation considers single objective parameter. This paper presents a bi-objective task allocation scheme for multi-robot network. It establishes the model of task allocation by considering the minimal utilization of time and battery energy resources. It proposes a numerical problem-solving method to obtain the optimal weight values to be used in the objective function. It encompasses the Linear Integer Programming optimization (LIP) technique for task allocation. Time consumption and energy consumption rates are considered as the evaluation parameters for the proposed scheme. The dynamic task allocation scheme is simulated in Webots virtual simulation environment. The experimental results express that the proposed Linear Integer Programming (LIP) -based task allocation technique minimizes the resource utilization and allocates the task effectively.**

Keywords—dynamic task allocation, constraint optimization technique, Linear Programming, multiple robot network

I. INTRODUCTION

To the extent of a decade it was considering a miracle where the mobile robot can move in an unknown environment at optimal speeds. But in recent years the advancements in both hardware and software optimization had it more capable of using in mobile robots [5]. Autonomous mobile robots are deployed in various situations and applications such as manufacturing industries, warehouses, search and rescue missions, exploration missions and military applications etc [8]. There is another topic gained attention where study is about the mobile-robot swarm where these robots does not have centralized control but instead of that it has robots where it operates independently. It does have local sensing and operating mechanism. With these this sort of approach all the robots will behave in a similar way to each other [6]. Therefore, the study of task allocation comes in development for swarm of robots. Introducing task allocation to robot swarms the swarm will start to do the task in more smart and intelligent way where each robot will be assigned with own task and it will have its own personal goal in respect of its priorities and resources and its final goal of the swarm [12]. But these all have its limitation where it can only be used in singular and idle situations. Therefore, the study of dynamic task allocation study is being developed in recent years.

Dynamic task allocation has become a major requirement for the swarm cooperative robots [12]. With the help of dynamic task allocation, the responses and actions of the robots can be changed dynamically in the unknown environment. Using this approach, the overall performance of the system can be enhanced. Where the robots will have the ability to observe the environment around it and respond, with this study it can also be developed in such a way where robots can establish communication between each other dynamically, if the situation does arise where the given task can't be performed with the allocated robot, the nearest neighbouring robot can cover its task and finish it.

This study observes how different the robot's response to the environments. It analyses the importance of time and battery resource utilization in task allocation. We develop the algorithm with multi-objective function and optimizing it with a linear minimization algorithm of the derived utilities.

This paper is organized as follows, chapter II presents literature review on multi-robot task allocation, chapter III describes the problem description as well as the utility functions in task allocation. Chapter IV details the linear programming task allocation strategy and chapter V illustrates the simulation result evaluation and discussion.

II. Literature Review

Multi-robot task allocation problem is a combinatorial optimization problem [15]. In the literature various heuristics-based task allocation strategies for multi-robot systems are found. Most of the heuristics strategies in the literature considered single objective-function for optimization [14]. Minimization of total travel distance is considered for optimization in [1]. Ant Colony Optimization algorithm [7], particle swarm optimization algorithms [2, 13], PSO variants [4] and genetic algorithm [1] are used for task allocation. Genetic algorithm optimization [3] for maximizing task completion rate is presented in [8]. Minimization of total tasks completion time using Particle Swarm Optimization is presented in [17].

Many authors presented single objective optimization function for task allocation. However, the multi-robot task allocation problem is a multi-objective optimization problem [18]. Multi-objective optimization task allocation improves the performance of the multi-robot systems [9]. Multi-objective optimization for task precedence constraint problems is presented in [11]. Minimization of task completion time, robot waiting time and idle time are the optimization factors considered. Another multi-objective optimization in order to minimize the battery power consumption and task completion time is presented in [15].

This paper proposes a multi-objective optimization task allocation strategy. Minimal utilization of robot battery power and task completing time are considered for task allocation. It involves Linear Integer optimization technique for single-task single-robot task allocation problems.

III. MULTI-ROBOT TASK ALLOCATION

A. *Problem description*

This paper proposes an optimization based multi-robot task allocation scheme. Consider $R = {}_1, R_2..R_n$ be the set of robots in the network. Let $T = T_1, T_2.._n$ be the set of tasks to be allocated for the robots. Let A be the $n \times 1$ matrix represents the task allocation set Eqn (1), where n is the number of robots and tasks. The multi-robot task allocation problem is a combinatorial optimization problem [17]. Any task allocation strategy assigns each task with each robot, such a way that, the utilization of cost to complete the tasks is minimum. At the same time the performance of the robots is maximum [19].

In this approach the number of robots and number of tasks is considered to be equal. The utilization of time and battery resources is the two major performance influencing parameters [16]. This study proposes a task allocation strategy based on Linear Integer Programming optimization techniques. It minimizes the utilization of the time and battery resources.

$$A = \begin{bmatrix} a_{11} \\ a_{22} \\ a_{33} \\ . \\ . \\ . \\ a_{nn} \end{bmatrix} \quad (1)$$

The constraints of task allocation are:

$$\sum_{j=1}^{n} a_{ij} = 1 \quad \text{where}, 1 \le j \le n \quad (2)$$

$$\sum_{i=1} {}_{ij} = 1 \quad \text{where}, 1 \le i \le n \quad (3)$$

$$a_{ij} \in 0,1 \quad \text{where}, 1 \le i,j \le n \quad (4)$$

The constraint given by Eqn (2) ensures that only one task is allocated to a robot. Similarly, the next constraint in Eqn (3) represents that a robot is allocated only to a single task. The final constraint in Eqn (4) states that the value of task allocation is binary either it 0 or it is 1.

IV. LINEAR PROGRAMMING INTASK ALLOCATION

A multi-robot system completes the whole task in a distributive manner. Efficient task allocation plays a vital influence in the better performance of the robot's team [18]. The time and energy utilization by the individual robots depend on the tasks allocated to them. However, the time and energy utility are contrast in nature [3]. Battery power utilization is higher for fast task completion and vice versa. Thus, these two utility functions form a pareto optimal task allocation problem. This paper aims to analyse the time and energy utilities relationship. This analysis supports to frame objective function in order to derive the global optimal task allocation.

Let $'T'$ and $'E'$ be the time and battery energy utility matrices, respectively. Where t_{ij} is the time utility required by the robot j to complete the task i. Similarly, e_{ij} is the battery energy utility required by the robot j to complete the task i (Eqn 5 & 6).

$$T = \begin{bmatrix} t_{11} & t_{12} & t_{13}\cdots & t_{1n} \\ t_{21} & t_{22} & t_{23}\cdots & t_{2n} \\ t_{31} & t_{32} & t_{33}\ldots & t_{31} \\ \ldots & \ldots & \ldots & \ldots \\ t_{n1} & t_{n2} & t_{n3}\ldots & t_{nn} \end{bmatrix} \quad (5)$$

$$E = \begin{bmatrix} e_{11} & e_{12} & e_{13}\cdots & e_{1n} \\ e_{21} & e_{22} & e_{23}\cdots & e_{2n} \\ {}_{31} & e_{32} & e_{33}\cdots & e_{31} \\ \ldots & \ldots & \ldots & \ldots \\ e_{n1} & e_{n2} & e_{n3}\ldots & e_{nn} \end{bmatrix} \quad (6)$$

The objective is to minimize the consumption of time and battery energy together. Therefore, the objective function is a minimizing weighted summation of time and energy utilities [4, 10](Eqn 7).

$$Q = (W_1 ({}_{ij}) + W_2 (E_{ij})) \quad (7)$$

$$Q = \begin{bmatrix} q_{11} & q_{12} & q_{13}\cdots & q_{1n} \\ q_{21} & q_{22} & q_{23}\cdots & q_{2n} \\ \ldots & \ldots & \ldots & \ldots \\ q_{n1} & q_{n2} & q_{n3}\cdots & q_{nn} \end{bmatrix} \quad (8)$$

Where Q is the 2D matrix consists of the consolidated time and energy utilities required for every robot to execute every task(Eqn 8). The proposed optimization strategy aims to minimize the utility requirement Q. Identification of optimal Q value relies on the time and energy weight parameters W_1 and W_2. The range of the weight values are considered between [0,1]. Initial values of $_1$ and W_2 are 0 and 1 respectively. Time utility requirement is inversely proportional to the energy requirement utility. Therefore, the value W_1 is uniformly incremented from 0 to 1 at the same time the value of W_2 is decremented from 1 to 0 at the same interval. The weight values are updated uniformly by means of a regular interval value. Let I be the interval, then the initial set if weight values is [0,1]. For the initial set of weight, the Q matrix is generated. Next the interval set is updated as [0+I, 1-I] for which the Q matrix is generated. This process ends when the updated weight set becomes as [1,0]. Thus, a number of Q matrices are created within the range [0,1]. The total number of Q matrices is based on the interval value. The average weight value $[W_{1avg,2avg}]$ is calculated from the set of weight values [W_1,W_2] within the range [0,1] with respect to the interval value. Simultaneously the average Q matrix is calculated from the set of Q matrices with respect to the interval value. The task allocation is performed by using the average Q matrix. . This method produces task allocation model with the minimal time and energy utility values.

The proposed bi-objective utility function identifies the task allocation scheme with minimal time and battery energy utility values. The previous chapter describes the objective function as well as the proposed numerical method of calculating the best weight values which derives the minimal utility values. By using this minimal Qmatrix the proposed single-robot single-task task allocation is performed in an optimal manner.

V. SIMULATION RESULTS ANALYSIS

The numerical simulation is performed to evaluate the proposed linear programming task allocation strategy. The simulation data is taken from the RoboCup 2D Soccer robot simulator [9]. The time and energy utility matrices are generated from the soccer game simulation. It is used for the analysis of trade-off weights' influence on the minimum utility function [9]. In the simulation there are 16 number of tasks and 16 number of robots are there. The time and energy utility matrices are given in Fig 1 & 2.

The proposed estimation technique of optimal weight values W_1 and W_2 is performed for four different interval values.
$I_1 = 0.002$,$I_2 = 0.001$,$I_3 = 0.02$,$I_4 = 0.01$.The minimal Q matrix is created by applying the proposed numerical calculation on the variant Q matrices generated for the four interval values. Linear integer programming-based task allocation is applied on the minimal Q matrix. It results the task allocation scheme which requires minimal time and energy utility values.

0.496	0.717	0.641	0.650	0.847	0.349	0.852	0.516	0.529	0.416	0.409	0.765	0.825	0.417	0.845	0.976
0.435	0.218	0.680	0.347	0.374	0.494	0.574	0.408	0.720	0.514	0.719	0.490	0.389	0.978	0.433	0.386
0.638	0.951	0.489	0.840	0.0.653	0.454	0.721	0.670	0.556	0.522	0.440	0.569	0.601	0.342	0.936	0.795
0.497	0.710	0.649	0.648	0.847	0.353	0.913	0.516	0.535	0.420	0.415	0.770	0.82	0.425	0.844	0.950
0.618	0.430	0.712	0.563	0.518	0.607	0.742	0.604	0.865	0.653	0.871	0.991	0.515	0.781	0.644	0.567
0.574	0.842	0.526	0.767	0.724	0.398	0.757	0.602	0.527	0.467	0.408	0.628	0.670	0.346	0.943	0.866
0.888	0.645	0.394	0.937	0.429	0.685	0.583	0.973	0.672	0.748	0.590	0.393	0.388	0.372	0.735	0.561
0.425	0.548	0.803	0.523	0.837	0.323	0.897	0.433	0.551	0.384	0.441	0.987	0.846	0.536	0.714	0.801
0.475	0.523	0.945	0.541	0.756	0.392	0.928	0.478	0.629	0.450	0.522	0.838	0.762	0.614	0.716	0.751
0.799	0.499	0.551	0.629	0.415	0770	0.637	0.729	0.989	0.817	0.899	0.455	0.398	0.638	0.629	0.498
0.981	0.506	0.385	0.739	0.344	0.828	0.536	0.943	0.752	0.891	0.703	0.339	0.311	0.434	0.630	0.462
0.819	0.357	0.319	0.590	0.222	0.978	0.430	0.775	0.736	0.945	0.753	0.237	0.195	0.439	0.491	0.329
0.653	0.420	0.661	0.572	0.477	0.657	0.703	0.631	0.915	0.793	0.926	0.542	0.470	0.739	0.630	0.535
0.690	0.537	0.674	0.669	0.571	0.624	0.794	0.681	0.893	0.682	0.723	0.608	0.597	0.647	0.747	0.646
0.214	0.420	0.703	0.335	0.934	0.109	0.683	0.224	0.344	0.170	0.239	0.876	0.891	0.396	0.532	0.680
0.277	0.498	0.701	0.408	0.893	0.161	0.738	0.290	0.385	0.224	0.276	0.957	0.965	0.416	0.604	0.757

Fig 1 Time utility requirement matrix for 16 tasks and 16 robots

0.692	0.198	0.537	0.623	0.179	0.393	0.239	0.318	0.548	0.348	0.660	0.269	0.186	0.485	0.168	0.283
0.612	0.931	0.678	0.315	0.643	0.481	0.508	0.564	0.261	0.687	0.181	0.607	0.494	0.132	0.416	0.457
0.661	0.144	0.504	0.162	0.366	0.587	0.216	0.565	0.338	0.507	0.562	0.691	0.514	0.599	0.204	0.281
0.356	0.262	0.699	0.429	0.215	0.313	0.251	0.634	0.673	0.356	0.502	0.061	0.168	0.515	0.199	0.210
0.326	0.599	0.225	0.450	0.493	0.393	0.238	0.329	0.148	0.348	0.089	0.699	0.421	0.187	0.323	0.517
0.547	0.138	0.530	0.299	0.251	0.324	0.097	0.462	0.659	0.411	0.417	0.690	0.545	0.439	0.211	0.262
0.248	0.390	0.418	0.056	0.631	0.585	0.545	0.058	0.508	0.251	0.304	0.635	0.651	0.433	0.184	0.341
0.670	0.326	0.287	0.584	0.294	0.432	0.123	0.416	0.328	0.420	0.685	0.217	0.092	0.365	0.144	0.081
0.396	0.535	0.178	0.604	0.156	0.549	0.268	0.593	0.519	0.379	0.307	0.173	0.183	0.533	0.080	0.220
0.202	0.633	0.548	0.692	0.625	0.254	0.373	0.191	0.252	0.286	0.206	0.547	0.476	0.511	0.526	0.379
0.271	0.387	0.408	0.133	0.540	0.209	0.364	0.064	0.150	0.080	0.275	0.567	0.678	0.363	0.326	0.669
0.241	0.419	0.631	0.328	0.840	0.060	0.657	0.210	0.181	0.299	0.102	0.840	0.770	0.496	0.348	0.362
0.538	0.551	0.6113	0.507	0.582	0.305	0.123	0.479	0.109	0.365	0.148	0.672	0.303	0.268	0.334	0.392
0.398	0.559	0.696	0.612	0.399	0.374	0.264	0.401	0.138	0.626	0.293	0.540	0.341	0.402	0.191	0.609
0.805	0.427	0.254	0.478	0.248	0.717	0.506	0.707	0.326	0.917	0.953	0.149	0.090	0.377	0.444	0.416
0.946	0.351	0.239	0.348	0.059	0.859	0.236	0.895	0.626	0.880	0.847	0.221	0.061	0.456	0.619	0.117

Fig 2 Battery energy utility requirement matrix for 16 tasks and 16 robots

The distribution of utility values with respect to $W_1$and W_2in an interval rate of change is plotted in Fig 3 and 4 for the intervals 0.002 and 0.001, respectively.

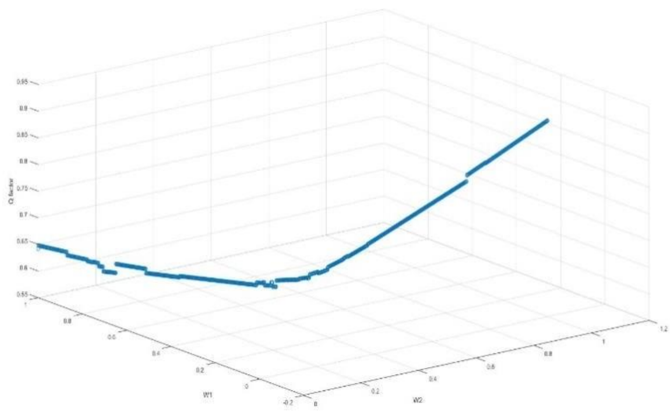

Fig 3 Weight values distribution in the interval 0.02

Table 1 represents the time utilization of the proposed linear programming-based task allocation scheme for the four different intervals. The results evidence that this technique consumes minimal time to compute the task allocation model. Table 2 shows that the calculated optimal weight values for the four different intervals.

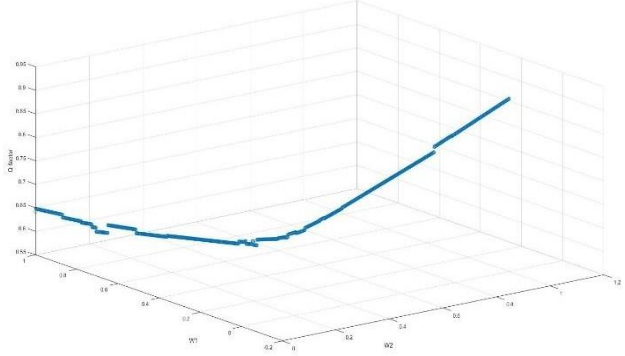

Fig 4 Weight values distribution in the interval 0.01

Table 1. Run time of proposed task allocation scheme

S.No	Weight Interval	Run Time(s)
1	0.002	0.0351
2	0.001	0.0628
3	0.02	0.0160
4	0.01	0.0125

Table 2 Resultant Optimal weights and allocation model

S.No	Weight Interval	Minimum utility value		Resultant Task allocation scheme
		Optimal W_1	Optimal W_2	
1	0.002	0.5340	0.4660	T1-R3;T2-R5; T3-R14 ;T4-R10; T5-R12; T6-R7; T7-R9; T8-R16; T9-R4; T10-R2; T11-R15; T12-R6; T13-R11; T14-R13; T15-R8; T16-R1
2	0.001	0.5330	0.4670	T1-R3;T2-R5;T3-R14;T4-R10;T5-R12;T6-R7;T7-R9;T8-R16;T9-R4;T10-R2;T11-R15;T12-R6;T13-R11;T14-R13;T15-R8;T16-R1
3	0.02	0.5800	0.4200	T1-R16;T2-R2;T3-R14;T4-R10;T5-R12;T6-R7;T7-R1;T8-R3;T9-R4;T10-R13;T11-R15;T12-R6;T13-R5;T14-R9;T15-R8;T16-R11
4	0.01	0.5500	0.4500	T1-R3; T2-R5;T3-R14;T4-R10;T5-R8;T6-R7;T7-R9;T8-R16;T9-R4;T10-R2;T11-R15;T12-R6;T13-R11;T14-R13;T15-R8;T16-R1

VI. DISCUSSION

The experiments results show that the proposed task allocation scheme based on linear programming provides the assurance of identifying optimal task allocation within minimum run time. The three-dimension plot from fig.1 to fig.2 depict the utility values distribution for average matrix generated for the uniformly varying weight values. Instead of using heuristic approach to find the optimal Qmatrix, this study proposes simple numerical strategy. According to the different interval results, minimum utility value is derived from almost equally weighted time and energy utilities. The proposed linear programming using best provides better task allocation with minimum time and energy utilization.

CONCLUSION

A multi-objective task allocation algorithm is proposed in this study. The time and battery energy utilization are the two important parameters influencing the robots task completion. This paper considers minimization of these two parameters as the objective function. Since it is a pareto optimization problem a weighted summation cost function is devised. However, the correct weight values are important for optimal task allocation. In order to identify the best weight values a numerical solving method is presented. This method successfully identifies the best weight values. The identified weight values are further used for task allocation. This study developed a linear programming-based task allocation scheme by using local best first search strategy. The proposed task allocation scheme is evaluated by a numerical simulation. The results clearly show that the proposed technique performs task allocation within minimal time and maximum accuracy. In the future the proposed concept is experimented with real robots and the performance is to be evaluated with the simulation results. The other objective parameters such as minimizing travel distance, maximizing task completion rate, maximizing the self-reconfigurability etc are to be included in the objective function for developing a robust task allocation strategy. This research can be extended to various parameter-differing objectives where the task allocation is dynamic, and the robots are heterogenous.

REFERENCES

[1] M. U. Arif and S. Haider, “An Evolutionary Traveling Salesman Approach for Multi-Robot Task Allocation Overview of MRTA,” 31st AAAI Conference on Artificial Intelligence, AAAI 2017, Vol. 1 pp. 5032-5033, 2017.

[2] S. Biswas, S. G. Anavatti, and M. A. Garratt, “Particle swarm optimization based co-operative task assignment and path planning for multi-agent system,” 2017 IEEE Symp. Ser. Comput. Intell. SSCI 2017 - Proc., vol. 2018-Janua, pp. 1–6, 2018.

[3] Chen Jianping, Yang Yimin, and Wu Yunbiao, “Multi-Robot Task Allocation Based On Robotic Utility Value and Genetic Algorithm”, IEEE

International Conference on Intelligent Computing and Intelligent Systems. I E E E, 2009.

[4] Chen Jianping, Y. Yang, and Y. Wu, “Multi-robot task allocation based on the modified particle swarm optimization algorithm,” Proc. - 2011 7th Int. Conf. Nat. Comput. ICNC 2011, vol. 3, no. 1, pp. 1744–1749, 2011.

[5] M. Elango, S. Nachiappan, and M. K. Tiwari, “Balancing task allocation in multi-robot systems using K-means clustering and auction-based mechanisms,” Expert Syst. Appl., vol. 38, no. 6, pp. 6486–6491, Jun. 2011.

[6] B. Fang, Q. Zhang, H. Wang, and X. Yuan, “Personality driven task allocation for emotional robot team,” Int. J. Mach. Learn. Cybern., vol. 9, no. 12, pp. 1955–1962, Dec. 2018.

[7] Hisayuki Sasaoka, “Evaluation of Hetero Sensitivity of Pheromone with Ant Colony Optimization in RoboCup Rescue Simulation System”, 2014 IEEE International Conference on Granular Computing (GrC) Evaluation

[8] Y. Huang, Y. Zhang, and H. Xiao, “Multi-robot system task allocation mechanism for smart factory,” Proc. 2019 IEEE 8th Jt. Int. Inf. Technol. Artif. Intell. Conf. ITAIC 2019, no. Itaic, pp. 587–591, 2019.

[9] Jianping Chen, Jianbin Wang, Qijun Xiao, and Changxing Chen,”A Multi-Robot Task Allocation Method Based on Multi-Objective Optimization” Institute of Electrical and Electronics Engineers and IEEE Control Systems Society, 2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV) : 18-21 Nov. 2018.

[10] R. T. Marler and J. S. Arora, “The weighted sum method for multi-objective optimization: New insights,” Struct. Multidiscip. Optim., vol. 41, no. 6, pp. 853–862, Jun. 2010.

[11] Panchu K.P, Rajmohan M, Sundar R, and Baskaran R (2018), “Multi-objective Optimisation of Multi-robot Task Allocation with Precedence Constraints,” Defence Science Journal, Vol 68, No 2, pp 175-182.

[12] Francesco Riccio, Emanuele Borzi, Guglielmo Gemignani, Daniele Nardi,” Multi-Robot Search for a Moving Target: Integrating World Modeling, Task Assignment and Context”, Francesco Robotics Society of Japan, IEEE Robotics and Automation Society, IEEE Industrial Electronics Society, and Institute of Electrical and Electronics Engineers., IROS 2016 : 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems : October 9-14, 2016, Daejeon Convention Center, Daejeon, Korea.

[13] A. Salman, I. Ahmad, and S. Al-Madani, “Particle swarm optimization for task assignment problem,” Microprocess. Microsyst., vol. 26, no. 8, pp. 363–371, 2002.

[14] S. Sarathambekai and K. Umamaheswari, “Multi-objective optimization techniques for task scheduling problem in distributed systems,” Comput. J., vol. 61, no. 2, pp. 248–263, Feb. 2018.

[15] Z. Shi, Q. Chen, S. Li, H. Cai and X. Shen, "Cooperative task allocation for multiple mobile robots based on multi-objective optimization method," 2010 3rd International Conference on Computer Science and Information Technology, Chengdu, 2010, pp. 484-489

[16] Tamer Abukhalil, Madhav Patil, Sarosh Patel, and Tarek Sobh, “Coordinating a Heterogeneous Robot Swarm Using Robot Utility-based Task Assignment (RUTA)” Institute of Electrical and Electronics Engineers, 2016 IEEE 14th International Workshop on Advanced Motion Control (AMC) : 22-24 April 2016.

[17] V. Trivedi, P. Varshney, and M. Ramteke, “A simplified multi-objective particle swarm optimization algorithm,” Swarm Intell., vol. 14, no. 2, pp. 83–116, 2020.

[18] A.T. Tolmidis and L. Petrou, “Multi-objective optimization for dynamic task allocation in a multi-robot system,” Eng. Appl. Artif. Intell., vol. 26, no. 5–6, pp. 1458–1468, May 2013.

[19] J. Turner, Q. Meng, G. Schaefer, A. Whitbrook, and A. Soltoggio, “Constraints for the Optimization of Total Task Allocations in a Multirobot System,” IEEE Trans. Cybern., vol. 48, no. 9, pp. 2583–2597, 2018.