Genetic Algorithms applied to the Job-shop Scheduling Problem

D. Duvivier* and Ph. Preux† and E-G. Talbi‡

Keywords: Evolutionary algorithms, combinatorial optimization, job-shop scheduling problem, hybrid search method

1 Introduction

Our work aims at assessing what Evolutionary Algorithms (EAs) are good for in the context of combinatorial optimization with regards to other search methods (both exact and non exact). We think that stochastic local search algorithms can bring something to this field. Exact methods are limited in the size of the problems they can handle because of their time/space requirements. We are perfectly aware of some limitations in the theoretical results to assess solutions found with local search. However, we think that hybrid algorithms might provide some useful methods, able to handle large instances.

To achieve our goal, we concentrate ourselves on the resolution of the job-shop-scheduling problem (JSP), the traveling salesman problem (TSP), and the quadratic assignment problem (QAP), and we investigate how the evolutionary paradigm may be used (see [BPT96] for the QAP).

In this paper, we will restrict ourselves to the JSP. There are several variants of the JSP (see [P94]). We consider here the simple JSP where J jobs, each composed of M operations are to be realized on M machines. Each job has an operation which has to be performed on each machine. Then, the aim of the search is to find a planning of occupation of the machines that minimizes the total time of realization of the J jobs (makespan). In section 2, we present the encoding of solutions. In section 3, we detail the algorithms we have used so far. Section 4 gives our current results. Finally, we will discuss these results and give our perspectives.

2 Encoding schemes

In this section, we present the encoding of solutions that we have used in our algorithms.

We have tested several encodings. Better results have been obtained with direct encoding (the data structure actually represents a planning). We use a two-dimensional direct encoding: for each machine, we have the list of operations that are performed with their time slot. This encoding makes it easier to detect available time slots. Using this encoding, we can detect the critical operations of a planning. A critical operation is one that, if started earlier, would reduce the makespan.

*Laboratoire d'Informatique du Littoral, BP 719, 62228 Calais Cedex, France, duvivier@lil.univ-littoral.fr

†Laboratoire d'Informatique du Littoral, preux@lil.univ-littoral.fr

‡Laboratoire d'Informatique Fondamentale de Lille, URA CNRS 369, Cité scientifique, 59655 Villeneuve d'Ascq Cedex, France, talbi@lifl.fr

Hence, critical operations are good target for local optimization of a planning.

The major drawback of such an encoding is the complexity of operators that have to be designed to maintain validity of solutions.

3 Algorithms

We have mainly worked with three classes of primitive algorithms: hill-climber, tabu search and evolutionary algorithms. We have also investigated hybrid algorithms tinkered with these three algorithms as building-blocks. Let us now turn to a brief presentation of these algorithms.

3.1 Multi-start Hill-climber

The hill-climber is considered as a very simple heuristic that does not require heavy computational resources. However, it may easily be trapped in a local optimum. We also use a variant of the basic hill-climber, namely a multi-start hill-climber (MSHC). Starting at a random point, the hill-climber tries to move uphill. When, after a certain amount of attempts, it always fails to climb, the current best point is recorded and the climbing is restarted from another point in the space, chosen at random.

The delicate part in using a hill-climber lies in the choice of the local move operator. Using our direct encoding, the basic move is an exchange of two operations on a randomly chosen machine.

3.2 TABU search

The TABU search is a hill-climber which is able to escape from local optima [Glo89a, Glo89b]. The basic move swaps two critical operations of two different jobs. The size of the TABU-list varies dynamically during the run according to [HW95].

No intensification/diversification mechanisms have been implemented. In fact, as the size of TABU-list varies dynamically, a short list favors intensification, while a long list improves the diversification mechanism.

3.3 Evolutionary algorithm

Evolutionary Algorithms (EAs) are stochastic search methods inspired by the biological evolution of living beings.

The operators, the objective function, the basic scheme of the algorithm have been tailored for the resolution of the JSP.

We use a recombination operator largely inspired by the GA/GT crossover introduced in [NY92]. This operator relies on Giffler and Thompson's algorithm [GT69].

The mutation performs a swap of two operations on a randomly chosen machine, and re-schedule operations as soon as possible in the planning. This mutation operator always produces (valid) semi-active plannings.

For the simple JSP, aiming at optimizing the makespan, the usual idea is to design the objective function as returning the makespan of a planning. However, close observation of the behavior of the EA in this case led us to work further the objective function. The number of critical operations may be used as a clue of the goodness of an individual: the less critical operations it has, the better it is. Hence, we use an objective function which combines the makespan and the number of critical operations. Using this new objective function, the heuristic can be better guided while seeking the optimum, and the results have been improved substantially.

Our algorithm has been tested with the following objective function:

$$\mathcal{F} = K \times C_{\max} + N_{\text{cop}}$$

where N_{cop} is the number of critical operations in the current planning. K is an upper bound for N_{cop} over all the plannings of the current instance of JSP, and $C_{\max}$ the makespan of current planning. Using this function, the makespan remains the most important criteria.

We have observed that, because we are typically using a high rate of mutation and a sophisticated recombination operator, the benefit of the application of the crossover is often destroyed by the mutation applied afterwards. Hence, we use a different scheme of application of the operators (see [DPT96] for further details) where crossover and mutation are exclusively applied instead of using a "serial" scheme.

3.4 Hybrid algorithms

Various hybridization schemes are possible (see [PT95]). We have currently experimented one scheme on the JSP, namely the synchronous hybridization which uses a search algorithm as a mutation operator.

We have used a simple hill-climber (HC) as the search algorithm in the hybrid denoted EA+HC below.

4 Results

To be able to compare our results with others, we used the ORLIB benchmarks [BEA]. In our benchmark suite, we use the Muth and Thompson's instances, the 8 Carlier's instances, as well as 6 Lawrence's instances, and 5 Taillard's instances of big size. The size of the instances ranges from 6×6 to 100×20.

Generally speaking, EA+HC always performs better than EA alone. The hybrid has found the optimal planning for some instances (MT6×6, MT10×10, car1, car2, car3, car4, car6, car7, la01, la02, la03, la04, and la05) while EA only found the optima of the MT6×6, la01, and la05 which are found by all the algorithms.

For their parts, EA+HC, MSHC, and TABU all achieve the same kind of results, lying within a few percents from the optimal planning. When considering the distribution of points found in a series of experiments with a given algorithm, it clearly appears that the standard deviation of makespan of optimal plannings is very different from one algorithm to another, and from one series of instances to another. MSHC, EA and EA+HC have a moderate stardard deviation, the TABU heuristic quite a high one.

Concerning the mean time of execution of the various algorithms, the very short time of execution of MSHC is noteworthy. EAs (both pure and hybrid) have moderate times of execution. For the TABU, no such conclusion might be easily drawn.

It is noteworthy that for big instances, the search space is huge and only a small fraction of it is visited by the algorithm.

We would like to put the emphasis on the performance achieved by the (multi-start) hill-climber. The best found plannings are either the optimal planning or very close to it. This prompts us with two remarks. First, the design of the hill-climber is very simple while the design of the EA is not so easy (notably because of the recombination operator). Second, the basic movement of MSHC is similar to the mutation that was used in the EA. Hence, this questions the efficiency of the GA/GT operator.

5 Discussion and perspectives

In this paper, we have compared different stochastic meta-heuristics on a test-suite composed of JSP instances ranging from small size to big size. The hybrid algorithm achieves better results than the pure EA. The TABU search as well as a simple multi-start hill-climber often obtains very good solutions.

Given the results obtained with the pure

EA and the hill-climber, we are prompted to question the efficiency of the recombination operator as long as this is the real difference between these two algorithms.

As it was observed in our experiments, different classes of problem hardness seem to exist. Very pragmatically, we would like to say that an instance is easy if it can be solved optimally with a simple algorithm. This way, we optimize the trade-off between the time of design of the algorithm and the quality of solutions it is able to find. In order to define a workable notion of hardness, we are currently working on the exhibition of several criteria to evaluate the hardness of an instance of the JSP for a given heuristic. Each search method has its particular way to move in the search space. This renders a search method more or less suited to find an optimum in a given search space. With regards to this, S. Kauffman has experimented the ability of recombination of hill-climbers and random search to find the optima on tailored search spaces [Kau89]. We hope to explain the performance of our algorithms with the use of measurable features of the search space associated to an instance. This would give us a way to predict, for a given instance, which algorithm should perform the best.

In the future, we will continue our study of hybridization. We will shortly have a hybrid EA+TABU. Further, we wish to experiment with the hybridization scheme discussed in [PT95] relying on the cooperation of search methods acting concurrently.

References

[BEA] J. E. Beasley. Orlib. available via anonymous ftp on `mscmga.ms.ic.ac.uk:/pub`.

[BPT96] Vincent Bachelet, Philippe Preux and El-Ghazali Talbi. Parallel Hybrid Heuristics: Application to the Quadratic Assignment Problem *Proc. of the Parallel Optimization Colloquium*, Versailles, France, March 1996.

[DPT96] D. Duvivier, Ph. Preux, and E-G. Talbi. Climbing up np-hard hills. Technical Report LIL-96-1, Laboratoire d'Informatique du Littoral, Calais, February 1996.

[Glo89a] F. Glover. Tabu search — part I. *ORSA Journal of Computing*, **1**(3):190–206, 1989.

[Glo89b] F. Glover. Tabu search — part II. *ORSA Journal of Computing*, **2**(1):4–31, 1989.

[GT69] B. Giffler and G. L. Thompson. Algorithms for solving production scheduling problems. *Operations Research*, 8:487–503, 1969.

[HW95] Alain Hertz and Marino Widmer. *La méthode TABOU appliquée aux problèmes d'ordonnancement.* Automatique, Productique, Informatique Industrielle, **29**(4–5), pages 353–378, 1995.

[Kau89] Stuart Kauffman. Adaptation on rugged fitness landscapes. In D.L. Stein, editor, *Lectures in the Science of Complexity*, pages 527–618. Addison-Wesley, 1989.

[NY92] Ryohei Nakano and Takeshi Yamada. A genetic algorithm applicable to large-scale job-shop problems. In R. Männer and B. Manderick, editors. *Proc. of the Second Conf. on Parallel Problem Solving in Nature.* Elsevier Science Publishers, Amsterdam, 1992, pages 281–290.

[P94] Bernard Penz. *Constructions agrégatives d'ordonnancements pour des job-shops statiques, dynamiques et réactifs.* PhD dissertation, Université de Grenoble I, 1994.

[PT95] Ph. Preux and E-G. Talbi. Assessing the evolutionary algorithm paradigm to solve hard problems. In *Constraint Processing, workshop on really hard problem solving*, September 1995. An extended version is available as report LIL-95-4 from the Laboratoire d'Informatique du Littoral.