International Journal of Informatics and Applied Mathematics
e-ISSN:2667-6990 Vol. 1, No. 1, 1-9

Hybrid Metaheuristic for Optimization Job-Shop Scheduling Problem

Benhamza Karima and Zedadra Ouarda

LabSTIC, Computer Science Department, 8 Mai 1945 Guelma University, POB 401 Guelma, Algeria
(benhamza.karima, zedadra.ouarda)@univ-guelma.dz

Abstract. Real Job-shop scheduling problem is one of the most difficult NP-Combinatorial issues. Exact resolution methods cannot handle large size cases. It is therefore necessary to use heuristic methods to solve them within a reasonable time. There are a large number of metaheuristic, which have the advantage of covering only part of the search space to find an acceptable solution. In this work, Genetic Algorithm and Simulated Annealing are used to solve Job-shop scheduling problem. The objective is to find the sequence of operations on the machines that will minimize the total time required to complete the set of jobs, also known as the "Makespan". Compared to traditional genetic algorithm, hybrid approach yields significant improvement in solution quality.

Keywords: Scheduling Problem, Hybrid Metaheuristic, Optimization

1 Introduction

Job Shop Problem (JSP) is an important industrial activity in which jobs are assigned to resources at specific times. Each job consists of a sequence of tasks, which must be performed in a given order. Generating a good job scheduling solution is particularly difficult in cases with large search space and with precedence constraints between operations.

Lenstra and Rinnooy Kan [9] demonstrated that JSP is a difficult problem classified as NP-hard, so it cannot be exactly solved within a reasonable computation time. To solve this problem, numerous approaches have been developed, incorporating exact and/or approximate methodologies. Exact methods such as branch and bound and dynamic programming take significant computing time to get the optimum solution.

Therefore, most presented JSP techniques have focused on the approximation method, including Artificial Intelligence (AI) approaches. In recent years, metaheuristic algorithms have proven their flexibilities and efficiency for many complex optimization problems. Several studies including Simulated Annealing, Tabu Search, Genetic Algorithm, Ant Colony Optimization and Particle Swarm Optimization have been devoted to JSP scheduling problem [5].

Metaheuristic algorithms usually start their optimization process by generating a population of random candidate solutions called individuals, then recombine these initial individuals over a predefined number of iterations. The main distinction between different metaheuristic algorithms lies in the mechanism of recombination of individuals, which is often inspired by nature, biology or physics. Genetic Algorithms (GA)[7] and Simulated Annealing (SA) [8] have become the leading methodologies for search and optimization problems in high-dimensional spaces. Genetic Algorithm (GA) is the best known and most widely used evolutionary computation technique. There are essential steps in basic GA that can be implemented differently according to the problem: population initialization, fitness function computation, population selection, crossover and mutation operators applied to a population of elements. These elements, called chromosomes or individuals, represent possible solutions to the problem [4].

Although classical GAs are more resistant to premature convergence than other search methods, GAs are not immune.

Hybrid methods offer advantages by avoiding this premature convergence and allowing acceleration towards the AG algorithm. This research aims to propose an efficient hybrid AG scheduling method to address this concern.

Moreover, Simulated Annealing (SA) has its origin in the fields of materials science and physics. It is a search process most commonly used in optimization problems. The objective function, similar to the energy of a material, is then minimized, by introducing fictive temperature, which is a simple and controllable parameter of the algorithm. SA algorithm starts by generating an initial solution and initializing the temperature parameter. Then, at each iteration, a solution is randomly selected in the neighborhood of the current solution. SA algorithm has the ability to find the optimal local result [10].

In this study, hybrid metaheuristic is used to optimize job-shop scheduling prob-

lems. Two metaheuristic are considered: Genetic Algorithms (GAs) and Simulated Annealing algorithm (SA). In JSP, the decision concerns how to sequence operations on the machines, which will minimize the total time required to complete the set of jobs, also known as the "Makespan".
The remainder of the paper is organized as follows: Section 2 presents the problem formulation. SA-GA-JSP Model is presented in Section 3, while experimental results are described in Section 4. Finally, Section 4 summarizes the results of this work and draws conclusions.

2 Problem Formulation

In JSP, a set of jobs is performed by a set of machines. Each job is formed by a sequence of consecutive operations, and each operation requires exactly one machine. Each machine is continuously available and can process one operation at a time without interruption. The decision concerns how to sequence the operations on the machines, which can minimize the makespan Cmax, i.e., the maximum time to complete the final operation in the schedule operations. JSP can be defined as follows [6]:
There is a set of n jobs: $J = J_1, J_2, J_3, \ldots, J_n$ and a set of m machines: $M = M_1, M_2, M_3, \ldots, M_m$. Each job J_i consists of a sequence of operations $O_{i1}, O_{i2}, O_{i3}, \ldots, O_{ini}$ where n_i is the number of operations that J_i comprises. Each operation $O_{ij}(i = 1, 2, , n; j = 1, 2, , n_i)$ must be processed by one machine on a given set of machines $M_{ij} \subseteq$ M. The respected hypotheses are summarized as follows:

− Jobs are independent, each machine can handle only one job at a time.
− All jobs and machines are available at time zero simultaneously, operations of one job cannot be processed simultaneously.
− After a job is processed on a machine, it is immediately transported to the next machine on its process, and the transmission time is assumed to be negligible.

3 SA-GA-JSP Model

In this work, sequential hybrid method based on GA and SA is exposed. Thus, SA is applied after the GAs to exploit the result of the previous exploration of the GA search space. The process adopted is to wait for the stabilization of the fitness in the GAs population. The best individuals found are the starting solutions of the SA algorithm. Figure 1 presents the general hybrid SA-GA-JSP model. The following section describes the main steps of SA-GA-JSP.

3.1 GA-JSP

In the GA, a suitable encoding (or representation) for the problem must first be devised. A fitness function is also required, which assigns a merit figure to

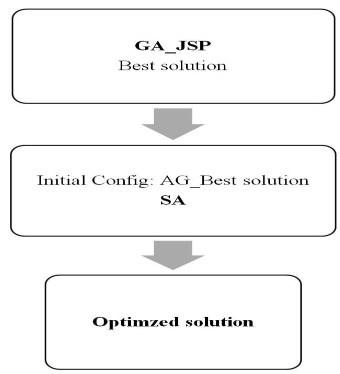

Fig. 1: SA-GA-JSP model

each encoded solution. The first population of solutions is generated randomly. During execution, parents must be selected for reproduction, and recombined to generate offspring (population of new solutions). The process is repeated until a good solution is provided. The stop criterion is generally based on the number of generations. These stages are detailed as follows:

a. Encoding representation: First task in GA-JSP is the solution representation as chromosome. An operation-based representation is used. In this formulation, each job number occurs m times in the permutation, i.e. as often as there are operations associated with this job.
For example, with 4 jobs x 3 machines problem (each job consists of three operations), a chromosome is given as [1 2 1 2 3 1 1 2 2 3 3 3] related to jobs representation [1 1 2 2 1 3 4 3 4 2 3 4] and associated operations-based representation[1 2 1 2 3 1 1 2 2 3 3 3].

b. Fitness function: GA assesses the solutions based on the fitness function. The better fitness an individual has, the higher the chance of being chosen for the next generation. In general, the fitness is relative to the objective function. In this paper, the minimal makespan represents a good objective measure function U(k). Then, the fitness function for solution k is as follows:

$$F(k) = \frac{1}{U(k)} \tag{1}$$

c. GA operation phases: In GA operation phase, an initial population is yielded randomly. Therefore, GA operates to produce new population by using basic genetic operations, i.e. selection, crossover and mutation.

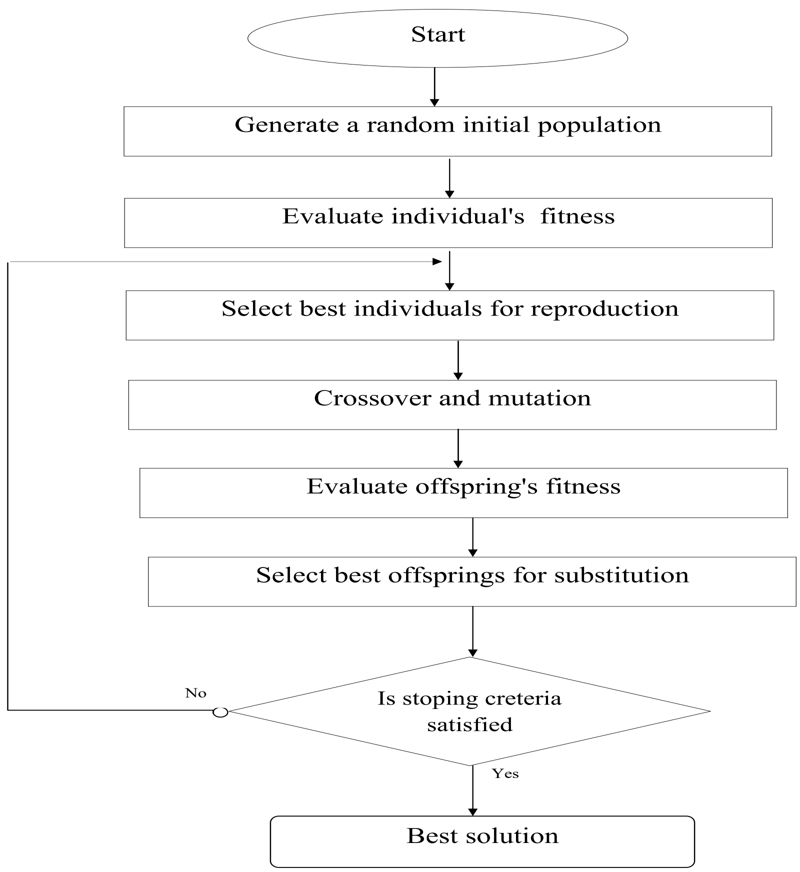

Fig. 2: GA-JSP Flow Chart

1. Selection operation: In GA, parent chromosomes are selected with a probability related to their fitness. The standard procedure is a fitness-proportional selection (or roulette wheel selection) of individuals to produce the offspring. A roulette wheel model is established to represent the survival probabilities for all the individuals in the population [7].
   The probability *P(i)* of selecting the *i-th* individual is given by:

$$P\left(i\right)=\frac{F\left(i\right)}{\sum_{k=1}^{ps}F\left(k\right)} \tag{2}$$

This means that each individual is interpreted as a segment of the wheel so that the size of this segment is proportional to the fitness. In general, by means of two chosen parents, one or two offspring can be generated.

2. Crossover operation: The crossover is the main operator applied in genetic algorithms. By crossover, the genetic structure of two selected individuals

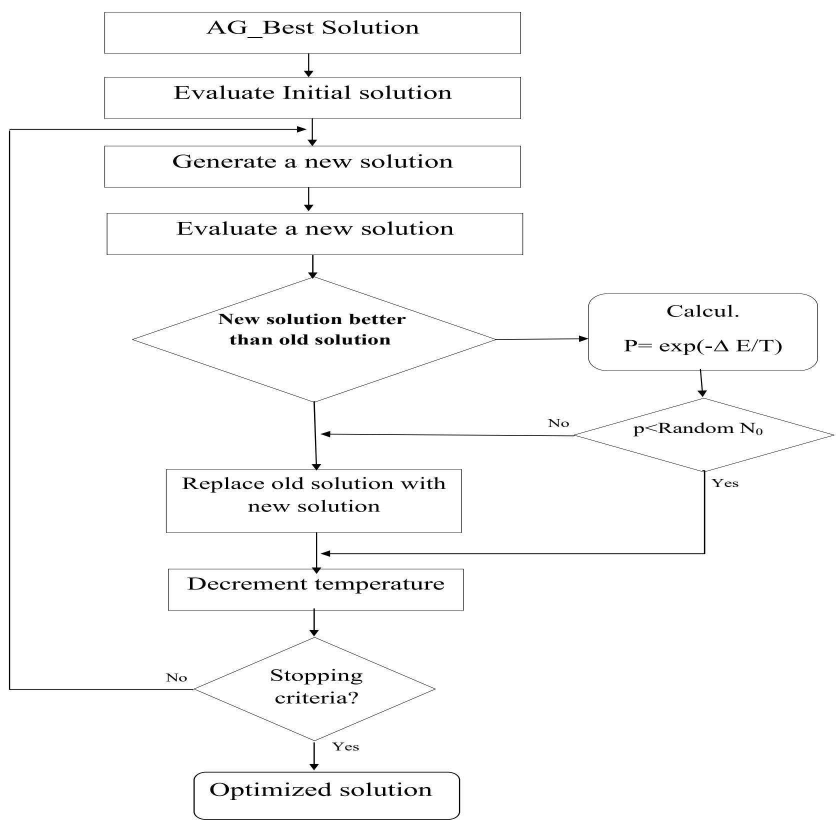

Fig. 3: SA-GA-JSP Flow Chart

of the current population is mixed. In this work, the idea for scheduling problems resulting in precedence preservative crossover operator (PPX) is integrated [3]. The principle is summarized as follows: A vector of equal length defines the order in which the operations are successively drawn from parent 1 and parent 2. First, offspring is initialized empty. Then, an operation is selected; it is deleted in both parents and appended to the offspring. The leftmost entry of the binary vector is also deleted. This procedure is repeated until the parent lists are empty and the offspring list contains all operations.

3. Mutation operation: Mutation is used only to produce small perturbations on chromosomes in order to maintain the diversity of population. For the operation sequence, two positions were selected and exchanged with each other. The mutation rate was applied with a probability of 0.1.

3.2 SA-GA-JSP

The best individual found in Step 1 was used as first solution. Initial temperature was set, then a loop was started until the threshold was reached. A neighbor was selected by making a small change of the current solution. Using the Metropolis rules, the transition to the neighbor was accepted or not. Finally, the temperature was slowly decreased until the thermodynamics equilibrium was reached. The solution converges as shown in Figure 3.

4 Experimental Results

A number of benchmark problems for JSP have been formulated in literature to compare the performance of new scheduling algorithms [2]. For the purpose

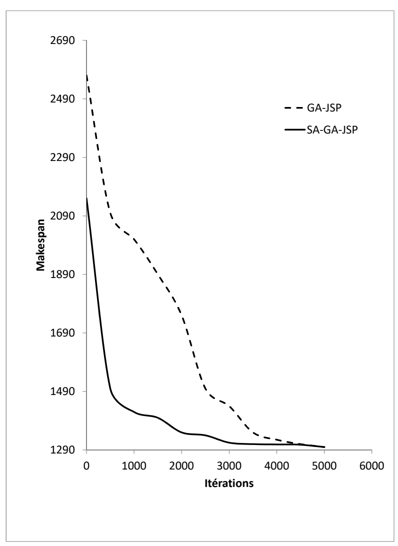

Fig. 4: The makespans evolutions for JSP example (25 jobs and 08 machines)

of this study, 30 instances were selected from OR-Library [1] as benchmarks to test SA-GA-JSP model. 180 chromosomes were used as initial population and the crossover rate was set at 0.6. Temperature T was initialized to 3000 to measure the quality of proposed system. Implementation was successfully completed to examine the convergence and effectiveness of the hybrid method. Figure 4 shows that the hybrid method reaches the optimal solution faster than GA-JSP method and produce less makespan value. Empirical results show that hybrid SA-GA-JSP method is effective enough to obtain a better solution than the genetic algorithm applied individually. SA-GA-JSP model converges towards the optimal solution for most instances of the benchmark problems.

5 Conclusion

Genetic Algorithms (GA) and Simulated Annealing (SA) represent powerful combinatorial optimization methods. In this work, an effective combination of GA and SA was developed to solve the job shop scheduling problem (JSP). This model has the advantage of avoiding premature convergence and allowing acceleration to AG algorithm. GA is suitable for traversing large search spaces in compared to other optimization methods. SA, alternatively attempts to avoid being trapped in a local optimum.

This hybrid metahaeuristic gives an excellent trade-off between solution quality, computing time and flexibility to take into account specific constraints in real JSP scheduling situations. Indeed, GA is effective for good exploration and SA is the best heuristic for task exploitation.

References

1. J. E. BEASLEY. Or-library. *http://people.brunel.ac.uk/mastjjb/jeb/info.html.*
2. Dennis Behnke and Martin Josef Geiger. Test instances for the flexible job shop scheduling problem with work centers, 2012.
3. Christian Bierwirth, Dirk C Mattfeld, and Herbert Kopfer. On permutation representations for scheduling problems. In *International Conference on Parallel Problem Solving from Nature*, pages 310–318. Springer, 1996.
4. Ilhem BoussaïD, Julien Lepagnot, and Patrick Siarry. A survey on optimization metaheuristics. *Information sciences*, 237:82–117, 2013.
5. Banu Çaliş and Serol Bulkan. A research survey: review of ai solution strategies of job shop scheduling problem. *Journal of Intelligent Manufacturing*, 26(5):961–973, 2015.
6. Liang Gao, CY Peng, C Zhou, and PG Li. Solving flexible job shop scheduling problem using general particle swarm optimization. In *Proceedings of the 36th CIE conference on computers & industrial engineering*, pages 3018–3027, 2006.
7. David E Goldberg. Genetic algorithms in search. *Optimization, and Machine-Learning*, 1989.
8. Scott Kirkpatrick. Optimization by simulated annealing: Quantitative studies. *Journal of statistical physics*, 34(5-6):975–986, 1984.

9. Jan K Lenstra and AHG Rinnooy Kan. Computational complexity of discrete optimization problems. In *Annals of Discrete Mathematics*, volume 4, pages 121–140. Elsevier, 1979.
10. Peter JM Van Laarhoven and Emile HL Aarts. Simulated annealing. In *Simulated annealing: Theory and applications*, pages 7–15. Springer, 1987.