Comparison of Methods for Generating Initial Solution for Simulated Annealing

Jaroslav Szabo

***Abstract*—** The article deals with the problem of choosing a method for creating first step solution for metaheuristic method of simulating annealing. Methods are verified on Taillard´s benchmark instances for flow-shop problematic. Three methods are compared based upon their speed, the best created solution and the number of iterations needed to improve result by simulated annealing.

***Keywords*—** flow shop problem, makespan, simulated annealing

I. Introduction

Simulated Annealing (SA) is a metaheuristic method for finding feasible solution. Origin of these methods tracks back to 1980s when Kirkpatrick, Gelatt and Vecchi and independently on them V. Černý described this method in [1] and [2]. Each approach was based on different background but both were tested on Traveling Salesman Problem (TSP). In our article we will discuss how much we can influence the working of SA method, by choosing different method for generating initial solution. Comparison of three different methods which can provide us with initial solution used in SA will be demonstrated on Taillard´s benchmarks for problem of Flow Shop scheduling [3]. We can compare already obtained optimal upper bound solutions with output of our SA method.

II. Simulated Annealing

Metaheuristic methods unlike the heuristic methods allow, under certain conditions, the transition to the solution of inferior value of the objective function, [4]. They consist of the two main parts. The first part is finding the initialisation solution and the calculation of the objective function. The second part is the process of improving the value of objective function in order to search for an optimal solution. We will deal with metaheuristic method of simulated annealing and finding initial solution. In the search for an initial solution we can use many different methods and procedures [5]-[8]. They may include simple heuristic methods or other general practices instead. The diagram below shows a complete scheme of a metaheuristic method. All of the best known metaheuristics have almost the same scheme for evaluating the best possible solution found. We will focus on simulated annealing metaheuristic.

J. Szabo Zilinska Univerzita, Zilina, Slovakia (e-mail: jaroslav.szabo@uniza.sk)

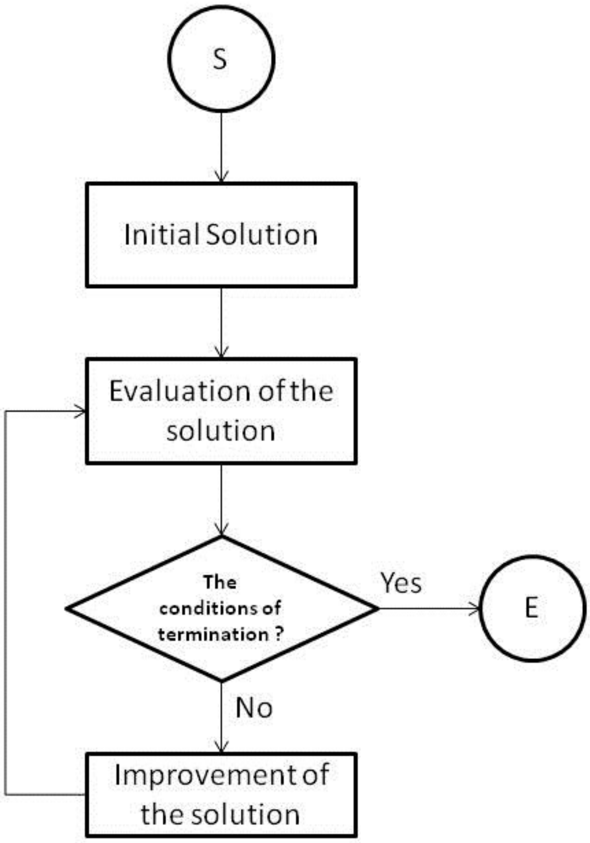

Fig 1 Diagram of metaheuristic

Let us show pseudocode for simulated annealing method.

```
function SA (x, t^max,T^max,β)
   x^best:=x; t:=0;
      T=T^max;
      repeat
            x'=GetRandom(x)
            if AcceptMove(x, x', T) then
                  x^best:=KeepBest(x', x^best)
                  x:=x';
            end-if
            T=T/(1+βT);
            t:=CPUTime;
      until t > t^max
      return x^best
```

- x^{best} - best found solution
- x - initial solution
- x'- next random solution
- t^{max} - time for method SA to run
- t - elapsed time
- T^{max} - maximum set temperature
- T - current temperature
- β – cooling index

For AcceptMove() method we can choose from several options:

1. The threshold approach when the accept random variable is set on constant value.
2. Monte Carlo approach.
3. Generic approach for bivalent variable.
4. Approach based on temperature. We used this approach for method AcceptMove().

$$p(x', x, T) = \begin{cases} e^{-(f(\mathbf{x}') - f(\mathbf{x}))/T}, & if \;\; f(\mathbf{x}') - f(\mathbf{x}) > 0; \\ 1 & otherwise. \end{cases}$$

III. LOW SHOP PROBLEMATIC

In general, the scheduling problem consists of positioning resource-demanding activities over time in such a way that the side constrains are respected and an object is minimised. Problems like this arise in diverse areas including production planning, civil engineering, computer science, etc. [9], [10]. In our work we will test effectiveness of initial solution creating methods on flow shop problematic. This is one of NP-hard optimization problems that cannot be solved by exact solving methods in polynomial time. This problem is one of basic scheduling problems of *m* machines and *n* jobs. Every machine *M* can handle only one operation *o* of any job *J* in defined time and only one operation of single job can by handled in defined time [11]. Also, order of operations handled by machines is strictly done so M_1 just does operations o_{i1} to o_{im} and so on. We will test initial methods on problems of 5 machines and 20 jobs, 50 jobs and 100 jobs [3].

IV. METHODS

When we are creating initial solutions for metaheuristic, we can use many different approaches. For example mathematical programming, heuristic methods or other metaheuristic methods. We will focus on three simple methods: Monte Carlo method and operations to rank the elements in the array according to the values of objective function. These methods we will call Weight method 1 and Weight method 2. Weight method 1 will evaluate objective function and rank elements from the best to the worst. Weight method 2 will do the opposite. Our concern will be how fast these methods can work, how good the created solution is compared to the best solution and how many steps are needed to obtain it.

V. RESULTS

All tests will be evaluated on computer Intel Core i7 (2,66GHz), 8GB RAM. First step will be comparison of three tested methods on problem of 5 machines and 20 jobs [3]. All measurement are averaged from 50 repetitions and rounded. In resulting tables we can find these data:

- VOC—Value of objective function created by method.
- VOE—Value of objective function evaluated by SA.
- NOI—Number of iteration of improvement.
- TCI—Time for creating initial solution [ns].

SA parameters were set on time - 250 ms, temperature - 100 and cooling index - 0.01.

Table 1 The results of experiment for 5 machines and 20 jobs

Method	VOC	VOE	NOI	TCI
Monte Carlo	1530	1282	18	4670
Weight method 1	1556	1281	21	24082
Weight method 2	1472	1280	17	17388

Second evaluation tests run on problem of 5 machines and 50 jobs [3].
SA parameters were set on time - 500 ms, temperature - 100 and cooling index - 0.01.

Table 2 The results of experiment for 5 machines and 50 jobs

Method	VOC	VOE	NOI	TCI
Monte Carlo	3198	2727	26	8210
Weight method 1	3119	2727	26	114768
Weight method 2	3240	2727	29	110394

Last set of tests run on problem of 5 machines and 100 jobs [3].
SA parameters were set on time - 1000 ms, temperature - 100 and cooling index - 0.01.

Table 3 The results of experiment for 5 machines and 100 jobs

Method	VOC	VOE	NOI	TCI
Monte Carlo	6198	5494	35	13415
Weight method 1	5981	5495	32	388354
Weight method 2	6258	5494	41	394184

VI. Conclusion

From results obtained in previous section we can draw several conclusions. The best initial values of objective functions of the tested problems were obtained by weighted methods. Monte Carlo is a random method, so obtained values can differ on whole spectrum. Method Monte Carlo is the fastest of the three compared methods. It is a result of the implementation of method Random() in Java programming language. Difference between the numbers of iterations of improvement is not significant enough to determine which method is better. We recommend using the Monte Carlo method which arose between tested methods by incomparable speed and therefore it is useful for creating initial solution for large problems.

References

[1] CERNY, V.: Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. Journal of Optimization Theory and Applications, 45, 41-51, 1985.

[2] KIRKPATRICK, S., GERLATT, Jr., C., VECCHI, M.: Optimization by Simulated Annealing. Science 220, 671-680, 1983.

[3] TAILARD, E.: Benchmarks for basic scheduling problems. European Journal of Operational Research 64, 278-285, 1993.

[4] JANACEK, J., HURTIK, J.: An impact of improvement-exchange heuristics to quality of probabilistic TSP. Communications - Scientific Letters of the University of Zilina, 4, 27-35, 2003.

[5] GLOVER, F., KOCHENBERGER, G.: Handbook of metaheuristics 57. Springer, International Series in Operations Research & Management Science, 2003.

[6] CZIMMERMANN, P.: A note on using graphs in regular scheduling problems. Communications - Scientific Letters of the University of Zilina, 4, 47-48, 2003.

[7] BLUM, C., ROLI, A.: Metaheuristics in combinatorial optimization: Overview and conceptual comparison. ACM Computing Surveys, 268-308, 2003.

[8] MARTINCOVA, P.: Performance of simulated grid scheduling algorithms. Journal of Information, Control and Management Systems, Vol. 5, No. 2, 261-270, 2007.

[9] PALUCH, S.: Bus scheduling as a graph coloring problem. Communications - Scientific Letters of the University of Zilina, 4, 16-20, 2003.

[10] JANOSIKOVA, L., HREBEN, M.: Mathematical programming vs. constraint programming for scheduling problems. Communications - Scientific Letters of the University of Zilina, 1, 39-43, 2013.

[11] GAREY, M.R., JOHNSON, D.S., SETHI, R.: The complexity of flowshop and jobshop scheduling. Mathematics of operational research, 1, 117-129, 1976.