

Analysis and Development of Various Evolutionary Computations for Multi-Objective Optimization in Job Shop Problems

Mamta
Research Scholar

Umesh Sehgal
Associate Professor

Abstract – The job shop problem is an optimization problem in computer science in which ideal jobs are assigned to resources at particular times. It is as follows: There are given n jobs J_1, J_2, \dots, J_n of varying sizes, which need to be scheduled on m identical machines, while trying to minimize the make span. The make span is the total length of the schedule (that is, when all the jobs have finished processing). It is mostly presented as an online problem, that is, each job is presented, and the online algorithm needs to make a decision about that job before the next job is presented. Job shop scheduling is concerned with finding a sequential allocation of competing resources that optimizes a particular objective function [1]. There are many variations of the problem exist, which can be summarized as follows:

- Machines can be related, independent, equal
- Machines can require a certain gap between jobs
- Machines can have sequence-dependent setups
- Objective function can be to minimize the make span, the L_p norm, etc
- Jobs may have constraints, for example a job i need to finish before job j can be started.
- Jobs and machines have mutual constraints, for example, certain jobs can be scheduled on some machines only.

Keywords – Multi Objective Optimization Problems.

I. INTRODUCTION

The simplest form of the **offline makes span minimization** problem deals with atomic jobs. The atomic jobs are not subdivided into multiple operations. It is equivalent to packing a number of items of various different sizes into a fixed number of bins, such that the maximum bin size needed is as small as possible. Hochbaum and Shmoys presented a polynomial-time approximation scheme in 1987 that finds an approximate solution to the offline make span minimization problem with atomic jobs to any desired degree of accuracy.

- Jobs consisting of multiple operations
- Multi-objective optimization

II. METHODS OF EVOLUTIONARY COMPUTATION

Evolutionary computation is iterative progress, such as growth or development in a population. This population is then selected in a guided random search using parallel processing to achieve the desired end. Such processes are often inspired by biological mechanisms of evolution. As evolution can produce highly optimized processes and

networks, it has many applications in computer science [3]. The forms of evolutionary computation include the following:

- Evolutionary algorithms such as genetic algorithms and genetic programming.
- Swarm intelligence algorithms such as ant colony, bee colony or particle swarm optimization.

III. EVOLUTIONARY ALGORITHMS

Evolutionary algorithms often perform well approximating solutions to all types of problems in fields such as engineering, art, biology, economics, marketing, genetics, operations research, robotics, social sciences, physics, politics and chemistry. **Evolutionary algorithm** (EA) is a part of evolutionary computation, a generic population-based metaheuristic optimization algorithm. *Artificial evolution* (AE) describes a process involving individual *evolutionary algorithms*; EAs are individual components that participate in an AE. An EA uses some mechanisms inspired by biological evolution: reproduction, mutation, recombination, and selection.

Evolutionary computation and algorithms have also been used as an experimental framework within which to validate theories about biological evolution and natural selection, particularly through work in the field of artificial life. Artificial life studies the logic of living systems in artificial environments [4]. Evolutionary algorithms are a practical application of the weak a life principle applied to problems. **Swarm intelligence (SI)** is the discipline that deals with collective behavior of decentralized, self-organized systems, natural or artificial. The expression was introduced by Gerardo Beni and Jing Wang in 1989, in the context of cellular robotic systems [5]. SI systems are typically made up of a population of simple agents interacting locally with one another and with their environment. The inspiration often comes from nature, especially biological systems. The agents follow very simple rules. The examples of systems studied by swarm intelligence are colonies of ants and termites, schools of fish, flocks of birds, herds of land animals and bacterial growth. Some human artifacts also fall into the domain of swarm intelligence, notably some multi-robot systems, and also certain computer programs that are written to tackle optimization and data analysis problems.

- *Altruism algorithm* is given by the researchers in Switzerland. They have developed an algorithm based on Hamilton's rule of kin selection. The algorithm shows how

altruism in a swarm of entities can, over time, evolve and result in more effective swarm behavior [6].

- *Ant colony optimization* is a class of optimization algorithms modeled on the actions of an ant colony. ACO methods are useful in problems that need to find paths to goals. Artificial 'ants'—simulation agents—locate optimal solutions by moving through a parameter space representing all possible solutions. Real ants lay down pheromones directing each other to resources while exploring their environment. The simulated 'ants' similarly record their positions and the quality of their solutions, so that in later simulation iterations more ants locate better solutions [7].

- *Artificial bee colony algorithm* is a swarm based meta-heuristic algorithm introduced by Karaboga in 2005 [8], and simulates the foraging behavior of honey bees. The ABC algorithm has three phases: employed bee, onlooker bee and scout bee. In the employed bee and the onlooker bee phases, bees exploit the sources by local searches in the neighborhood of the solutions selected based on deterministic selection in the employed bee phase and the probabilistic selection in the onlooker bee phase. In the scout bee phase which is an analogy of abandoning exhausted food sources in the foraging process, solutions that are not beneficial anymore for search progress are abandoned, and new solutions are inserted instead of them to explore new regions in the search space.

- *Artificial immune systems* concerns the usage of abstract structure and function of the immune system to computational systems, and investigating the application of these systems towards solving computational problems from mathematics, engineering, and information technology.

- *Charged system search (CSS)* is a new optimization algorithm based on some principles from physics and mechanics [9]. CSS utilizes the governing laws of Coulomb and Gauss from electrostatics and the Newtonian laws of mechanics.

- *Cuckoo search (CS)* mimics the brooding behavior of some cuckoo species. This cuckoo search algorithm [10] can outperform other algorithms such as particle swarm optimization.

- *Intelligent water drops algorithm (IWD)* is a swarm-based nature-inspired optimization algorithm [11], which has been inspired by natural rivers and how they find almost optimal paths to their destination. The IWD algorithm is generally a constructive population-based optimization algorithm [12].

- *Particle swarm optimization (PSO)* is a global optimization algorithm for dealing with problems in which a best solution can be represented as a point or surface in an n-dimensional space. Hypotheses are plotted in this space and seeded with an initial velocity, as well as a communication channel between the particles [13]. Particles then move through the solution space, and are evaluated according to some fitness criterion after each time step. The main advantage of such an approach over

other global minimization strategies such as simulated annealing is that the large numbers of members that make up the particle swarm make the technique impressively resilient to the problem of local minima.

IV. A REVIEW WORK

The behaviour of a single ant, bee, termite and wasp often is too simple, but their collective and social behaviour is of paramount significance. The national Geographic TV Channel reveals that advanced mammals including lions also enjoy social lives, perhaps for their self-existence at old age and in particular when they are wounded. The collective and social behaviour of living creatures motivated researchers to undertake the study of today what is known as *Swarm Intelligence*. Historically, the phrase Swarm Intelligence (SI) was coined by Beni and Wang in late 1980s (Beni and Wang, 1989) in the context of cellular robotics. A group of researchers in different parts of the world started working almost at the same time to study the versatile behaviour of different living creatures and especially the social insects. There is a lot of research done in the area of centralized job scheduling. As many approaches are out of the scope of this research, we only provide a brief overview of relevant ones.

In 1959, Pierre-Paul Grassé invented the theory of Stigmergy to explain the behavior of nest building in termites [14]. In 1983, Deneubourg and his colleagues studied the collective behavior of ants [15]. Moyson Manderick have an article on self-organization among ants [16] in 1988 and the work of Goss, Aron, Deneubourg and Pasteels on the collective behavior of Argentine ants, which will give the idea of Ant colony optimization algorithms in 1989 [17]. The implementation of a model of behavior for food is given by Ebling and his colleagues in 1989 [18]. In 1991, M. Dorigo proposed the Ant System in his doctoral thesis (which was published in 1992 [19]). A technical report extracted from the thesis and co-authored by V. Maniezzo and A.H Colorni was published five years later. In 1996, publication of the article on Ant System [20], and then also Hoos and Stützle invent the MAX-MIN Ant System in 1996 [21]. In 1997, Dorigo and Gambardella publish the Ant Colony System [22]. Schoonderwoerd and his colleagues developed the first application to telecommunication networks in 1997 [23]. Dorigo in 1998, launches first conference dedicated to the ACO algorithms [24] 1998, Stützle proposes initial parallel implementations [25]. In 1999, Bonabeau, Dorigo and Theraulaz publish a book dealing mainly with artificial ants [26]. The special issue of the Future Generation Computer Systems journal on ant algorithms is published in 2000 [27]. Gutjahr in 2000 provides the first evidence of convergence for an algorithm of ant colonies [28]. In 2001, the first use of COA Algorithms is presented by companies (Eurobios and AntOptima). IREDA and his colleagues published the first multi-objective algorithm [29]. In 2002, Bianchi and her colleagues suggested the first algorithm for stochastic problem [30]. In 2004,

Zlochin and Dorigo show that some algorithms are equivalent to the stochastic gradient descent, the cross-entropy and algorithms to estimate distribution [31]. The first applications on protein folding problems are given in 2005. Apinanthana Udomsakdigool (2010) examine in the real world situation any scheduling problems faced by decision maker are involved more than one aspect and therefore multiple criteria analysis is required.

4 a) Artificial Bee Colony (ABC) Algorithm

There have been extensive studies of scheduling algorithms and heuristics in both static and dynamic job shops for decades by researchers and practitioners (Gere 1966, Blackstone et. al. 1982, Rajendran and Holthaus 1999, Jain and Meeran 1999). The objective of scheduling is to determine the job schedules that minimize (or maximize) a measure (or multiple measures) of performance (Rajendran and Holthaus 1999). Nakrani and Tovey (2004) have given a new honey bee algorithm for dynamic allocation of internet servers. Since 2005, D. Karaboga and his research group [32] have been studying the ABC algorithm and its applications to real world problems. Karaboga and Basturk have investigated the performance of the ABC algorithm on unconstrained numerical optimization problems and its extended version for the constrained optimization problems and Karaboga et al. applied ABC algorithm to neural network training. In 2010, Hadidi et al. employed an Artificial Bee Colony (ABC) Algorithm based approach for structural optimization [33].

b) Particle swarm optimization

The first antecedents in the field of computer graphics of particle swarm optimization can be traced back to the work of Reeves (1983), who proposed particle systems to model objects that are dynamic and cannot be easily represented by polygons or surfaces. The examples of such objects are fire, smoke, water and clouds. In these systems, particles are independent of each other and their movements are governed by a set of rules. Some years later, Reynolds (1987) used a particle system to simulate the collective behavior of a flock of birds. In a similar kind of simulation, Heppner and Grenander (1990) included a *roost* that was attractive to the simulated birds. Both models inspired the set of rules that were later used in the original particle swarm optimization algorithm. Particle swarm optimization was introduced by Kennedy and Eberhart (1995). It has roots in the simulation of social behaviors using tools and ideas taken from computer graphics and social psychology research [34].

The rules that govern the movement of the particles in a problem's search space can also be seen as a model of human social behavior in which individuals adjust their beliefs and attitudes to conform to those of their peers (Kennedy & Eberhart 1995). Social psychology research, in particular the dynamic theory of social impact (Nowak, Szamrej & Latané, 1990), was another source of inspiration in the development of the first particle swarm optimization algorithm (Kennedy, 2006). **Ventresca and B.M. Ombuki (2004)** examines an application of the recently proposed adaptive metaheuristic Ant Colony

Optimization (ACO) [35] for the job shop scheduling problem (JSSP). In the static JSSP, a finite number of jobs are to be processed by a finite number of machines. Each job consists of a predetermined sequence of task operations, each which needs to be processed without interruption for a given period of time on a given machine. The tasks of the same job cannot be processed concurrently and each job must visit each machine exactly once. A feasible schedule is an assignment of operations to time slots on a machine without violation of the job shop constraints. A makespan is defined as the maximum completion time of the jobs. The objective of the JSSP is to find a schedule that minimizes the makespan. A good schedule is one that minimizes the total amount of time machines are idle. According to complexity theory [36], the JSSP is characterized as NP-hard combinatorial optimization problem. Obtaining exact solutions for such problems is computationally intractable. Inspired by the study of Argentine ants conducted by Goss et al. [37], Marco Dorigo developed the ACO which uses principles of cooperative behaviour found in real ants colonies to solve hard combinatorial optimization problems. Indirect communication and information exchange between ants is achieved by mimicking the communicative behaviour (so-called *foraging and recruiting behaviour*) of real ant colonies. In the ACO, each ant constructively builds a solution by several stepwise probabilistic decisions until a solution is reached.

The ACO meta-heuristic has been applied to various hard combinatorial optimization problems. For example, in the scheduling field, ACO has effectively been applied to the Flow-shop problem [38], Resource Constraint project Scheduling problem [39], the Single Machine Total Tardiness problem [40]. ACO has also been shown to solve other permutation scheduling problems such as the Travelling Salesman problem [41] and Vehicle routing problems. However, the application of the ACO to Shop scheduling problems such as the JSSP and open shop scheduling problem has proved to be quite difficult and very few papers report work on ACO implementation for the JSSP. Colorni et. al were the first to apply an ant system (AS) to the JSSP in [42]. Although the contribution of the results obtained using the ACO for the JSSP are significant to the scheduling community, "the algorithm was far from reaching state-of-the-art performance". Sjoerd van der Zwaan stated that Al [43] next developed an ACO for the JSSP where he used genetic algorithm for ACO parameter tuning. More recently C. Blum et. al. have investigated the application of ACO to shop scheduling problems including the JSSP [44]. In this, we develop an ant colony optimization approach to the job shop scheduling problem that utilizes a local search technique (we call *Foot Stepping*) which aids in exploration of the solution space. Furthermore, unlike most local search techniques which are problem specific, the foot stepping based strategy is more flexible as it does not alter the ant system algorithm, but rather the information gathered by the ants to improve on the current solution.

V. A BEE COLONY OPTIMIZATION ALGORITHM TO JOB SHOP SCHEDULING

5a) *Chin Soon Chong (2006)* examine that in the intense competition of global market has resulted in challenging manufacturing environment with lower product costs, shorter product life cycles and more product variety [47]. The conflicting objectives of maintaining low inventory level to reduce costs, and quick response to customer demand to remain competitive calls for an effective scheduling algorithm for production shop floor. In this respect, there have been extensive studies of scheduling algorithms and heuristics in both static and dynamic job shops for decades by researchers and practitioners (Gere 1966, Blackstone et. al. 1982, Rajendran and Holthaus 1999, Jain and Meeran 1999) [48]. A scheduling problem can be characterized by a set of jobs, each with one or more operations. The operations of a job are to be performed in a specified sequence on specific machines. The objective of scheduling is to determine the job schedules that minimize (or maximize) a measure (or multiple measures) of performance (Rajendran and Holthaus 1999) [49]. Due to factorial explosion of possible solutions, job shop scheduling problems are considered to be a member of a large class of intractable numerical problems known as NP-hard (Jain and Meeran 1999) [50]. The commonly used performance measures that are related to job shop scheduling include machine utilization, cycle time, and throughput rate and inventory level. Of these measures, utilization of manufacturing resources is of vital importance to any manufacturing enterprise in the global competition of today. Improving resource utilization leads to better throughput rate and lower product cost. An alternative measure of resource utilization is makespan of a schedule, which is often studied by research community in job shop scheduling problems. Solution techniques for shop scheduling problems range from simple priority dispatching rules such as FIFO (first in first out) and SPT (shortest processing time) to more elaborate techniques such as Branch and Bound (Brucker et. al.1994), tabu search (Nowicki and Smutnicki 1996), shifting bottleneck algorithms (Balas and Vazacopoulos 1998), and ant colony (Blum and Sampels 2004) [51]. Meta-heuristics such as tabu search and shifting bottleneck procedure have been very successful. These approaches excel in solution quality as well as in computation time. Other meta-heuristics that work well when computation time is unconstrained are evolutionary computation approaches such as ant colony. This work aims to explore an evolutionary computation approach, which is based on nectar collection in honey bee colonies, to job shop scheduling problems. This research is inspired by the work done by Nakrani and Tovey (2004) [52], on using a new honey bee algorithm for dynamic allocation of internet servers. In their algorithm, servers and HTTP request queues in an Internet server colony are modeled as foraging bees and flower patches respectively. The

experimental results show that the algorithm performs reasonably well in the dynamic allocation problem. Based on similar idea of honey bee colonies and the behavior of forager bees, which is characterized by decentralized and elementary interactions, we adapt the algorithm to job shop scheduling problem.

5 b) *Chong C (2009)* examine that the honey bees' effective foraging strategy can be applied to job shop scheduling problems [53]. A feasible solution in a job shop scheduling problem is a complete schedule of operations specified in the problem. Each solution can be thought of as a path from the hive to the food source. The figure illustrates such an analogy

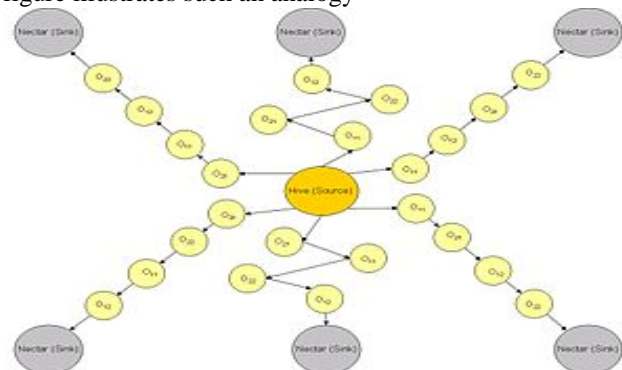


Fig.3.1. X make span.

The X make span of the solution is analogous to the profitability of the food source in terms of distance and sweetness of the nectar. Hence, the shorter the make span gives the higher the profitability of the solution path.

We can thus maintain a colony of bees, where each bee will traverse a potential solution path. Once a feasible solution is found, each bee will return to the hive to perform a waggle dance. The waggle dance will be represented by a list of "elite solutions", from which other bees can choose to follow another bee's path. Bees with a better makespan will have a higher probability of adding its path to the list of "elite solutions", promoting a convergence to an optimal solution. Using the above scheme, the natural honey bee's self organizing foraging strategy can be applied to the job shop scheduling problem.

VI. CONCLUSION

In the face of globalization and rapidly shrinking product life cycle, manufacturing companies are trying different means to improve productivity through management of machine utilization and product cycle-time. Job shop scheduling is an important task for manufacturing industry in terms of improving machine utilization and reducing cycle-time.

We are proposing here with the use of honey bee foraging model to solve the job shop problem. We will test the implementation of self organization of algorithms to test on electronics parts manufacturing scheduling problems. Further we propose a comparative study with honey bee colony and ant colony algorithms for the same.

We are proposing the use of ant colony optimization metaheuristic to the job shop scheduling problem that utilizes local search techniques “Footstepping” which aids in exploration of solution space.

A rudimentary Ant System was examined along with a Max-Min Ant System approach. The latter is a well known extension of the AS, that utilizes pheromone boundaries and the transition from iteration to global best solution for pheromone updating.

Further we are proposing three heuristic algorithms for solving job-shop scheduling problems, job-shop scheduling problems with multi-purpose machines, and open-shop scheduling problems.

All these algorithms are based on the particle swarm optimization algorithm, and generate solutions that belong to the class of parameterized active schedules via their specific decoding procedures. A comparative study of proposed algorithm with the existing algorithms is also being proposed. Scheduling involves the allocation of resources to activities over time. Most organizations must schedule resources on a recurrent basis and this creates considerable demand for good scheduling techniques. Since the mid 1950s, researchers have been advocating the use of formal optimization algorithms to find solutions to scheduling problems. Unfortunately, after fifty years of work, these methods can still only guarantee optimality for a very limited set of problems.

There are two main reasons for the limited success of traditional optimization algorithms to scheduling. First, most scheduling problems belong to the class of NP-hard problems. This class of problems is distinguished by rapid growth in the number of potential solutions with modest growth in the number of resources to be scheduled. The growth is so quick that even the fastest computer could not search through every potential solution to large-scale problems in a reasonable amount of time. Second, for many practical scheduling problems, it is difficult to capture the problem formulation in a closed-form mathematical expression. This difficulty is perhaps the reason why most scheduling is still done in an ad hoc manner. Because of these two difficulties, many researchers have thus turned their attention to population-based stochastic search methods, e.g., genetic algorithms, ant colony optimizations, and particle swarm optimizations, which are able to find near-optimal solutions within an acceptable computation time. In various optimization techniques such as ant colony optimization, bee optimization and particle swarm optimization, we will optimize job shop problem.

REFERENCES

- [1] Graham, R. (1966), "Bounds for certain multiprocessing anomalies" (PDF), *Bell System Technical Journal* 1563–1581.
- [2] Sawaragi, Y.; Nakayama, H. and Tanino, T. (1985), "Theory of Multiobjective Optimization (vol. 176 of Mathematics in Science and Engineering). Orlando, FL: Academic Press.
- [3] Bäck, T., Fogel, D., Michalewicz, Z. (1997), "Handbook of Evolutionary Computation", Oxford Univ. Press.
- [4] Langton, C. G. (1992), "Artificial Life", Addison-Wesley, section 1
- [5] Beni, G., Wang, J. (1989), "Swarm Intelligence in Cellular Robotic Systems", Proceed. NATO Advanced Workshop on Robots and Biological Systems, Tuscany, Italy, 26–30.
- [6] Waibel M, Floreano D and Keller L (2011), "A quantitative test of Hamilton's rule for the evolution of altruism", *PLoS Biology*, 9(5): e1000615. doi:10.1371/journal.pbio.1000615
- [7] Marco Dorigo and Thomas Stützle (2004), "Ant Colony Optimization", MIT Press, ISBN 0-262-04219-3
- [8] Karaboga, Dervis (2010), "Artificial bee colony algorithm", *Scholarpedia*, 5(3): 6915.
- [9] Kaveh, A.; Talatahari, S. (2010). "A Novel Heuristic Optimization Method: Charged System Search". *Acta Mechanica* 213 (3-4): 267–289. doi:10.1007/s00707-009-0270-4.
- [10] Yang X.-S. And Deb S. (December 2009). "Cuckoo search via Lévy flights". (NaBIC 2009). IEEE Publications. pp. 210–214. arXiv:1003.1594v1.
- [11] Yang X. S., (2008), "Nature-Inspired Metaheuristic Algorithms", Frome: Luniver Press. ISBN 1905986106.
- [12] Shah-Hosseini, Hamed (2009). "The intelligent water drops algorithm: a nature-inspired swarm-based optimization algorithm". In *Journal of Bio-Inspired Computation* (1/2): 71–79.
- [13] Parsopoulos, K. E.; Vrahatis, M. N. (2002). "Recent Approaches to Global Optimization Problems Through Particle Swarm Optimization". *Natural Computing* 1 (2-3): 235–306. doi:10.1023/A:1016568309421.
- [14] P.-P. Grassé (1959), "La théorie de la Stigmergie : Essai d'interprétation du comportement des termites constructeurs", *Insectes Sociaux*, numéro 6, p. 41–80.
- [15] J.L. Deneubourg, J.M. Pasteels et J.C. Verhaeghe (1983), "Probabilistic Behaviour in Ants : a Strategy of Errors?", *Journal of Theoretical Biology*, numéro 105.
- [16] F. Moysen, B. Manderick (1988), "The collective behaviour of Ants : Example of Self-Organization in Massive Parallelism", *Symposium on Parallel Models of Intelligence*, Stanford, Californie.
- [17] S. Goss, S. Aron, J.-L. Deneubourg et J.-M. Pasteels (1989), "Self-organized shortcuts in the Argentine ant", *Naturwissenschaften*, volume 76, pages 579-581.
- [18] M. Ebling, M. Di Loreto, M. Presley, F. Wieland, et D. Jefferson (1989), "An Ant Foraging Model Implemented on the Time Warp Operating System", *Proceedings of the SCS Multiconference on Distributed Simulation*.
- [19] M. Dorigo (1992), "Optimization, Learning and Natural Algorithms", PhD thesis, Politecnico di Milano, Italie.
- [20] M. Dorigo, V. Maniezzo, et A. Colomi (1996), "Ant system: optimization by a colony of cooperating agents", *IEEE Transactions on Systems, Man, and Cybernetics--Part B*, volume 26, numéro 1, pages 29-41.
- [21] T. Stützle et H.H. Hoos (2000), "MAX MIN Ant System", *Future Generation Computer Systems*, volume 16, pages 889-914.
- [22] M. Dorigo et L.M. Gambardella (1997), "Ant Colony System : A Cooperative Learning Approach to the Traveling Salesman Problem", *IEEE* volume 1, numéro 1, pages 53-66.
- [23] R. Schoonderwoerd, O. Holland, J. Bruen et L. Rothkrantz (1997), "Ant-based load balancing in telecommunication networks", *Adaptive Behaviour*, volume 5, numéro 2, pages 169-207.
- [24] M. Dorigo (1998), "From Ant Colonies to Artificial Ants : First International Workshop on Ant Colony Optimization", *ANTS 98*, Bruxelles, Belgique, octobre 1998.
- [25] T. Stützle (1998), *Parallelization Strategies for Ant Colony Optimization*, *Proceedings of PPSN-V, Fifth International Conference on Parallel Problem Solving from Nature*, Springer-Verlag, volume 1498, pages 722-731.
- [26] É. Bonabeau, M. Dorigo et G. Theraulaz (1999), "Swarm intelligence", Oxford University Press.
- [27] M. Dorigo, G. Di Caro et T. Stützle (2000), *Special issue on "Ant Algorithms"*, *Future Generation Computer Systems*, volume 16, numéro 8.

- [28] W.J. Gutjahr (2000), "A graph-based Ant System and its convergence", *Future Generation Computer Systems*, volume 16, pages 873-888.
- [29] S. Iredi, D. Merkle et M. Middendorf (2001), "Bi-Criterion Optimization with Multi Colony Ant Algorithms", *Evolutionary Multi-Criterion Optimization*, First International Conference (EMO'01), Zurich, Springer Verlag, pages 359-372, 2001.
- [30] L. Bianchi, L.M. Gambardella et M.Dorigo (2002), "An ant colony optimization approach to the probabilistic traveling salesman problem", *PPSN-VII, Seventh International Conference on Parallel Problem Solving from Nature*, Lecture Notes in Computer Science, Springer Verlag, Berlin, Allemagne.
- [31] M. Zlochin, M. Birattari, N. Meuleau, et M. Dorigo (2004), "Model-based search for combinatorial optimization: A critical survey", *Annals of Operations Research*, vol. 131, pp. 373-395.
- [32] D. Karaboga (2005), "An Idea Based On Honey Bee Swarm for Numerical Optimization", Technical Report-TR06, Erciyes University, Engineering Faculty, Computer Engineering Department.
- [33] Ali Hadidi, Sina Kazemzadeh Azad, Saeid Kazemzadeh Azad (2010), "Structural optimization using artificial bee colony algorithm", 2nd International Conference on Engineering Optimization, 2010, September 6 – 9, Lisbon, Portugal.
- [34] Pisut Pongchairerks (2009), "Particle swarm optimization algorithm applied to scheduling problems", *ScienceAsia* 35 (2009): 89–94
- [35] M.Dorigo and G.Di Caro (1999), "The ant colony optimization meta-heuristic", in *New Ideas in Optimization*, McGraw-Hill, pp. 11-32, 1999.
- [36] M.R.Garey, and D.S (1979), "Computers and Intractability, A Guide to the Theory of NP-Completeness", W.H Freeman and Company.
- [37] S. Goss, S.Aron, J. L. Deneubourg, and J. M.Pasteels (1990), "Self-organized shortcuts in the Argentine" *Ant. Naturwissenschaften*, 76:579-581.
- [38] T. Stutzle (1998), "An ant approach for the flow shop problem", In *Fifth European Congress on Intelligent Techniques Soft Computing (EUFIT'98)*, vol.3, Verlag Mainz, Aachen, pp. 1560-1564.
- [39] D. Merkle and M. Middendorf and H. Schmeck (2002), "Ant Colony Optimization for Resource Constrained Project Scheduling", *IEEE Transactions on Evolutionary Computation*, 6(4):333-346.
- [40] D. Merkle and M. Middendorf (2003), "Ant Colony Optimization with Global Pheromone Evaluation for Scheduling a Single Machine", *Journal of Applied Intelligence*, Kluwer Academic Publishers, pp, 105-111.
- [41] L. M, Gambardella, E. Taillard and G. Agazzi (1999), "MACS-VRPTW: A Multiple Ant Colony System for Vehicle Routing Problems with Time Window", *New Ideas in Optimization*, pp.63-76, McGraw-Hill, London,.
- [42] Colomi A., M. Dorigo, V. Maniezzo and M. Trubian (1994), "Ant system for Job-shop Scheduling", *JORBEL -Belgian Journal of Operations Research, Statistics and Computer Science*, 34(1):39-53.
- [43] S.van der Zwaan, C. Marques (1999), "Ant Colony Optimization for Job Shop Scheduling", In *proceedings of Third workshop on Genetic Algorithms and Artificial Life*.
- [44] C. Blum and M. Sampels (2002), "An ant Colony Optimization Algorithm for FOP Shop Scheduling: A case study on different pheromone representations", *Proceedings of the 2002 Congress on Evolutionary Computation*, CEC'02, pp.1558-1563.
- [45] Sjoerd van der Zwaan (2004), "Ant Colony Optimisation for Job Shop Scheduling", *proceedings on scheduling'04*.
- [46] Apinanthana Udomsakdigool (2011), "Ant colony algorithm for multi-criteria job shop scheduling to minimize makespan, mean flow time and mean tardiness" ISSN 1750-9653, England, UK, *International Journal of Management Science and Engineering Management*, 6(2): 117-123, <http://www.ijmsem.org>
- [47] Chin Soon Chong (2006), "Bee colony algorithm for job shop scheduling *Proceedings of the 2006 Winter Simulation Conference*.
- [48] Gere, W. S.(1966), Jr., "Heuristics in jobshop scheduling," *Management Science*, Vol. 13, No. 1 (1966), pp. 167-175.
- [49] Rajendran, C. and Holthaus (1999), O., "A comparative study of dispatching rules in dynamic flowshops and jobshops," *Journal of Operational Research* Vol. 116, pp. 156-170.
- [50] Jain. A. S. and Meeran. S. (1999), "Deterministic job shop scheduling:past, present and future," *European Journal of Operational Research*, Vol. 113, No. 2 (1999), pp.390-434
- [51] Blum, C. and M. Sampels. (2004), "Mathematical Modelling and Algorithms", 2004, 3(3):p. 285 - 308.
- [52] Nakrani, S. and C. Tovey, "On honey bees and dynamic allocation in an internet server colony. *Adaptive Behavior*", 2004. 12(3-4): p. 223-240.
- [53] Chong C (2009) "Bee optimizations" *proceedings in simulation conference* page(s): 1954-1961, Monterey, CA ISBN:1-4244-0500-9
- [54] Pisut Pongchairerks (2009), "Particle swarm job scheduling" *ScienceAsia* 35 (2009): pages 89–94.
- [55] J. Kennedy (2006). "Swarm Intelligence In *Handbook of Nature-Inspired and Innovative Computing: Integrating Classical Models with Emerging Technologies*", pages 187-219, Springer US, Secaucus, NJ, 2006.