# BRADO algorithm for task allocation in multi-robot systems

Samaneh Hosseini, *Isfahan University Of Technology*
Fatemeh Nikbakht, *Isfahan University Of Technology*

***Abstract*—With the development of information technology, the capability and application fields of robots become wider. Optimal task allocation among the robots is one of the key issues to be investigated for the of multi-robot systems. On the other hand, task allocation is a np problem that can be solved using swarm methods. Common methods such as genetic algorithm, ant colony algorithm, particle swarm optimization algorithm, etc. have been used to allocate tasks to robots. In this paper, we use the lesser-known BRADO algorithm to allocate tasks; which on average has yielded better results than the above algorithms and has a better ability to get out of local minima. The results show that this algorithm gives much better results than the genetic algorithm when the number of tasks is more than 20. Also, with the increase in the number of tasks, the improvement of the results was more evident.**

***Index Terms*—Multi-robot systems (MRS), Task allocation, Swarm algorithms, BRADO algorithm, Genetic Algorithm**

## I. Introduction

DURING the last few years, the field of research in mobile robotics has encountered a significant shift to multi-robot systems rather than single robot systems. This increased interest in the community of mobile robotics research towards MRS comes from the significant advantages and higher potential provided by MRS than single robot systems [1]. Task allocation is one of the key issues to be addressed for multi-robot systems. Generally, tasks are assumed to be indivisible and can be executed by single robot. But often a task consists of several sub tasks. Thus, robots available in multi-robots system may cooperate and coordinate together to complete such tasks [2].

### A. NP problems

The NP problems set of problems whose solutions are hard to find but easy to verify and are solved by Non-Deterministic Machine in polynomial time. These issues include two types: NP-complete and NP-hard; Solving NP-hard problems is much more difficult than solving NP-complete problems. The following techniques can be applied to solve computational problems in general, and they often give rise to substantially faster algorithms: Approximation, Randomization, Restriction, Parameterization, Heuristic and Metaheuristic approaches [3].

S. Hosseini is an Associate Professor at Isfahan University of Technology.
F. Nikbakht is a postgraduate student at Isfahan University of Technology.

### B. Task allocation in multi-robot systems

In general there are task allocation ways of allocating T tasks to P processors (robots or agent or sensor), and the problem of finding an optimal feasible allocation is known to be NP-Complete [4]. There are several definitions for allocating tasks in multi-robot systems, which we will introduce in this section.

*1) Allocate tasks between homogeneous robots:* All agents or robots have the same ability to perform different tasks.

*2) Allocate tasks between heterogeneous robots:* Agents or robots have different abilities to perform different tasks; This means that each robot performs each task at a specific time according to its characteristics.
In addition to the above, there may be restrictions on tasks and agents, such as the order in which tasks are performed.

### C. Swarm intelligence algorithms

According to what has been said, one of the existing methods for allocating tasks is to use Meta-heuristic methods. Fig 1 shows the common meta-heuristic methods. The BRADO algorithm we have used in this article is a meta-heuristic method that is an evolutionary computing [5].

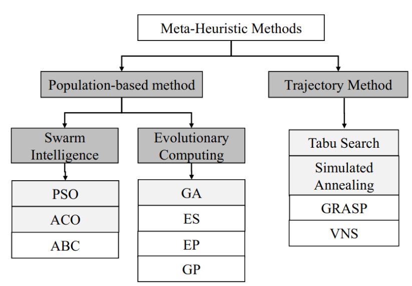



Fig. 1. Meta-Heuristic Methods

BRADO (BRAin Drain Optimization) algorithm as a socioinspired evolutionary computing approach, in which the search algorithm is inspired by the process of brain drain phenomenon.
In order to evaluate the BRADO performance, it was applied to several benchmark optimization functions and the results produced by BRADO, particle swarm optimization, imperialist competitive algorithm and GA have been compared. Results show the BRADO superiority to avoid the regions around local minima and dealing with high dimensionality problems.
The most important factors of brain drain phenomenon which

have viewed as the source of inspiration to propose the BRADO algorithm are as follows:

- Given their skills, people prefer to migrate to where they can be most productive
- Unsatisfied highly skilled workers migrate from the developing to the developed countries
- A migrant person influence on the future migration flow among his/her acquaintances
- Experts who return home influence on the future migration flow among their acquaintances

Generally, given an objective function f(x), optimization is finding the best available values for vector x given a defined domain to discover an optimum of f(x). Each solution x i is a D-dimensional vector in which D exhibits the number of optimization parameters. Fig. 2 shows the flowchart of the BRADO algorithm.

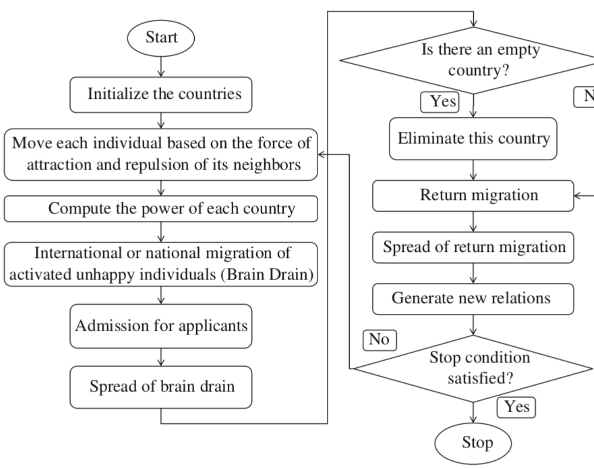



Fig. 2. Flowchart of the BRADO algorithm [5].

The initial individuals of our algorithm are divided into some collections which are called country. Each country has a social network which specifies neighbors for each individual. The individuals try to find increasingly better area of the search space. If an individual is unhappy with its situation, it will try to change it and go to somewhere else better than the source country. This migration influences on the formation of future migration flow among the individuals acquaintances. The reason of this event is spread of influence among social network members. Spread of influence, as a kind of network diffusion process, is a chain of decisions where it is most favorable for people to be affected by decisions of their friends. This process may be at least part of the interpretation for many kinds of imitation in a social network such as fashions and fads. However, in the BRADO algorithm spread of influence process occurs when an individual goes abroad and the action of the migrant impacts on the decisions of its neighbors. Some migrants will return home and bring skills and expertness learned abroad. During these migrations, if developing countries could not improve their situations, will be omitted and powerful ones take possession of their individuals. One of the most important parts of this algorithm is the motion section, which we see in the following formulation: The introduced movement method in this algorithm works based on three layers. Consider an individual who has the lowest cost among its neighbors. This individual is called a brain. Other individuals in each country are ordinary individuals. Based on these definitions, the position of each individual is changed according to its own cost and that of its neighbors. There is a three layered movement method for ordinary, brain and the best individuals.

*1) Movement method for ordinary individuals:* If an individual is an ordinary one, its position is changed according to the force of attraction and repulsion of its neighbors. Based on this method, we divide its neighbors into two groups: good neighbors and bad ones. The good neighbors attract him towards themselves and on the contrast the bad ones repulse it. So, all of its neighbors impact on the direction of its next movement.

*2) Movement method for brain individuals:* As mentioned before, the individual who has the lowest cost among its neighbors is called a brain individual. This individual will remain as a brain until there does not exist an individual among its neighbors with lower cost during the next iterations of the algorithm. It is obvious that after migration of a brain node (brain drain), according to the social network relations in the source country, a new brain (brains) will be created.

*3) Movement method for best individual:* By time step t, if no better solution exists than the one a specific individual presents, it will move randomly. The position of this individual is altered by changing one of its parameters. If the new position has a lower cost, the individual confirms it, otherwise ignores it.

The BRADO algorithm has unique optimization properties including the ability to make a right balance between intensification and diversification and the ability to escape local minima, which make it a suitable candidate for the problem being addressed [5].

The paper is organized as follows: Section II provides a brief overview of the methods for assigning tasks. In Section III, we model the problem. In Section IV, we look at the different parts of the algorithm and the changes we need to make. Experimental results in comparison with the results of genetic algorithm are presented in Section V. Finally, Section VI concludes the article.

## II. Literature Review

Tasks allocation is not a new issue and has been discussed in various fields and applications for many years; thus many methods have been proposed for it.

Auction and market-based mechanisms are among the most popular methods for distributed task allocation in multi-robot systems. A. Khams et al. presented a market-based approach to complex task allocation. Both centralized and hierarchical allocations are investigated as winner determination strategies for different levels of allocation and for static and dynamic search tree structures [6], [7]. Liu, L et al. presented a new market-based multi-robot task allocation algorithm that produces optimal assignments [8].

GA have been found quite efficient in solving such complex computational problems. G. Chen et al. presented a new task allocation algorithm that is based on the principles of GA [9]. A. Rauniyar et al. have proposed algorithms based on dynamic genetics [10].

A. Hussein and A. Khamis has conducted a comparative study for allocating tasks is implemented between the proposed market-based approach and two optimization-based approaches (genetic algorithm and simulated annealing algorithm), the results show that the optimization-based approaches outperformed the market-based approach in terms of best allocation and computational time [11].

li et al. proposed a new solution for MRTA based on ant colony algorithm, built up the model of the algorithm and described the task allocation process in details. They realized the simulation of ant colony algorithm based on MATLAB, and then compared the robustness and the best incomes of the four algorithms(Market-Based Approaches: M+ [12]–[14], Behaviors-based approaches: [13], [15], [16]). The simulation results showed that, ant colony algorithm is a high degree of ability and stability for solving MRTA [17].

W. Guo et al.proposed a task allocation algorithm for WSNs. In the proposed algorithm, the construction process of discrete particle swarm optimization (DPSO) is achieved through adopting a binary matrix encoding form, minimizing tasks execution time, saving node energy cost, balancing network load, and defining a fitness function for improving scheduling effectiveness and system reliability [18]. Other papers have used pso algorithm in combination such as [19], [20].

In addition to the above heuristics, other methods are also offered, such as: s. hosseini et al. porposed Distributed Constraint Optimization approach [21] , Approaches based on linear programming [22].

As mentioned before task allocation is an NP problem. Thus, we have to rely on meta-heuristic or approximation approaches to solve this problem. The majority of the literature focuses on approximation algorithms, but, this problem remains a hot topic in the optimization research. The successful application of meta-heuristic algorithms for solving many NP-hard problems, motivated us to apply BRADO and some well-known meta-heuristic algorithms to the task allocation problem.

## III. Problem Formulation and Modeling

In this section, we enter a deeper definition of the problem and then model the problem. Given a set of robots (agents) R, and a set of tasks T each requires one robot. A function A:T→S, mapping each task to a agent in order to be executed. The goal is to assign tasks to robots so as to maximize overall expected performance, taking into account the priorities of the tasks and the skill ratings of the robots. The problem of task allocation can be formulated in many ways. Given our multi-robot systems application domain, it can be formulated as follows:

- *R*: a set of robots $r_i$ , i = 1, 2, . . . n.
- *T*: a set of tasks $t_j$ , j = 1, 2, . . . m.
- *X*: a set that if task j assign to robot $x_{ij}$ = 1, otherwise $x_{ij}$ = 0, j = 1, 2, . . . m, i = 1, 2, . . . n.
- *A*: a set of agents abilities, $a_{ij}$ is the ability of robot i to execute task j.

The problem is to find the optimal allocation of a set of tasks to a subset of sensors, which will be responsible for accomplishing it:

$$A : T \to S \tag{1}$$

The goal is to find a suitable minimum cost assignment of robots to tasks:

$$Minimize(U) \tag{2}$$

Subject to:

- Each task is assigned to only one robot:

$$\sum_{i=1}^{n} x_{ij} = 1, \forall j = 1, 2, ...m \tag{3}$$

- Each robot performs tasks that can minimize the time to perform all its tasks:

$$\sum_{j=1}^{m} x_{ij} a_{ij} \le U, \forall i = 1, 2, ...n \tag{4}$$

If necessary, we can add other constraint to the problem. Next, we solve the problem with BRADO algorithm.

## IV. Proposed Solution

In this section, we explain how to adapt the BRADO algorithm to solve the tasks allocation problem. Here, we focus on those aspects of BRADO that are relevant to the current application.

*1) Representation:* One of the important parts of a successful algorithm is the representation step.
An individual $I_i$ (i =1, 2, ..., $N_{ind}$ ) in the BRADO algorithm is considered as a M-dimensional array of variable values, where M is the number of tasks and $m_i \in$ R denotes the selected robot for task $t_i$.
Table 1 depicts the applied representation. Which displays the $I_i$=[1,4,2,2,3,3,5] in such a way that each dimension is a robot that indicates that the task for that dimension has been assigned to it. For example in 7-dimensional array, Task 4 is assigned to Robot 2.

TABLE I
THE APPLIED INDIVIDUAL REPRESENTATION FOR TASKS ALLOCATION

| Task 1 | Task 2 | Task 3 | **Task 4** | Task 5 | Task 6 | Task 7 |
|---|---|---|---|---|---|---|
| 1 | 4 | 2 | **2** | 3 | 3 | 5 |

*2) Movement method:* According to what was described, there are three different movement methods for ordinary, brain and best individuals in the BRADO algorithm. Here, the three movement methods applied to tasks allocation are described.

- ***Movement method for ordinary individuals***: For ordinary individuals, we use the same basic algorithm in this problem, the position of an ordinary individual is changed based on the force of attraction and repulsion of its neighbors. However, since values associated to an individual indicate expert numbers, a real value such as 4.25 is meaningless. Therefore, in the algorithm these numbers are rounded to the nearest expert number.

$$4.25 \to 4 \tag{5}$$

- ***Movement method for the brain individuals***: In the original BRADO algorithm, the brain individuals move toward the best individual based on the distances between the corresponding variable values. After computing each variables distance from the best, an alteration vector is added to the position of the individual. But, this only works for the numerical variables. For our application in which the parameters are positive integers, and each value indicates an robot number, it is necessary to use another approach. This is because here the distance is meaningless. In this paper, *crossover and mutation operators* of genetic algorithm, are used as the tool to change the position of a brain individual. More specially, crossover produces new individuals by merging sub-individuals from the brain and the best individual in the country. During crossover, a randomly chosen position divides the individuals (A and B) in two parts. The first new individual obtains its first part from A and the second one from B. The other new individual obtains the inverse sub-individuals. In other words, two new tasks allpcation configurations (individuals) arise each receiving sections of their new task assignments from the other assignment. After applying the crossover, one of the two produced team configurations will be omitted randomly. Thereafter, mutation operator randomly changes the value of one task index from M existing task positions. Table II and Table III indicate the crossover operators and table VI indicates mutation operators.

TABLE II
BRAIN AND BEST INDIVIDUALS

| parent | Task 1 | Task 2 | Task 3 | Task 4 | Task 5 | Task 6 |
|---|---|---|---|---|---|---|
| brain | 1 | 4 | 2 | 2 | 3 | 3 |
| best | 5 | 5 | 3 | 2 | 1 | 4 |

TABLE III
THE CROSSOVER OPERATOR APPLIED TO MOVE THE BRAIN INDIVIDUALS WITH SELECTED POSITION FOR DIVISION BETWEEN TASK 3 AND TASK 4

| child | Task 1 | Task 2 | Task 3 | Task 4 | Task 5 | Task 6 |
|---|---|---|---|---|---|---|
| child 1 | 1 | 4 | 2 | 2 | 1 | 4 |
| child 2 | 5 | 5 | 3 | 2 | 3 | 3 |

TABLE IV
THE MUTATION OPERATOR APPLIED TO MOVE THE BRAIN INDIVIDUALS WITH SELECTED POSITION TASK 3

| | Task 1 | Task 2 | Task 3 | Task 4 | Task 5 | Task 6 |
|---|---|---|---|---|---|---|
| brain | 1 | 4 | 2 | 2 | 3 | 3 |
| brain + | 1 | 4 | 5 | 2 | 3 | 3 |

- ***Movement method for the best individual***: For the best individual in each country, a task index i is selected from M existing tasks. Then its value is changed to a randomly selected $m_i \in R$.

## V. Performance Evaluation

In order to evaluate the proposed approach, We considered 7 datasets that randomly consider the cost of each task performed by each robot. Datasets respectively 10, 20, 40, 60, 120, 200 and 400 are tasks. The number of robots for all experiments is 5. We also considered another special data set with 60 tasks so that the cost in the optimal case was 45 and a wrong choice would increase the cost by at least 900.
To show the results of BRADO algorithm, we have shown only solutionthe of the best country.
In two separate sections, we first present the results and then discuss.

*1) Result:* Figure 3 shows the results of the BRADO algorithm and the genetic algorithm for the data set with different numbers of tasks. As we can see, in sets with tasks of 10 and 20, genetic algorithms performed somewhat better; But for a larger data set, we see that the results of the BRADO algorithm are better, and we also have an incremental improvement trend.
Figure 4 shows the results for the mean country and the minimum country of the BRADO algorithm versus the genetic algorithm for the dataset with an optimal solution of 45. We see that the BRADO algorithm has found the optimal answer, while the genetic algorithm has found the answer that is far from optimal.

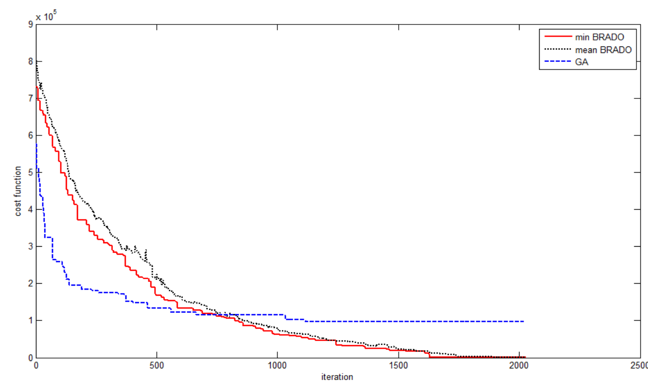
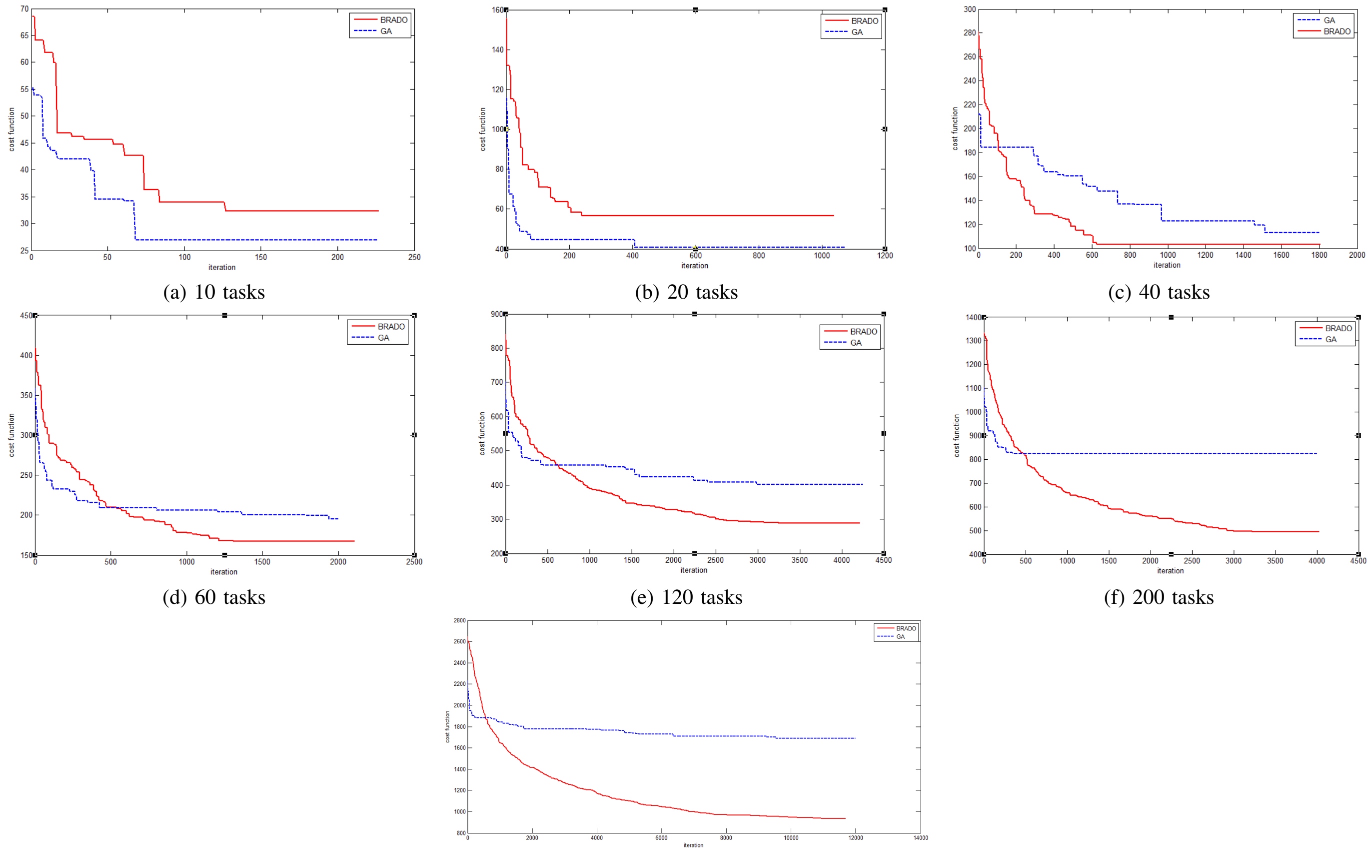



Fig. 4. The results of allocating tasks in datasets with 60 different tasks and And find the optimal solution.

*2) Discuss:* According to Figure 3 of the algorithm genetics, the ratio of the BRADO algorithm gets stuck in the local optimization much later when the number of tasks is high. In BRADO algorithm, we consider a threshold for international migration. It seems, when the number of tasks is small; All solutions quickly converge to one country. Because the number of parameters is small, brain drain is faster. As a result, we have reduced exploration and increased exploitation.
On the other hand, the improvement of the solution compared to the genetic algorithm by increasing the number of tasks in Figure 5 and Table V can be clearly seen.




(a) 10 tasks (b) 20 tasks (c) 40 tasks

(d) 60 tasks (e) 120 tasks (f) 200 tasks

(g) 400 tasks

Fig. 3. The results of allocating tasks in datasets with 10, 20, 40, 60, 120, 200 and 400 different tasks.

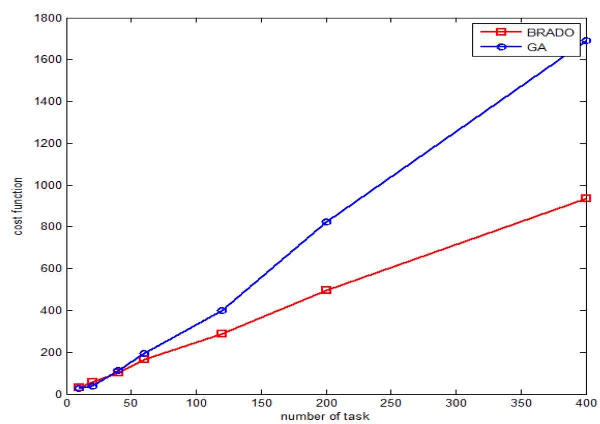



Fig. 5. Comparison of solutions for task allocation BRADO algorithm and algenetic gorithm for different number of tasks

Another point is that in Broadway algorithm according to what is shown in Figure 4 and related explanations; The probability of finding the optimal answer is much higher than the genetics of the algorithm.

## VI. Conclusions and Recommendations

In this paper, we studied the problem of finding task allocation that minimizes the time cost to perform tasks. We introduce BRADO, a new proposed swarm intelligence algorithm, to address this problem. In BRADO, the search algorithm is inspired by the process of real brain drain phenomenon as a human's social behaviour. One of the mainm properties of BRADO is its ability to make a right balance between intensification and diversification. This characteristic helps dealing with task allocation problem which is an NP-hard and high dimensional problem embracing many local optima.

Between the results of genetic algorithm and BRADO algorithm in the dataset with less than 20 tasks, genetic algorithm was slightly better, but in the dataset with more tasks by increasing the number of tasks, we saw a significant improvement in the BRADO algorithm results. The genetic algorithm got stuck in local minima very quickly compared to BRADO algorithm.

Another noteworthy point in this article was that with BRADO algorithm, in dataset where the number of tasks is high, there is a possibility of finding an optimal solution.

## References

[1] L. E. Parker *Multiple mobile robot systems*, in Springer Handbook of Robotics. Springer, pp. 921-941, 2008.

[2] Vig, Lovekesh, and Julie A. Adams. *Coalition formation: From software agents to robots.*, Journal of Intelligent and Robotic Systems 50.1, 2007: 85-118.

[3] N. Maaroju *Thesis Choosing the best heuristic for a NP problem*, Thapar University,Patiala, Department of Computer Engineering and Science, Tapar University, PATIALA-147004, June 2009.

[4] K. Tindell, A. Burns, A. Wellings *Allocating Hard Real Time Tasks (An NP-Hard Problem Made Easy)*, Real Time Systems Research Group Department of Computer Science University of York England.

TABLE V
TASK ALLOCATION BRADO ALGORITHM VS. GA ALGORITHM

| Number Of Tasks | BRADO Algorithm | Genetic Algorithm | Number Of Iteration |
|---|---|---|---|
| 10 | 32.29 | **26.89** | 225 |
| 20 | 56.6 | **40.72** | 1040 |
| 40 | **103.3** | 112.7 | 1800 |
| 60 | **167** | 195.1 | 2000 |
| 120 | **287.6** | 401.6 | 4000 |
| 200 | **495.2** | 823.6 | 4000 |
| 400 | **934.3** | 1689 | 12000 |


[5] J. Basiri, F. Taghiyareh, *Introducing a Socio-inspired Swarm Intelligence Algorithm for Numerical Function Optimization*, Department of Electrical and Computer Engineering University of Tehran, 2014.

[6] A. Khamis, A. M. Elmogy, F. O. Karray *Market-based Dynamic Task Allocation in Mobile Surveillance Systems*,J Intell Robot Syst, Springer.10.1007/s10846-010-9536-2, March 2010.

[7] M. Badreldin, A. Hussein, A. Khamis, *A Comparative Study between Optimization and Market-Based Approaches to Multi-Robot Task Allocation*,Hindawi Publishing Corporation Advances in Artificial Intelligence Volume, Article ID 256524, 2013.

[8] Liu, L. and Shell, D.A. *Optimal Market-based Multi-Robot Task Allocation via Strategic Pricing.* ,2013, June, In Robotics: Science and Systems (Vol. 9, No. 1, pp. 33-40).

[9] G. Chen and J. B. Cruz, Jr *Genetic algorithm for task allocation in UAV cooperative control*, AIAA Guidance, Navigation, and Control Conference and Exhibit 11-14 August 2003, Austin, Texas, AIAA 2003-5582.

[10] A. Rauniyar and P. K. Muhuri *Multi-Robot Coalition Formation Problem:Task Allocation with Adaptive Immigrants Based Genetic Algorithms*, 2016 IEEE International Conference on Systems, Man, and Cybernetics, Budapest, Hungary, 978-1-5090-1897, October 9-12, 2016.

[11] A. Hussein and A. Khamis *Market-based Approach to Multi-robot Task Allocation*, in Springer, International Conference on Individual and Collective Behaviors in Robotics.IEEE, 978-1-4799-28, 2013.

[12] Kalra N, Martinoli, *A Comparative Study Of Market- Based And Threshold-Based Task Allocation,* Distributed Autonomous Robotic Systems, Tokyo: Springer, 2006.

[13] F. Tang, *Automated Synthesis of Multi-Robot Task Solution through Software Reconfiguration,* Proc of IEEE Int Conf on Robotics and Automation, Barcelona, Spain, pp:1501-1508, 2005.

[14] Botelho.S, Alami.R, *M+: a scheme for multi-robot cooperation through negotiated task allocation and achievemen,* Proc of the 1999 IEEE International Conf on Robotics and Automation, Detroit, Michigan, pp: 1234-1239, 1999.

[15] L. E. Parker, *ALLIANCE: An Architecture for Fault Tolerant Multi-robot Cooperation,* IEEE Transactions on Robotics and Automation, Vol. 14, No. 2, pp. 220-240, April 1998.

[16] B. B. Werger, M. J. Matari, *Broadcast of Local Eligibility for Multi-Target Observation,* in Distributed Autonomous Robotic Systems, Springer, pp: 347-356, 2001.

[17] J. P. Wang and Y. Gu and X. M. LI*Multi-robot Task Allocation Based on Ant Colony Algorithm*, , Journal Of Compputer, VOL. 7, NO. 9, September 2012.

[18] W. Guo, J. Li, Senior Member, IEEE, G. Chen, Y. Niu, and C. Chen *A PSO-optimized Real-time Fault-tolerant Task Allocation Algorithm in Wireless Sensor Networks*, IEEE Transactions on Parallel and Distributed Systems, 2014.

[19] X. Wang, X. Zhong, Li Li, R. Lu and Y. Zheng*PSOGT: PSO and Game Theoretic Based Task Allocation in Mobile Edge Computing,* IEEE Computer Society, Volume: 1, Pages: 2293-2298, 2019.

[20] A. Salman,I.Ahmad, S. Al-Madani*Particle swarm optimization for task assignment problem,* Microprocessors and Microsystems Volume 26, Issue 8,363-371, 10 November 2002.

[21] S. H. Semnani and O. A. Basir*Target to sensor allocation: A hierarchical dynamic Distributed Constraint Optimization approach* Computer Communications 36 1024-1038, 2013.

[22] M. Dorigo, M. Birattari, and T. Stitzle, *Ant Colony Optimization: Arificial Ants as a Computational Intelligence Technique,* IEEE computational intelligence magazine, November, 2006.