Heuristic Optimization: implementation exercise 2

Samuel Buchet: 000447808

May 2017

1 Introduction

Two stochastic local search algorithms are implemented for this second implementation exercise. The first one is a simulated annealing, which belong to the class of simple stochastic algorithms. The second one is a Greedy Randomised Adaptive Search Procedures (GRASP), which is a hybrid stochastic algorithm.

2 Metaheuristics used

2.1 Simulated annealing

2.1.1 Principle

The first metaheuristic implemented is the simulated annealing algorithm. This algorithm is made on the basis of a local search. The neighbourhood used is the transpose one because it is the smallest one. Thanks to this small neighbourhood, it is faster to explore most of the neighbours. The acceptance criterion of this method is based on the metropolis condition. If the new solution found improve the best found so far, it replaces it. Otherwise, there is a probability to accept a new solution, which is equal to $e^{\frac{sol_{best} - sol_{actual}}{T}}$. sol_{best} is the best solution found by the algorithm so far and $sol_{current}$ is the current solution. T is a variable called temperature and it depends on three paramaters : the initial temperature (T_0), the length of a plateau (N) and the cooling rate (α). At the begining of the algorithm, T is equal to T_0 and after each N iterations, T is updated as following : $T \leftarrow T * \alpha$.

2.1.2 Parameter tuning

As the acceptance criterion depends on the probability $e^{\frac{sol_{best} - sol_{actual}}{T}}$, a new solution has more chance to be accepted if it is close to the best found so far and if the temperature is not too low.

It has been found that the initial temperature shouls be quite high. Otherwise, very few solutions were accepted. As seen in class, the parameter N should be proporitonal to the number of neighbours. It has been set to twice the size of the neighbourhood (~ 100 for the instances of size 50, ~ 200 for the instances of size 100).

Several values have been tested for the α parameter. It has been noticed that the cooling rate should be slow. Otherwise, the temperature become low too fast and no more solutions are accepted. The final value used is 0.998.

2.1.3 Addition component

It has been noticed that even with a slow cooling rate, the temperature decreases very fast and the algorithm does not accept new solutions long before the end of the algorithm. In addition, as

the algorithm is stochastic, it follows different paths each at execution (with a different seed) and good solutions can be missed. To improve the result, when the temperature is too low, it is set to the original temperature again. The limit temperature, which is equal to 100, has been decided by testing the number of solutions accepted.

2.2 GRASP

2.2.1 Principle

Greedy Randomised Adaptive Search Procedures (GRASP) is a metaheuristic based on a constructive step and a local search step. The algorithm build a new solution and applies a local search to it. This step is repeated until the termination criterion is met.

The local search step is done using the algorithms from the first implementation exercise. According to the first report, best improvement doesn't improve, so first improvement is chosen. The insert neighbourhood is also used because it was the best one.

The constructive step is based on the simplified rz heuristic. The construction depends on a parameter α which represents the amount of randomness of the solution. The first step of the algorithm remains the same : the jobs are ordered according to the weighted sum of the processing times. After this step, the insertion position of the jobs is uniformly chosen amoung the positions whose cost is lower or equal than $\alpha * cost_{worst} + (1 - \alpha) * cost_{best}$, where $cost_{best}$ is the cost of the partial solution using the best insertion position and $cost_{worst}$ is the cost using the worst position. If α is equal to 0, the solution is equal to the heuristic solution (if no ties) and if α is equal to 1, the solution is almost random.

2.2.2 Parameter tuning

There is only one paramter for this algorithm. However, the tuning is complicated since the optimal α is not the same on for instances. It has been observed that a low value is better. The final value used in the algorithm is 0.4.

2.2.3 Addition component

It has been noticed that like simulated annealing, the algorithm does not improve the solution long before the termination criterion is met. In order to use the time budget more efficiently, an additional searching step has been used. This additional step consists in multiple local search runs starting from the greedy randomised solution, after a perturbative step. This method has been used to exploit the promising solutions generated by the construction step. The process is executed if the solution selected is close enough to the best solution found so far. The percentage of proximity to the best solution and the number of searches done have been tuned so that the algorithm generates enough greedy randomised solutions.

The perturbation step has been selected from [1]. It consists in applying random insert moves. According to [1], the number of moves applied should be low. Indeed, tests have shown that the best solutions are obtained with about 5 insert moves.

2.3 Stoping critera

mean instances 50 : 60.17067, x500 : 30085.33
mean instances 100 : 563.8932, x500 : 281946.6

3 better solution found

50_20_03
1 28 24 14 48 15 34 45 19 5 18 37 11 29 31 9 39 10 25 33 30 44 2 43 7 41 20 40 49 47 38 4 42 17 16 22 6 26 8 50 13 21 32 23 46 36 35 3 12 27
591806

Références

[1] Q.-K. Pan, R. Ruiz. 2012. Local search methods for the flowshop scheduling problem with flowtime minimization. European Journal of Operational Research 222(1), 31-43.