Sequential and parallel simulated annealing to solve the VRPTW

Bożena Wieczorek*, Marcin Połomski†

*Faculty of Electrical Engineering, Silesian University of Technology, Akademicka 10, Gliwice, Poland, e-mail: bozena.wieczorek@polsl.pl
†Faculty of Electrical Engineering, Silesian University of Technology, Akademicka 10, Gliwice, Poland, e-mail: marcin.polomski@polsl.pl

Abstract **This work presents sequential and parallel simulated annealing algorithm to solve the vehicle routing problem with time windows (VRPTW). The VRPTW is a combinatorial optimization problem in which the number of vehicles and the total distance travelled by the vehicles are to be minimized. The aim is to explore speedups and investigate how the shorter annealing chains in the parallel algorithm of independent searches influence the accuracy of solutions to the problem. The accuracy of solutions is measured by their proximity to the optimum solution found so far. The computational experiments were carried out on the test instances by Solomon.**

Keywords **Simulated annealing, vehicle routing problem with time windows, bicriterion optimization, parallel computing.**

I. Introduction

The vehicle routing problem with time windows can be formulated as follows. There is a central depot of cargo and n customers (nodes) located at some distances from the depot. The locations of the depot $(i = 0)$ and the customers $(i = 1, 2, \dots, n)$, the shortest distances $d_{i,j}$ and the corresponding travel times $t_{i,j}$ between any two locations i and j are given. The cargo have to be delivered to each customer i according to the delivery demand q_i by a fleet of K vehicles. Each vehicle serves a subset of customers on the route which begins and ends at the depot. The vehicles have the same capacity Q. For each customer a service time window $[a_i,\ b_i]$ and a service time s_i are defined. a_i and b_i determine, respectively, the earliest and the latest time for start servicing. The objective is to find the set of routes which guarantees the delivery of cargo to all customers and satisfies the time window and vehicle capacity constraints. Furthermore, the size of the set equal to the number of vehicles needed (primary goal) and the total travel distance (secondary goal) should be minimized. More formally, the aim is to:

$$\text{minimize} \quad K, \quad \text{and then} \tag{1}$$

$$\text{minimize} \quad \textstyle\sum_{i=0}^{n}\sum_{j=0,j\neq i}^{n}\sum_{k=1}^{K} d_{i,j}x_{i,j,k}, \tag{2}$$

where a decision variable $x_{i,j,k}$ is 1 if vehicle k travels from customer i to j, and 0 otherwise. The R101 test instance by Solomon and the best solution for this test are presented on Fig. 1 and 2.

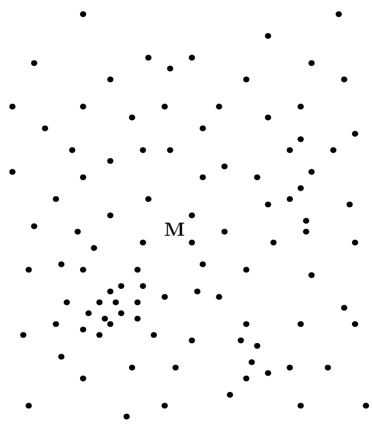

Figure 1. The R101 test instance by Solomon

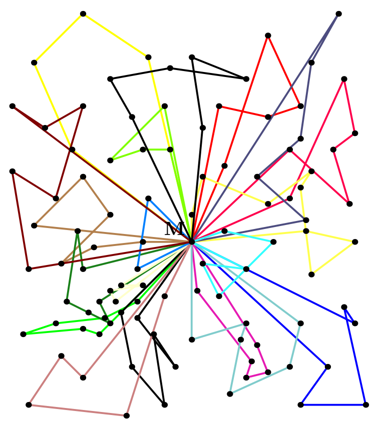

Figure 2. The best solution for the R101 test instance

II. Sequential simulated annealing

The algorithm of simulated annealing is a variant of local search in which the neighbour solutions of lower costs are always accepted. The solutions of higher costs are accepted with the probability:

$$e^{-\frac{\delta}{T_i}} \tag{3}$$

where T_i, $i = 0, 1, \dots, i_{\max}$, is a parameter called a temperature of annealing, which falls from some initial value T_0 according to the formula $T_{i+1} = \beta T_i$, where β $(\beta < 1)$ is a constant and δ denotes an increase of the solution cost. A sequence of steps for which a temperature of annealing remains constant is called a cooling stage (chain). The application of simulated annealing to solve the VRPTW is as follows. An initial solution to the problem is found by making use of some heuristics. The process of simulated annealing is divided into two phases. Each phase consists of some number of cooling stages, and each cooling stage consists of some number of annealing steps (Fig. 3).

```
Create the initial_solution;
solution ← initial_solution;
best_solution ← initial_solution;
for f = 1 to 2 do {phase 1 and 2}
  T ← T_{0,f}; {initial temperature}
  repeat
    for i = 1 to L do
      annealing_step_f(solution, best_solution);
    end for;
    T ← β_f · T; {temperature reduction}
  until a_f cooling stages are executed;
end for;
```

Figure 3. Sequential simulated annealing for the VRPTW

The main goal of phase 1 is minimizing the number of routes of the solution, whereas phase 2 minimizes in the first place the total length of these routes. On every annealing step a neighbour solution is determined by moving one or more customers among the routes. Both the customers and the routes are chosen randomly. The sequential algorithm of annealing halts if a specified number of cooling stages is executed. Contrary to the classical approach we memorize the best solution found during the whole annealing process.

III. Parallel independent searches

Let us assume that p processes, $p = 4, 8, 16$ and 20, can be executed and each of them is capable of generating its own annealing chains. In the algorithm of independent searches (PIS) processes P_j, $j = 1, 2, \ldots, p$ carry out the independent annealing searches using different initial solutions. As the final result the best solution among the solutions found by the processes is chosen. In order to keep the cost of parallel computations constant the length of a cooling stage is decreased with an increase of the number of processes. At each temperature process P_j executes $L = 4E/p$ annealing steps, where $E = 10^5$ is a constant. The computational experiments were performed on a cluster of 11 nodes, each equipped with Intel® Core$^{\text{TM}}$2 Quad $2,4$GHz processor.

IV. Experimental results

The parallel algorithm of independent searches was implemented using C language and Intel MPI library. The computational experiments were carried out on the R109, R110, R202 and RC102 test instances by Solomon. The overall number of experiments performed for each test instance was 5000, 5 series of 1000 experiments (for $p = 4, 8, 16, 20$ and the sequential algorithm, $p = 1$). For each test and a given number of processes, p, the mean value of total travel distances of routes, $\bar{y}$, the number of hits into the best solution, H, the total execution time, T and the speedup, S, were calculated. The results of experiments are shown in Tab. I. Comparing the sequential and the PIS algorithms it can be seen that shorter annealing chains in the parallel algorithm give inferior solutions with respect to the mean value of total distances of routes. As the number of parallel processes grows up the value of $\bar{y}$ and the number of hits into the best solution worsen. Only for the R202 and RC102 test instances and the number of processes $p = 4$ and $p = 4, 8$ respectively, the value of $\bar{y}$ is better. Similarly, for the test R109 and the number of hits into the best solution found so far, H. However the final results obtained by the PIS algorithm are not worse than by 2% as compared to the sequential algorithm. The speedups achieved for this MPI implementation of the PIS algorithm should be better as the parallel processes do not communicate while executing their annealing chains. This problem needs further research.

Table I
Results for tests R109, R110, R202, RC102

Test	p	$\bar{y}$	H	T	S
R109	1	1203.26	164	183.1	1
	4	1205.25	214	47.3	3.9
	8	1209.78	131	27.8	6.6
	16	1216.85	43	15.5	11.8
	20	1219.64	21	12.5	14.7
R110	1	1136.30	99	192.6	1
	4	1138.90	48	50.7	3.8
	8	1142.37	17	30.3	6.4
	16	1153.23	2	16.8	11.5
	20	1156.07	2	12.0	16.1
R202	1	1199.75	119	174.3	1
	4	1198.79	63	48.9	3.6
	8	1203.50	18	26.5	6.6
	16	1202.92	11	12.6	13.8
	20	1211.58	10	10.9	16.0
RC102	1	1559.25	478	172.0	1
	4	1556.64	429	46.5	3.7
	8	1558.00	268	23.5	7.3
	16	1560.06	78	12.3	14.0
	20	1562.03	47	10.5	16.4

V. Conclusions

It is considered as the main result of this work that the shorter annealing chains in the parallel algorithm of independent searches yields solutions of worse accuracy to the problem as compared to the sequential algorithm. The results obtained by the PIS algorithm however are not worse than by 2%. The areas of further investigations include modification of the PIS algorithm in order to simulate the behaviour of the sequential simulated annealing and examination of other MPI communication modes to achieve better speedups.

References

[1] Arbelaitz. O., Rodriguez, C., Zamakola. I., Low cost parallel solutions for the VRPTW optimization problem, Proc. of the Intern. Conf. on Parallel Processing Workshops, (2001).

[2] Azencott, R., Parallel simulated annealing: An overview of basic techniques, in Azencott, R. (Ed.), Simulated annealing. Parallelization techniques, J. Wiley, NY, (1992), 37–46.

[3] Czech, Z.J., Wieczorek, B., Frequency of co-operation of parallel simulated annealing processes, Proc. of the 6th International Conference on Parallel Processing and Applied Mathematics (PPAM'05), (September 11–14, 2005), Poznań, Poland.