View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by IntechOpen

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,800
Open access books available

122,000
International authors and editors

135M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

CLARIVATE ANALYTICS BOOK CITATION INDEX INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

13

A parallel simulated annealing algorithm as a tool for fitness landscapes exploration

Zbigniew J. Czech
Silesia University of Technology and University of Silesia
Poland

1. Introduction

Solving a discrete optimization problem consists in finding a solution which maximizes (or minimizes) an objective function. The function is often called the fitness and the corresponding landscape the fitness landscape. We are concerned with statistical measures of a fitness landscape in the context of the vehicle routing problem with time windows (VRPTW). The measures are determined by using a parallel simulated annealing algorithm as a tool for exploring a solution space. This chapter summarizes our experience in designing parallel simulated annealing algorithms and investigating fitness landscapes of a sample NP-hard bicriterion optimization problem.

Since 2002 we have developed several versions of the parallel simulated annealing (SA) algorithm (11)-(19). Each of these versions comprises a number of parallel SA processes which co-operate periodically by passing and exploiting the best solutions found during the search. For this purpose a specific scheme of co-operation of processes has been devised. The methods of parallelization of simulated annealing are discussed in Aarts and van Laarhoven (2), Aarts and Korst (1), Greening (20), Abramson (3), Boissin and Lutton (8), and Verhoeven and Aarts (35). Parallel simulated annealing to solve the VRPTW is applied by Arbelaitz et al. (4). Onbaşoğlu and Özdamar (26) present the applications of parallel simulated annealing algorithms in various global optimization problems. The comprehensive study of parallelization of simulated annealing is given by Azencott et al. (5)

The parallel SA algorithm allowed us to discover the landscape properties of the VRPTW benchmarking tests (33). This knowledge not only increased our understanding of processes which happen during optimization, but also helped to improve the performance of the parallel algorithm. The usage of the landscape notion is traced back to the paper by Wright (37). The more formal treatments of the landscape properties are given by Stadler (32), Hordijk and Stadler (22), Reidys and Stadler (31). Statistical measures of a landscape are proposed by Weinberger (36). The reviews of the landscape issues are given by Reeves (30) and Reeves and Rowe (29).

Section 2 of this chapter formulates the optimization problem which is solved. Section 3 describes a sequential SA algorithm. In section 4 two versions of the parallel SA algorithm, called independent and co-operating searches, are presented. Section 5 is devoted to the statistical measures of the fitness landscapes in the context of the VRPTW. In subsections 5.1-5.2 some basic notions are introduced, and in subsection 5.3 the results of the experimental study are discussed. Section 6 concludes the chapter.

2. Problem formulation

The VRPTW is an extension to the capacitated vehicle routing problem (CVRP) which is formulated as follows (34). There is a central depot of goods and n customers (nodes) geographically scattered around the depot. The locations of the depot ($i = 0$) and the customers ($i = 1$, 2, …, n), as well as the shortest distances d_{ij} and the corresponding travel times t_{ij} between any two customers i and j are given. Each customer asks for a quantity q_i of goods which has to be delivered (or picked up from) by a vehicle of capacity Q. Because of this capacity limit, the vehicle after serving a subset of customers has to return to the depot for reloading. The vehicle effects the whole service on a number of routes. Each route starts and terminates at the depot. A solution to the CVRP is a set of routes of minimum travel distance (or travel time) which visits each customer i, $i = 1, 2, \ldots, n$, exactly once. The total demand for each route cannot exceed Q.

The CVRP is extended into the VRPTW by introducing for each customer and the depot a service time window $[e_i, f_i]$ and a service time s_i ($s_0 = 0$). The values e_i and f_i determine, respectively, the earliest and the latest time for start servicing. The customer i has to be served within the time window $[e_i, f_i]$ and the service of all customers should be accomplished within the time window of the depot $[e_0, f_0]$. The vehicle can arrive to the customer before the time window but then it has to wait until time e_i, when the service can begin. The latest time for arrival of the vehicle to customer i is f_i. It is assumed that the routes are traveled simultaneously by a fleet of K homogeneous vehicles (i.e. of equal capacity), each vehicle assigned to a single route. A solution to the VRPTW is the set of routes which guarantees the delivery of goods 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 objective) and the total travel distance (secondary objective) should be minimized.

More formally, there are three types of decision variables in this two-objective optimization problem. The first decision variable, $x_{i,j,k}$, $i, j \in \{0,1,\ldots,n\}$, $k \in \{1,2,\ldots,K\}$, $i \neq j$, is 1 if vehicle k travels from customer i to j, and 0 otherwise. The second decision variable, t_i, denotes the time when a vehicle arrives at customer i, and the third decision variable, b_i, denotes the waiting time at that customer. The aim is to:

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

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

subject to the following constraints:

$$\sum_{k=1}^{K} \sum_{j=1}^{n} x_{i,j,k} = K, \text{ for } i = 0, \tag{3}$$

$$\sum_{j=1}^{n} x_{i,j,k} = \sum_{j=1}^{n} x_{j,i,k} = 1, \text{ for } i = 0 \text{ and } k \in \{1,2,\ldots,K\}, \tag{4}$$

$$\sum_{k=1}^{K} \sum_{j=0, j\neq i}^{n} x_{i,j,k} = \sum_{k=1}^{K} \sum_{i=0, i\neq j}^{n} x_{i,j,k} = 1, \text{ for } i, j \in \{1,2,\ldots,n\}, \tag{5}$$

$$\sum_{i=1}^{n} q_i \sum_{j=0, j\neq i}^{n} x_{i,j,k} \leq Q, \text{ for } k \in \{1,2\ldots,K\} \tag{6}$$

$$\sum_{k=1}^{K} \sum_{i=0, i\neq j}^{n} x_{i,j,k}(t_i + b_i + h_i + t_{i,j}) \leq t_j, \quad \text{for } j \in \{1,2,\ldots,n\} \tag{7}$$
$$\text{and } t_0 = b_0 = h_0 = 0,$$

$$e_i \leq (t_i + b_i) \leq f_i, \text{ for } i \in \{1,2,\ldots,n\}. \tag{8}$$

Formulas (1) and (2) define the minimized functions. Eq. (3) specifies that there are K routes beginning at the depot. Eq. (4) expresses that every route starts and ends at the depot. Eq. (5) assures that every customer is visited only once by a single vehicle. Eq. (6) defines the capacity constraints. Eqs. (7)–(8) concern the time windows. Altogether, eqs. (3)–(8) define the feasible solutions to the VRPTW.
Lenstra and Rinnooy Kan (24) proved that the VRP and the VRPTW are NP-hard discrete optimization problems.

3. Sequential simulated annealing

The algorithm of simulated annealing which can be regarded as a variant of local search was first introduced by Metropolis et al. (25), and then used to optimization problems by Kirkpatrick, Gellat and Vecchi (23), and Cěrny (10). A comprehensive introduction to the subject can be found in Reeves (27) and Azencott (5).
Let $C\colon S \mapsto \mathbb{R}$ be a cost function which is to be minimized, defined on some finite solution set (search space) S. Let $N(X)$, $N(X) \subset S$, be a set of neighbors of solution X for each $X \in S$. Usually the sets $N(X)$ are small subsets of S. For the VRPTW the members of $N(X)$ are constructed by moving one or more customers among the routes of solution X. The way in which these sets are created influences substantially the accuracy of results obtained by a simulated annealing algorithm. While constructing the sets $N(X)$ we make sure that their members are built through deep modifications of X. Let R be a transition probability matrix, such that $R(X,Y) > 0$ if and only if $Y \in N(X)$. Let (T_i), $i = 0, 1, \ldots$ be a sequence of positive numbers, called the temperatures of annealing, such that $T_i \geq T_{i+1}$ and $\lim_{i\to\infty} T_i = 0$. The sequence (T_i) is called the cooling schedule, and a sequence of annealing steps within which the temperature of annealing stays constant is called the cooling stage. Consider the sequential annealing algorithm for constructing a sequence (or chain) of solutions (X_i), $X_i \in S$, defined as follows. An initial solution X_0 is computed using e.g. some heuristics. Given the current solution X_i, a potential next solution Y_i is chosen from set $N(X_i)$ with probability $R(X_i, Y_i)$. Then in a single annealing step solution X_{i+1} is set as follows (cf. Fig. 2):

$$X_{i+1} = \begin{cases} Y_i & \text{if } C(Y_i) \leq C(X_i), \\ Y_i & \text{with probability } p_i, \text{ if } C(Y_i) > C(X_i), \\ X_i & \text{otherwise,} \end{cases}$$

where

$$p_i = \exp(-(C(Y_i) - C(X_i))/T_i). \tag{9}$$

If X_{i+1} is set to Y_i and $C(Y_i) > C(X_i)$, then we say that an uphill move is carried out. Eq. (9) implies that uphill moves are performed more often when temperature T_i is high. When T_i is close to zero uphill moves occur sporadically. Simulated annealing can be described formally by non-homogeneous Markov chains. In these chains the probability of moving from one state to another depends not only on these states but also on the temperature of annealing.
A solution $X \in S$ is said to be a local minimum of the cost function C, if $C(X) \leq C(Y)$ for all $Y \in N(X)$, and to be a global minimum of C, if $C(X) = \inf_{Y \in S} C(Y)$. Let $S_{\min}$ be the set of global minima of C. We say that the process of simulated annealing converges, if $\lim_{i \to \infty} P(X_i \in S_{\min}) = 1$. It was proved (21) that the convergence is guaranteed by the logarithmic cooling schedules of the form: $T_i \geq \frac{R}{\log(i+1)}$ for some constant R which depends on the cost function landscape. It was also shown (5; 9) that for the logarithmic cooling schedules the speed of convergence is given by:

$$P(X_i \notin S_{\min}) \sim \left(\frac{K}{i}\right)^{\alpha} \tag{10}$$

for i large enough, where $K > 0$ and $\alpha > 0$ are suitable constants. Both constants are connected to the cost function landscape, and for large solution spaces constant K is very large and constant α is very small (5; 9). This implies that the process of simulated annealing converges very slowly. According to Eq. (10) a global minimum is attained only if the process of annealing is infinite. For this reason the question of how to accelerate simulated annealing by making use of parallelism is crucial.
In the sequential simulated annealing algorithm to solve the VRPTW, the chain (X_i) is constructed in two phases. The goal of phase 1 is to minimize the number of routes of the VRPTW solution, whereas phase 2 minimizes the total length of the routes. However in phases 1 and 2 it may happen that both the number of routes and the total length of routes are reduced. The cost of solution X_i in phase 1 is computed as: $C_1(X_i) = c_1 N + c_2 D + c_3(r_1 - \bar{r})$, and in phase 2 as: $C_2(X_i) = c_1 N + c_2 D$, where N is the number of routes (vehicles) of solution X_i, D – the total travel distance of the routes, r_1 – the number of customers of a randomly chosen route which is to be shorten and perhaps eliminated from the current solution, $\bar{r}$ – the average number of customers in all routes, c_1, c_2, c_3 – some constants. For simplicity, instead of the logarithmic an exponential cooling schedule is used, i.e. the temperature of annealing is decreased as $T_{k+1} = \beta_f T_k$, for $k = 0, 1, \ldots, a_f$, and some constants β_f ($\beta_f < 1$) and a_f ($f = 1$ and 2 denote phase 1 and 2).

4. Parallel simulated annealing algorithm

4.1 Independent searches

In the parallel algorithm of independent searches (IS), p independent simulated annealing processes $\mathcal{P}_0$, $\mathcal{P}_1$, …, $\mathcal{P}_{p-1}$ are executed. Every process performs its computations like in the sequential algorithm. On completion, the processes pass their best solutions found to the master process, which selects the best solution among solutions it received. This solution constitutes the final result of the IS algorithm.
More formally, suppose that i steps of sequential simulated annealing is taken. Then in parallel IS, p annealing chains of $z = i/p$ steps each are executed. As the result p terminal solutions $\{X_{z,0}, X_{z,1}, \ldots, X_{z,p-1}\}$ are computed, from which the final solution Y_i is selected by: $Y_i = X_{z,0} \otimes X_{z,1} \otimes \ldots \otimes X_{z,p-1}$, where $\otimes$ is the operator of choosing the better solution with respect

to the total length of routes[1]. In terms of convergence we have (5):

$$P(Y_i \notin S_{\min}) = \prod_{0 \le j \le p-1} P(X_{z,j} \notin S_{\min}). \tag{11}$$

Assuming that each simulated annealing chain j of z steps converges at speed determined by Eq. 10: $P(X_{z,j} \notin S_{\min}) \sim \left(\frac{K}{z}\right)^{\alpha}$, we get (5):

$$P(Y_i \notin S_{\min}) \sim \left(\frac{Kp}{i}\right)^{\alpha p}. \tag{12}$$

Consider a chain of $i = 10^7$ steps of sequential simulated annealing, and let $K = 100$ and $\alpha = 0.01$. Then according to Eq. 10 the speed of convergence is equal $\left(\frac{K}{i}\right)^{\alpha} \approx 0.89$. If one uses $p =$ 5, 10, 15 and 20 processes, then by Eq. (12) the speeds of convergence of IS are: $\left(\frac{Kp}{i}\right)^{\alpha p} \approx 0.61$, 0.40, 0.27 and 0.18, respectively. Thus the parallel independent searches converge much faster than the sequential algorithm.

4.2 Co-operating searches

The parallel algorithm of co-operating searches (CS) executes in the form of p processes $\mathcal{P}_0$, $\mathcal{P}_1, \ldots, \mathcal{P}_{p-1}$ (Figs. 1-3). A process generates its own annealing chain divided into two phases (lines 6–19 in Fig. 1). A phase consists of a number of cooling stages, and a cooling stage consists of a number of annealing steps. The processes co-operate with each other every ω annealing step passing their best solutions found to date (lines 12–16 in Fig. 1, and Fig. 3). The chain of annealing steps of process $\mathcal{P}_0$ is entirely independent (Fig. 4). The chain of process $\mathcal{P}_1$ is updated at steps $u\omega$, $u = 1, 2, \ldots, u_{\mathrm{m}}$, to the better solution between the best solutions found by processes $\mathcal{P}_0$ and $\mathcal{P}_1$ to date. Similarly, process $\mathcal{P}_2$ chooses as the next point in its chain the better solution between its own best and the one obtained from process $\mathcal{P}_1$. Thus the best solution found by process $\mathcal{P}_l$ is piped down for further enhancement to processes $\mathcal{P}_{l+1} \ldots \mathcal{P}_{p-1}$. Clearly, after step $u_m\omega$ process $\mathcal{P}_{p-1}$ holds the best solution X_b found by all the processes. To our best knowledge the speed of convergence of co-operating searches given e.g. by equations similar to Eq. (10) and (12) are not known.

As mentioned before, the temperature of annealing decreases according to the equation $T_{k+1} = \beta_f T_k$ for $k = 0, 1, 2, \ldots, a_f$, where a_f is the number of cooling stages. In this work we investigate two cases in establishing the points of process co-operation with respect to temperature drops. In the first case, of regular co-operation, processes interact at the end of each cooling stage ($\omega = L$) (lines 12–13 in Fig. 1). The number of annealing steps executed within a cooling stage is set to $L = (5E)/p$, where $E = 10^5$ is a constant established experimentally, and $p = 5$, 10, 15 and 20, is the number of processes (line 3 in Fig. 1). Such an arrangement keeps the parallel cost of the algorithms constant when different numbers of processes are used, provided the co-operation costs are neglected. Therefore in this case as the number of processes becomes larger, the length of cooling stages goes down, what means that the frequency of co-operation increases. In the second case, of rare co-operation, the frequency is constant and the processes exchange their solutions every $\omega = E$ annealing step (lines 14–15 in Fig. 1). For the number of processes $p = 10$, 15 and 20, the co-operation takes place after 2, 3 and 4 temperature drops, respectively.

[1] In this analysis it is assumed that each chain achieves a solution with the minimum (best known) number of routes.

```
1   parfor P_j, j = 0, 1, ..., p − 1 do
2     Set co-operation mode to regular or rare depending on a test set;
3     L := (5E)/p; {establish the length of a cooling stage; E = 10^5}
4     Create the initial_solution using some heuristics;
5     current_solution_j := initial_solution; best_solution_j := initial_solution;
6     for f := 1 to 2 do {execute phase 1 and 2}
        {beginning of phase f}
7       T := T_{0,f}; {initial temperature of annealing}
8       repeat {a cooling stage}
9         for i := 1 to L do
10          annealing_step_f(current_solution_j, best_solution_j);
11        end for;
12        if (f = 1) or (co-operation mode is regular) then {ω = L}
13          co_operation;
14        else {rare co-operation: ω = E}
15          Call co_operation procedure every E annealing step
              counting from the beginning of the phase;
16        end if;
17        T := β_f T; {temperature reduction}
18      until a_f cooling stages are executed;
        {end of phase f}
19    end for;
20  end parfor;
21  Produce best_solution_{p−1} as the solution to the VRPTW;
```

Fig. 1. Parallel simulated annealing algorithm of co-operating searches

```
1   procedure annealing_step_f(current_solution, best_solution);
2     Create new_solution as a neighbor to current_solution
        (the way this step is executed depends on f);
3     δ := C_f(new_solution) − C_f(current_solution);
4     Generate random x uniformly in the range (0, 1);
5     if (δ < 0) or (x < e^{−δ/T}) then
6       current_solution := new_solution;
7       if C_f(new_solution) < C_f(best_solution) then
8         best_solution := new_solution;
9       end if;
10    end if;
11  end annealing_step_f;
```

Fig. 2. Annealing step procedure

The exchange of solutions between processes can be considered as exploitation of the search results, whereas exploration takes place when a process penetrates the search space freely. Let us call a sequence of ω annealing steps executed by a process between points of co-operation as a chain of free exploration. Taking into account Eq. (10) the longer these chains the better.

```
1   procedure co_operation;
2     if j = 0 then Send best_solution_0 to process P_1;
3     else {j > 0}
4       receive best_solution_{j-1} from process P_{j-1};
5       if C_f(best_solution_{j-1}) < C_f(best_solution_j) then
6         best_solution_j := best_solution_{j-1};
7         current_solution_j := best_solution_{j-1};
8       end if;
9       if j < p − 1 then Send best_solution_j to process P_{j+1}; end if;
10    end if;
11  end co_operation;
```

Fig. 3. Procedure of co-operation of processes

$$X_0 \rightarrow \left\{ \begin{array}{ccccccccccc} X_0^{(0)} & \rightarrow & X_0^{(\omega)} & \rightarrow & X_0^{(2\omega)} & \rightarrow & \bullet & \bullet & \rightarrow & X_0^{(u_m\omega)} & \\ & & \downarrow & & \downarrow & & & & & \downarrow & \\ X_1^{(0)} & \rightarrow & X_1^{(\omega)} & \rightarrow & X_1^{(2\omega)} & \rightarrow & \bullet & \bullet & \rightarrow & X_1^{(u_m\omega)} & \\ & & \downarrow & & \downarrow & & & & & \downarrow & \\ \bullet & & \bullet & & \bullet & & \bullet & \bullet & & \bullet & \\ \bullet & & \bullet & & \bullet & & \bullet & \bullet & & \bullet & \\ X_{p-2}^{(0)} & \rightarrow & X_{p-2}^{(\omega)} & \rightarrow & X_{p-2}^{(2\omega)} & \rightarrow & \bullet & \bullet & \rightarrow & X_{p-2}^{(u_m\omega)} & \\ & & \downarrow & & \downarrow & & & & & \downarrow & \\ X_{p-1}^{(0)} & \rightarrow & X_{p-1}^{(\omega)} & \rightarrow & X_{p-1}^{(2\omega)} & \rightarrow & \bullet & \bullet & \rightarrow & X_{p-1}^{(u_m\omega)} & \rightarrow X_b \end{array} \right.$$

Fig. 4. Scheme of co-operation of processes (X_0 – initial solution; X_b – best solution among the processes)

Note that due to co-operation, a process after having completed a chain with solution X, may be forced to explore the search space from a—probably more promising—solution different from X. In order to obtain good results during parallel search the proper balance between exploitation and exploration has to be maintained.

A series of experiments was carried out in order to establish how the number of processes, the length of chains of free exploration, and the frequency of processes co-operation influence the accuracy of solutions to the VRPTW (16). For the experiments, 39 out of 56 benchmarking tests[2] elaborated by Solomon (33) were used. The tests are grouped into three major problem sets named R, C and RC. The geographical coordinates for customers in sets R, C and RC are generated randomly, in a clustered manner, and as a mix of random and clustered structures, respectively. Each of these sets is divided into two subsets, R1, R2, C1, C2, RC1, RC2. The subsets R1, C1 and RC1 have short time windows and permit relatively large numbers of routes (between 9 and 19) in the solutions. The time windows for subsets R2, C2 and RC2 are wider allowing less routes (between 2 and 4) in the solutions. Every test involves 100 customers and the distances are measured using Euclidean metric. It is assumed that travel times are equal to the corresponding distances.

[2] The tests in set C are easy to solve, so they were omitted in the experiments.

In the series of the experiments, the IS and CS algorithms[3] were executed at least 1000 times for each test, a given number of processes p, a number of annealing steps L_2 fixed for it, and a period of communication ω. Based on each sample of results[4] the average of total travel distances of routes $\bar{y}$ and the standard deviation s were calculated. The experiments showed that depending on the test instance, the minimum of the mean value $\bar{y}$ appeared for different values of parameters p, L_2 and ω. E.g. the minimum of $\bar{y}$ for test R101 was obtained for $p = 20$ and $L_2 = \omega = E/4$ (Table **??**). Whether these specific values of parameters give

p	L_2	ω	R101	R102	R103	R104	R105	R106
5	E	E	13.9	17.6	11.4	0.8	1.1	10.6
10	$E/2$	$E/2$	6.5	11.6	3.9	0.5	0.1	5.3
15	$E/3$	$E/3$	1.4	3.3	min	0.7	2.3	2.1
20	$E/4$	$E/4$	min*	min*	0.6*	min	0.3	min
10	$E/2$	E	10.3	13.6	5.5	0.7	1.4	6.7
15	$E/3$	E	15.3	19.9	9.7	1.1	1.0	3.0
20	$E/4$	E	13.8	20.1	8.8	0.6*	min*	0.9*
p	L_2	ω	R107	R108	R109	R110	R111	R112
5	E	E	0.8	1.0	min*	min*	0.3	1.2
10	$E/2$	$E/2$	min	0.1	6.5	3.2	min	1.1
15	$E/3$	$E/3$	1.1	min	6.7	3.7	1.4	min
20	$E/4$	$E/4$	0.7*	1.2*	10.4	5.3	1.9*	0.8
10	$E/2$	E	1.9	1.5	4.1	2.7	1.5	1.6
15	$E/3$	E	3.1	2.7	8.8	3.6	3.0	0.7
20	$E/4$	E	4.6	4.2	10.3	5.5	3.6	1.3*

Table 1. Values of test statistic Z for CS algorithm and set R1;'*' marks the best choice of parameters p, L_2 and ω

statistically superior results can be proved by testing the hypotheses $H_0 : \mu_i \leq \mu_m$ versus an alternative hypothesis $H_a : \mu_i > \mu_m$, where μ denotes the mean value of a population of total travel distances of routes; i – populations whose samples have worse mean values (e.g. cases $p = 5$ and $L_2 = \omega = E$; $p = 10$ and $L_2 = \omega = E/2$; etc. for test R101); m – a population for which the minimum mean value of a sample was observed (i.e. case $p = 20$ and $L_2 = \omega = E/4$ for test R101). In the cases where H_0 are rejected one can claim that their values of parameters p, L_2 and ω give inferior solutions with respect to the values for which $\bar{y} = \bar{y}_{\min}$ occur, or equivalently, the population with $\bar{y} = \bar{y}_{\min}$ comprises superior solutions as compared to other

[3] It was observed (15) that for some Solomon's tests the probability of finding a solution with the minimum number of routes was very low. Therefore phase 1 of the algorithms was executed in the CS fashion with $a_1 = 50$ cooling stages and $L_1 = 10^5$ annealing steps in each stage. In phase 2 the IS and CS modes were used with $a_2 = 100$ and L_2 depending on the number of processes. The following values of parameters were fixed: $c_1 = 40000$, $c_2 = 1$, $c_3 = 50$, $\beta_1 = 0.95$, $\beta_2 = 0.98$.

[4] For some tests the size of the sample was smaller than 1000, since only solutions with the minimum number of routes were considered.

A.	p	L_2	ω	R109	R110	R202	RC102	RC104	RC108	RC202
IS	5	E	–	2.1*	2.7	5.6	2.4	3.0	min*	3.3
	10	$E/2$	–	2.9	4.8	9.4	4.0	6.5	8.5	4.4
	15	$E/3$	–	6.8	6.5	12.0	5.2	11.1	13.6	3.6
	20	$E/4$	–	8.8	9.5	13.1	6.3	11.7	20.3	4.5
CS	5	E	E	min	min*	min*	min	min*	2.4	min*
	10	$E/2$	$E/2$	6.5	3.2	7.2	0.8*	2.7	7.1	4.0
	15	$E/3$	$E/3$	6.7	3.7	10.4	3.4	5.2	12.1	6.7
	20	$E/4$	$E/4$	10.4	5.3	12.8	4.1	8.2	15.2	8.2
CS	10	$E/2$	E	4.1	2.7	4.7	5.3	3.3	5.1	2.5
	15	$E/3$	E	8.8	3.6	7.8	3.5	6.2	10.4	3.4
	20	$E/4$	E	10.3	5.5	9.7	4.3	6.8	13.6	3.9

Table 2. Values of test statistic Z for IS and CS algorithms

populations. For the test statistic:

$$Z = \frac{\bar{y}_i - \bar{y}_m}{\sqrt{\frac{s_i^2}{n_i} + \frac{s_m^2}{n_m}}}$$

the hypotheses H_0 are rejected at the $\alpha = 0.01$ significance level, if $Z > Z_{0.01} = 2.33$ (n_i and n_m are numbers of experiments over which s_i and s_m values are calculated). Table **??** shows the values of Z for set R1 (results for sets R2, RC1 and RC2 are reported in (16)), where min indicates values of p, L_2 and ω which give the minimum of $\bar{y}$. The framed values denote rejections of hypotheses H_0, what means that for the corresponding values of parameters p, L_2 and ω, the results of statistically worse total travel distances of routes are achieved. It can be seen that the values of Z for test R101 and parameters $p = 15$, $L_2 = \omega = E/3$, and $p = 20$, $L_2 = \omega = E/4$, are less than 2.33. So it is justified to claim that these values of parameters give statistically the best solutions to the VRPTW. In other words, using $p = 20$ or 15 processes co-operating after every cooling stage enable us to obtain quickly solutions of the best accuracy.
It follows from the experiments (16) that for most Solomon's tests the results of high accuracy can be achieved by making use of $p = 20$ processes. The exceptions are tests R109, R110, R202, RC102, RC104, RC108 and RC202. For these tests the minimum of $\bar{y}$ occurs when $p = 5$ and most of other numbers of processes yield statistically worse results. As already indicated, to keep the cost of parallel computations constant, the number of annealing steps taken by processes between points of co-operation was decreased along with an increase of the number of processes. The results of the experiments prove that for the tests listed above the execution of shorter annealing chains of free exploration of length from $L_2 = E/4$ to $L_2 = E/2$ are not compensated—in terms of accuracy—by the co-operation between processes.
The annealing chains of free exploration are substantially longer in the algorithm of independent searches (IS), in which the processes do not co-operate and execute chains as long as $L_2 = Ea_2$, where a_2 is the fixed number of cooling stages[5]. Table 2 compares the results obtained by the IS and CS algorithms for the specific tests mentioned above. It can be seen that an increase of the length of chains and lack of co-operation in the IS algorithm, make

[5] Note that altogether each process of the IS and CS algorithms executes $a_1 + a_2$ cooling stages.

the results worse for tests R110, R202, RC104 and RC202. Applying the IS algorithm—of low communication cost—can be justified only for tests R109 and RC108.
Considering the results of the experiments and the objective of computing good quality solutions to the VRPTW in a short time, Solomon's tests can be divided into 3 groups:

I – tests which can be solved quickly (e.g. using $p = 20$ processes) to good accuracy with rare co-operation ($\omega = E$). To this group belong 24 tests, out of 39, not listed in groups II and III specified below.

II – tests which can be solved quickly (e.g. with $p = 20$) but the co-operation should take place after every cooling stage (we call this co-operation regular) (e.g. $\omega = E/4$ for $p = 20$) to achieve good accuracy of solutions. This group comprises 8 tests: R101, R102, R103, R107, R108, R111, R207 and RC105.

III – tests whose solving cannot be accelerated as much as for the tests in groups I and II. The solutions of best accuracy are obtained for less than $p = 20$ processes[6]. To this group belong 7 tests: R109, R110, R202, RC102, RC104, RC108 and RC202.

5. Fitness landscape

5.1 Basic definitions

Let C, S and $N(X)$ be a cost function, a search space and a set of neighbors of solution X, respectively, as defined in section 3. A solution $X^o \in S$ is said to be a local minimum of function C, if $C(X^o) \leq C(Y)$ for all $Y \in N(X^o)$, and to be a global minimum X^* of C, if $C(X^*) = \inf_{Y \in S} C(Y)$. In evolutionary optimization function C is often called the fitness and the associated landscape a fitness landscape. More formally (29), a landscape $\mathcal{L}$ for the function C is a triple $\mathcal{L} = (S, C, d)$ where d denotes a distance measure $d\colon S \times S \mapsto \mathbb{R}^+ \cup \{\infty\}$ which for any solutions $P, Q, R \in S$ satisfies the conditions: $d(P,Q) \geq 0$, $d(P,Q) = 0 \Leftrightarrow P = Q$ and $d(P,R) \leq d(P,Q) + d(Q,R)$. If d is symmetric, i.e. $d(P,Q) = d(Q,P)$ for all $P, Q \in S$ then d is a metric on space S.
Discrete optimization can be performed by neighborhood search where the process of searching starts at some initial solution and converges to a local optimum, or an attractor. The searching process is described by a function $\mu\colon S \mapsto S^o$, where $X \in S$ is an initial solution and $\mu(X)$ is the optimum that it reaches (29). A basin of attraction of solution X^o is the set $B(X^o) = \{X\colon \mu(X) = X^o\}$. The set contains the initial solutions from which the search leads to a specified attractor. The basins of attraction for a given function are not unique. They depend on a method adopted for landscape exploration and can be established only if the method is deterministic. Therefore the notion of the basin is of limited use for methods with a good deal of randomization, like simulated annealing.

5.2 Statistical measures of fitness landscape

The nature of a fitness landscape can be unveiled either by mathematical analysis or by gathering some statistical data during the process of searching it. In this work we follow the latter approach. Several statistical measures have been proposed in the literature. Weinberger (36) observed that some characteristics could be obtained from a random walk. Let C_t be the fitness of the solution visited at time t. Then the autocorrelation function of the landscape during

[6] There are two open questions here: whether less than $p = 5$ processes could give solutions of better accuracy for some tests in group III, and whether finding solutions for tests in groups I-II can be speeded up even further by making use of more than $p = 20$ processes with no loss of solutions accuracy.

a random walk of length T is:

$$a_j = \frac{\sum_{t=1}^{T-j}(C_t - \bar{C})(C_{t+j} - \bar{C})}{\sum_{t=1}^{T}(C_t - \bar{C})^2}$$

where $\bar{C}$ is the mean fitness of the T solutions visited, and j is the lag. For smooth landscapes, with neighbor solutions of similar fitness, and small lags, the values of a_j are close to 1. As the lag increases the values of autocorrelation are likely to diminish. The values of a_j are close to zero at all lags for rugged landscapes, where close solutions have unrelated fitness.
A useful indicator of the difficulty of an optimization problem is the number of optima appearing in a corresponding landscape. Indeed, the more optima in the landscape, the harder is to find the global optimum. Suppose that for a given optimization problem the search is restarted r times with random initial solutions. Most likely these solutions lay in different basins of attraction, so as the result of the search a number of different local solutions k, $k \leq r$, will be found. Based on the values of r and k one may estimate the number of optima ν present in a given landscape. Assuming that the probability of encountering each solution is the same, it is easy to show that the random variable K which takes the number of distinct solutions in a series of r independent searches has the Arfwedson distribution (28):

$$P[K = k] = \frac{\nu!}{(\nu - k)!\,\nu^r} rk \tag{13}$$

where $1 \leq k \leq \min(r, \nu)$, with the mean:

$$\mathrm{E}K = \nu[1 - (1 - 1/\nu)^r]. \tag{14}$$

After having measured EK one can solve numerically Eq. (14) and find an estimate for $\hat{\nu}$. Reeves (28) gives an approximation of it as: $\hat{\nu} \approx (\bar{K}^2 - r)/(2(r - \bar{K}))$, where $\bar{K}$ is a measured estimation of EK. When the value of ν is small one may ask how many searches W should be done to be sure with some certainty that all optima have been found. The waiting time W_k for the $(k+1)$th solution provided that k of them have been already found has a geometric distribution, and the mean of the waiting time for ν solutions is (28):

$$\mathrm{E}W \approx \nu(\ln\nu + \gamma) \tag{15}$$

where $\gamma \approx 0.577$ is Euler's constant. The formulas (13)–(15) are derived under the assumption that the probability of encountering each solution is the same, or in other words that solutions are isotropically distributed in the landscape. Unfortunately in many optimization problems, also in the VRPTW, this assumption is not valid.

5.3 Experimental study

The objective of the study was to gather statistical data concerning the fitness landscapes for 39 (out of 56) VRPTW tests by Solomon.

Fitness landscape characteristics

In the course of experiments the parallel simulated annealing algorithm was executed at least 4200 times (see column Exp. in Table 4) for each test in sets R and RC. The VRPTW is a two-objective optimization problem in which both, the number of routes and the total travel distance, should be minimized. For the landscape studies only solutions with the minimum

number of routes were taken into account. Most of Solomon's tests are relatively easy to solve with respect to the first objective function (the exceptions are tests R104, R112, RC101, RC105, and RC106, see paragraph "Difficulty of test instances"). The minimum number of routes for each test is generally known. Since the VRPTW problem is NP-hard, there is some probability that these numbers are not global minima. However for simplicity, we shall name them as 'minima' instead of 'known minima'.

Table 3 contains the histograms of numbers of solutions[7] produced by the algorithm with the total travel distance y worse by 0-1%, 1-2% etc. than the distance $y_{\min}$ of the best-known solution. The columns denoted by $\bar{\tau}$ and $\tau_{\max}$ show the values of $(\bar{y} - y_{\min})/y_{\min}$ and $(y_{\max} - y_{\min})/y_{\min}$, where $\bar{y}$ and $y_{\max}$ are the average and maximum total travel distances obtained for a given test, respectively. All values in Table 3 are expressed in per cents, and the tests are ordered according to 0-1% column. It can be seen e.g. for test R112 that there is 30% chance of getting a solution with distance y worse by 2-3% than $y_{\min}$. This is because the number of distinct solutions in terms of y for this test, discovered in ranges from 0-1% to >4% were 102, 149, 179, 89 and 81, respectively. Clearly, the distribution of solutions in the fitness landscape is not isotropic, i.e. they are not uniformly scattered across every direction in the search space. There exists a relatively large number of solutions with $y \in [1.02y_{\min}, 1.03y_{\min})$, what increases the probability that the algorithm will finish its computations at a local optimum with y in this range. Fig. 5 plots the distances d of a sample of solutions from the best solution found $X_{\min}$, against the total travel distances of solution routes[8]. As a metric for measuring the distance d between solutions we use the minimum number of customer movements among the routes necessary to convert one solution into another (see subsection 5.1). It was observed that the solutions of all VRPTW tests were not sampled with equal probability. For instance, the majority of solutions of test R112[9] were hit only a few times, but 5 solutions were reached at least 10 times (marked by white circles in Fig. 5). Most likely the sizes of basins of attraction of more popular solutions are larger, although the notion of such a basin is vague in the context of simulated annealing where random uphill moves may take place. The characteristics of the fitness landscape depend also on the search algorithm. Note that the solutions of test R112 reached most often (at least 10 times) are located in range 0-2%, i.e. range of good accuracy (Fig. 5), partly due to good convergence, as we believe, of the parallel algorithm which favors solutions of higher quality. In general, the shape of the landscape which is discovered is as good as thorough is an exploration of the landscape conducted by the algorithm. On the other hand, an excellent search algorithm can give a biased picture of the landscape, since the worse local optima are then found less frequently—if at all—than the better ones. Similar results to that of test R112 were obtained for other Solomon's tests characterized by "long histograms" (see columns 0-1% ... >4% of Table 3). For instance, the numbers of distinct solutions discovered for tests R211 and RC202 in ranges from 0-1% to >4% were 335, 1047, 926, 723, 1351 and 7349, 3105, 14281, 19246, 9027, respectively. The attractors (marked by white circles) were observed in ranges 0-3% (test R211) and 0-5% (test RC202) (Figs. 6–7).

[7] Note that each of these solutions is a local minimum to the VRPTW problem with respect to the total travel distance.

[8] Note that two separate series of experiments were conducted. In the first series the data contained in Tables 3-4 were gathered. The goal of the second series of experiments was to find, up to 700 best solutions to the selected VRPTW tests. The results of these experiments are depicted in Figs. 5-12.

[9] Overall 9200 executions of the algorithm were carried out for this test, 600 executions produced solutions with the number of routes equal 9, which is likely to be minimum, and 399 of these solutions were distinct.

Test	0-1 %	1-2 %	2-3 %	3-4 %	>4 %	$\bar{\tau}$	$\tau_{\max}$
R112	17	25	30	15	13	2.4	6.5
R110	45	21	18	7	9	1.6	11.4
R108	47	37	13	2	1	1.2	5.9
R107	61	37	2	0	0	0.8	7.7
R109	70	16	7	3	4	0.8	10.6
R111	72	6	13	5	4	0.9	10.3
R104	82	17	1	0	0	0.5	2.4
R106	91	9	0	0	0	0.6	1.8
R103	96	4	0	0	0	0.6	3.8
R102	100	0	0	0	0	0.4	1.0
R105	100	0	0	0	0	0.2	1.6
R101	100	0	0	0	0	0.1	0.5
R211	8	24	21	16	31	3.4	12.4
R207	25	26	11	20	18	2.4	11.4
R210	41	44	15	0	0	1.2	3.2
R203	56	43	1	0	0	0.9	3.4
R204	75	3	14	8	0	0.9	4.9
R208	77	23	0	0	0	0.6	2.9
R202	88	1	5	5	1	0.5	6.3
R209	97	3	0	0	0	0.2	2.5
R206	99	1	0	0	0	0.4	2.6
R201	100	0	0	0	0	0.1	1.3
R205	100	0	0	0	0	0.0	3.4
RC108	63	25	9	2	1	0.9	11.6
RC104	69	7	24	0	0	0.8	3.1
RC106	72	10	14	4	0	0.7	4.9
RC101	89	11	0	0	0	0.2	2.1
RC102	96	0	1	3	0	0.3	7.6
RC105	99	1	0	0	0	0.3	1.4
RC103	100	0	0	0	0	0.1	3.4
RC107	100	0	0	0	0	0.0	0.4
RC202	14	6	27	36	17	3.2	13.5
RC203	64	23	11	2	0	0.8	4.0
RC206	89	8	3	0	0	0.5	3.3
RC205	91	9	0	0	0	0.5	2.5
RC207	94	4	1	1	0	0.3	4.9
RC201	96	4	0	0	0	0.3	2.8
RC208	97	3	0	0	0	0.2	2.3
RC204	100	0	0	0	0	0.1	3.7

Table 3. Histograms of numbers of solutions in specified ranges 0-1%, 1-2%, ..., >4%, $\bar{\tau} = (\bar{y} - y_{\min})/y_{\min}$, $\tau_{\max} = (y_{\max} - y_{\min})/y_{\min}$ (all values in per cent; tests are ordered according to 0-1% column)

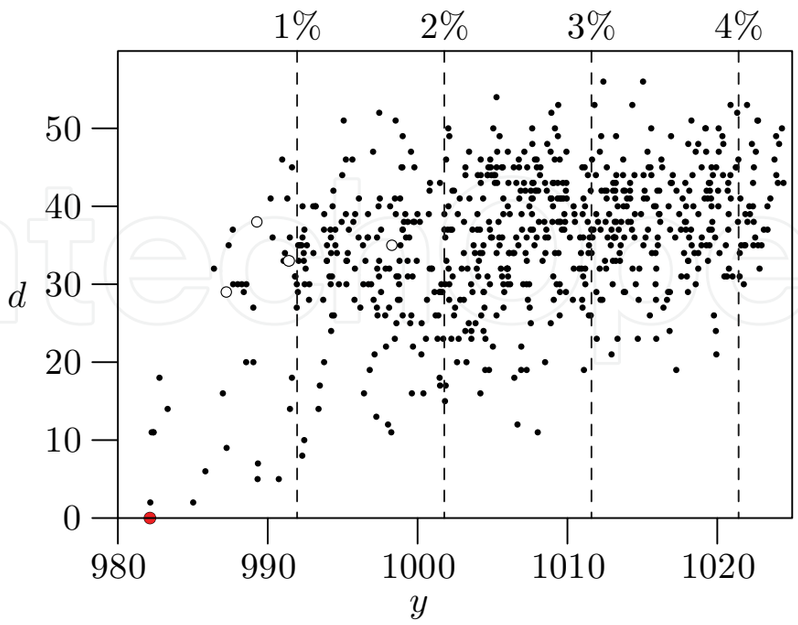

Fig. 5. Distance d from the best solution marked by a shaded circle vs. total travel distance y for test R112 (700 solutions, $X_{\min} = (N_{\min}, y_{\min}) = (9, 982.14)$, $N_{\min}$ is the minimum number of routes)

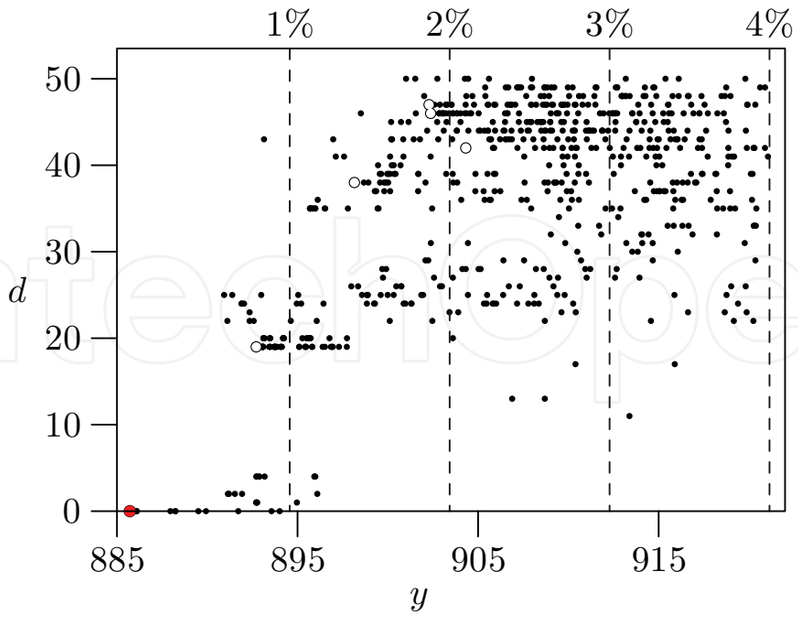

Fig. 6. Distance d from the best solution vs. total travel distance y for test R211 (700 solutions, $X_{\min} = (2, 885.71)$)

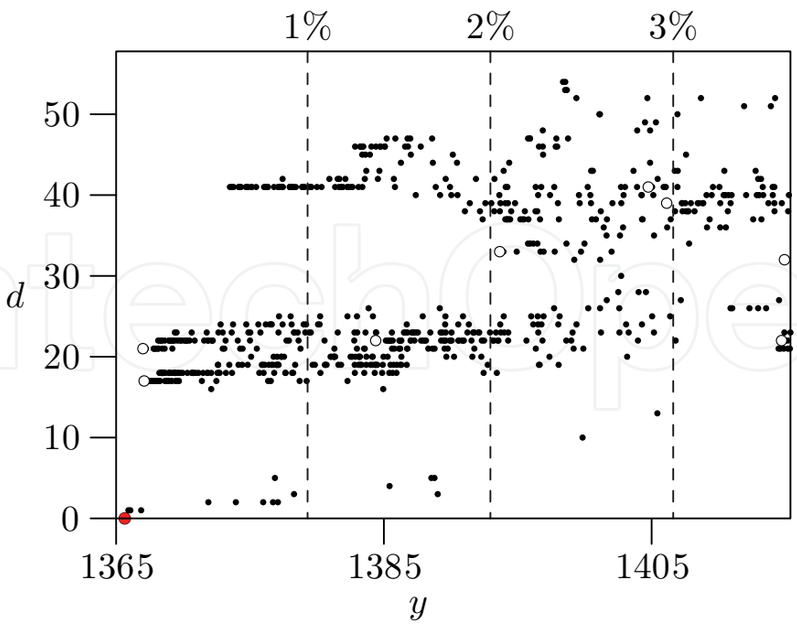

Fig. 7. Distance d from the best solution vs. total travel distance y for test RC202 (700 solutions, $X_{\min} = (3, 1365.64)$)

"Big valley" structure

Several experimental studies in discrete optimization revealed correlations among locations of local optima which suggest existence of a globally convex or "big valley" structures in the fitness landscapes (7). The experiments indicated that local optima are closer (in terms of distance d) to the global optimum than are random points in a search space. The local optima are also closer to each other and form a "big valley" structure with the best local (or global) optimum appearing in a center of the valley. The phenomenon can be illustrated by plotting a graph of fitness against average distance from all other optima. The graph in Fig. 8 shows that the best solution (marked by a shaded circle) has almost minimum distance $\bar{d}$ what implies it is located near the center of the valley. However this is not the case for the graphs in Figs. 9 and 10 where many local optima are much closer to the center than the best solution.

Approximate and exact solutions

Suppose that a hard optimization problem is to be solved. Then getting an approximate solution worse no more than by 0-1% with respect to the optimum can be considered as adequate. In such circumstances a good indicator of the problem difficulty is the value of $\bar{\tau} = (\bar{y} - y_{\min})/y_{\min}$ which exhibits the shift of the average cost $\bar{y}$ of solutions from $y_{\min}$ attained by solving the problem repeatedly. The value of $\tau_{\max} = (y_{\max} - y_{\min})/y_{\min}$ provides some insight into the depth of local optima. If $\bar{\tau} \leq 1\%$ is observed then a problem can be thought of as easy to solve. Assuming that 1% accuracy of solution approximation is acceptable, all VRPTW tests, except R112, R110, R108, R211, R207, R210 and RC202, can be classified as easy[10] (Table 3). Fig. 11 drawn for test R102 shows that all its 700 best solutions found, have their y values within 0.28% margin from $y_{\min}$, what indicates that the fitness landscape

[10] Remembering of course that they are instances of the NP-hard problem being solved by the advanced parallel algorithm.

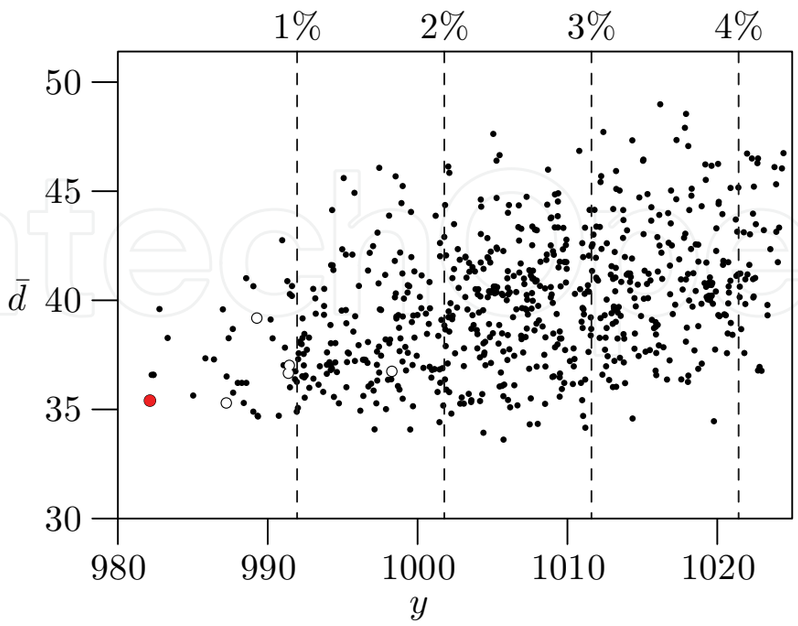

Fig. 8. Average solution distance $\bar{d}$ from the remaining 699 solutions vs. total travel distance y for test R112 (700 solutions)

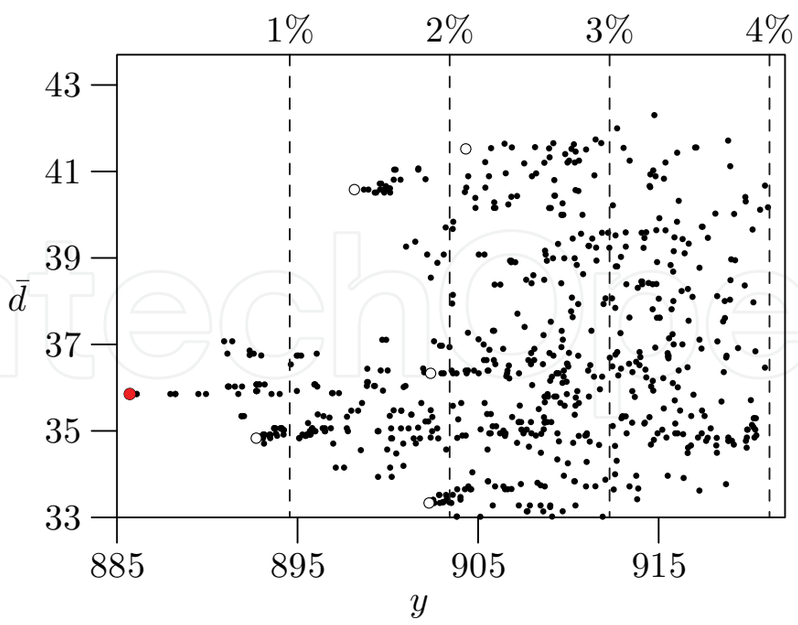

Fig. 9. Average solution distance $\bar{d}$ from the remaining 699 solutions vs. total travel distance y for test R211 (700 solutions)

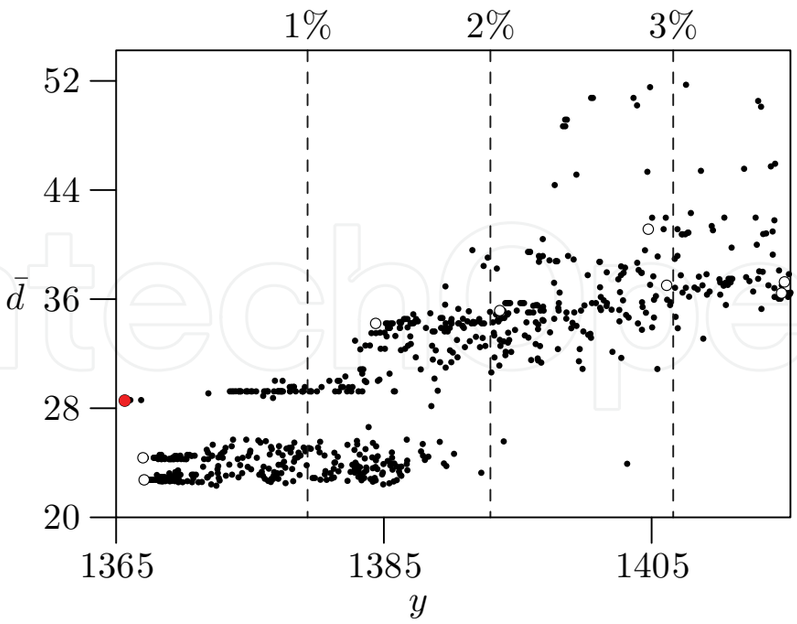

Fig. 10. Average solution distance $\bar{d}$ from the remaining 699 solutions vs. total travel distance y for test RC202 (700 solutions)

ot this test is quite smooth. Thus test R102 can be ranked as easy to solve if an approximate solution is sought. However it turns out to be the hardest one in Solomon's set if the optimum is searched for. The smoothness of the landscape is an advantage if one wants to solve a problem with some limited accuracy. In the case when the absolute optimum is desired, the key role plays the number of local optima ν appearing in the landscape. For test R102 the number of these optima was as large as 44773. Table 4 contains the numbers of local optima unveiled for Solomon's tests in the first series of our experiments. Fig. 15 shows the plot of average $\bar{K}$ as a function of the number of unveiled optima ν (see subsection 5.1).

Difficulty of test instances

Since the VRPTW is a two-objective optimization problem, the difficulty of its instances can be estimated by probabilities P_1, P_2 and P_3 that after execution of a searching algorithm i) a solution with the minimum number of routes is found, ii) a solution with the minimum number of routes and minimum distance allowing 1% accuracy is found, and iii) the best-known solution is found, respectively (see Table 4, where $\bar{K}$ – average number of distinct solutions observed in a sample of series of $r = 100$ experiments, ν – number of local optima unveiled; Exp. – number of experiments conducted; $\bar{K}$ and ν are calculated over solutions with minimum number of routes; tests are ordered according to P_2 column). Note that probability P_2 is a product of P_1 and the value of the 1st column of Table 3 scaled to range $[0,1]$. The probability P_3 is counted as a ratio of the number of times the best-known solution is found to ν. If the best-known solution is not found, then probability P_3 is determined by considering the best solution obtained for a given test by the parallel algorithm, see http://sun.aei.polsl.pl/~zjc/best-solutions.html. The hardest test to solve with 1% accuracy is R104 ($P_2 = 0.002$) and there are many easy tests in this respect, R101, R102, etc. As mentioned before, it is very difficult to solve test R102 to its optimality ($P_3 = 2 \cdot 10^{-5}$). The easy tests in this regard are R205 ($P_3 = 0.997$) and R105

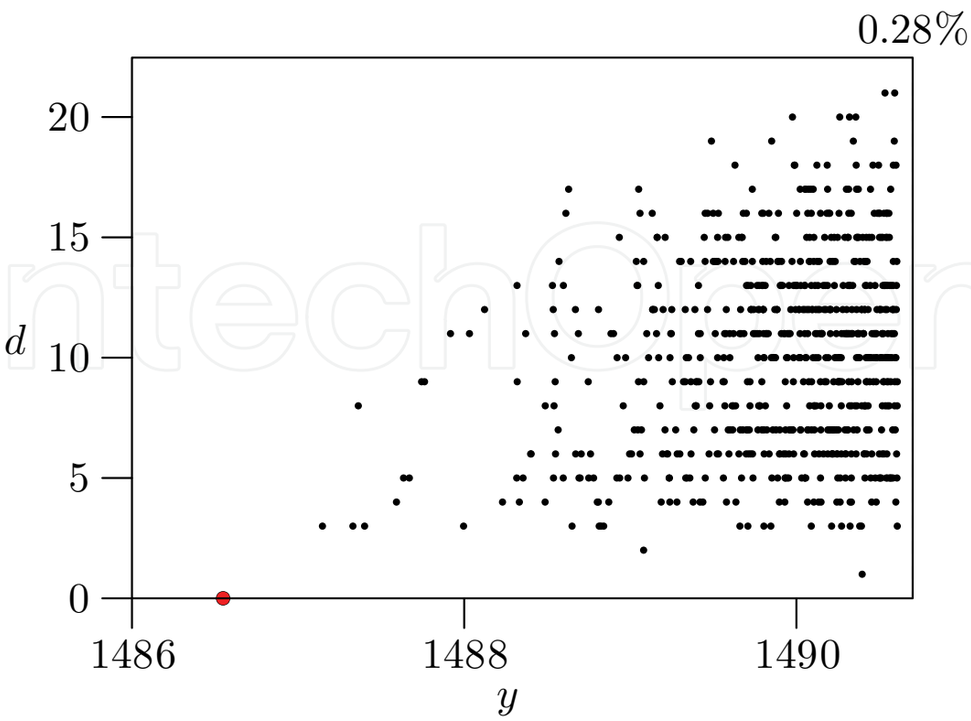

Fig. 11. Distance d from the best solution vs. total travel distance y for test R102 (700 solutions, $X_{\min} = (17, 1486.55)$)

$(P_3 = 0.559)$. As can be seen in Fig. 12 there are many solutions of test R105 located at small distances d from the minimum. Clearly, such a dense distribution of good quality solutions surrounding the optimum one, facilitates the gradual improvements of the configuration of a current solution during the process of simulated annealing. In contrast, there are not many neighbor solutions close to the minima for tests R112, R211, RC202 and R102 (see Figs. 5-7 and Fig. 11). Each of these minima belongs to a "small valley" of solutions which occurs away from the "big valley" structure. As the result, reaching those minima from an arbitrary initial solution by a process of small enhancements is not easy, and sometimes not possible at all.
The plots in Fig. 13 and 14 show the difficulty of 39 tests by Solomon. For the tests: R104, R112, RC101, RC105, RC106, both probabilities (P_1, P_2 in Fig. 13, and P_1, P_3 in Fig. 14) are less than 0.5. Thus these tests can be classified as the most difficult to solve in Solomon's set.

Taking advantage of landscape properties

In this paragraph we ponder how the features of the fitness landscape can be exploited to improve the performance of the parallel simulated annealing algorithm solving the VRPTW problem. Boese et al. (7) proposed an adaptive multi-start algorithm for the traveling salesman problem. It consists of two phases. In the first, initial phase, a set of R random local minima is computed. In the second phase, which is executed a specified number of times, based on the k ($k \leq R$) best local minima found so far, an adaptive starting solution is constructed. This solution is then improved A times using the greedy descent algorithm, what results in a set of $k + A$ local minima. From this set, the k best minima are selected, a new adaptive starting solution is formed, and the second phase is repeated. An adaptive starting solution is created

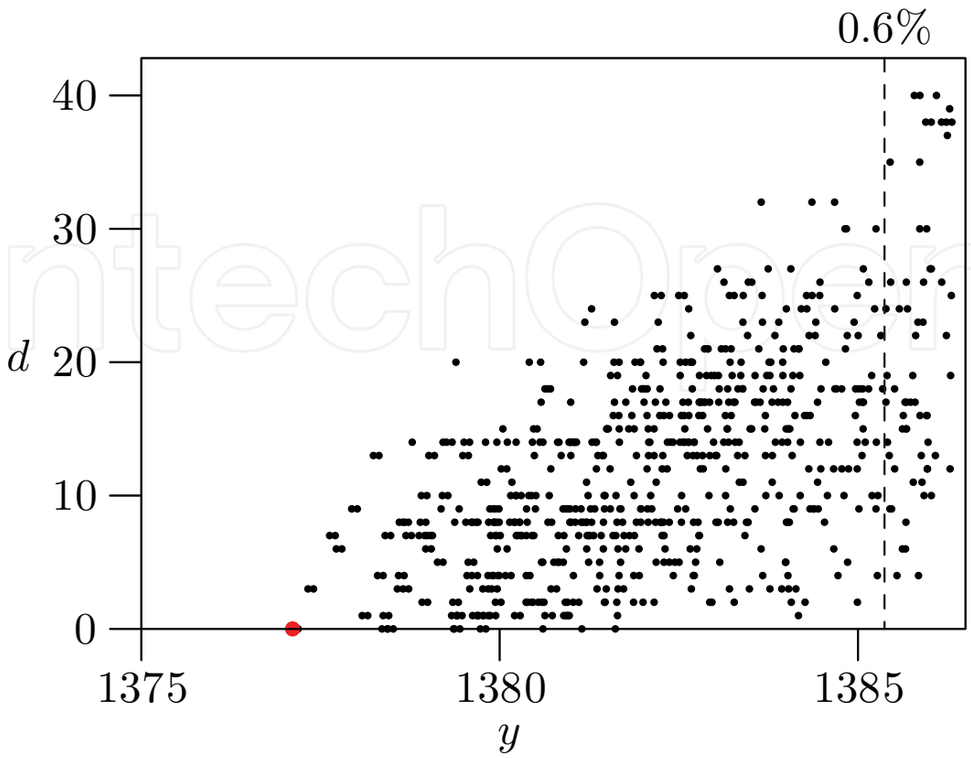

Fig. 12. Distance d from the best solution vs. total travel distance y for test R105 (700 solutions, $X_{\min} = (14, 1377.11)$)

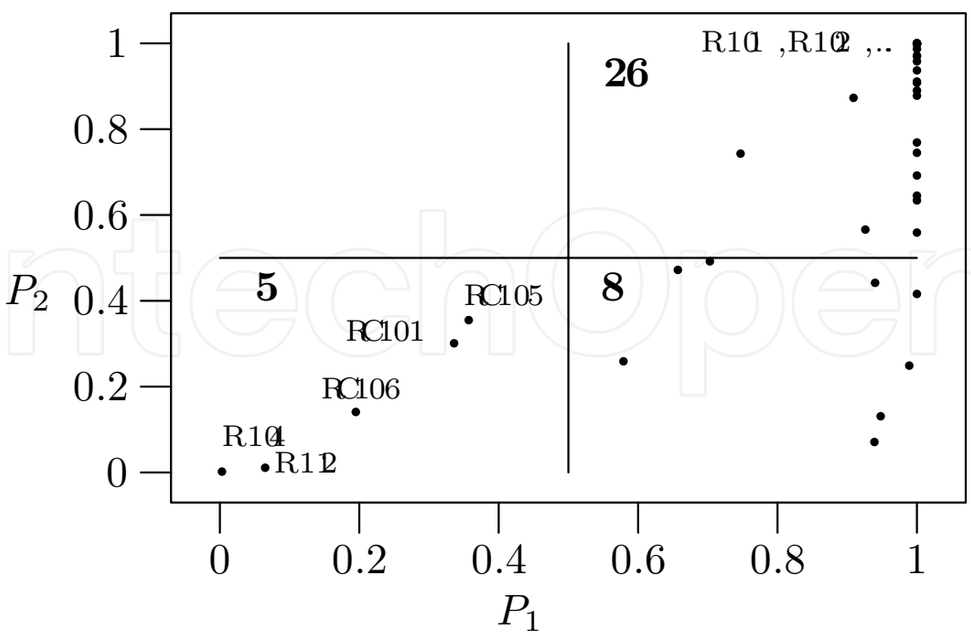

Fig. 13. Difficulty of tests measured by probabilities P_1 and P_2 (1% approximate solution is desired)

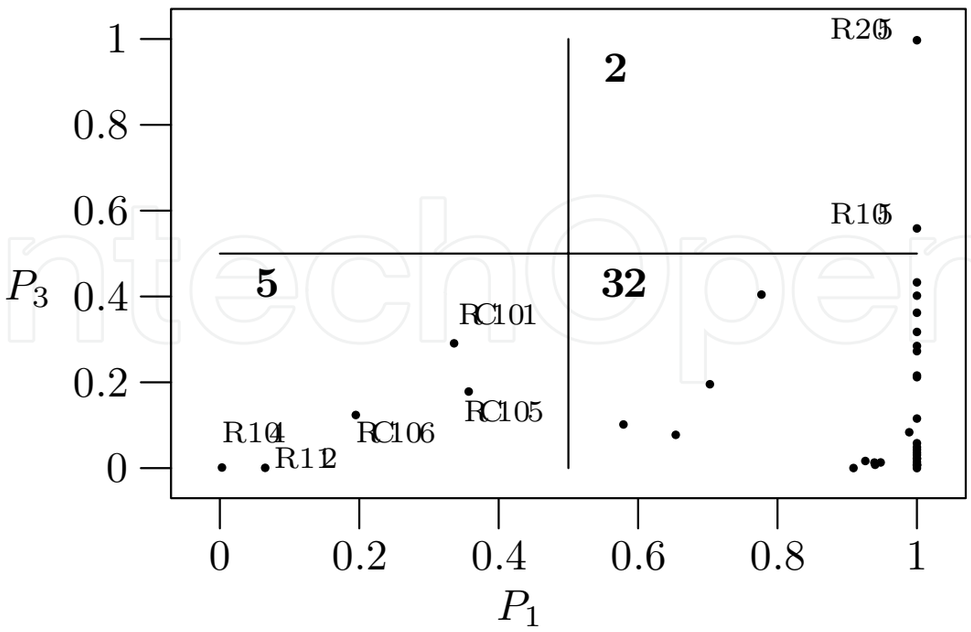

Fig. 14. Difficulty of tests measured by probabilities P_1 and P_3 (best solution is desired)

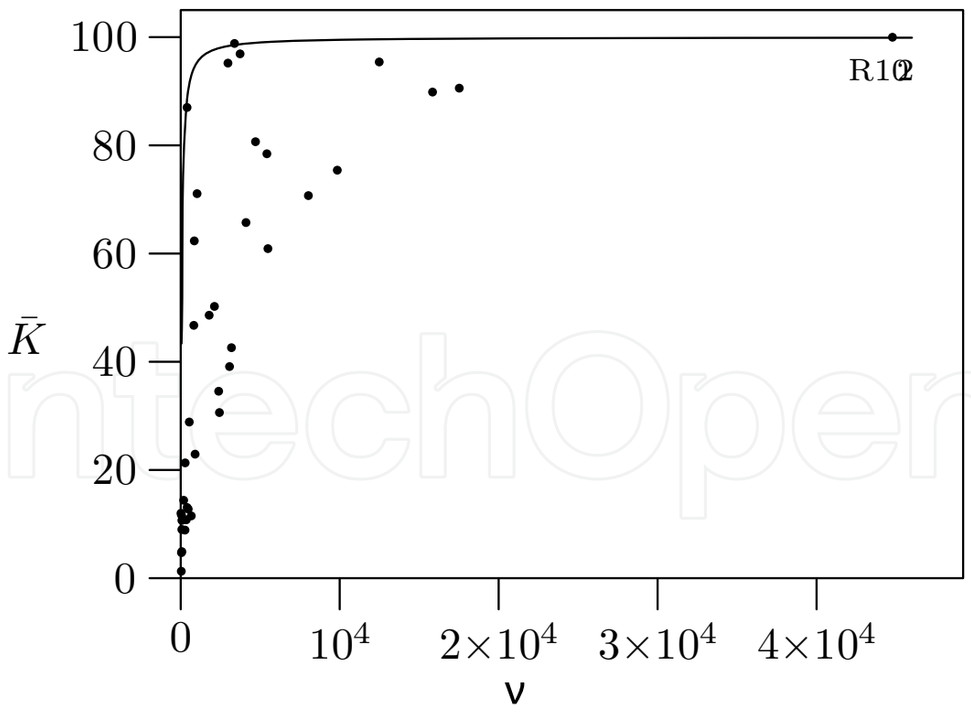

Fig. 15. Average number of distinct solutions $\bar{K}$ observed in a sample of series of $r = 100$ experiments vs. number of unveiled local optima ν for 39 Solomon's tests (for reference, solid line plots Eq. 14)

Test	P_1	P_2	P_3	$\bar{K}$	ν	Exp.
R104	0.003	0.002	0.001	12.00	15	41600
R112	0.065	0.011	$7 \cdot 10^{-4}$	87.00	399	9200
R110	0.579	0.259	0.102	60.91	5486	60600
R108	0.940	0.442	0.008	95.20	2971	4900
R111	0.654	0.472	0.078	65.73	4106	38700
R109	0.703	0.492	0.195	30.61	2435	61400
R107	0.926	0.566	0.017	80.67	4705	19400
R103	0.909	0.873	$2 \cdot 10^{-4}$	98.83	3384	4500
R106	1.000	0.908	0.008	90.57	17523	72200
R105	1.000	0.999	0.559	14.40	185	9100
R101	1.000	1.000	0.006	95.41	12488	56000
R102	1.000	1.000	$2 \cdot 10^{-5}$	99.98	44773	48600
R211	0.939	0.071	0.013	71.07	1028	4700
R207	0.989	0.249	0.084	78.44	5420	23300
R210	1.000	0.416	0.058	70.71	8031	68400
R203	1.000	0.559	0.006	96.91	3734	5300
R204	1.000	0.745	0.216	48.60	1789	10400
R208	1.000	0.769	0.008	75.41	9850	71100
R202	1.000	0.878	0.402	39.11	3070	37200
R209	1.000	0.970	0.433	21.31	279	4200
R206	1.000	0.987	0.049	62.34	854	4400
R205	1.000	0.998	0.997	1.28	36	36500
R201	1.000	1.000	0.317	10.69	72	4500
RC106	0.195	0.141	0.124	11.68	45	22700
RC101	0.336	0.300	0.291	4.91	70	33300
RC105	0.357	0.355	0.178	9.00	69	8900
RC108	1.000	0.634	0.285	42.60	3192	46800
RC104	1.000	0.692	0.031	89.85	15842	40100
RC102	0.777	0.743	0.404	11.51	664	76800
RC103	1.000	1.000	0.022	46.73	823	17700
RC107	1.000	1.000	0.036	4.72	46	15600
RC202	0.948	0.131	0.013	34.55	2387	56000
RC203	1.000	0.645	0.043	50.23	2121	25600
RC206	1.000	0.890	0.273	10.79	351	27800
RC205	1.000	0.911	0.115	22.92	904	42100
RC207	1.000	0.937	0.212	8.89	270	22000
RC201	1.000	0.958	0.362	12.77	472	50100
RC208	1.000	0.971	0.014	13.06	401	18700
RC204	1.000	0.999	0.022	28.86	538	68300

Table 4. Selected statistical measures of fitness landscapes P_1 – probability that solution has minimum number of routes, P_2 – probability that solution has minimum number of routes and minimum distance allowing 1% accuracy, P_3 – probability that solution is the best-known or best-achieved, $\bar{K}$ – average number of distinct solutions observed in a sample of series of $r = 100$ experiments, ν – number of local optima unveiled ($\bar{K}$ and ν are calculated over solutions with minimum number of routes; tests are ordered according to P_2 column)

out of as many frequently occurring edges within the best local minima (salesman's tours) as possible, because it is believed that if the "big valley" structure holds, then very good solutions are located near other good solutions.
Boese et al.'s approach cannot be directly used for the VRPTW problem, since its instances may not have the "big valley" structure (see Fig. 9 and 10). Moreover, an initial solution is not enhanced in simulated annealing into a better local minimum, like in greedy descent. It is rather a starting point for a random walk which ends up at some local optimum. The correlation between the quality of this optimum and the quality of the initial solution where the search began is quite weak.
However a shape of the fitness landscape provides some insight into the procedure which finds the set of neighbors $N(X)$ of a current solution X (see section 3). Figs. 6 and 7 indicate that the optimum solution can be a member of a "small valley" of solutions whose distances from all other solutions—measured by d—are large. Therefore in order to reach any solution in such an isolated "valley", the procedure finding the neighbors should create them through deep modifications of a current solution. This gives some guarantee that both close and distant neighbors will be constructed with equal probability.
The information concerning the ruggedness of the fitness landscape is used to establish the initial temperature of annealing in our parallel algorithm, what is a standard practice. Since the algorithm consists of two phases, the temperature $T_{0,f}$ is computed at the beginning of each phase ($f = 1,2$). The procedure finding a neighbor solution is executed a specified number of times and the average increase of solution cost Δ is computed. The initial temperature $T_{0,f}$ is fixed in such a way that the probability of worsening the solution cost by Δ in the first annealing step: $e^{-\Delta/T_{0,f}}$, is not larger than a predefined constant—in our case 0.01 (15). If this probability is too large then the convergence of simulated annealing is slow.

6. Concluding remarks

The fitness landscape is a useful notion in discrete optimization. It increases the understanding of processes which happen during optimization and helps to improve the performance of optimization algorithms. The experiments conducted for the VRPTW benchmarking tests by Solomon showed that the optimum solution can be located inside a "small valley" placed far away from the "big valley" containing the predominant number of solutions. In order to be able to find such an optimum one should assure that among the neighbors of a current solution built during an optimization process, there are not only the close neighbor solutions but also the distant ones. At the beginning of the process of simulated annealing the initial value of the temperature has to be fixed. It is usually done by taking into account the degree of ruggedness of the fitness landscape of a problem instance being solved. Statistical measures of the fitness landscape can be helpful in establishing the difficulty of instances of the problem. The analysis of this difficulty has several facets. One may ask how hard is to find the exact solution to the problem. In this case the key role plays the number of local optima occurring in the landscape. This number can be estimated by detecting distinct solutions in a series of experiments. The larger is the numer of these solutions, the more local optima are present in the landscape, and the problem instance is harder to solve. If one wants to solve the problem with some accuracy, then the smoothness of the landscape is crucial. An indicator here can be the value of $\bar{\tau} = (\bar{y} - y_{\min})/y_{\min}$ which exhibits the shift of the average cost $\bar{y}$ of solutions from $y_{\min}$ attained by solving the problem repeatedly. For two-objective minimization problems, like the VRPTW, one can ask what are the probabilities that in a final solution produced by an optimization algorithm both objective functions are minimized, or

stay within some accuracy limits. For example, we found that among the VRPTW tests these probabilities are the smallest for test R104, and the largest for test R205. Last but not least, the amenability of the problem and its instances for parallelization can be investigated. If the simulated annealing paradigm is used, then shortening the parallel execution time in order to get speedup, decreases the chains of steps of free exploration of the solution space carried out by processes. However short chains cause deterioration of quality of search results, because the convergence of simulated annealing is relatively slow. This difficulty can be alleviated by making processes co-operate. For this goal a suitable scheme of co-operation and its frequency are to be devised. It follows from our experiments that solving most of the VRPTW tests can be accelerated by using up to 20 processes. However for some tests (group III, see subsection 4.2) solutions of best accuracy are obtained for less than 20 processes. We believe that this issue requires further investigation.

7. Acknowledgments

We thank the following computing centers where the computations of our project were carried out: Academic Computer Centre in Gdansk TASK, Academic Computer Centre CYFRONET AGH, Kraków (computing grants 027/2004 and 069/2004), Poznań Supercomputing and Networking Center, Interdisciplinary Centre for Mathematical and Computational Modelling, Warsaw University (computing grant G27-9), Wrocław Centre for Networking and Supercomputing (computing grant 04/97). The research of this project was supported by the Minister of Science and Higher Education grant No 3177/B/T02/2008/35.

8. References

[1] Aarts, E.H.L., and Korst, J.H.M., Simulated annealing and Boltzmann machines, Wiley, Chichester, 1989.
[2] Aarts, E.H.L., and van Laarhoven, P.J.M., Simulated annealing: Theory and applications, Wiley, New York, 1987.
[3] Abramson, D., Constructing school timetables using simulated annealing: sequential and parallel algorithms, Man. Sci. 37, (1991), 98–113.
[4] Arbelaitz, O., Rodriguez, C., and Zamakola, I., Low cost parallel solutions for the VRPTW optimization problem, Proceedings of the International Conference on Parallel Processing Workshops, (2001).
[5] Azencott, R., Parallel simulated annealing: An overview of basic techniques, in Azencott, R. (Ed.), Simulated annealing. Parallelization techniques, J. Wiley, NY, (1992), 37–46.
[6] Azencott, R., and Graffigne, C., Parallel annealing by periodically interacting multiple searches: Acceleration rates, In: Azencott, R. (Ed.), Simulated annealing. Parallelization techniques, J. Wiley, NY, (1992), 81–90.
[7] K. D. Boese, A. B. Kahng, and S. Muddu, A new multi-start technique for combinatorial global optimization, Operations Research Letters 16, (1994), 101–113.
[8] Boissin, N., and Lutton, J.-L., A parallel simulated annealing algorithm, Parallel Computing 19, (1993), 859–872.
[9] Catoni, O., Grandes déviations et décroissance de la température dans les algorithmes de recuit simulé, C. R. Ac. Sci. Paris, Ser. I, 307 (1988), 535-538.
[10] Cěrny, V., A thermodynamical approach to the travelling salesman problem: an efficient simulation algorithm, J. of Optimization Theory and Applic. 45, (1985), 41-55.

[11] Czech, Z.J., and Czarnas, P., A parallel simulated annealing for the vehicle routing problem with time windows, Proc. 10th Euromicro Workshop on Parallel, Distributed and Network-based Processing, Canary Islands, Spain, (January, 2002), 376–383.
[12] Czarnas, P., Czech, Z.J.,and Gocyla, P., Parallel simulated annealing for bicriterion optimization problems, Proc. of the 5th International Conference on Parallel Processing and Applied Mathematics (PPAM 2003), (September 7–10, 2003), Czestochowa, Poland, Springer LNCS 3019/2004, 233–240.
[13] Czech, Z.J., and Wieczorek, B., Solving bicriterion optimization problems by parallel simulated annealing, Proc. of the 14th Euromicro Conference on Parallel, Distributed and Network-based Processing, (February 15–17, 2006), Montbéliard-Sochaux, France, 7–14 (IEEE Conference Publishing Services).
[14] Czech, Z.J., and Wieczorek, B., Frequency of co-operation of parallel simulated annealing processes, Proc. of the 6th Intern. Conf. on Parallel Processing and Applied Mathematics (PPAM 2005), (September 11–14, 2005), Poznań, Poland, Springer LNCS 3911/2006, 43–50.
[15] Czech, Z.J., Speeding up sequential annealing by parallelization, Proc. of the International Conference on Parallel Computing in Electrical Engineering, PARELEC 2006, (September 13–17, 2006), Bialystok, Poland, 349–354 (IEEE Conference Publishing Services).
[16] Czech, Z.J., Co-operation of processes in parallel simulated annealing, Proc. of the 18th IASTED International Conference on Parallel and Distributed Computing and Systems (PDCS 2007), (November 13-15, 2006), Dallas, Texas, USA, 401–406.
[17] Debudaj-Grabysz, A., and Czech, Z.J., Theoretical and practical issues of parallel simulated annealing, Proc. of the 7th Intern. Conf. on Parallel Processing and Applied Mathematics (PPAM 2007), (September 9–12, 2007), Gdansk, Poland, Springer LNCS 4967/2008, 189–198.
[18] Czech, Z.J., Statistical measures of a fitness landscape for the vehicle routing problem, Proc. of the 22nd IEEE International Parallel and Distributed Symposium (IPDPS 2008), 11th Intern. Workshop on Nature Inspired Distributed Computing (NIDISC 2008), (April 14–18, 2008), Miami, Florida, USA, 1–8.
[19] Czech, Z. J., Mikanik, W., and Skinderowicz, R., Implementing a parallel simulated annealing algorithm, (2009), 1–11 (submitted).
[20] Greening, D.R., Parallel simulated annealing techniques, Physica D 42, (1990), 293–306.
[21] Hajek, B., Cooling schedules for optimal annealing, Mathematics of Operations Research 13, 2, (1988), 311–329.
[22] W. Hordijk and P. F. Stadler, Amplitude spectra of fitness landscapes, J. Complex Systems 1, (1998), 39–66.
[23] Kirkpatrick, S., Gellat, C.D., and Vecchi, M.P., Optimization by simulated annealing, Science 220, (1983), 671-680.
[24] Lenstra, J., and Rinnooy Kan, A., Complexity of vehicle routing and scheduling problems, Networks 11, (1981), 221–227.
[25] Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., and Teller, E., Equation of state calculation by fast computing machines, Journ. of Chem. Phys. 21, (1953), 1087-1091.
[26] Onbaşoğlu, E.,and Özdamar, L., Parallel simulated annealing algorithms in global optimization, Journal of Global Optimization 19: 27–50, 2001.

[27] Reeves, C.R., (Ed.) Modern Heuristic Techniques for Combinatorial Problems, McGraw-Hill, London, 1995.
[28] Reeves, C. R., Direct statistical estimation of GA landscape properties, in: Foundations of Genetic Algorithms 6, FOGA 2000, Martin, W. N., Spears, W. M. (Eds.), Morgan Kaufmann Publishers, (2000), 91–107.
[29] Reeves, C. R. and Rowe, J. E., Genetic algorithms: Principles and Perspective. A Guide to GA Theory, Kluwer Academic Publishers, (2003), 231–263.
[30] C. R. Reeves, Fitness landscapes, In: Search methodologies. Introductory Tutorials in Optimization and Decision Support Techniques, Burke, E. K. and Kendall, G. (Eds.), 587–610, Springer-Verlag, Berlin, (2005).
[31] C. M. Reidys and P. F. Stadler, Combinatorial landscapes, SIAM Rev. 44, (2002), 3–54.
[32] P. F. Stadler, Towards a theory of landscapes, In: Complex Systems and Binary Networks, Lopéz-Peña, R., Capovilla, R., García-Pelayo, R., Waelbroeck, H., and Zurteche, F. (Eds.), 77–163, Springer-Verlag, Berlin, (1995).
[33] Solomon, M.M., Algorithms for the vehicle routing and scheduling problems with time window constraints, Operations Research 35, (1987), 254–265, see also http://w.cba.neu.edu/˜msolomon/problems.htm.
[34] Toth, P., and Vigo, D., (Eds.), The vehicle routing problem, SIAM Monographs on Discrete Mathematics and Applications, Philadelphia, PA, 2002.
[35] Verhoeven, M.G.A., and Aarts, E.H.L., Parallel local search techniques, Journal of Heuristics 1, (1996), 43–65.
[36] Weinberger, E. D., Correlated and uncorrelated landscapes and how to tell the difference, Biol. Cybernet. 63, (1990), 325–336.
[37] S. Wright, The role of mutation, inbreeding, crossbreeding and selection in evolution, In: 6th Int. Congress on Genetics 1, Jones, D. (Ed.), (1932), 356–366.

IntechOpen

IntechOpen

IntechOpen

Parallel and Distributed Computing

Edited by Alberto Ros

ISBN 978-953-307-057-5

Hard cover, 290 pages

Publisher InTech

Published online 01, January, 2010

Published in print edition January, 2010

The 14 chapters presented in this book cover a wide variety of representative works ranging from hardware design to application development. Particularly, the topics that are addressed are programmable and reconfigurable devices and systems, dependability of GPUs (General Purpose Units), network topologies, cache coherence protocols, resource allocation, scheduling algorithms, peertopeer networks, largescale network simulation, and parallel routines and algorithms. In this way, the articles included in this book constitute an excellent reference for engineers and researchers who have particular interests in each of these topics in parallel and distributed computing.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Zbigniew J. Czech (2010). A Parallel Simulated Annealing Algorithm as a Tool for Fitness Landscapes Exploration, Parallel and Distributed Computing, Alberto Ros (Ed.), ISBN: 978-953-307-057-5, InTech, Available from: http://www.intechopen.com/books/parallel-and-distributed-computing/a-parallel-simulated-annealing-algorithm-as-a-tool-for-fitness-landscapes-exploration

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2010 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License, which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen