SWARM INTELLIGENCE FOR SCHEDULING: A REVIEW

Ana Madureira
*amd@isep.ipp.pt**
GECAD
ISEP/ IPP
Porto, Portugal

Nelson Sousa
nffs@isep.ipp.pt
GECAD
ISEP/ IPP
Porto, Portugal

Ivo Pereira
iasp@isep.ipp.pt
GECAD
ISEP/ IPP
Porto, Portugal

Abstract: Swarm Intelligence generally refers to a problem-solving ability that emerges from the interaction of simple information-processing units. The concept of *Swarm* suggests multiplicity, distribution, stochasticity, randomness, and messiness. The concept of *Intelligence* suggests that problem-solving approach is successful considering learning, creativity, cognition capabilities. This paper introduces some of the theoretical foundations, the biological motivation and fundamental aspects of swarm intelligence based optimization techniques such Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO) and Artificial Bees Colony (ABC) algorithms for scheduling optimization.

Keywords: Decision support system(s), Meta-Heuristics, Scheduling tools for production management, Optimization techniques for manufacturing production.

INTRODUCTION

Evolutionary Computation (EC) techniques have been applied to the scheduling of orders in manufacturing systems, leading to a research area called Evolutionary Scheduling that is at the interface of Artificial Intelligence (AI) and Operational Research.

Scheduling problems are generally complex, large scale, constrained, and multi-objective in nature, and classical operational research techniques are often inadequate at solving them effectively. With the advent of computation intelligence, there is a renewed interest in solving scheduling problems through evolutionary computational techniques. Several EC techniques have been proposed. A summary of the most significant chronology milestones of evolutionary computation techniques is presented in table 1.

Evolutionary Computation is a growing research field of AI. EC is the general term for several computational techniques which use ideas and get inspiration from natural evolution/adaptation, such as natural selection and genetic inheritance. EC could be categorized in two main areas: the Evolutionary Algorithms (EA) and the Swarm Intelligence (SI).

An emerging area of research of Evolutionary Computation is the Swarm Intelligence (SI). SI is a new computational and behavioural paradigm for solving distributed problems based on self-organization. While its main principles are similar to those underlying the behaviour of natural systems consisting of many individuals, such as ant colonies and flocks of birds, SI is continuously incorporating new ideas, algorithms, and principles from the engineering and basic science communities.

Table 1 - Evolutionary Techniques Chronology

Evolutionary Technique	Authors	Year	Refs.
Evolutionary programming	Fogel, Owens and Walsh	1966	[1]
Genetic Algorithms	Holland	1975	[2]
Scatter Search	Glover	1977	[3]
Artificial Immune Systems	Farmer, Packard and Perelson	1986	[4]
Swarm Intelligence	Beni and Wang	1989	[5]
Memetic Algorithms	Moscato	1989	[6]
Ant Colony Search Algorithm	Colorni, Dorigo and Maniezo	1992	[7]
Cultural Algorithms	Reynolds	1994	[8]
Particle swarm optimization	Kennedy and Eberhart	1995	[9]
Path Relinking	Glover	1996	[10]
Diferencial Evolution	Storn and Price	1996	[11]
Bees Algorithms	Pham, Ghanbarzadeh, Koc, Otri, Rahim and Zaidi	2005	[12]
Artificial Bee Colony	Karaboga and Akay	2005	[13]

This paper aims at review some of the most recent contributions to the Evolutionary Scheduling research area.

The remaining sections of this paper are organized as follows: initially the theoretical foundations, the biological motivation and fundamental aspects of SI paradigm with focalization on the design and implementation of the Particle Swarm Optimization (PSO) Ant Colony Optimization (ACO) and Artificial Bees Colony (ABC) algorithms are summarized. Then, some recent applications of SI optimization methods to scheduling resolution are presented and, finally, the paper presents some conclusions.

SWARM INTELLIGENCE OPTIMIZATION METHODS

Swarm Intelligence is a relatively new approach to problem solving that takes inspiration from the collective intelligence of swarms of biological populations, and was discovered through simplified social behaviours model simulation of insects and of other animals [14]. Among the most promising SI inspired optimization techniques are ACO, PSO and ABC optimization algorithms.

Ant Colony Optimization

Ant Colony Optimization (ACO) takes inspiration from the foraging behaviour of some ant species. These ants deposit pheromone on the ground in order to mark some favourable path that should be followed by other members of the colony. ACO exploits a similar mechanism for solving optimization problems.

The ACO algorithm is a probabilistic technique for solving computational problems which can be reduced to finding good paths through graphs. This algorithm, initially proposed by Marco Dorigo in his PhD thesis [7], is a member of ant colony algorithms family, in SI methods, and it constitutes some Meta-Heuristic optimizations.

Table 2 - Analogy between Natural and Artificial Ants

Natural Ant Colony	*Artificial Ant Colony*
Ant	Agent
Ant Colony	Set of Ants/Iterations
Pheromone	Diversity Mechanism
Path	Solution
Evaporation	Pheromone update

The first ACO algorithm is known as Ant System was aiming to search for an optimal path in a graph. It was based on the foraging behaviour of ants seeking a path between their colony and a source of food. The original idea has since diversified to solve a wider class of numerical problems, and as a result, several problems have emerged, drawing on several aspects of the behaviour of ants [15]. The general ACO algorithm is described in Algorithm 1 (Table 3). After initialization, the metaheuristic iterates over three phases: at each iteration, a number of solutions are constructed by the ants; these solutions could be then improved through a local search (this step is optional), and finally the pheromone is updated through evaporation and by increasing the pheromone levels associated with a chosen set of good solutions.

Table 3 –Ant Colony Optimization Algorithm

Algorithm 1: Ant Colony Optimization Metaheuristic

```
Set ACO parameters.
Initialize pheromone trails
While termination criteria not met do
     Construct AntSolutions
     Apply Localsearch (optional)
     Update Pheromones
EndWhile
```

Ant System is the first ACO algorithm proposed in the literature [7]. Its main characteristic is that, at each iteration, the pheromone values are updated by all the *m* ants that have built a solution in the iteration itself. The pheromone τ_{ij}, associated with the edge joining cities *i* and *j*, is updated as follows:

$$\tau_{ij} = (1 - \rho) * \tau_{ij} + \sum_{k=1}^{m} \Delta\tau_{ij}^{k} \quad (1)$$

where ρ is the evaporation rate, *m* is the number of ants, and τk_{ij} is the quantity of pheromone laid on edge (*i, j*) by ant *k*:

$$\Delta\tau_{ij}^{k} = \begin{cases} Q/L_k & \text{if ant k uses edge (i, j) in its tour} \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

where *Q* is a constant, and L_k is the length of the tour constructed by ant *k*.

In the construction of a solution, ants select the following city to be visited through a stochastic mechanism. When ant *k* is in city *i* and has so far constructed the partial solution S^p, the probability of going to city *j* is given by:

$$p_{ij}^{k} = \begin{cases} \dfrac{\tau_{ij}^{\alpha} * \eta_{ij}^{\beta}}{\sum_{c_{ij} \in N(S^p)} \tau_{ij}^{\alpha} * \eta_{ij}^{\beta}} & \text{if } c_{ij} \in N(S^p) \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

where $N(s^p)$ is the set of feasible components; that is, edges (*i*, *j*) where *j* is a city not yet visited by the ant *k*. The parameters α and β control the relative importance of the pheromone versus the heuristic information η_{ij}, which is given by:

$$\eta_{ij} = \frac{1}{d_{ij}} \quad (4)$$

where d_{ij} is the distance between cities *i* and *j*.

Particle Swarm Optimization

Particle Swarm Optimization (PSO) is a population based stochastic optimization technique developed by Eberhart and Kennedy [9], inspired by social behaviour of bird flocking or fish schooling. PSO shares many similarities with evolutionary computation techniques such as Genetic Algorithms (GA).

Table 4 - Analogy between Birds and PSO

Bird Flocking	Particle Swarm Optimization
Bird	Particle
Bird Flocking	Particle Swarm
Area overflown by birds	Search space
Bird localization during flight	Position
Bird localization where it found food or the nest	Optimal solution
Flight capabilities	Fitness
Best known position of bird	*pBest* (Individual Experience)
Best known position of oh whole population	*gBest* (Collective experience)

The algorithm is initialized with a population of random solutions and searches for optimal solution by updating generations. However, unlike GA, PSO has no evolution operators such as crossover and mutation. In PSO, the potential solutions, called particles, fly through the problem space by following the current optimum particles. The detailed information will be given in following sections. Compared to GA, the advantages of PSO are that it is easier to implement and there are few parameters to adjust.

The particle swarm concept originated as a simulation of simplified social system. The original intent was to graphically simulate the choreography of a bird block or fish school. However, it was found that particle swarm model can be used as an optimizer.

Table 5 - Particle Swarm Optimization Algorithm

Algorithm 2: Particle Swarm Optimization Metaheuristic

```
Initialize particles population in hyperspace
While termination criteria not met do
      Evaluate fitness of individual particles
      Modify velocities based on previous best
      and global best
EndWhile
```

In PSO, instead of using genetic operators, as in the case of GAs, each particle (individual) adjusts its flying according with its own and group experiences. The general PSO algorithm is described in Algorithm 2 (Table 5).

Each particle is treated as a point in a D-dimensional space and is manipulated as described below in the original PSO algorithm:

$$v_{id} = v_{id}+c_1 * rand()(p_{id}-x_{id})+c_2\, rand()(p_{gd}-x_{id}) \quad (5)$$

$$x_{id} = x_{id} + v_{id} \quad (6)$$

where c_1 and c_2 are positive constants and *rand*() is a random function in the range [0,1], $X_i = (x_{i1}, x_{i2}, \ldots, x_{iD})$ represents the ith particle, $P_i = (p_{i1}, p_{i2}, \ldots, p_{iD})$ is the best previous position (position giving the best fitness value) of the particle, the symbol g represents the index of the best particle among all particles in the population, and $V_i = (v_{i1}, v_{i2}, \ldots, v_{iD})$ is the rate of the position change (velocity) for particle i.

Equation (5) describes how the velocity is dynamically updated and equation (6) the position update of the “flying” particles. Equation (5) is divided in three components, namely the momentum, the cognitive and the social component. In the first component, the velocity cannot be changed abruptly: it is adjusted based on the current velocity. The second component represents the learning from its own flying experience. The third component consists on the group learning flying experience [9].

Most applications of optimization algorithms are tailored to static problems. Many real-world systems, however, change its state frequently. These system state changes result in a requirement for frequent, sometimes almost continuous, re-optimization. It has been demonstrated that PSO can be successfully applied to tracking and optimizing dynamic systems [16].

The first parameter added into the original PSO algorithm is the inertia weight ω. The dynamic equation of PSO with inertia weight is modified to be:

$$v_{id} = \omega v_{id}+c_1\, rand()(p_{id}-x_{id})+c_2\, rand()(p_{gd}-x_{id}) \quad (7)$$

$$x_{id} = x_{id} + v_{id} \quad (8)$$

where ω constitutes the inertia weight that introduces a balance between the global and the local search abilities. A large inertia facilitates a global search while a small inertia weight facilitates the local search.

Bees Based Algorithms

The Bees Algorithm is a new population-based search algorithm, first developed in 2005 by Pham et al. [12] and Karaboga et al. [13] independently. The algorithm mimics the food foraging behaviour of swarms of honey bees. In its basic version, the algorithm performs a kind of neighbourhood search combined with random search and can be used for optimization problems.

In 2005, Pham proposed a *Bees Algorithm* in a technical report [12] inspired in the foraging behaviour of honey bees to find food sources, has an optimization algorithm to find a optimal solution. At the same time Karaboga [13] proposes a very similar Artificial Bee Colony (ABC) algorithm that proposes a similar inspiration in the foraging behaviour of the bees.

Table 6 - Analogy between Natural and Artificial Bees

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

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.

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 is 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 correspond to the quality of the solution.

Table 7 - ABC Algorithm

Algorithm 3: ABC Algorithm

```
Initialization of Bee Population
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
EndWhile
```

Initialization of bee population

In the initialization phase the algorithm generates randomly an initial distributed solutions, with $\frac{sn}{2}$ solutions were sn is the size of the population, which will be the food field for the employed bees. Each x_i (i=1,2,…,$\frac{sn}{2}$) it's a dimensional vector D. The ABC algorithm does not assure that all initial solution is realizable to increase the performance of the algorithm. These types of solution will disappear by the normal acting of the algorithm. Values between the limits of the parameterization are assigned to the solution and a $failure_i$ value is addicted to analyze when this solution i must be abandoned.

After the ABC validates the population do a repeated cycles of searches of employed, onlooker and scout bees.

Employed bee phase

An employed bee performs an alteration in their position of food source based in an equation and evaluates the nectar amount in the new position.

$$v_{ij} = \begin{cases} x_{ij} + \emptyset(x_{ij} - x_{kj}), \ if\ R_j < MR \\ x_{ij},\ otherwise \end{cases} \quad (9)$$

where $k \in \{1,2,\dots,sn\}$ is randomly chosen index that has to be different from i and $\emptyset_{ij}$ is uniformly distributed random real number in the range of [−1,1]. Rj 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.

Fig. 1 - Equation for new solution [13]

After a new solution the algorithm select the solution by the follow rules:

- Two realizable solutions – select 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 bee share the information with the onlooker bees and the solutions are select based in a probabilistic selection by the value of fitness or violation of the solutions depending if they are realizable or unrealizable.

Onlooker bee phase

The onlooker bee select is own food source based in a probabilistic rate according to the nectar solution amount. That uses the same equation (9) to create a new food source validate and adjusting the new solution according to the parameterization.

Scout bee phase

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

SWARM INTELLIGENCE FOR SCHEDULING

Scheduling is a decision-making process that is used on a regular basis in many manufacturing and services industries. It deals with the allocation of resources to tasks over given time periods and its goal could be the optimization of one or more optimization objectives. In current and global competitive environments effective sequencing and scheduling has become imperative for survival in the market-place. Scheduling appears in diverse areas of science, engineering and industry and takes different forms depending on the restrictions and optimization criteria of the operational environments.

Table 8 - A non exhaustive genealogy of scheduling contributions

Work/Technique	Authors	Year	Refs
Book "Work, Wages, and Profits"	Gantt	1916	[17]
Various Optimizers for Single Stage Production	Smith, Johnson and Jackson	1956	[18]
Scheduling presentation to AI community	Fox	1982	[19]
Genetic algorithms	Davis,	1985	[20]
	Yamada and Nakano	1991	[21]
Shifting bottleneck	Adams, Balas and Zawack	1988	[22]
Simulated Annealing	Matsuo, Suh and Sullivan	1988	[23]
Tabu Search	Barnes, Laguna and Glover	1992	[24]
Ant Colony Optimization	Van der Zwaan and Marques	1999	[25]
Particle Swarm Optimization	Jerald, Asolcan, Prabaharan, et al.	2004	[26]
	Cagnina, Esquivel and Gallard		[27]
Bee based Algrithms	Chong, Low, Sivakumar and Gay	2006	[28]
	Pan, Tasgetiren, Suganthan and Chua	2010	[29]

References to scheduling contributions in manufacturing could be situated at the beginning of 20th century with the work of Henry Gantt and other pioneers. In 1916, in his book "Work, Wages, and Profits" [17], Henry Gantt explicitly discusses scheduling, especially in the Job-shop environment.

Some of the first scheduling publications appear in the industrial engineering and operations research literature associated to Naval Research Logistics Quarterly in the early fifties and contained results by W.E. Smith, S.M. Johnson and J.R. Jackson [18]. The scheduling problem was introduced with some impact to the community of Artificial Intelligence in 1982 by Mark S. Fox, through the paper titled "Job-Shop Scheduling: An Investigation in Constraint-Directed Reasoning" [19].

Since then several proposals through Exact Methods and AI based optimization, generally referred as Meta-heuristics or Nature Inspired Techniques, have been referred in literature. A non exhaustive genealogy of the most significant contributions to scheduling resolutions is presented in table 8.

Swarm Intelligence-based techniques have been applied to a wide range of combinatorial optimization and search problems in which the task is to accommodate a set of entities such as events, activities, resources and people into a time-space so that the available resources are utilized as efficiently as possible and the additional constraints are satisfied. Examples of scheduling problems are production scheduling, personnel scheduling, transport scheduling, scheduling for the web, project scheduling, etc.

In this work we concentrate our review to the scheduling resolution in manufacturing systems.

Ant Colony Optimization

Ant Colony optimization is probably the most successful example of artificial/engineering swarm intelligence system with numerous applications to real-world problems. ACO is one of the most successful techniques in the wider field of swarm intelligence. The significant research efforts on ACO have established it as a mature metaheuristic that can lead to very effective algorithms for many difficult optimization problems.

Van der Zwaan and Marques [25] developed an improved Ant System for the Job-Shop scheduling problem and study the difference that parameterization makes in an ACO. Huang and Liao [30] propose hybridization between ACO and Tabu Search in a Job Shop problem using a specific pheromone trail definition inspired from Shifting Bottleneck method. Recently, Sun, Wang and Fang [31] present an approach based on ACO optimization to a discrete Job-Shop Scheduling.

Merkle and Middendorf [32] describe a contribution for solving permutation problems to Single Machine Problems for Total Weighted minimization. Blum [33] proposes an implementation of ACO in an Open Shop problem using a hybrid approach combining ACO with Bean Search.

Liao and Juan [34] propose an ACO to minimize the tardiness in a Single Machine Problems with utilization of setup times. In Yagmahan and Yenisey [35] a multi-objective scheduling problem approach based on ACO for scheduling to reduce the total scheduling cost is proposed.

Madureira and Pereira [36] proposed a novel approach for the resolution of Dynamic Scheduling Problem by combining different techniques, like

Autonomic Computing (AC), Multi-Agent Systems (MAS), Case-based Reasoning (CBR), and Bio-Inspired Optimization Techniques (mainly Ant Colony Optimization).

Particle Swarm Optimization

PSO has been applied to different problems and is another example of successful artificial/engineering swarm intelligence paradigm. Within little more than a decade hundreds of papers have been reported successful applications of PSO to scheduling. An extensive survey of PSO applications is made by Poli [37].

The application to scheduling problems reports to Tasgetiren et al. [38] work presenting an application of PSO to Single Machine Problem for Total Weighted Tardiness optimization. Sha and Hsu [39] presents a hybrid PSO technique adjusting the features of some parameters to a better utilization of PSO in Job Shop scheduling problem and applying Tabu Search to depurate the final solution. More recently in [40] same authors present a work using PSO in an Open Shop scheduling problem where they propose a “new” PSO method with some improvements in particle comportment. In Liao, Tseng and Luarnb [41] a utilization of PSO in a Flow-Shop scheduling problem using an application of PSO with Local Search is described. The authors performed a test suite comparing PSO with some Genetic Algorithms techniques. Ercan [42] explores in his work the implementation of PSO, Hybrid PSO and other optimization techniques for a hybrid Flow-Shop scheduling problem.

In 2010, Chen et al. [43] proposes an improved PSO approach to solve the resource-constrained scheduling problem. Low Hsu and Su [44] present a work with the application of PSO in a Single-Machine Scheduling Problem with periodic maintenance. The authors adjust the original algorithm to a most efficient application considering features of a Single Machine Problem. Lin et al. [45] proposes an “efficient” Job-Shop algorithm based in PSO. The authors refer a MPSO algorithm that is a combination of PSO, multi-type individual enhancement scheme and random-key encoding scheme.

Madureira et al. [46] proposed a hybrid approach for Dynamic Manufacturing Scheduling Optimization through Collective Intelligence and Swarm Intelligence. The proposed system is applied to the problem of jobs scheduling on dynamic manufacturing environments.

Bees Based Algorithms

The Bee based algorithms are the most recent SI optimization methods under study in this work. A few scheduling applications to this algorithm have been referred on literature.

Chong et al. [28] propose 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 [47] propose in 2008 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 [48]. Results show that it is comparable to these approaches.

Pan et al. [29] proposes 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 neighbouring generation. Huang and Lin [49] presents an Open-Shop scheduling problem work “with an idle-time-based filtering scheme”, a system that can automatically adapt their behaviour stopping the search in solutions with insufficient fitness, decreasing “time–cost for the remaining partial solution time-cost“.

CONCLUSIONS AND FURTHER WORK

Evolutionary Computation techniques have been applied to the scheduling of orders in manufacturing systems, leading to a research area called Evolutionary Scheduling that is at the interface of Artificial Intelligence and Operational Research.

Theoretical foundations of swarm intelligence paradigm with main focus on the implementation and illustration of ACO, PSO and ABC optimization algorithms have been discussed in detail, followed by an overview of optimization techniques highlighting the first main applications of Nature Inspired Optimization techniques for scheduling resolution. Finally, some of the most current and noteworthy applications of Swarm Intelligence based algorithms for scheduling resolution in manufacturing systems are surveyed.

ACKNOWLEDGMENT

The authors would like to acknowledge FCT, FEDER, POCTI, POSI, POCI, POSC, and COMPETE for their support to R&D Projects and GECAD.

References

[1] Fogel, L.J., Owens, A.J., and Walsh, M.J., Artificial Intelligence through Simulated Evolution. 1966, John Wiley, New York, NY.
[2] Holland, J. H., Adaptation in Natural and Artificial Systems. 1975, University of Michigan Press. Ann Arbor.
[3] Glover, F., Heuristics for integer programming using surrogate constraints. 1977, Decision sciences, 8:pp. 156-166.
[4] Farmer, J.D., Packard, N., and Perelson, A., The immune system, adaptation and machine learning. 1986, Physica D, vol. 2, pp. 187–204
[5] Beni, G. and Wang, J., Swarm Intelligence in Cellular Robotic Systems. 1989, Proceed. NATO Advanced Workshop on Robots and Biological Systems, Tuscany, Italy, June 26–30.
[6] Moscato, P., On Evolution, Search, Optimization, Genetic Algorithms and Martial Arts: Towards Memetic Algorithms. 1989, Caltech Concurrent Computation Program (report 826).
[7] Dorigo, M., Optimization, Learning and Natural Algorithms. 1992, PhDThesis, Politecnico di Milano, Italy, (in Italian).
[8] Reynolds, R.G., An Introduction to Cultural Algorithms. 1994, in Proceedings of the 3rd Annual Conference on Evolutionary Programming, World Scienfific Publishing, pp 131–139.
[9] Kennedy, J. and Eberhart, R.C., Particle Swarm Optimization. 1995, in Proceedings of the IEEE International Conference Neural Networks, pp 1942-1948.
[10] Glover, F., Tabu Search and adaptive memory programmimg, Advances, Applications and Chalenges, Kluwer. 1996, pp. 1-75.
[11] Storn, R., On the usage of differential evolution for function optimization. 1996, Biennial Conference of the North American Fuzzy Information Processing Society (NAFIPS), pp. 519–523.
[12] Pham, D.T., Ghanbarzadeh, A., Koc, E., Otri, S., Rahim, S., and Zaidi, M., The Bees Algorithm. 2005, Manufacturing Engineering Centre, Cardiff University, United Kingdom.
[13] Karaboga, D. and Akay, B., A modified Artificial Bee Colony (ABC) algorithm for constrained optimization problems. 2011, Applied Soft Computing, vol. 11, pp. 3021-3031
[14] Dorigo, M., Swarm Intelligence. 2007, Springer New York.
[15] Dorigo, M., Birattari, M., and Stützle, T., Ant Colony Optimization - Artificial Ants as a Computational Intelligence Technique. 2006, IEEE Computational Intelligence Magazine.
[16] Eberhart, R.C. and Shi, Y., Tracking and optimizing dynamic systems with particle swarms. 2001, Proc. Congress on Evolutionary Computation, Seoul, Korea.
[17] Gantt, H.L., Work, Wages, and Profits, second edition, Engineering Magazine Co., New York. 1916, Reprinted by Hive Publishing Company, Easton, Maryland, 1973.
[18] Smith, W.E., Various Optimizers for Single Stage Production. 1956, Naval Research Logistics Quarterly, Vol. 3, pp. 59–66.
[19] Zweben, M. and Fox, M.S., Intelligent Scheduling. 1994, Morgan Kaufman Publishers, San Francisco.
[20] Davis, L., Job shop scheduling with genetic algorithms. 1985, in J.J. Grefenstette (ed.), Proceedings of an International Conference on Genetic Algorithms and Their Applications, Carnegie-Mellon University, Pittsburgh, 136-140.
[21] Yamada, T. and Nakano, R., A genetic algorithm applicable to large-scale job-shop problems. 1991, in R. MSnner and B. Manderick (eds.), Parallel Problem Soloing from Nature 2, Elsevier, Amsterdam, 281-290.
[22] Adams, J., Balas, E., and Zawack, D., The shifting bottleneck procedure for job shop scheduling. 1988, Management Science 34, 391-401.
[23] Matsuo, H., Suh, C.J., and Sullivan, R.S., A controlled search simulated annealing method for the general jobshop scheduling problem. 1988, Working paper 03-04-88, University of Texas at Austin.
[24] Barnes, J.W, Laguna, M., and Glover, F., An overview of tabu search approaches to production scheduling problems. 1992, Proc. Symposium Intelligent Scheduling Systems, San Francisco, 30-50.
[25] Van der Zwaan, S. and Marques, C., Ant colony optimization for job shop scheduling. 1999, Proc 3rd Workshop in Genetic Algorithms and Artificial Life.
[26] Jerald, J., Asolcan, P., Prabaharan, G., et al. Scheduling Optimization of Flexible Manufacturing Systems Using Particle Swarm Optimization Algorithm. 2004, Advanced Manufacturing Technology
[27] Cagnina, L., Esquivel, S., and Gallard, R. Particle Swarm Optimization for Sequencing Problems: A Case Study. 2004, Proceedings of 2004 Congress on Evolutionary Computation, Oregon, Portland, pp. 536-540.
[28] 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.
[29] 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.
[30] Huang, K-L. and Liao C-J., Ant colony optimization combined with taboo search for the job shop scheduling problem. 2006, Computers & Operations Research 35, pp.1030–1046.
[31] Sun, B., Wang, H., and Fang, Y., Application of Ant Colony Algorithm in Discrete Job-shop Scheduling. 2009, Second International Symposium on Knowledge Acquisition and Modeling, IEEE.
[32] Merkle, D. and Middendorf, M. On solving permutation scheduling problems with ant colony optimization. 2005, International Journal of Systems Science Vol. 36, No. 5, p. 255–266.
[33] Blum, C., Beam-ACO—hybridizing ant colony optimization with beam search: an application to open shop scheduling. 2005, Computers & Operations Research 32, 1565-1591.
[34] Liao, C. and Juan, H. An ant colony optimization for single-machine tardiness scheduling with sequence-dependent setups. 2007, Computers & Operations Research 34, 1899–1909.

[35] Yagmahan, B. and Yenisey, M.M., Ant colony optimization for multi-objective flow shop scheduling problem. 2008, Computers & Industrial Engineering 54, 411-420.
[36] Madureira, A. and Pereira, I., Intelligent Bio-Inspired System for Manufacturing Scheduling under Uncertainties. 2011, International Journal of Computer Information Systems and Industrial Management Applications, ISSN 2150-7988, volume 3, pp. 072-079.
[37] Poli, R. Analysis of the publications on the applications of particle swarm optimisation. 2008, Journal of Artificial Evolution and Applications: 1–10.
[38] Tasgetiren, M.F., Sevkli, M., Liang, Y.C., and Gencyilmaz, G., Particle swarm optimization algorithm for single machine total weighted tardiness problem. 2004, in Proceedings of the IEEE congress on evolutionary computation, vol. 2, Oregon: Portland.
[39] Sha, D.Y. and Hsu C-Y., A hybrid particle swarm optimization for job shop scheduling problem. 2006, Computers & Industrial Engineering 51 pp.791–808.
[40] Sha, D.Y. and Hsu C-Y, A new particle swarm optimization for the open shop scheduling problem. 2008, Computers & Operations Research 35 p.3243 – 3261.
[41] Liao, C-J., Tseng, C-T., and Luarn, P. A discrete version of particle swarm optimization for flowshop scheduling problems. 2007, Computers & Operations Research 34, pp.3099–3111.
[42] Ercan, M.F., Particle Swarm Optimization and Other Metaheuristic Methods in Hybrid Flow Shop Scheduling Problem. 2009, InTech Publication.
[43] Chen, R.M., Wu, C-L., Wang, C-M., and Lo, S-T., Using novel particle swarm optimization scheme to solve resource-constrained. 2010, Expert Systems with Applications 37 p. 1899–1910.
[44] Low, C., Hsu, C-J., and Su, C-T., A modified particle swarm optimization algorithm for a single-machine scheduling problem with periodic maintenance. 2010, Expert Systems with Applications 37, pp.6429–6434.
[45] Lin, T-L., Horng, S.J., Kao, T.W., Chen, Y-H., Run, R-S., Chen, R-J., Lai, J-L., and Kuo, I-H. An efficient job-shop scheduling algorithm based on particle swarm optimization. 2010, Expert Systems with Applications 37 pp. 2629–2636.
[46] Madureira, A., Pereira. I., and Sousa, N., Collective Intelligence on Dynamic Manufacturing Scheduling Optimization. 2010, IEEE Fifth International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA 2010), Liverpool, United Kingdom.
[47] 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.
[48] Taillard, E., Benchmarks for Basic Scheduling Problems. 1993, European Journal of Operational Research, vol. 64, pp. 278-285.
[49] 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.