

# Towards Scheduling Optimization through Artificial Bee Colony Approach

Ana Madureira  
GECAD Research Group-School of  
Engineering  
Polytechnic Institute of Porto  
Porto, Portugal  
amd@isep.ipp.pt

Ivo Pereira  
GECAD Research Group-School of  
Engineering  
Polytechnic Institute of Porto  
Porto, Portugal  
iaspe@isep.ipp.pt

Ajith Abraham  
Machine Intelligence Research  
Labs (MIR Labs)  
Scientific Network for Innovation  
and Research Excellence, USA  
ajith.abraham@ieee.org

**Abstract**— In this paper an Artificial Bee Colony Approach for Scheduling Optimization is presented. The adequacy of the proposed approach is validated on the minimization of the total weighted tardiness for a set of jobs to be processed on a single machine and on a set of instances for Job-Shop scheduling problem. The obtained computational results allowed concluding about their efficiency and effectiveness. The ABC performance and respective statistical significance was evaluated.

**Keywords** - Scheduling, Optimization, Swarm Intelligence, Self-organization, Artificial Bee Colony.

## I. INTRODUCTION

In the last decade, with the advent of Computational Intelligence, there is a renewed interest in solving NP-complete scheduling problems through Bio-Inspired Optimization Techniques involving self-organized behavior. Swarm Intelligence (SI) represents a family of approximate optimization techniques that gained a lot of popularity in the past two decades in Evolutionary Computation research area [1][22-24]. This class of techniques is among the most promising and successful optimization techniques, which has been continuously incorporating new ideas, algorithms, and principles from the engineering and basic science communities [2][3].

A new research area, in this field, is known as Bee based Optimization techniques that are inspired by the intelligent foraging behavior of honey bees. In its basic version, the algorithm performs a kind of neighborhood search combined with random search [2]. Bee based Optimization Techniques is the name given to the mechanism or system where a group of bees work in collaboration with each other to find optimal solution for the problem. The Artificial Bee Colony (ABC) algorithm is a swarm based meta-heuristic algorithm that was introduced by Karaboga [2] for optimizing numerical problems. It has been used for combinatorial optimization problems [3], and for unconstrained and constrained optimization problems [4].

The main motivation of this work arises from the interest to define simple and robust approaches and for validate the adequacy of Artificial Bee Colony (ABC) for resolution of hard scheduling optimization problems. This paper presents an ABC based approach for solving real world scheduling problems. Its performance was validated

on Single Machine Scheduling Problem (SMSP) and Job-Shop Scheduling Problem (JSSP) instances.

The remaining sections are organized as follows. Theoretical foundations, the biological motivation and fundamental aspects of SI paradigm with focalization on the design and implementation of an ABC algorithm and some recent applications of ABC to scheduling applications are summarized in section II. In Section III the ABC proposed approach developing for WT is described. Section IV presents computational study and discusses obtained results. Finally, the paper presents some conclusions and puts forward some ideas for future work.

## II. BEES BASED OPTIMIZATION ALGORITHMS

The bees based optimization techniques are a new population-based paradigm of search algorithms that mimics the food foraging behaviour of swarms of honey bees. They were first proposed in 2005 by Pham and collaborators [3] and Karaboga [2] independently. In its basic version, those algorithms perform a kind of neighbourhood search combined with random search and can be used for optimization problems. Pham et al. [3] proposed a Bees Algorithm technical report inspired in the foraging behaviour of honey bees to find food sources. At the same time, Karaboga [2] proposed an ABC algorithm that proposes a similar inspiration in the foraging behaviour of the bees.

Natural bees are social insects living in an organized group called hive and they are known for the production of honey and for the adaptation to changes in the environment in a collective intelligent way. Bees have photographic memory; proper navigation systems use the nectar for energy and use the pollen as protein for the creation of their offspring [5][6].

Generally, a bee colony contains a reproductive female (queen), a few thousand males (drones), and many thousands of sterile females (workers). After mating with multiple drones, the queen produces many young bees ushering in a new generation [5][6]. The queen has a life expectancy of 3 to 5 years, is the mother of all members of the colony, and its main task is mating with the drones, a reproductive operation that is known as mating flight. The

fertilized eggs become females (workers and future queen) and unfertilized eggs become males (drones). Drones are the males in the hive, have a life expectancy of around 90 days and do not live longer than six months. Its main task is to mate with the queen and die after having done it successfully. Worker bees are females without reproductive capacity, representing most of the bees in the hive, and usually live between 4-9 months. In the first half of his life, they are responsible for many tasks in the hive, from maintenance to the defence of same. In the second half of his life, the main task of these bees is looking for food, and initially makes short flights in order to learn the location of the hive and the topology of the environment.

Searching for food is the most important task of a bee colony [6]. This process begins with the search for food sources by exploiting a group of bees, with the aim of extracting nectar. The value of a food source depends on several factors such as the proximity of the hive, the concentration of energy, and ease of food extraction. For the sake of simplicity, the profitability of a food source can be represented by a single numerical value which represents the quality [2]. After unloading the nectar, the explorer bee who found the food source becomes "used" and performs a special move, a sort of dance, in order to share information about the source of food for bees recruit spectators to go exploring this source. Besides the wealth of food source, this shared information also involves the direction and distance from the hive [6].

The sharing of information between the bees is the most important part in the formation of collective knowledge. So there are three different types of dances that bees can do: round dance, waggle dance, and tremble dance [2][6]. If the distance of the food source in relation to the hive is less than 100 meters, the bees perform a round dance, otherwise they perform a waggle dance. While the round dance gives no information about the direction of the food source, the waggle dance gives information about the direction to the sun, the distance from the hive and the quality of food source. Finally, the tremble dance is used when the bee detects delays when discharging its nectar.

Spectators bees watch various dances from different bees and decide which food source they want to use, and there is a higher probability of choosing more profitable sources. Once they choose and locate the food source, spectators bees become in workers bees by exploiting this source to extract its nectar. When the food source is exhausted, employed bees become in explorers [2].

#### A. Artificial Bee Colony algorithm

Real bees are social insects living in organized group called hive. In a beehive, the individuals have some specific tasks performed by specialized individuals. The goal of this organization is to maximize the amount of nectar in the colony getting the utmost of the food sources. The bases of the model of ABC are three types of specialized bees

Employed, Onlooker and Scout that represent a minimal model of the real swarm intelligent forage selection. Table I describes the analogy between artificial and real bees.

TABLE I. ANT ANALOGY BETWEEN NATURAL AND ARTIFICIAL BEES

<i>Natural Bee Colony</i>	<i>Artificial Bee Colony</i>
Food Source	Solution
Quality of nectar	Objective Function
Onlookers	Exploitation of search
Scout	Exploration of search

The ABC algorithm (Table II) have three main phases, corresponding to three types of specialized bees, Employed, Onlooker and Scout, that represent a minimal model of the real swarm intelligent forage selection. Employed bees are in the same number of food sources (solutions) and are responsible to explore one and only one food source at the time and give information to other bees. When an employed bee left his food source becomes a scout bee. Onlooker bees turret in the hive for a information of a employed bees to establish a good food source. Scouts bees seek environment trying to find a new food source depending on an internal motivation or external clues or randomly. Half of the hive is composed by employed bees and the other half by onlooker bees. The food source position represents a solution that is measured by the nectar amount corresponds to the quality of the solution (Table I).

TABLE II. ABC ALGORITHM

Algorithm 3: ABC Algorithm
Initialization of Bee Population
Cycle = 1
While cycle <> Maximum Cycle Number
Employed Bees Phase
Calculate Probabilities for Onlookers
Onlooker Bees Phase
Scout Bees Phase
Memorize the best solution achieved so far
Increment Cycle
EndWhile

In the initialization phase, the algorithm randomly generates  $\frac{sn}{2}$  initial solutions, where  $sn$  is the size of the population, which will be the food field for the employed bees. Each  $x_i$  ( $i=1,2,\dots,\frac{sn}{2}$ ) is a dimensional vector  $D$ . Values between the limits of the parameterization are assigned to the solution and a  $failure_i$  value is also added to analyze when this solution  $i$  must be abandoned. After validating the population, the algorithm repeats a specified number of cycles of employed, onlooker and scout bees phases.

#### 1) Employed bees phase

An employed bee performs a change in their position of food source based on equation (1) and evaluates the nectar amount in the new position/solution [4]:

$$v_{ij} = \begin{cases} x_{ij} + \emptyset(x_{ij} - x_{kj}), & \text{if } R_j < MR \\ x_{ij}, & \text{otherwise} \end{cases} \quad (1)$$

where  $k \in \{1, 2, \dots, sn\}$  is a randomly chosen index that has to be different from  $i$ , and  $\emptyset_j$  is a uniformly distributed random real number in the range of  $[-1, 1]$ .  $R_j$  is uniformly distributed random real number in the range of  $[0, 1]$  and  $MR$  is a control parameter of ABC algorithm in the range of  $[0, 1]$  which controls the number of parameters to be modified.

Then the algorithm selects the solution by the following rules:

- Two realizable solutions – selects the one with the best amount of nectar (fitness) value;
- One solution realizable and one unrealizable – select the realizable;
- Two unrealizable solution - select the one with the smaller degradation factor.

Finished the search, the employed bees share the information with the onlooker bees and the solutions are selected based on a probability by the value of fitness or violation of the solutions depending if they are realizable or not.

### 2) Onlooker bees phase

Onlooker bees select their own food source based on a probabilistic rate according to the amount of nectar on the solution. The algorithm uses the same equation (1) to create a new food source, validating and adjusting the new solution according to the parameterization.

### 3) Scout bees phase

After the above steps, all food sources that will not be explored anymore are abandoned. The employed bees that left the food source get a new position from scouts search.

## B. Bee based approaches for Scheduling resolution

At this section some of the more recent ABC applications will be outlined. According to the literature, bees based algorithms have been applied to a few scheduling applications. Chong et al. [7] proposed an application of Bee Colony Optimization (BCO) to a Job-Shop Scheduling making a comparison of BCO with ACO and Tabu Search. In this work the Tabu search results are clearly more effective than the others in study.

Wong and Chong [8] proposed an improvement to BCO with Big Valley landscape exploitation. The results were compared with Shifting Bottleneck Heuristic, Tabu Search Algorithm and Bee Colony Algorithm with Neighbourhood Search on Taillard JSSP benchmark [9]. Pan et al. [10] proposed an ABC for a Flow-Shop scheduling problem presenting an improvement of the original ABC. In this work were considered different source of food not as a

solution but as discrete job permutation and different neighboring generation. Pansuwan et al. [11] proposed an ABC algorithm based on just in time philosophy in order to minimize the combination of earliness and tardiness penalties cost. The analysis of variance on the experimental results indicated that the performance can be improved dramatically after adopting an optimum parameter setting. Huang and Lin [12] presented an Open-Shop scheduling problem work “with an idle-time-based filtering scheme”, a system that can automatically adapt their behavior stopping the search in solutions with insufficient fitness, decreasing “time–cost for the remaining partial solution time-cost”. Banharnsakun et al. [13] proposed an effective scheduling method based on best-so-far ABC for solving the Job-Shop Scheduling Problem. The authors biased the solution direction toward the best-so-far solution rather a neighboring solution as proposed in the original ABC method. The authors also used the set theory to describe the mapping of their proposed method to the problem in the combinatorial optimization domain. Tasgetiren et al. [14] presented a discrete ABC algorithm to solve the economic lot scheduling problem under extended basic period approach and power-of-two policy. The algorithm provides a cyclic production schedule of  $n$  items to be produced on a single machine such that the production cycle of each item is an integer multiple of a fundamental cycle.

## III. COMPUTATIONAL STUDY

A software tool was developed to perform the computational study aiming to analyse and evaluate the performance of ABC, on resolution of SMSPP for minimization of total weighted tardiness (WT) and Job-Shop Scheduling Problem for *makespan* ( $C_{\max}$ ) minimization. The computational tests were carried out on a PC with Intel Xeon W3565 at 3.20 GHz, with the ABC coded in Java.

In this section, a computational study is carried out in order to analyse SI based algorithms – ABC - on the resolution of benchmark problems considering quality of solutions and computational times.

The ABC based algorithm is evaluated on the resolution of WT instances and its efficiency and effectiveness will be analysed. Statistical analysis is performed in order to estimate the significance and confidence of the obtained results.

We consider that academic benchmark problems are an effective evaluation framework since they have been used by multiple authors and diverse application areas over the years, allowing an efficient comparing framework with previous work related on literature. Additionally, they permit an insight of global behavior and performance on a class of scheduling problems which are our main objective.

### A. Parameters Tuning

The ABC algorithm has a certain number of parameters that need to be set appropriately [15]. As such, we

performed a preliminary study to identify which set of values would yield better results for minimizing total weighted tardiness, for each size in consideration.

After some preliminary parameter tuning we define the parameters ABC algorithms in analysis, considering identical computational effort, to allow a better comparison in efficiency (computational time) and effectiveness (quality of solution).

The solutions are encoded by the natural representation (string), each position corresponds to a job index and the position of the job index is the correspondent processing order. The number of positions on the string corresponds to the number of jobs (problem size).

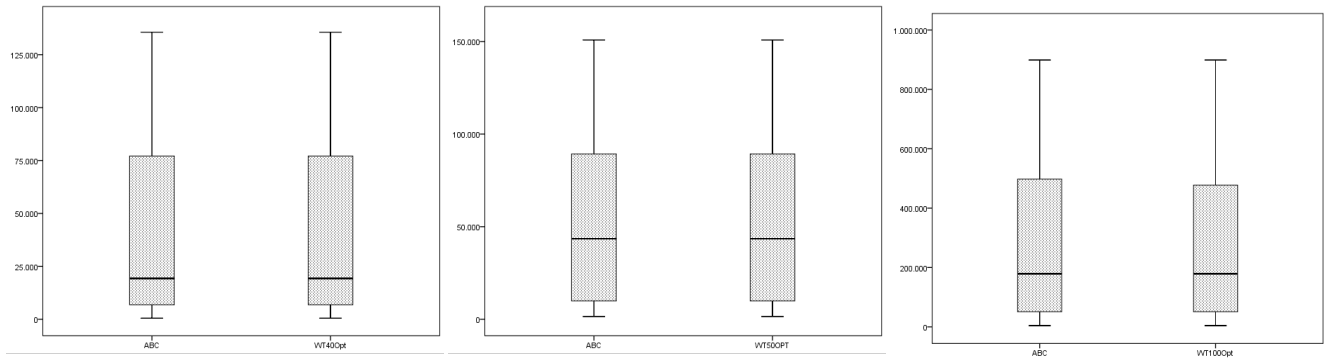
We consider the analogy with Genetic Algorithms populations. The initial bee colony generation process consists in applying some mechanism generator to a starting individual. The initial solution is defined by the priority rule EDD rule, in which an initial solution (bee) is defined by the due dates increasing ordering, thus giving priority to tasks with small due dates. ABC general parameterization is summarized in Table III.

**TABLE III.** ABC PARAMETERIZATION

Parameter	ABC	
	SMSP	JSSP
Size of population	20 (60)	50 (100)
Maximum failure	1000(3000)	1000(2000)
Number of cycles	3000(9000)	3000(4500)

### B. Single Machine Scheduling Problem

At this section we pretend to evaluate the adequacy of Swarm Intelligence to the Weighted Tardiness resolution. ABC effectiveness and efficacy are compared on the resolution of 75 benchmark instances of WT problem for different sizes (40, 50 and 100 jobs).



**Figure 1:** Distribution of Weighted Tardiness for instances with 40, 50 and 100 jobs

Such conclusion could be supported mainly on %error that indicates average deviation from optimal values for instances with 40 jobs ABC obtains in all instances the optimal solutions. For instances with 50 jobs, ABC obtains 22 (in 25) optimal solutions. For instances with 100 jobs, ABC obtains 10 (in 25) optimal solutions.

### 1) Experimental design

For this study, were used 75 instances of Weighted Tardiness Scheduling problems for different sizes available at OR-Library [16]. The first 25 instances for each size were select for testing and not the instances where better results were obtained. The optimal solutions or the best known (ORLibrary) for Weighted Tardiness (WT) will be used as a comparing framework. Performance analysis will be based on quality of solutions and computational times.

### 2) Discussion of Results

We developed n=5 simulations for each instance under analysis. In order to analyze the obtained results, performance measures were computed: the best, the average, and worst value and the deviation error from the best obtained value to the optimal (best known available in OR-Library [16]). The relative percentage of deviation error is determined by %error= (Best-Optimal)/Best formula.

The obtained solutions values by the proposed ABC algorithm are presented in Figure 1. In Table IV is presented the average computational time, the standard deviation, and the number of optimum values.

**TABLE IV.** RESULTS FOR 5 RUNS

No. jobs	Average time (s)	%error	No. of optimal
40	~1	0%	25
50	1.1	0.01%	22
100	12.5	0.2%	10

Additionally, some statistical sampling were retrieved to summarize important features from the observed instances. From the analysis of statistical sampling summary based on WT minimization, it is possible to conclude that does not exist statistic evidence of the difference significance between optimal solutions and ABC performance on WT resolution for 40, 50 and 100 jobs(Figure 1 and Figure 2).

The boxplot from Figure 2 allows the analysis of confidence interval, making its synthesis and compared mean values by ABC and optimal solutions in terms of minimization for Weighted Tardiness (WT). From the boxplot analysis we can conclude that ABC has been effective on the resolution of WT considering that its

obtained mean values were similar to the optimal solutions (in some instances the best known).

This analysis makes possible to conclude that there are not significant differences between ABC obtained results when compared with optimum solutions. This statistical evidence can be observed even on median and dispersion indicators.

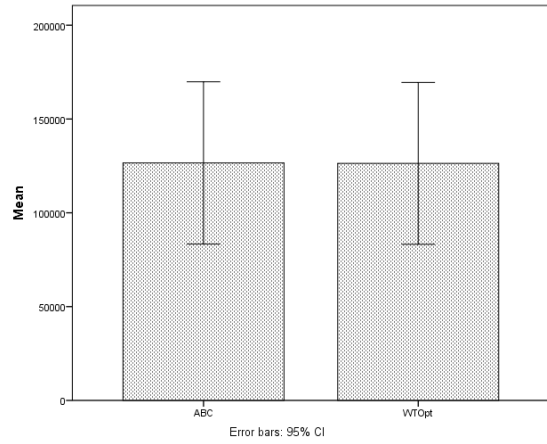


Figure 2: Distribution of Weighted Tardiness for SMSP instances

For the purpose statistical analysis the following hypothesis have been formulated:

- $H_0$  - null hypothesis: mean values of ABC obtained solutions and optimal solutions are not statistically different;
- $H_1$  - alternative hypothesis: mean values of ABC obtained solutions and optimal solutions are statistically different.

A paired-samples  $t$ -test indicated that, with a confidence level of 95%, there is no statistically significant difference between performance of ABC, and optimal solutions related on literature [the median of differences between (ABC-optimal) = 287,720;  $t(74) = 1.069$ ;  $p = 0.289$ ]. There is no statistically significant (null hypothesis is not rejected). These results could suggest that ABC have been effective on WT resolution for all observed instances, considering that does not exist statistical evidence that its performance is significantly different than optimal solutions for WT minimization.

In general most of the instances were solved in relatively short computational time. For  $n=40$  the average time was 1 second, as for  $n=50$  the average time was 1.1 seconds. Only the instances of  $n=100$  took more time to be solved (~12.5 seconds).

### C. Job-Shop Scheduling Problem (JSSP)

At this section the will be described the computational involving a set of instances of Job-Shop Scheduling Problem (JSSP) will be described.

At this section we pretend to evaluate the adequacy of Swarm Intelligence to the Job-Shop Scheduling Problem.

ABC effectiveness and efficacy are compared on the resolution of 38 benchmark instances of JSSP problem.

#### 1) Experimental design

Additionally, the performance was tested on 38 benchmark instances of JSSP from different sizes, available at OR-Library [16]. The instances were selected based on their dimension (number of jobs). For this study we used different problem instances from Fisher and Thompson [17], Lawrence [18], Adams, Balas and Zawack [19], Storer, Wu and Vaccari [20] and Yamada and Nakano [21].

#### 2) Discussion of Results

The boxplot from Figure 3 allows the analysis of confidence interval, making its synthesis and compared mean values by ABC and optimal solutions in terms of minimization of makespan ( $C_{max}$ ) for Job-Shop.

From the bar graph analysis we can conclude that ABC has been effective on the resolution of JSSP considering that its obtained mean values were similar to the optimal solutions (in some instances the best known). It seems, from the graph analysis, that exist significant difference of the performance of ABC when compared with optimal solutions.

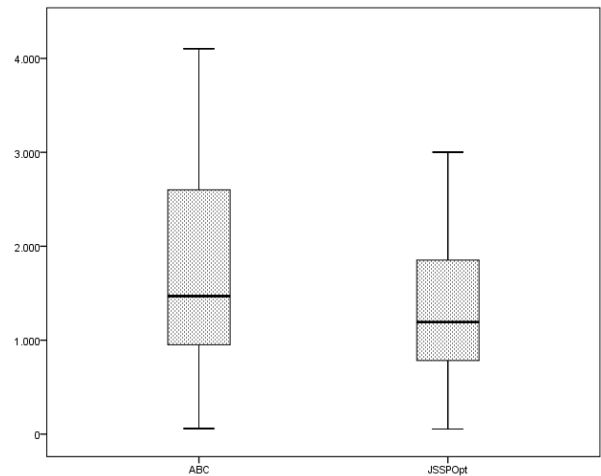


Figure 3: Distribution of makespan for JSSP instances

Considering a significance level  $\alpha=5\%$ , it is possible to conclude that, in general the difference in performance from proposed ABC and optimal solution is significant. For the purpose statistical analysis the following hypothesis have been formulated:

- $H_0$  - null hypothesis: mean values of ABC obtained solutions and optimal solutions are not statistically different;
- $H_1$  - alternative hypothesis: mean values of ABC obtained solutions and optimal solutions are statistically different.

A paired-samples  $t$ -test indicated that, with a confidence level of 95%, there is statistically significant difference between performance of ABC, and optimal solutions related

on literature [the median of differences (ABC – optimal) = 559,132, SD = 552,031)  $t(37) = 6.244$ ;  $p < 0.001$ ]. These results suggest that exists statistical evidence that its performance is significantly different than optimal solutions for JSSP *makespan* optimization (null hypothesis is rejected).

This conclusion indicates that proposed ABC needs additional improvements in order to incorporate domain knowledge for JSSP optimization.

#### IV. CONCLUSIONS AND FUTURE WORKS

The main motivation of this work arises from the interest to define simple and robust approaches and for validate the adequacy of Artificial Bee Colony (ABC) for resolution of hard scheduling optimization problems.

In this paper the effectiveness and efficiency of ABC in the optimization of total weighted tardiness for SMSP and for minimization of *makespan* on JSSP is validated.

More than developing algorithms with unquestionable practice utility, the main purpose of this paper was to illustrate, through more simple scheduling problems, the potential effectiveness and efficiency of using SI approaches, with special emphasis on Bee based techniques for scheduling problem solving.

The obtained results show that proposed ABC algorithm was effective for the instances studied, being possible to find good solutions in short time, i.e., a few CPU seconds.

As future work we will extend our ABC algorithm to the Real World Scheduling problem, where the unpredictable events could occur and change the current scheduling plans.

#### ACKNOWLEDGMENTS

This work is supported by FEDER Funds through the “Programa Operacional Factores de Competitividade - COMPETE” program and by National Funds through FCT “Fundação para a Ciência e a Tecnologia” under the project: FCOMP-01-0124-FEDER-PEst-OE/EEI/UI0760/2011 and PTDC/EME-GIN/109956/2009.

#### REFERENCES

- [1] M. Dorigo, “Swarm Intelligence,” New York: Springer, 2007.
- [2] D. Karaboga, An Idea Based On Honey Bee Swarm For Numerical Numerical Optimization, Technical Report-TR06 (Erciyes University, Engineering Faculty, Computer Engineering Department), 2005.
- [3] D.T. Pham, A. Ghanbarzadeh, E. Koc, S. Otri, S. Rahim and M. Zaidi, The Bees Algorithm. Manufacturing Engineering Centre, Cardiff University, United Kingdom, 2005.
- [4] D. Karaboga, and B. Akay, A modified Artificial Bee Colony (ABC) algorithm for constrained optimization problems. Applied Soft Computing, vol. 11, pp. 3021-3031 2011
- [5] El-Ghazali Talbi, Metaheuristics - From Design to Implementation, Wiley, 2009
- [6] Dervis Karaboga and Bahriye Akay. 2009. A survey: algorithms simulating bee swarm intelligence. Artif. Intell. Rev. 31, 1-4 (June 2009), 61-85.
- [7] Chong, C.S., Low, M.Y.H., Sivakumar, A.I., and Gay, K.L. A bee colony optimization algorithm to job shop scheduling. 2006, Winter Simulation Conference IEEE.
- [8] Wong, L-P. and Chong, C.S., Bee colony optimization algorithm with big valley landscape exploitation for job shop scheduling problems. 2010, Winter Simulation Conference IEEE.
- [9] E.Taillard, Benchmarks for Basic Scheduling Problems. European Journal of Operational Research, vol. 64, pp. 278-285, 1993.
- [10] Pan, Q-K, Tasgetiren, M.F., Suganthan, P.N., and Chua, T.J., A discrete artificial bee colony algorithm for the lot-streaming flowshop scheduling problem. 2010, Information Sciences, Elsevier, Netherlands, In Press.
- [11] P. Pansuwan, N. Rukwong, P. Pongcharoen, Identifying Optimum Artificial Bee Colony (ABC) Algorithm's Parameters for Scheduling the Manufacture and Assembly of Complex Products, Second International Conference on Computer and Network Technology (ICCNT), pp. 339-343, 2010
- [12] Huang, Y-M. and Lin, J-C. A new bee colony optimization algorithm with idle time-based filtering scheme for open shop-scheduling problems. 2011, Expert Systems with Applications 38 pp.5438–5447.
- [13] A. Banharsakun, B. Sirinaovakul, T. Achalakul, Job Shop Scheduling with the Best-so-far ABC, Engineering Applications of Artificial Intelligence, 2011.
- [14] Tasgetiren, M.F.; Bulut, O.; Fadiloglu, M.M.; , "A discrete artificial bee colony algorithm for the economic lot scheduling problem," Evolutionary Computation (CEC), 2011 IEEE Congress on , vol., no., pp.347-353, 5-8 June 2011.
- [15] M. Pirlot, “General Local Search Method,” European Journal of Operational Research, vol. 92, 1996, pp. 493-522.
- [16] OR-Library - <http://people.brunel.ac.uk/~mastjib/jeb/info.html>.
- [17] H. Fisher and G.L. Thompson, Probabilistic learning combinations of local job-shop scheduling rules, J.F. Muth, G.L. Thompson (eds.), Industrial Scheduling, Prentice Hall, Englewood Cliffs, New Jersey, pp.225-251, 1963.
- [18] S. Lawrence, Resource constrained project scheduling: an experimental investigation of heuristic scheduling techniques (Supplement), Graduate School of Industrial Administration, Carnegie-Mellon University, Pittsburgh, Pennsylvania, 1984.
- [19] J. Adams, E. Balas, and D. Zawack, The shifting bottleneck procedure for job shop scheduling, Management Science 34, pp. 391-401, 1988.
- [20] R.H. Storer, S.D. Wu, R. Vaccari, New search spaces for sequencing instances with application to job shop Management Science 38, pp. 1495-1509, 1992.
- [21] T. Yamada, R. Nakano, A genetic algorithm applicable to large-scale job-shop instances, R. Manner, B. Manderick (eds.), Parallel instance solving from nature 2, North-Holland, Amsterdam, pp. 281-290, 1992.
- [22] Hongbo Liu, Ajith Abraham and Maurice Clerc, Chaotic Dynamic Characteristics in Swarm Intelligence, Applied Soft Computing Journal, Elsevier Science, Volume 7, Issue 3, pp. 1019-1026, 2007.
- [23] R. Thangaraj, M. Pant, A. Abraham and P. Bouvry, Particle Swarm Optimization: Hybridization Perspectives and Experimental Illustrations, Applied Maths and Computation, Elsevier Science, Netherlands, Volume 217, No. 1, pp. 5208-5226, 2011.
- [24] H. Liu, A. Abraham, O.K. Choi and S.H.Moon, Variable Neighborhood Particle Swarm Optimization for Multi-objective Flexible Job-shop Scheduling Problems, The Sixth International Conference on Simulated Evolution And Learning (SEAL06), China, Springer Verlag, Germany, Lecture Notes in Computer Science, T.D.Wang et al. (Eds.): SEAL 2006, LNCS 4247, pp. 197-204, 2006.