Solving the Real-life Vehicle Routing Problem with Time Windows Using Self Organizing Migrating Algorithm[1]

*Zuzana ČIČKOVÁ – Ivan BREZINA – Juraj PEKÁR**

Abstract

This article deals with self-organizing migrating algorithm (SOMA) for solving the vehicle routing problem with time windows (VRPTW). Our interest in VRPTW arises from a real-life distribution problem in one of the regions of Slovakia (individual customers' commodities delivering times were restricted by their available service time), where the previous distribution was realized on the base of solution derived with heuristic Clarke & Wright's savings algorithm with time windows. The importance of that problem follows from many practical applications as well as from its computational complexity, therefore the use of optimization techniques seems to be relatively complicated, and nowadays many researchers turn their attention to applications of alternative computational techniques that are inspired by evolutionary biology. The obtained solution allows reducing the total time needed by 16.2%. The presented approach could be used also for solving various economic problems with time restrictions in the field of distribution.

Keywords*: evolutionary algorithms, heuristics, self-organizing migrating algorithm, vehicle routing problem with time windows*

JEL Classification: C02, C61, C63, L91

Introduction

Nowadays, the threat of depletion of non-renewable resources which are necessary for car propulsion is the reason for development and utilization of instruments that take advantage of the optimization. The efficiency may be increased

* Zuzana ČIČKOVÁ – Ivan BREZINA – Juraj PEKÁR, University of Economics in Bratislava, Faculty of Economic Informatics, Department of Operations Research and Econometrics, Dolnozemská cesta 1, 852 35 Bratislava 5, Slovak Republic, e-mail: cickova@euba.sk; brezina@euba.sk; pekar@euba.sk

[1] This paper is supported by the Grant Agency of Slovak Republic – VEGA, grant No. 1/0104/12 *Modeling supply chain pricing policy in a competitive environment*.

by force of quantitative approaches that are aimed at optimization of physical distribution of the commodities. Related optimization problems are routing and scheduling problems (e.g. shortest path problem, travelling salesman problem, vehicle routing problem, etc.). The management of physical distribution of commodities is interesting not only for its practical relevance in the fields of transportation, distribution and logistics, but also for theoretical research, because a lot of related problems belong to the NP-hard problems.

This article deals with the real world routing problem that has appeared in Slovakia. The distribution centre was situated in the city of Banská Bystrica, and the distribution was made to 29 municipalities, in which stores with a daily demand for certain number of crates of merchandise were situated. Also a desired "window" of delivery of goods was known for each municipality, i.e. the start and end time when it was possible to realize the supply of goods. The goal was to determine how many vehicles a day must be used so that the demand of all stores must be met and so that the travelled distance is as low as possible.

The routing and scheduling problems generally involve the assignment of vehicle (fleet of vehicle) to trips such that corresponding costs are as low as possible. The vehicle routing problem (VRP) is one of the most intensively studied problems in optimization. The standard VRP is the generalization of travelling salesman problem, where we consider that the capacity of vehicle (fleet of vehicles) is limited and that the non-negative demands of nodes (customers) are known. This problem consists in designing the optimal set of routes for a vehicle in order to serve a given set of customers. Each customer has a certain demand and each vehicle has a certain capacity (the orders of customers have to be delivered in full, partial fulfilment of orders is not allowed), and they are located in a certain depot. Further on, there exist a matrix **C** that represents the minimum distance (length, cost, time) between all the pairs of customers and also between the customers and the depot. The goal is to find optimal vehicle routes (usually minimum distance). The routes must be designed in such a way that each customer is visited only once by exactly one vehicle; all routes start and end at the depot and the total demands of all customers on one particular route must not exceed the capacity of the vehicle.

The practical problems of physical distribution often include the need to respect the time restriction. Frequently we consider time restrictions that are a consequence of the earliest possible time of service, the latest possible time of service or the need to serve during the given time interval. The above mentioned terms are known as time windows and the corresponding problem is known as vehicle routing problems with time windows (VRPTW). If it is necessary to consider only the earliest possible time of service or the last possible time of service, the problem is known as the problem with soft delivery time windows, if we are dealing with time interval with given lower and upper limit, those problems are known as problems with hard delivery time windows, e.g. Desrosiers et al. (1995). Another description of soft window can be found e.g. in Russell and Urban (2008), where the violation of time restriction is allowed, although incurring some cost. Garey and Johnson (1979) have shown that VRP is NP-hard problem and the NP-harness of the VRPTW demonstrated Savelsbergh (1985), and in this work he also proposed the arc interchange heuristic to solve it. Generally, NP-hard problems could be solved by exact methods (branch and bound algorithms, cutting plane method, etc. (e.g. Bard, Kontoravdis and Yu, 2002; Achuthan, Caccetta and Hill, 2003), that work well for small-size problem, or heuristics with more or less success. The heuristics could be classified into two general groups: classical heuristics and metaheuristics. In general, metaheuristics consist of general search procedures whose principles allow them to escape local optimality with the help of built-in stochastic component. The significant part of metaheuristics is the group of evolutionary techniques. Nowadays, evolutionary algorithms are considered to be effective tools that can be used to search for solutions of optimization problems. Significant advantage over traditional methods is that they are designed to find global extremes and that no auxiliary information, such as convexity, gradients etc. are needed.

Popularity of applications of various metaheuristics for solving the routing and scheduling problems is documented in numerous articles. From all, we can mention the application of self-organizing migration algorithm to vehicle routing problem (Čičková and Brezina, 2008), differential evolution to travelling salesman problem (Peško, 2006), ant colony optimization to vehicle routing problem (Yu, Yang and Yao, 2009), ant colony optimization to vehicle routing problem with soft time windows and stochastic travel times (Russell and Urban, 2008) or to open vehicle routing problem (Li, Tian and Leung, 2008), Tabu search to the heterogeneous vehicle routing problem (Tarantilis, Zachariadis and Kiranoudis, 2008) or to open vehicle routing problem (Derigs and Reuter, 2009).

A lot of work has been done to develop both exact and heuristic algorithm for VRPTW. Branch and bound algorithm and application of dynamic programming are developed in Christofides, Mingozzi and Toth (1981a; 1981b) or in Baker (1983), where the authors reported problems up to 50 vertices, but current research still focuses on heuristic approaches due to the computational complexity of VRPTW. Generally, heuristic approaches can be divided into two areas: classical heuristics (that belong to the class of deterministic methods) or metaheuristics (that belong to the class of combined methods that include deterministic and also stochastic components). For example, well-known Solomon's insertion heuristics that is presented (Solomon, 1987) belongs to the classical heuristics. Solomon proposes a benchmark problem set for the vehicle routing problem with time windows and conducts a computational study of several heuristic algorithms using the set. While Solomon's insertion heuristic builds a route in a serial manner, in Potvin and Rousseau's (1993) the routes are built in parallel. Weigel and Cao (1999) present a case study of application of VRPTW algorithms for sears home delivery problem and technician dispatching problem with the use of a cluster-first-route-second method and an algorithm called multiple-insertion. Campbell and Savelsbergh (2005) introduced optimization models of the home delivery problem, which is more closely related to real-world applications. In Poot, Kant and Wagelmans (2002) several non-traditional constraints (among others also multiple time windows) are presented and the authors propose a savings-based method for solving corresponding problems. Tung and Pinnoi (2000) modified Solomon's insertion algorithm and applied it to a waste collection problem with time constraints. Also the popularity of metaheuristics utilization for solving the VRPTW is documented in a number of works. From all we mention the following: Rochat and Taillard (1995) present a probabilistic Tabu search method. The use of simulated annealing and Tabu search for the VRPTW is given in Thangiah (1999), the particle swarm optimization is presented in Ai and Kachitvichyanukul (2009), ant colony optimization is used in Russell and Urban (2008). Illustration of using neural network to this problem is given in Zhang and Tang (2007). Also the use of genetic algorithm remains popular as is documented e.g. Chang and Chen (2007), Ghoseiri and Ghannadpour (2010), Ombuki, Ross and Hanshar (2006), etc.

1. Problem Formulation

The classical version of VRPTW is defined on graph

$$G = (V_c \cup V_d, A)$$

where

$V_c = \{v_1, v_2 \ldots v_n\}$ – represents set of customers,

$V_d = \{v_0\}$ – represents the origin,

$A = \{(v_i, v_j): v_i, v_j \in V_c \cup V_d, i \neq j\}$ – the arc set of G.

A cost or travel time c_{ij} is associated with every arc of the graph. A fleet of r vehicles of the same capacity V is located at v_0. Denote in our case:

$N = \{1, 2 ... n\}$	– set of served nodes; n – number of nodes except the origin;
$N_0 = N \cup \{0\}$	– set of all nodes; 0 – origin;
$H = \{1, 2 ... r\}$	– set of vehicles; r – number of vehicles;
$K = \{1, 2 ... 2n+1\}$	– order of arc in sequence of h-th vehicle $h \in H$;
$2n + 1$	– maximal number of arcs in a sequence.

Each customer has a certain demand (g_i, $i \in N$) and a service duration (t_i, $i \in N$). Further on, there is the known time window of each customer: as the earliest possible start of service in different nodes (f_i, $i \in N$) and the last acceptable time of service in different nodes (l_i, $i \in N$). The demand is fulfilled from initial node ($i = 0$) – origin. The goal is to determine the minimal number of vehicles so that the total travelled time or distance is as low as possible (we suppose that there is known the shortest time distance between all nodes c_{ij}, $i, j \in N_0$) with respect to the following restrictions: the origin represents initial node and also the final node of every route, from the origin the demand g_i, $i \in N$ of all the other nodes is met within their time windows (the earliest possible start of service f_i, $i \in N$, the last acceptable time of service l_i, $i \in N$, each node (except central node) is visited exactly once and total demand on route must not exceed the capacity of the vehicle (V). The total time of the route of a vehicle could not exceed the given time (T).

Further on, we suppose that the service time at the centre is set to t_0 (this time is added to the total time only in case that the vehicle returns to the origin due to violation of capacity limit, and it is able to serve the nodes on the next route). The model takes into account the waiting time so that the vehicle is allowed to wait for service if it arrives before the earliest possible start of service.

Decision variables:

x_{ijkh} – a binary variables equal to one if the edge between node i and node j is used by vehicle h as k-th in sequence and zero otherwise, where $i, j \in N_0$, $h \in H$, $k \in K$;

w_j – non-negative variables that indicates waiting time at node j, $j \in N$;

b_j – variables that indicates real starting time of service at node j, $j \in N_0$;

u_j – variables that represents remaining capacity of vehicle at the node j, $j \in N_0$.

Mathematical formulation of the model is given below:

$$\min \sum_{i \in N_0} \sum_{j \in N_0} \sum_{h \in H} \sum_{k \in K} c_{ij} x_{ijkh} + \sum_{i \in N} w_i + \sum_{i \in N} t_i + \sum_{j \in N} \sum_{k \in K-1} \sum_{h \in H} t_0 x_{0jkh} + M \sum_{i \in N} \sum_{h \in H} x_{0j1h} \quad (1)$$

Subject to:

$$\sum_{j \in N_0} \sum_{k \in K} \sum_{h \in H} x_{ijkh} = 1, \; i \in N, \; i \neq j \quad (2)$$

$$\sum_{i \in N_0} \sum_{k \in K} \sum_{h \in H} x_{ijkh} = 1, \; j \in N, \; i \neq j \quad (3)$$

$$\sum_{i \in N_0} x_{ijkh} = \sum_{i \in N_0} x_{ji \; k+1 \; h}, \; j \in N, \; h \in H, \; k \in K, \; i \neq j \quad (4)$$

$$b_i + t_i + w_j + c_{ij} - b_j \;\; \cdot x_{ijkh} = 0, \; i, j \in N, \; h \in H, \; k \in K - \; 1 \;, \; i \neq j \quad (5)$$

$$b_i + t_i + w_j + c_{i0} + t_0 + c_{0j} - b_j \;\; \cdot x_{i0kh} \cdot x_{0j \; k+1 \; h} = 0, \; i, j \in N, \; h \in H, \; k \in K, \; i \neq j \quad (6)$$

$$w_j + c_{0j} - b_j \;\; \cdot x_{0j1h} = 0, \; j \in N, \; h \in H \quad (7)$$

$$b_j + t_j + d_{0j} \;\; \cdot x_{j0kh} \leq T, \; j \in N, \; h \in H, \; k \in K \quad (8)$$

$$u_i + g_j - u_j \;\; \cdot x_{ijkh} = 0, i \in N_0, \; j \in N, \; h \in H, \; k \in K, \; i \neq j \quad (9)$$

$$\sum_{i \in N} x_{0j \; k+1 \; h} \leq \sum_{i \in N} x_{j0kh}, \; j \in N, h \in H, \; k \in K \quad (10)$$

$$\sum_{j \in N} x_{0j1h} \leq 1, \; h \in H \quad (11)$$

$$x_{0j1h} \geq x_{0jkh} \quad j \in N, \; h \in H, \; k \in H \quad (12)$$

$$u_0 = 0 \quad (13)$$

$$u_i \leq V, \; i \in N \quad (14)$$

$$b_0 = 0 \quad (15)$$

$$f_i \leq b_i, \; i \in N \quad 16)$$

$$b_i + t_i \leq l_i, \; i \in N \quad (17)$$

$$x_{ijh} \in \; 0, 1 \;, \; i, j \in N_0, \; h \in H \quad (18)$$

$$w_i \geq 0, u_i \geq 0 \;\; i \in N \quad (19)$$

The objective function (1) minimizes the total duration travelled and also the number of vehicles (*M* is a big positive number). Equations (2) and (3) ensure that a vehicle leaves each node and vehicle enters each node except the origin exactly ones. Equation (4) and (10) ensures the connectivity of the route. Equation

(5) calculates the real starting time of service for the next node on the route (except the origin) on the base of previous node. Equation (6) ensures the calculation of starting time of service for the next node on the route, in case that the route goes through origin. Equation (7) calculates the real starting time of service of the first node on the route of the vehicle. Equation (8) ensures that the total vehicle time travelled must not exceed the given time (T). Equations (9), (13) and (14) ensure that all demands on the route must not exceed the capacity of the vehicle. Equations (11) and (12) ensure that each route starts at the origin exactly once. Equations (15), (16) and (17) ensure that the time windows of all nodes on the route are met.

2. Self-Organizing Migrating Algorithm

Self-organizing migrating algorithm (SOMA) belongs to the class of optimization techniques. It can be classified as evolutionary algorithm, despite the fact that no new individuals are created during simulations, because it is based on the self-organizing behaviour of individuals in a social environment (e.g. a herd of animals looking for food, a group of animals such as wolves or other predators may be a good example). If they are looking for food, they usually cooperate and compete so that if one member of the group is successful, the other animals of the group change their trajectories towards the most successful member. It is repeated until all members meet around one food source. SOMA, as well other evolutionary algorithms, is working on a population of individuals (*np* – number of individuals in the population). Each individual represents one candidate solution for the given problem, i.e. a set of arguments of objective function (d – number of arguments). Associated with each individual is also the *fitness*, which represents the relevant value of objective function. A population is usually randomly initialized at the beginning of evolutionary process. And the first iteration (1st migration loop) begins. So, an individual with the highest fitness (called Leader) is chosen and the others begin to move towards him. Geometrically speaking, sequences of the new positions are generated in the d dimensional hyperplane (but some parameters of an individual could be frozen depending on one of the control parameters *prt*). At the end of jumping, the individual returns to the position with the highest fitness and this is its position for the next migration loop. The detailed steps how SOMA actually works are described below.

SOMA was introduced by Zelinka in 1999 and has been successfully tested on various types of test functions (e.g. Rosenbrock's saddle, De Jong functions, Schwefel's function etc.). Self-organizing migration algorithm, as well as other evolutionary techniques, works well on solving non-constrained problems that contain continuous variables, but nowadays there were developed few approaches that involve the solving of constrained problems with integer or binary variables.

The principle of basic version of SOMA[2] can be described by following pseudocode:

BEGIN
SETTING of control parameters;
INITALIZATION of population;
EVALUATION of each individual;
  WHILE (*STOPPING CRITERION* is not satisfied) **DO (***MIGRATION LOOPS***)**
    SELECT leader
    FOR (each individual of the population except leader) **DO**
      JUMPING individual toward the leader
      EVALUATE fitness of individual after each jump
      MOVE individual on the position with the best fitness
    ENDFOR
  ENDWHILE
  EVALUATE process of calculating
END

The steps of the algorithm can be briefly summarized as follows (according to Onwubolu and Babu (2004); Zelinka (2002), where it is also possible to find recommended values for different parameters):

Setting of the control parameters. A disadvantage of SOMA, as well as other evolutionary techniques, is that the efficiency of SOMA has a dependence on the setting of control parameters. The control parameters are described below:

- *d* – dimensionality; number of parameters of individual.
- *np* – population size; number of individuals in population.
- *mig* – migrations; represent the maximum number of iteration.
- *mass* – path length, $mass \in \langle 1.1, 3 \rangle$; represents how far an individual stops behind the leader, recommended value is 3.
- *step* – $step \in \langle 0.11,\ mass \rangle$; defines the granularity with what the search space is sampled, recommended value is 0.11.
- *prt* – perturbation, $prt \in \langle 0, 1 \rangle$; determines whether an individual travels directly towards the leader or not.

Initialization. The population is usually randomly initialized at the beginning of evolutionary process and the *fitness* is calculated for each individual.

The test of stopping condition. In its canonical form, the most used stopping criterion is to reach the maximal number of migration loops (represented by parameter *mig*).

[2] In literature (Onwubolu and Babu, 2004) could be found also another variations of SOMA.

Migration loop. SOMA works in migration loops (*ml*), where no new individuals are created and only existing ones are moving over the search space. In each migration loop, the individual with the highest fitness – leader (x_L^{ml}) is chosen and the other individuals begin to jump towards the leader according to the *step*:

$$x_{i,j}^{ml+1} = x_{i,j,start}^{ml} + (x_{L,j}^{ml} - x_{i,j,start}^{ml}).t.prt_j\,,\ t \in \langle 0,\ \text{by } step \text{ to},\ mass \rangle$$

$$i = 1, 2\ldots np, j = 1, 2\ldots d$$

where *np* represents the number of individuals in population and *d* represents the number of parameter of individual.

Each individual is evaluated after each jump with the *fitness* (relevant value of objective function f_c). The jumping continues, until new position defined by the *mass* is reached. Then the individual returns to that position, where the best fitness was found:

$$x_{i,j}^{ml+1} = \min \quad f_c(x_{i,j}^{ml}),\ f_c(x_{i,j,start}^{ml}) \quad , \quad i = 1, 2\ldots np, j = 1, 2\ldots d$$

Before individual begins to jump, a random number for each individual component is generated and is compared with *prt*. If the random number is equal or less then *prt*, then the associated component of the individual is set to 0. Let *k* be the number of components that are equal to 0. Hence, the individual moves in $d - k$ dimensional subspace. This fact establishes a higher robustness of the algorithm. Vector **prt** is created before the individual begins to move in the search space.

$$prt_j = \begin{cases} 1, \text{ if } rand_j \langle 0,1 \rangle > prt \\ \\ 0, \text{ otherwise} \end{cases} \qquad j = 1, 2 \ldots d$$

So that process continues in each generation for all individuals except the leader. The result is a new generation with the same number of individuals.

Evaluation. The whole process of reproduction continues until the last (users specified) number of generations is reached. The value of the best individual from each generation is reflected to *history vector*, which shows the progression of an evolutionary process.

3. Self-Organizing Migrating Algorithm for the VRPTW

SOMA is an algorithm that works well in the case of non-constrained problems with continuous variables, so if we want to apply it for solving NP-hard problems, it is necessary to consider the following factors:

Selection of an appropriate representation of individual. We chose a natural representation of individual that is particularly known from genetic algorithms (where it has been often used with success by solving much known travelling salesman problem). In response to the vehicle routing problem, each node (city) except centre node (depot) is assigned with integer from 1 to *d* (*d* represents the number of nodes except depot), which represents corresponding node in individual. Each individual is then represented by *d*-dimensional vector of integers, representing the sequence of visiting of the nodes. Then, the initial population $P^{(0)}$ is generated as follows:

$$P^{0} = x_{i,j}^{0} = \text{randperm}\; d \qquad i = 1,\ 2 ... np\;\; j = 1,\ 2 ... d$$

where the function randperm(*d*) ensure the establishment of a random permutation of *d* integers, so it is the random permutation of the sequence of nodes. Each individual in the population is also assigned with its fitness that represents total cost of the route.

Formulation of objective function. The computation of objective function value of an individual is realized in two steps with respect to the following facts:

- The total demand on the route must not exceed the capacity of vehicle.
- All nodes except origin are served within their known time window that is given by the range (earliest possible time of service, the last possible time of service).
- The information about its service time is known for each node except origin.
- It is a known due for reverse arrival to the origin, which means the total time on one route must not exceed this restriction.

Goal: to minimize number of vehicles, so that the total distance travelled by vehicles is as low as possible.

Transformation the parameters of individual to the real numbers. Because SOMA was originally designed to solve problems with continuous variables and the used natural representation consists of integer variables, it is desirable to transform integers to real numbers. The used method for transformation was presented in (Onwubolu and Babu, 2004) for solving travelling salesman problem.

Transformation of unfeasible solutions. The use of SOMA for VRPTW does not require the formation of feasible solution in case of natural representation of individual; therefore it is necessary to choose an appropriate method of transformation of the unfeasible solutions (see Čičková, Brezina and Pekár, 2008).

The algorithms were implemented in MATLAB 7.1. Two functions were created: SOMA adapted for solving VRPTW and the function for calculation of objective function value. All the experiments were run on PC INTEL(R) Core(TM) 2 CPU, E8500 @ 3.16 GHz, 3.25 GB RAM under Windows XP.

Setting of the control parameters of SOMA. The control parameters were set on the base of the article (Čičková, Brezina and Pekár, 2011), which describes the possibility of setting the parameters with the help some statistical methods e.g. Kruskal-Wallis test, Bartlett's test, Cochran-Hartley's test.

4. VRPTW Experiments

The efficiency of discovered set of control parameters (*step* = 0.9, *prt* = 0.7 and *f* = 200) that is based on the before mentioned article (Čičková, Brezina and Pekár, 2011) we also tested on the Solomon's VRPTW benchmark, which are 100-customer problem sets.[3] The problems are categorised into six classes, namely C1, C2, R1, R2, RC1 and RC2. Problems which fall into C categories are clustered data, meaning nodes are clustered either geographically or in terms of time windows. Problems from R categories are uniformly distributed data and those from RC categories are hybrid problems that have the features of both C and R categories. In addition, C1, R1 and RC1 problem sets have narrower time window for the depot, whereas C2, R2 and RC2 sets have wider time window for the depot. Because of the similarity structure of our above mentioned real-life problem, we tested the classes C2 (C201.50 – C209.50) and (RC201.50 – RC207.50). The control parameter *np* was set to 10.*d* (where *d* represents number of nodes) and the parameter *mig* was set to 5000. The results were compared with the known optimal solution. The percentage deviation was less than 7.5% for the C2 set and it was less than 9.8% for the RC2 set.

Based on these results it can be stated that SOMA is able to return the acceptable solution for the VRPTW and the identified values of the control parameters can be used for solving real-life vehicle routing problem with time windows that follows in the next section.

5. Real-life Vehicle Routing Problem with Time Windows

The problem deals about the real distribution scheduling in the region of Banská Bystrica in Slovakia. Distribution centre was situated in the city of Banská Bystrica and distribution was made to 29 municipalities, in which were situated the stores with a daily calling for certain number of crates of merchandise. Time distances in minutes between the centre and individual citizens themselves were known. And also "window" of delivery of merchandise was known for each municipality, i.e. start and end time when it was possible to realize the

[3] See <http://web.cba.neu.edu/~msolomon/problems.htm> (12. 12. 2011).

supply. Further on, it was also estimated service time in certain municipalities (unloading time of vehicle). The capacity of the available vehicles was set to 80 crates. The goal was to determine how many vehicles a day must be used so that demand of all stores must be met.

Input data: number of delivery points d = 29, distribution centre i = 0, the shortest time distance between all municipalities and between distribution centre and each municipality c_{ij} (i = 0, 1, 2... 29, j = 0, 1, 2... 29), service time at the centre t_0 = 30 min., operating times in different stores t_1, t_2... t_{29}, the earliest possible start of delivery of goods in different stores f_1, f_2... f_{29}, the last acceptable time of delivery of merchandise in different municipalities l_1, l_2... l_{29}, vehicle capacity V = 80, the demand of individual stores g_1, g_2... g_{29}. The whole transportation of crates had to be realized between 6.00 a.m. (f_i = 0, i = 1, 2... 29) and 9.00 a.m. (l_i = 180, i = 1, 2... 29). The goal was to minimize the total time and to determine the minimal number of vehicles, with respect the following restrictions: the origin 0 is the initial node and also the final node for each route, the demands of the supply nodes must be met within their time windows, the vehicle is allowed to wait for service when a vehicle arrives before the earliest limit, each supply node (except origin) is visited exactly once, total demand on route must not exceed the capacity of vehicle and the orders of customers have to be delivered in full.

The distribution was previously made on the basis of the solution that was derived with heuristic Clarke & Wright's savings algorithm with time windows[4] with the use of four vehicles and the total duration was 555.7 min. Individual routes were: Route 1: 0–8–22–27–15–0, duration 133.4; Route 2: 0–3–10–12–9–17–24–25–18–0, duration 134.3; Route 3: 0–1–6–19–2–11–28–21–26–29–0, duration 144.8; Route 4: 0–5–4–13–16–7–14–23–20–0, duration 143.2 min.
Our solution of presented problem is based on same principles as was mentioned above with the setting of control parameters (*mass*, *step* and *prt*) with *np* = 3 000 and *mig* = 5 000. The only difference is in formulation of objective function (fitness), where we consider that the final route must satisfy the following:

- If the vehicle arrives earlier than the lower bound of window, the vehicle is allowed to wait for service, and the waiting time is added to the total time of the corresponding route.
- The supply nodes are included in the same route only in case that the savings $s_{ij} > 0$, the computation of savings is based on heuristic Clarke & Wright's savings algorithm so that: $s_{ij} = c_{0j} + c_{i0} - c_{ij}$, $i, j = 1, 2... d$.
- If the vehicle returns to the origin in the case of:

[4] See <http://www.ise.ncsu.edu/kay/matlog/> (1. 2. 2012).

– The capacity of a vehicle is exceeded ⇒ the service time in the origin is added to the total time of the corresponding route (in practice, the next route will be realized with the same vehicle).
– Violating of the last possible service time, the objective function is penalized by penalty constant, which represents the fact that the next route is realized by another vehicle. The real duration of distribution is calculated by subtracting the total of these penalties.

The best result obtained from realized simulations allows the use only 3 vehicles with the total duration 465.76 min. The use of heuristic Clarke & Wright's savings algorithm with time windows returns the total time of distribution 555.7 min. with the need of 4 vehicles. Based on those results it can be stated that our approach allows decreasing the number of vehicles simultaneously with the savings in the total time (the saving is 16.2%).

The routes are described in Tables 1 – 4:

Route A: Customer sequence: 0; 20; 5; 4; 13; 7; 28; 11; 21; 26; 29; 0 (Banská Bystrica; Staré Hory; Riečka; Tajov; Králiky; Špania Dolina; Podkonice; Slovenská Ľupča; Lučatín; Medzibrod; Brusno; Banská Bystrica), total capacity 74, return due to upper time limit of a window, total time of route 154.65 minutes, objective value 154.65 minutes ⇒ the next route will be realized by another vehicle.

T a b l e 1

Route A

Number	**20**	**5**	**4**	**13**	**7**	**28**	**11**	**21**	**26**	**29**
	Staré Hory	Riečka	Tajov	Králiky	Š. Dolina	Podkonice	S. Ľupča	Lučatín	Medzibrod	Brusno
Demand q_i	9	4	10	3	4	7	15	7	4	11
Saving t_i	5	3	5	3	3	3	8	3	3	5
Earliest limit l_i	0	0	0	60	60	0	0	30	120	0
Latest limit u_i	180	120	180	180	120	180	180	150	180	180
Arrival	12.4	35.27	43.45	51.45	78.55	104.25	113.25	124.85	129.85	134.05
Start of service	12.4	35.27	43.45	60	78.55	104.25	113.25	124.85	129.85	134.05
End of service	17.4	38.27	48.45	63	81.55	107.25	121.25	127.85	132.85	139.05
Capacity	9	13	23	26	30	37	52	59	63	74

Source: Own compilation.

Route B: Customer sequence: 0; 23; 14; 0 (Banská Bystrica; Dolný Harmanec; Harmanec; Banská Bystrica), total capacity 29, return due to savings = 0, total time of route 38.30 minutes, objective value 192.95 minutes ⇒ the route C will be realized with the same vehicle, service time at the centre is 30 min.

Route C: Customer sequence: 0; 1; 2; 6; 19; 15; 22; 27; 8; 0 (Banská Bystrica; Kynceľová; Nemce; Selce; Priechod; Môlča; Dolná Mičiná; Čerín; Horná Mičiná; Banská Bystrica), total capacity 54, return due to upper time limit, total time of route C 94.22 minutes, objective value 317.17 minutes ⇒ the next route will be realized by another vehicle.

Route D: Customer sequence: 0; 3; 17; 12; 16; 10; 9; 24; 25; 18; 0 (Banská Bystrica; Malachov; Hronsek; Badín; Kordíky; Horné Pršany; Vlkanová; Sliač; Kováčová; Sielnica; Banská Bystrica), total capacity 77, total time of route 148.59 minutes, objective value 465.76 minutes.

T a b l e 2

Route B

Number	**23**	**14**
	D. Harmanec	Harmanec
Demand q_i	12	17
Saving t_i	5	8
Earliest limit l_i	0	0
Latest limit u_i	180	180
Arrival	12.65	20.38
Start of service	12.65	20.38
End of service	17.65	28.38
Capacity	12	29

Source: Own compilation.

T a b l e 3

Route C

Number	**1**	**2**	**6**	**19**	**15**	**22**	**27**	**8**
	Kynceľová	Nemce	Selce	Priechod	Môlča	D. Mičiná	Čerín	H. Mičiná
Demand q_i	5	4	12	4	4	10	5	10
Saving t_i	3	3	5	3	3	5	3	5
Earliest limit l_i	0	60	30	60	120	0	60	0
Latest limit u_i	120	180	180	150	180	180	180	180
Arrival	72.96	76.71	83.46	92.96	113.71	132.61	139.79	149.34
Start of service	72.96	76.71	83.46	92.96	120	132.61	139.79	149.34
End of service	75.96	79.71	88.46	95.96	123	137.61	142.79	154.34
Capacity	5	9	21	25	29	39	44	54

Source: Own compilation.

T a b l e 4

Route D

Number	**3**	**17**	**12**	**16**	**10**	**9**	**24**	**25**	**18**
	Malachov	Hronsek	Badín	Kordíky	Horné Pršany	Vlkanová	Sliač	Kováčová	Sielnica
Demand q_i	6	5	11	6	4	6	18	16	5
Saving t_i	3	3	5	3	3	3	8	8	3
Earliest limit l_i	0	30	0	60	30	60	0	0	120
Latest limit u_i	120	90	180	180	180	120	180	180	180
Arrival	6	26.31	38.25	63.48	85.99	102.45	112.75	123.45	134.27
Start of service	6	30	38.25	63.48	85.99	102.45	112.75	123.45	134.27
End of service	9	33	43.25	66.48	88.99	105.45	120.75	131.45	137.27
Capacity	6	11	22	28	32	38	56	72	77

Source: Own compilation.

Conclusion

Distribution is evidently one of the major dimensions of many firms. Logistic costs constitute a significant share of the total costs of every organization. This amount varies from 10% to 25% of the total costs depending on the given industry and country; that's why many managers start to pay attention to optimization techniques that involve the reduction of logistic cost. Many variants of routing and scheduling problems that can be very rewarding are known in the field of logistics.

This paper deals with a real-data vehicle routing problem with time windows. The vehicle routing problem with time windows belongs to NP-hard problems, so no algorithm has been known to solve it in the polynomial time, even though with the development of information technology the number of problems that can be solved by exact algorithms has been increased. The alternative is, except for classical heuristics, the use of evolutionary algorithm, which can give after finite number of iteration an "effective" solution.

Nowadays, we follow the increased interest in methods, which are inspired by different biological evolutionary processes in nature. This technology is covered by the common name of "evolutionary algorithms". But their application to constrained problems requires some additional modifications of theirs basic versions. The paper was focused on application of self-organizing migrating algorithm (SOMA) to real-life vehicle routing problem with time windows in Slovakia. The special factors that involve the use of that algorithm were presented and the efficiency of calculations has been validated on the basis of publicly available instances. The result was also compared with the known solution based on heuristic Clarke & Wright's savings algorithm with time windows. Based on these results the following can be stated: the number of vehicles was decreased (from 4 to 3) and the total time of distribution was improved by 16.2%.

This approach can be also applied to a wide variety of distribution problems with time restrictions.[5] The economic impact of optimization of these problems can lead to considerable savings in logistics costs and in that way to increase the competitive advantage of many companies.

[5] E.g. travelling salesman problem with time windows, multiple salesman problem with time windows, multi depot travelling salesman problem with time windows, group traveling salesman problem with time windows, one of set travelling salesman problem with time windows, travelling purchaser problem with time windows, open travelling salesman problem with time windows, vehicle routing problem with time windows, multi depot vehicle routing problem with time windows, fleet size and mix vehicle routing problem with time windows, open vehicle routing problem period vehicle routing problem with time windows, period vehicle routing problem with simultaneous delivery and pickup with time windows, inventory vehicles routing problem with time windows and stochastic vehicle routing problem with time windows, etc.

References

ACHUTHAN, N. R. – CACCETTA, L. – HILL, S. P. (2003): An Improved Branch-and-Cut Algorithm for the Capacited Vehicle Routing Problem. Transportation Science, *37*, No. 2, pp. 153 – 169.

AI, T. J. – KACHITVICHYANUKUL, V. (2009): A Particle Swarm Optimization for the Vehicle Routing Problem with Simultaneous Pickup and Delivery. Computers & OR, *36*, No. 5, pp. 1693 – 1702.

BAKER, E. K. (1983): An Exact Algorithm for the Time-Constrained Traveling Salesman Problem. Operations Research, *31*, No. 5, pp. 938 – 945.

BARD, J. F. – KONTORAVDIS, G. – YU, G. (2002): A Branch-and-Cut Procedure for the Vehicle Routing Problem with Time Windows. Transportation Science, *36*, No. 2, pp. 250 – 269.

BREZINA, I. – ČIČKOVÁ, Z. – GEŽÍK, P. – PEKÁR, J. (2009): Modelovanie reverznej logistiky – optimalizácia procesov recyklácie a likvidácie odpadu. Bratislava: Vydavateľstvo Ekonóm.

CAMPBELL, A. M. – SAVELSBERGH, M. (2005): Decision Support for Consumer Direct Grocery Initiatives. Transportation Science, *39*, No. 3, pp. 313 – 327.

CHANG, Y. – CHEN, L. (2007): Solve the Vehicle Routing Problem with the Time Windows via a Genetic Algorithm. Discrete and Continuous Dynamical Systems. Supplement, pp. 240 – 249. Available at: <http://www.aimsciences.org/journals/pdfs.jsp?paperID=2806&mode=full>.

CHRISTOFIDES, N. – MINGOZZI, A. – TOTH, P. (1981a): Exact Algorithms for the Vehicle Routing Problem, Based on Spanning Tree and Shortest Path Relaxations. Mathematical Programming, *20*, No. 1, pp. 255 – 282.

CHRISTOFIDES, N. – MINGOZZI, A. – TOTH, P. (1981b): State Space Relaxation Procedures for the Computation of Bounds to Routing Problems. Networks, *11*, No. 2, pp. 145 – 164.

ČIČKOVÁ, Z. – BREZINA, I. (2008): An Evolutionary Approach for Solving Vehicle Routing Problem. In: Quantitative Methods in Economics: Multiple Criteria Decision Making XIV. Bratislava: Vydavateľstvo Ekonóm, pp. 40 – 44.

ČIČKOVÁ, Z. – BREZINA, I. – PEKÁR, J. (2008): Alternative Method for Solving Traveling Salesman Problem by Evolutionary Algorithm. Management Information Systems, *3*, No. 1, pp. 17 – 22.

ČIČKOVÁ, Z. – BREZINA, I. – PEKÁR, J. (2011): A Memetic Algorithm for Solving the Vehicle Routing Problem. In: Mathematical Methods in Economics 2011. [29th International Conference, Jánska Dolina, Slovakia.] Prague: Professional Publishing, pp. 125 – 128.

DAVENDRA, D. – ZELINKA, I. (2008): Flow Shop Scheduling using Self Organizing Migration Algorithm. In: Modelling and Simulation. [Proceedings of the European Conference.] Nicosia: European Council of Modelling and Simulation, pp. 195 – 200.

DERIGS, U. – REUTER, K. (2009): A Simple and Efficient Tabu Search Heuristic for Solving the Open Vehicle Routing Problem. Journal of the Operational Research Society, *60*, No. 12, pp. 1658 – 1669.

DESROSIERS, L. –DUMAS, Y. – SOLOMON, M. M. – SOUMIS, F. (1995): Time Constrained Routing and Scheduling. In: BALL, M. O. et al. (eds): Network Routing. Handbooks in Operations Research and Management Science. Vol. 8. Amsterdam: North-Holland, pp. 35 – 139.

GAREY, M. R. – JOHNSON, D. S. (1979): Computers and Intractability. A Guide to the Theory of NP-Completeness. New York: Freeman.

GHOSEIRI, K. – GHANNADPOUR, S. F. (2010): Multi-objective Vehicle Routing Problem with Time Windows Using Goal Programming and Genetic Algorithm. Applied Soft Computing, *10*, No. 4, pp. 1096 – 1107.

LI, X. Y. – TIAN, P. – LEUNG, S. CH. (2008): An ant Colony Optimization Metaheuristic Hybridized with Tabu Search for Open Vehicle Routing Problems. Journal of the Operational Research Society, *60*, No. 7, pp. 1012 – 1025.

OMBUKI, B. – ROSS, B. J. – HANSHAR, F. (2006): Multi-Objective Genetic Algorithms for Vehicle Routing Problem with Time Windows. Applied Intelligence, *24*, No. 1, pp. 17 – 30.

ONWUBOLU, G. C. – BABU, B. V. (2004): New Optimization Techniques in Engineering. Berlin – Heidelberg: Springer-Verlag.

PEŠKO, Š. (2006): Differential Evolution for Small TSPs with Constraints. In: Challenges in Transport and Communication. [4th International Scientific Conference.] Pardubice, pp. 989 – 994.

POOT, A. – KANT, G. – WAGELMANS, A. P. M. (2002): A Savings Based Method for Real-life Vehicle Routing Problems. Journal of the Operational Research Society, *53*, No. 1, pp. 57 – 68.

POTVIN, J. Y. – ROUSSEAU, J. M. (1993): A Parallel Route Building Algorithm for the Vehicle Routing and Scheduling Problem with Time Windows. European Journal of Operations Research, *66*, No. 3, pp. 331 – 340.

ROCHAT, Y. – TAILLARD, E. D. (1995): Probabilistic Diversification and Intensification in Local Search for Vehicle Routing. Journal of Heuristics, *1*, No. 1, pp. 147 – 167.

RUSSELL, R. A. – URBAN, T. L. (2008): Vehicle Routing Problem with Soft Time Windows and Erlang Travel Times. Journal of the Operational Research Society, *59*, No. 9, pp. 1220 – 1228.

SAVELSBERGH, M. W. P. (1985): Local Search for Routing Problems with Time Windows. Annals of Operations Research, *4*, No. 1, pp. 285 – 305.

SOLOMON, M. M. (1987): Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints. Operations Research, *35*, No. 2, pp. 254 – 265.

TARANTILIS, C. D. – ZACHARIADIS, E. E. – KIRANOUDIS, C. T. (2008): A Guided Tabu Search for the Heterogeneous Vehicle Routing Problem. Journal of the Operational Research Society, 59, No. 12, pp. 1659 – 1673.

THANGIAH, S. R. (1999): A Hybrid Genetic Algorithms Simulated Annealing and Tabu Search Heuristic for Vehicle Routing Problems with Time Windows. In: CHAMBERS, L. (ed.): Practical Handbook of Genetic Algorithms. Volume 3: Complex Structures. Boca Raton, Florida: CRC Press, pp. 347 – 381.

TUNG, D. V. – PINNOI, A. (2000): Vehicle Routing-scheduling for Waste Collection in Hanoi. European Journal of Operational Research, *125*, No. 3, pp. 449 – 468.

YU, B. – YANG, Z. – YAO, B. (2009): An Improved Ant Colony Optimization for Vehicle Routing Problem. European Journal of Operational Research, *196*, No. 1, pp. 171 – 176.

WEIGEL, D. – CAO, B. (1999): Applying GIS and OR Techniques to Solve Sears Technician-dispatching and Home-delivery Problems. Interfaces, *29*, No. 1, pp. 112 – 130.

ZELINKA, I. (2002): Umělá intelligence v problémech globální optimalizace. Praha: BEN-technická literatura.

ZHANG, Z. – TANG, L. (2007): Solving Prize-collecting Traveling Salesman Problem with Time Windows by Chaotic Neural Network. In: Neural Networks: Part II – Advances in Neural Networks. [ISNN '07 Proceedings of the 4th International Symposium..] Berlin – Heidelberg: Springer-Verlag, pp. 63 – 71.