



**HAL**  
open science

# Scheduling in Real-Time Systems Using Hybrid Bees Strategy

Yahyaoui Khadidja, Bouri Abdenour

► **To cite this version:**

Yahyaoui Khadidja, Bouri Abdenour. Scheduling in Real-Time Systems Using Hybrid Bees Strategy. 6th IFIP International Conference on Computational Intelligence and Its Applications (CIIA), May 2018, Oran, Algeria. pp.375-386, 10.1007/978-3-319-89743-1\_33 . hal-01913890

**HAL Id: hal-01913890**

**<https://inria.hal.science/hal-01913890>**

Submitted on 6 Nov 2018

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

# Scheduling in Real-Time Systems Using Hybrid Bees Strategy

Yahyaoui Khadidja <sup>[1]</sup> and Bouri Abdenour <sup>[2]</sup>  
Department of computer science, Exact sciences of Faculty,  
University Mustapha Stambouli, Algeria

<sup>[1]</sup> yahyaouikhadidja@yahoo.fr

<sup>[2]</sup> bouriabdenour17@yahoo.fr

**Abstract.** In the last decade, stochastic and meta-heuristic algorithms have been extensively used as intelligent strategies to resolve different combinatorial optimization problems. Honey Bee Mating Optimization is one of these most recent algorithms, which simulate the mating process of the queen of the hive. The scheduling algorithm is of paramount importance in a real-time system to ensure desired and predictable behavior of the system. Within computer science real-time systems are an important while often less known branch. Real-time systems are used in so many ways today that most of us use them more than PCs, yet we do not know or think about it when we use the devices in which they reside. Finding feasible schedules for tasks running in hard, real-time computing systems is generally NP-hard. In this work, we are interested in hybridizing this HBMO algorithm with other metaheuristics: Genetic Algorithms (GA), Greedy Random Adaptive Search Procedure (GRASP), Tabu Search (TS) and Simulated Annealing (SA) to resolve a real-time scheduling problem and obtain the optimal tasks schedule with respecting all temporal constraints. This is a complex problem which is currently the object of research and applications. In this scheduling problem, each task is characterized by temporal, preemptive and static periodicity constraints. The quality of the proposed procedure is tested on a set of instances and yields solutions which remain among the best.

**Keywords:** Real-time systems, scheduling, HBMO, Optimization, Metaheuristics.

## Introduction

Finding feasible schedules for tasks running in hard, real-time computing systems is generally NP-hard [1]. A vast amount of work has been done in the area of real-time scheduling by both operations research and computer science communities.

The proposed approach is based on an optimization algorithm inspired from the Honey Bees Mating Optimization (HBMO). It combines an HBMO algorithm, Genetic Algorithms (GA), Simulated Annealing (SA), Tabu Search (TS) and the Greedy Randomized Adaptive Search Procedure (GRASP). The scheduling algorithms can be rated based on different parameters like makespan, flowtime, communication cost, reliability cost, and makespan [2]. The task duplication scheduling scheme has been utilized to minimize the processor idle time and lessen the mean task response time in [3]. In this paper, we aim to obtain a schedule called feasible if both of the following conditions are met 1) No precedence constraints are violated and 2) all temporal

constraints are satisfied. In order to realize this goal, we have used a new approach based on the hybridization of several algorithms. Our strategy uses the HBMO as a base algorithm which combines a number of different procedures. Each of them corresponds to a different phase of the mating process of the honey bees.

In the next section, we see also that the honey bee mating algorithm has never been applied to resolve real-time tasks scheduling problems. This prompted us to use this strategy and hybridizing it with other metaheuristics in order to see its performance to resolve this kind of problem. Our contribution consists in combining the global search represented by HBMO with some local search metaheuristics for the resolution of real-time task scheduling problems in order to intensify the search in promising zones detected by the HBMO exploration process. This hybridization is realized with methods which have been applied in isolation in the resolution of real-time task scheduling problems: GA, TS, SA and GRASP [4, 5] First, we have to select the population of honey bees that will make up the initial hive. In the proposed approach, this population is created by using GA alone, or GRASP alone, or random selection selected. Then a percentage is selected from these three populations using a hybridization of these processes. The best member of each initial population obtained is selected as the queen of the hive. All the other members of each population are the drones.

The two local search strategies SA and TS are used as workers for improving the broods.

The paper is organized in the following way. The related works are presented in the second section. The presentation and formulation of real-time tasks scheduling problems is described in Section 3. Section 4 presents the proposed Scheduling Approach. The resolution process of the scheduling real-time tasks problem is detailed in Section 5. The sixth section presents the simulation and experimental results arrived at. The paper finishes with the conclusion and recommendations for future research.

## **Related works**

The task scheduling problem is defined as one of the many popular academic NP-hard problems. In [6], different aspects in scheduling and issues in various levels of real time systems are described. The use of metaheuristic methods shows their efficiency and effectiveness to solve these categories of complex problems [7]. In the literature, many metaheuristics have been proposed based on methods and approaches to task scheduling. In [8], E. G. Talbi et al. have hybridized Simulated Annealing (SA) with Genetic Algorithms (GA). The authors in [9] resolve the scheduling task problem through hybridization of Particle Swarm Optimization (PSO) with GA.

The comparison between various scheduling algorithms and simulations results is given in [10]. An scheduling algorithm has multiobjective to minimize the total tardness and total number of processors is used in [11]. The authors combine Adaptive Weight Approach with genetic algorithm and simulated annealing method. In order to explore the computing power of quantum computation, an approach based on the hybrid Quantum genetic algorithm is used to resolve the real-time scheduling problem in multiprocessor environment In [12]. In [13], the authors applied Bee Colony

Optimization (BCO) as a heuristic algorithm to solve the problem of static scheduling of independent tasks on identical machines. This strategy is based on the intelligent behavior of honey bees in the foraging process. Koudil et al. adapted the MBO algorithm to solve integrated partitioning/scheduling problems in co-design in [14]. This algorithm gives good results in terms of solution quality and execution time. The Fuzzy logic is also used to solve the real time tasks scheduling problem. In [15], the authors propose a model which in the input stage consists of three linguistic variables i.e. CPU time, deadline and communication overhead.

### 3 Partitioning and scheduling

A real-time application is normally composed of multiple tasks with different levels of criticality : Missing deadlines is not desirable. In this context, three categories of real-time applications are defined [2,16]:

- Soft real-time applications : The system could still work correctly Although the tasks could miss some deadlines. However, missing some deadlines for soft real-time tasks will lead to paying penalties.
- Hard real-time applications : The tasks can not miss any deadline, otherwise, undesirable or fatal results will be produced in the system.
- Firm real-time applications : The tasks, which are such that the sooner they finish their computations before their deadlines, the more rewards they gain.

In this paper, a real-time task  $t_{ij}$  graph is defined by a set of temporal constraints such as  $\{ st(t_{ij}^k), c(t_{ij}^k), ft1(t_{ij}^k), dl(t_{ij}^k) \}$  as well as preemption and static periodicity constraints. The table 1 reports the temporal constraints or parameters description of real-time system to study.

Table 1 Description of the real-time tasks adopted

Temporal parameters	Temporal Parameters description	Temporal parameters definition
$st(t_{ij})$	Start time	Time at which a $j^{i^{eme}}$ task of $i^{eme}$ graph starts its execution
$c(t_{ij})$	Computing time	Time necessary for completion of a $j^{i^{eme}}$ task of $i^{eme}$ graph instance
$ft(t_{ij})$	Completion time	Time at which a $j^{i^{eme}}$ task of $i^{eme}$ graph instance finishes its execution
$dl(t_{ij})$	Deadline time	Time by which execution of the $j^{i^{eme}}$ task of $i^{eme}$ graph instance should be completed, after the task is released.
$\tau_{ij}$	the period of $\tau_{ij}$	The periodic behavior of task, the study be restricted to the. hyperperiod H which is the least common multiple of the periods. $\tau_i$

The task model real-time system is usually described by a set of task graphs. The related concepts are defined as follows :

**System:** A real-time system is modeled as a set  $G = \{G_i, i = 1, \dots, g\}$  of periodic task graph.

**Task graph:** A task graph  $G_i$  is a directed acyclic graph (DAG) of period  $\tau_i$  and defined by a 2-tuple  $G_i = \{T_i, E_i, \tau_i\}$  where :

- $T_i = \{t_{ij}, j = 1, \dots, n_i\}$  represents the set of tasks,
- $E_i = \{e_{ijl}, j = 1, \dots, n_i\}$  represents the set of communications.

**Task :** A task  $t_{ij}$  is a node of task graph and defined by  $t_{ij}^k$  denote the processor implementation ( instance of task).

The problem consists of three parts :

Part 1 : Partitioning  $P$ , which determines the assignement of each task to  $p_u, u \in \{1, \dots, p\}$ , processors :

$$\forall t_{ij}, P(t_{ij}) = p_u, u \in \{1, \dots, p\} \quad (1)$$

Part 2 : Scheduling on processors  $SC_{1_u}$ , which associates each task instance assigned on processors  $\{1, \dots, p\}$  a start time  $st$ .

$$\forall P(t_{ij}) = p_u, t_{ij} \rightarrow SC_{1_u}(t_{ij}^k) = st(t_{ij}^k), \quad (2)$$

$$u \in \{1, \dots, p\}$$

Part 3 : Scheduling on interconnexion system  $SC_{2_l}$ , Which associates each inter-processors communication a start time  $st$ . The communication defined by  $e_{ijl}$  is a directed edge, which means task  $t_{ij}$  precedes  $t_{il}$ . Each communication is associated with a data transmission time on a specific communication channel. It is commonly assumed, in multiprocessor systems research, that communication between tasks assign to the same processor is effectively instantaneous, compared with inter processors communication.

$$e_{ijl}^k \rightarrow SC_{2_l}(e_{ijl}^k) = st(e_{ijl}^k), \{e_{ijl}^k \in E_i^k / P(t_{ij}) \neq P(t_{il})\} \quad (3)$$

Scheduling in real-time systems requires that the order of execution be stipulated. A schedule is called feasible if both of the following conditions are met :

- 1) All precedence constraints are respected and
- 2) All timing constraints are satisfied such as the deadline defined by  $dl(t_{ij}^k)$ . which means the task must be finished before the deadline, otherwise the system will fail. In the multiprocessor system The task scheduling is a mapping process of all tasks to  $p$  processors. The goal of the scheduler is to schedule the system's tasks on these  $p$  processors, so that every task is completed before the expiration of the deadline  $dl(t_{ij}^k)$ .

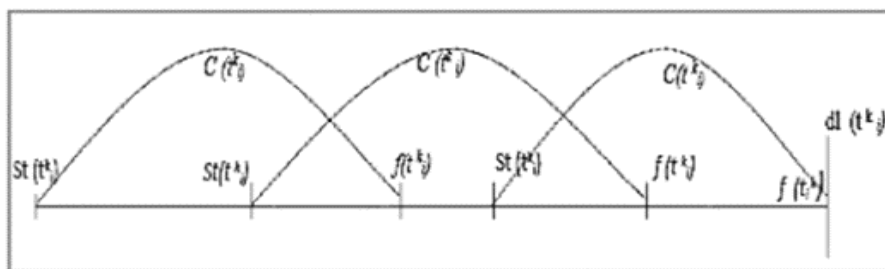
Here we use the similar cost function as [17]. The cost function formulated by equation (4) serves to calculate the tardiness scheduling of the real time system.

$$Tardiness = \sum_k \sum_i \sum_j fcost(t_{ij}^k) = \begin{cases} |dl(t_{ij}^k) - ft(t_{ij}^k)| & \text{if } ft(t_{ij}^k) > dl(t_{ij}^k) \\ 0 & \text{if } ft(t_{ij}^k) < dl(t_{ij}^k) \end{cases} \quad (4)$$

With :

$$\begin{cases} ft(t_{ij}^k) = st(t_{ij}^k) + c(t_{ij}^k) \\ st(t_{ij}^{k-1}) \leq st(t_{ij}^k) \leq dl(t_{ij}^k) - c(t_{ij}^k) \end{cases} \quad (5)$$

The Real-time tasks complete their computing time before their deadlines as early as possible. Different compartments are showed in Figure 1.



**Fig. 1.** Compartments of the real time instances of the task

#### 4 Hybrid HBMO for real time scheduling resolution problem

The HBMO algorithm was proposed by Abbass in [17]. Since then it has been used on a number of different applications[18][19][20][21][22]. HBMO was found to outperform some better known algorithms. However, it has not been applied to real-time scheduling problems. A honey-bee colony consists of the queen(s), drones, worker(s) and broods. The Honey-bee Mating Optimization algorithm mimics the natural mating behavior of the queen bee when she leaves the hive to mate with drones in the air. After each mating, the genetic pool of the queen is enhanced by adding sperm to her spermatheca. Before the mating flight begins, the queen is initialized with a certain amount of energy and only ends her mating flight when the energy level drops below a threshold level (which is close to zero) or when her spermatheca is full. The probability of a drone mating with a queen obeys the following annealing function:

$$Prob(D) = e^{[-\Delta(f\Delta)/speed(t)]} \quad (6)$$

where  $\Delta(f)$  is the absolute difference between the fitness of  $D$  and the fitness of the queen and  $Speed(t)$  is the speed of the queen at a given time  $t$ . After each flight, the queen's speed and energy evolve according to the following equations:

$$Speed(t + 1) = \alpha \times Speed(t) \quad (7)$$

$$Energy(t + 1) = Energy(t) \quad (8)$$

where factor  $\alpha \in (0,1)$  is the amount of speed and energy reduction after each flight. The workers are presented as heuristics whose functionality is to improve the broods produced during the mating process.

To diversify the initial HBMO population and select the best solution as the queen, Genetic Algorithms (GA) and GRASP method are used. The former, GA, works very well on mixed (continuous and discrete) combinatorial problems. Genetic Algorithms create a population of solutions and apply genetic operators such as mutation and crossover to improve the solutions in order to find the best one(s) [24]. The latter, GRASP, is a multi-start or iterative process, in which each GRASP iteration consists of two phases, a construction phase, in which a feasible solution is produced, and a local search phase, in which a local optimum in the neighborhood of the constructed solution is sought. GRASP has proved its efficiency in research and computer science applications as well as in industrial applications.

To improve the broods produced during the queen's mating, we use two workers based on local search metaheuristics: Tabu Search (TS) and Simulated Annealing (SA).

## 6 PRESENTATION OF THE RESOLUTION PROCESS

In this section, we present the approach adapted to resolve the issues addressed. To start with, a set of parameters must be defined:

- The bee population represents the set of scheduling plans of real-time tasks with temporal constraints,
- The queen represents the tasks best scheduling plan in the population generated by GA, by GRASP, randomly or hybridization of a percentage from each of the three populations obtained using the three aforementioned procedures and gradually improved by the implementation of the neighborhood generations iterative procedures (crossover in the HBMO) and two workers SA and TS.
- The drones make up the remaining task scheduling plans

Algorithm 2 describes The HBMO approach for scheduling

*Algorithm 2.* Hybrid HBMO scheduling approach

---

1. Initialization
2. **Generate the initial population of the bees using separately GA, GRASP, Random or through Hybridization of percentages selected from these procedure;**

3. Evaluate the fitness of the plan;
  4. Select the best plan of the population of plans-scheduling which represents the queen;
  5. Selection of the best bee as the queen;
  6. Sperm: Size of the spermatheca;
  7. **E (t)** et **S (t)**: Energy and Speed between respectively [0.5, 1];
  8.  **$\alpha$**  : factor of reduction of energy and speed between [0,1];
  9. **M**: maximum number of mating flights;
  10. **For** i=0 **to** **M** (mating flights)
  11. **do while** E(t) > 0 and Sperm is not full
  12. Select a drone
  13. **if** the drone passes the probabilistic condition  $Prob(D) = e^{[-\Delta(f\Delta)/S(t)]}$
  14. **Add** sperm (plan) of the drone in the spermatheca
  15. **endif**
  16.  $S(t+1) = \alpha \times S(t)$
  17.  $E(t+1) = \alpha \times E(t)$
  18. **enddo**
  19. **do** j = 1 Size of Spermatheca
  20. Select a sperm from the spermatheca;
  21. Generate a brood by crossover the queen's genotype with the selected sperm;
  22. **Improve the brood's fitness** by applying the **workers (Tabu Search /Simulated Annealing)**
  23. **if** the brood's fitness is better than the queen's fitness
  24. **Replace** the queen with the brood
  25. **else Add** the brood to the population of drones
  26. **Endif**
  27. **Replace the drones by the broods**
  28. **Enddo ( for)**
  29. **enddo**
  30. **return** The Queen (Best Solution Found)
- 

A summary of differences and similarities between the HBMO approach applied for the resolution of the real time scheduling problem and the original algorithm are showed in table 1.

Table 1. Differences and similarities between our HBMO-real time scheduling and the original HBMO

Parameters	HBMO-Basic algorithm [17]	HBMO-real time scheduling adopted
Number of queen	01	01
Initial population	Randomly	Randomly, Genetic algorithm, Grasp, Hybridization
Generation Process		
Local search	Greedy SAT (GSAT)	stochastic process of the drone selection
Brood Generation	Crossover haploid & mutation	Genetic crossover operator



Improvement Process (local search)	Mutation FLIP	Simulated annealing, Tabu search
Resultant broods	All broods are killed	All broods will be used in the next mating flight
Fitness function	Fitness function	Objective function <i>lateness(G)</i> of the real time system

## 7 SIMULATION AND EXPERIMENTAL RESULTS

This part is devoted to the implementation and testing of the algorithmic methods developed above. In order to assess the effectiveness of the approaches under study, we constructed a testbench consisting three task systems, which are randomly generated. The systems were composed of 40, 60 and 80 task graphs. Each task graph had its own temporal characteristics. The parameters of the algorithm have been selected after thorough testing. A number of different values were tested and the ones selected are those that yielded the best results in both solution quality and computational time. The best values for these parameters appear in the table 2 and table 3.

Table 2. Parameters and their values for the HBMO approach adopted

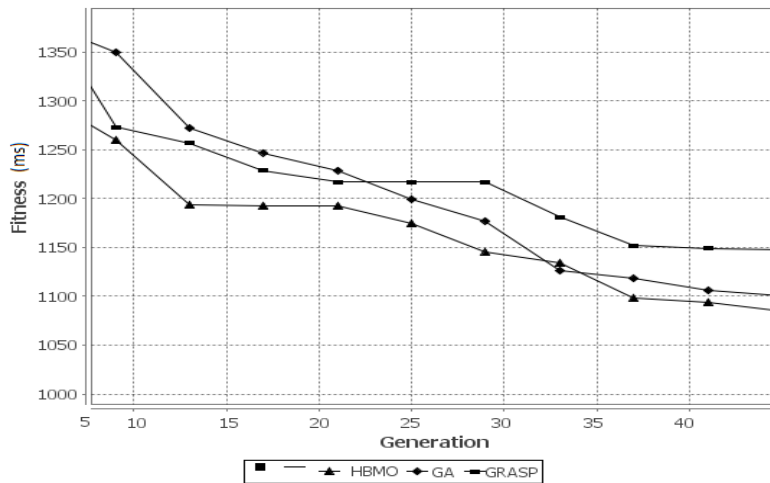
Parameters	Values
Size of population	50
Mating flight	100
Spermatheca	6
Speed	0.80
Energy	0.70
Alpha ( $\alpha$ )	0.20

Table 3. Parameters and their values for the used approaches

Algorithms	Parameters	Values
GA	Size of population	50
	Probability of crossing	0.60
	Probability of Mutation	0.20
	Size of population	70
GRASP	Size of the population	50
	Algorithm of amelioration	Local search
SA	Initial Temperature	Fitness of brood
	Final Temperature	0
	Number of iterations	Until the amelioration or temperature=0

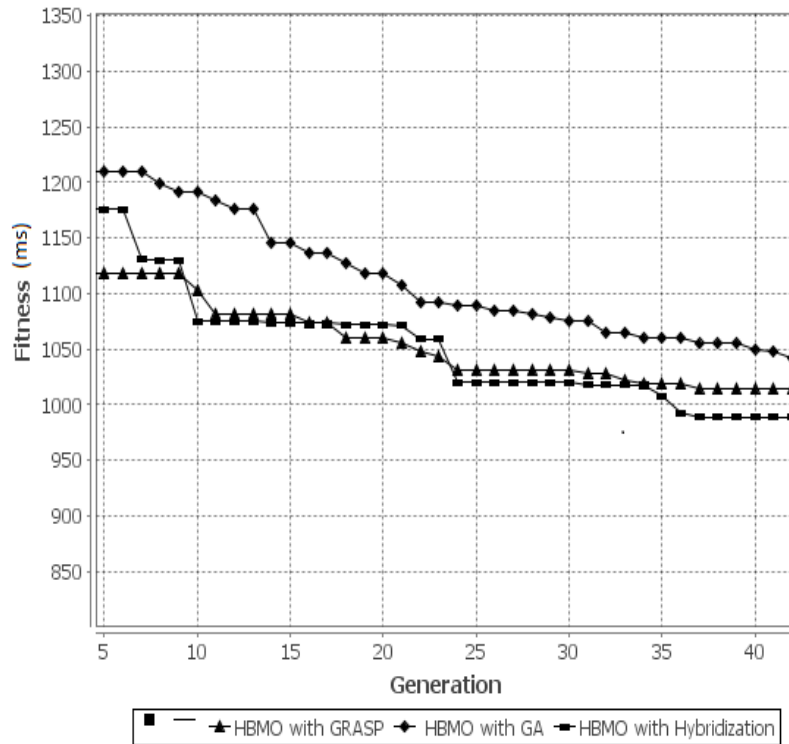
	Dimension of the list	5
TS	Number of iterations	5
	Type of neighborhood	With mutation and recoil

In the first time, we generate the initial population of the proposed algorithm randomly and with GA. After 10 runs , The figure 2 illustrate the graphic results obtained through the application of HBMO, GA and GRASP approaches. The best real-time task scheduling approach is to be selected taking into consideration the Cost function criteria. The graphics results show that the HBMO approach converge to the optimal results comparing with the GA and GRASP methods.



**Fig. 2.** Graphic results obtained through different hybridization processes example of 60 tasks

In order to ameliorate scheduling real-time problem results obtained through the application of basic HBMO (HBMO\_Random) , we have hybridized the HBMO algorithm with other metaheuristics: Genetic algorithms GA, GRASP and hybridization (three sub-populations made up of one percentage from random, one percentage from AG, one percentage from GRASP) in the stage of initial population generation. The figure 3 presents all graphic results obtained through different hybridization processes. The graphics results show that the results given by HBMO-hybridization approach were best than the results given by the HBMO with GA and HBMO with GRASP algorithms. We notice also that the results given by the HBMO-GRASP and HBMO- hybridization were nearly the same. (in the iterations number 34, 24, 16 and 9) the values obtained by HBMO\_ hybridization and the HBMO\_GRASP are the same ).



**Fig. 3.** Graphic results obtained through different hybridization processes (example of 80 tasks).

## 8 CONCLUSION

In this paper, an approach is proposed to solve real-time task scheduling problems. This approach is based on the hybridization of several algorithms inspired from nature such as Genetic, HBMO, Tabu Search and Simulated Annealing and GRASP algorithms. In the stage of generating initial solutions, the adopted approach uses Genetic Algorithms alone, random selection alone, GRASP alone, and hybridization of the three processes selecting a percentage of the population from each. In the stage of solution improvement by the workers, HBMO uses Tabu Search and Simulated Annealing algorithms. The proposed approach was tested on three real-time task scheduling instances with 40 tasks, 60 tasks and 80 tasks where each task has its own temporal constraints. The gain coefficient from the optimal solution is used as the standard to evaluate the quality of the solutions obtained. Results confirm the positive impact of using a hybrid strategy with regard to function objective quality and computing time in comparison with the hybridization algorithms object of this paper.

The results arrived at show that for real-time scheduling problems (with a number of tasks 80), the best approach is HBMO-Hybridization. In the future works, we will focus our searches to resolve the problem of minimizing the energy consumption in the real-time embedded systems by using the proposed algorithms.

## REFERENCES

- 1 Azerou M. and Akl S. G. Scheduling Algorithms for Real-Time Systems , Technical Report No. 2005-499, School of Computing, Queen's University Kingston, Ontario Canada K7L 3N6, July 15, 2005.
- 2 Rathna Devi M and Anju A. Multiprocessor Scheduling of Dependent Tasks to Minimize Makespan and Reliability Cost Using NSGA-II". International Journal in Foundations of Computer Science and Technology (IJFCST), Vol.4(2), March 2014.
- 3 Singh R. An Optimized Task Duplication Based Scheduling in Parallel System. International Journal Intelligent Systems and Applications (IJISA) Vol.8-N8-4, PP.263~7.2016.
- 4 Fleischer and Mark. *Simulated Annealing: Past, Present, and Future*. Proceedings of the Winter Simulation Conference, Department of Engineering Management, Old Dominion University, Norfolk, VA.1995.
- 5 Casey S and Thompson J. GRASP the examination scheduling problem. Volume 2740 of Lecture Notes in Computer Science, pp. 2322~44. Springer, PATAT 2002.
- 6 George DI, Amalarethnam A, Josphin M. Dynamic Task Scheduling Methods in Heterogeneous Systems- A Survey. International Journal of Computer Applications Volume 110 (6), 2015.
- 7 Talbi E G. *Metaheuristics: From Design to Implementation*. Wiley & Sons, Hoboken, New Jersey, 2009.
- 8 Davidovic T, Selmic M and Teodorovic D. Scheduling Independent Tasks: Bee Colony Optimization Approach. 17th Mediterranean Conference on Control & Automation Makedonia Palace, Thessaloniki, Greece June, 2009.
- 9 Sri Raj Pradhan, Sharma S and Konar D. A Comparative Study on Dynamic Scheduling of Real-Time Tasks in Multiprocessor System using Genetic Algorithms. International Journal of Computer Applications. Volume 120 (20), June 2015.
- 10 Yoo M, Yokoyama T. Multiobjective GA for Real Time Task Scheduling. Proceedings of the International MultiConference of Engineers and Computer Scientists (IMECS 2016). Vol I. Hong Kong, 2016.
- 11 Konar D, Bhattacharyya S , Sharma K, Sri Raj Pradhan SR. An Improved Hybrid Quantum-Inspired Genetic Algorithm (HQIGA) for Scheduling of Real-Time Task in Multiprocessor System. Applied Soft Computing · December 2016
- 12 Davidovic T, Ramljak D, Selmic M and Teodorovic D. Bee colony optimization for the p-center problem. Computers and Operations Research Vol. 38, pp. 13671~376, 2011.
- 13 Koudil M, Benatchba K, Tarabet A and Sahraoui E.B. Using Artificial Bees to Solve Partitioning and Scheduling Problems in Codesign. Applied Mathematics and Computation, vol. 186(2), pp.17101~722,2007.
- 14 Nirmala H, Dr.Girijamma H A. Fuzzy Scheduling Algorithm for Real -Time multiprocessor system International Journal of Scientific & Engineering Research, (IJSER). Vol 5 (7) 2014.
- 15 Y. Monnier, J.-P. Beauvais, and A.-M. Deplanche, "A genetic algorithm for scheduling tasks in a real-time distributed system," in Proc. 24th EUROMICRO Conf., pp. 708~714, 1998.
- 16 Garrison W. Greenwood and Lang Hurley C L S. Scheduling Tasks in Real-Time Systems Using Evolutionary Strategies. Proceedings of the 3rd Workshop on Parallel and Distributed Real-Time Systems IEEE 1995.
- 17 Abbass H A. A Single Queen Single Worker Honey-Bees Approach to 3-SAT. The genetic and Evolutionary computation conference, GECCO, san Francisco, USA, 2001.

- 18 Sahoo R R, RakshitMd P and Haidar T. Navigational Path Planning of Multi-Robot using Honey Bee Mating Optimization Algorithm (HBMO). International Journal of Computer Applications (0975 – 8887), Vol. 27(11), pp. 0975~8887, August 2011.
- 19 Marinakis Y , Marinaki M and Dounias G. Honey Bees Mating Optimization algorithm for large scale vehicle routing problems. Nat Comput,vol (9),pp.52~57,2010.
- 20 Sabar N, Ayob M and Kendall G. Solving Examination Timetabling Problems using Honey-bee Mating Optimization (ETP-HBMO). Multidisciplinary International Conference on Scheduling : Theory and Applications (MISTA), Dublin Ireland, 2009.
- 21 Yousefi N and Ebrahimian H. Optimal design of multi-machine power system stabilizers using interactive honey bee mating optimization. Trends in life science (TLS) Dama international Journal. Vol.4 (1), 2015.
- 22 Abedinia O, Naderi M and Ghasemi A. Robust LFC in deregulated environment: Fuzzy PID using HBMO. 10th International Conference Environment and Electrical Engineering (EEEIC),2011.
- 23 Holland J H. Adaptation in natural and artificial systems. MI: University of Michigan Press, 1975.