



Scheduling in Real-Time Systems Using Hybrid Bees Strategy

Yahyaoui Khadidja^(✉) and Bouri Abdenour

Department of Computer Science, Exact Sciences of Faculty,
University Mustapha Stambouli, Mascara, Algeria
yahyaouikhadidja@yahoo.fr, 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

1 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 Sect 2. The presentation and formulation of real-time tasks scheduling problems is described in Sect. 3. Section 4 presents the proposed Scheduling Approach. The resolution process of the scheduling real-time tasks problem is detailed in Sect. 5. Section 6 presents the simulation and experimental results arrived at. The paper finishes with the conclusion and recommendations for future research.

2 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], 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 tardiness 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 witch 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^{ieme} task of i^{eme} graph starts its execution
$c(t_{ij})$	Computing time	Time necessary for completion of a j^{ieme} task of i^{eme} graph i instance
$ft(t_{ij})$	Completion time	Time at which a j^{ieme} task of i^{eme} graph instance finishes its execution
$dl(t_{ij})$	Deadline time	Time by which execution of the j^{ieme} task of i^{eme} graph instance should be completed, after the task is released
τ_{ij}	The period of τ_{ij}	The periodic behavior of task, the study be restricted to the hyperperiod H which is the least common multiple of the periods. τ_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 $G = \{G_i, i = 1, ..g\}$ of periodic task graph.

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

- $T_i = \{t_{ij}, j = 1, ..ni\}$ represents the set of tasks,
- $E_i = \{e_{ijl}, j = 1, ..ni\}$ 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 assignment 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 $SC1_u$, which associates each task instance assigned on processors $\{1, \dots, p\}$ a start time st .

$$\forall P(t_{ij}) = p_u, t_{ij} \rightarrow SCh_{1_u}(t_{ij}^k) = st(t_{ij}^k), \quad u \in \{1, \dots, p\} \quad (2)$$

Part 3: Scheduling on interconnexion system SCh_2_I , 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.

$$e_{ijl}^k \rightarrow SCh_2I(e_{ijl}^k) = st(e_{ijl}^k), \left\{ e_{ijl}^k \in E_i^k / P(t_{ij}) \neq P(t_{il}) \right\} \quad (3)$$

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 Eq. (4) serves to calculate the tardiness scheduling of the real time system.

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

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 comportments are showed in Fig. 1.

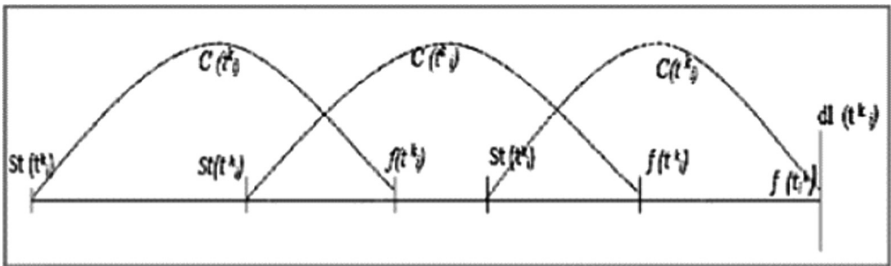


Fig. 1. Comportments of the real time instances of the task

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 [18–22]. 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).

5 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. α : 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 2.

Table 2. 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 generation process	Randomly	Randomly, Genetic algorithm, Grasp, Hybridization
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 $lateness(G)$ of the real time system

6 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 Tables 3 and 4.

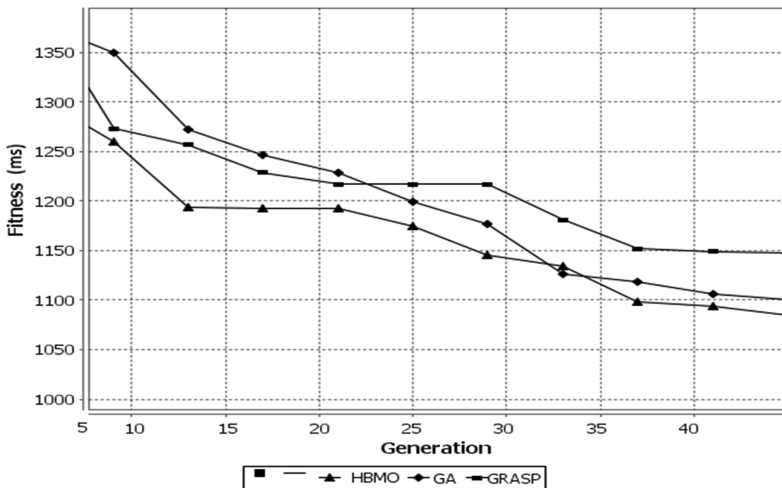
In the first time, we generate the initial population of the proposed algorithm randomly and with GA. After 10 runs, The Fig. 2 illustrate the graphic results obtained through the application of HBMO, GA and GRASP approaches. The best real-time task scheduling approach is to be selected taking into consideration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

Table 3. 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 (α)	0.20

Table 4. 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
TS	Dimension of the list	5
	Number of iterations	5
	Type of neighborhood	With mutation and recoil

**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 Fig. 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).

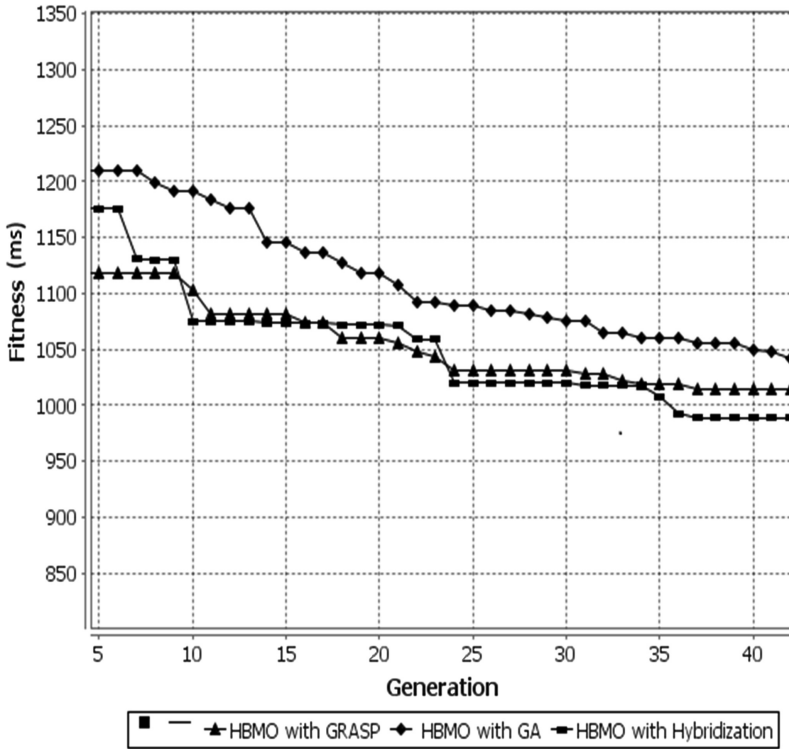


Fig. 3. Graphic results obtained through different hybridization processes example of 80 tasks.

7 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., 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, 15 July 2005
2. Rathna Devi, M., Anju, A.: Multiprocessor scheduling of dependent tasks to minimize makespan and reliability cost using NSGA-II. *Int. J. Found. Comput. Sci. Technol. (IJFCST)* **4**(2), 27–39 (2014)
3. Singh, R.: An optimized task duplication based scheduling in parallel system. *Int. J. Intell. Syst. Appl. (IJISA)* **8**(8), 26–37 (2016)
4. Fleischer, M.: Simulated annealing: past, present, and future. In: *Proceedings of the Winter Simulation Conference*, Department of Engineering Management, Old Dominion University, Norfolk, VA (1995)
5. Casey, S., Thompson, J.: GRASping the examination scheduling problem. In: Burke, E., De Causmaecker, P. (eds.) *Practice and Theory of Automated Timetabling IV*. LNCS, vol. 2740, pp. 232–244. Springer, Heidelberg (2002)
6. George, D.I., Amalarethinam, A., Josphin, M.: Dynamic task scheduling methods in heterogeneous systems - a survey. *Int. J. Comput. Appl.* **110**(6), 12–18 (2015)
7. Talbi, E.G.: *Metaheuristics: from Design to Implementation*. Wiley, Hoboken (2009)
8. Davidovic, T., Selmic, M., Teodorovic, D.: Scheduling independent tasks: Bee colony optimization approach. In: *17th Mediterranean Conference on Control & Automation* Makedonia Palace, Thessaloniki, Greece, June 2009
9. Pradhan, S.R., Sharma, S., Konar, D.: A comparative study on dynamic scheduling of real-time tasks in multiprocessor system using genetic algorithms. *Int. J. Comput. Appl.* **120** (20), 3 (2015)
10. Yoo, M., Yokoyama, T.: Multiobjective GA for real time task scheduling. In: *Proceedings of the International MultiConference of Engineers and Computer Scientists (IMECS 2016)*, vol. 1, Hong Kong (2016)
11. Konar, D., Bhattacharyya, S., Sharma, K., Pradhan, S.R.: An improved hybrid quantum-inspired genetic algorithm (HQIGA) for scheduling of real-time task in multiprocessor system. *Appl. Soft Comput.* **53**, 296–307 (2016)
12. Davidovic, T., Ramljak, D., Selmic, M., Teodorovic, D.: Bee colony optimization for the p-center problem. *Comput. Oper. Res.* **38**, 1367–1376 (2011)
13. Koudil, M., Benatchba, K., Tarabet, A., Sahraoui, E.B.: Using artificial bees to solve partitioning and scheduling problems in codesign. *Appl. Math. Comput.* **186**(2), 1710–1722 (2007)
14. Nirmala, H., Girijamma, H.A.: Fuzzy scheduling algorithm for real – time multiprocessor system. *Int. J. Sci. Eng. Res. (IJSER)*. **5**(7) (2014)
15. Monnier, Y., Beauvais, J.-P., Deplanche, A.-M.: A genetic algorithm for scheduling tasks in a real-time distributed system. In: *Proceedings of 24th EUROMICRO Conference*, pp. 708–714 (1998)
16. Greenwood, G.W., Lang Hurley, C.L.S.: Scheduling tasks in real-time systems using evolutionary strategies. In: *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. In: *The Genetic and Evolutionary Computation Conference, GECCO, San Francisco, USA* (2001)
18. Sahoo, R.R., Rakshit, P., Haidar, T.Md.: Navigational path planning of multi-robot using honey bee mating optimization algorithm (HBMO). *Int. J. Comput. Appl.* **27**(11), 0975–8887 (2011)

19. Marinakis, Y., Marinaki, M., Dounias, G.: Honey Bees Mating Optimization algorithm for large scale vehicle routing problems. *Nat. Comput.* **9**, 52–57 (2010)
20. Sabar, N., Ayob, M., Kendall, G.: Solving examination timetabling problems using Honey-Bee Mating Optimization (ETP-HBMO). In: *Multidisciplinary International Conference on Scheduling: Theory and Applications (MISTA)*, Dublin, Ireland (2009)
21. Yousefi, N., Ebrahimian, H.: Optimal design of multi-machine power system stabilizers using interactive Honey Bee Mating Optimization. *Trends Life Sci. (TLS) Dama Int. J.* **4**(1) (2015)
22. Abedinia, O., Naderi, M., Ghasemi, A.: Robust LFC in deregulated environment: fuzzy PID using HBMO. In: *10th International Conference Environment and Electrical Engineering (EEEIC)* (2011)
23. Holland, J.H.: *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor (1975)