FILIP TANER, B.Eng.
E-mail: filip.taner@gmail.com
ANTE GALIĆ, B.Eng.
E-mail: ante.galic@fpz.hr
TONČI CARIĆ, Ph.D.
E-mail: tonci.caric@fpz.hr
University of Zagreb,
Faculty of Transport and Traffic Sciences
Vukelićeva 4, 10000, Zagreb, Croatia

Distribution Logistics
Review
Accepted: July 13, 2011
Approved: July 5, 2012

SOLVING PRACTICAL VEHICLE ROUTING PROBLEM WITH TIME WINDOWS USING METAHEURISTIC ALGORITHMS

ABSTRACT

This paper addresses the Vehicle Routing Problem with Time Windows (VRPTW) and shows that implementing algorithms for solving various instances of VRPs can significantly reduce transportation costs that occur during the delivery process. Two metaheuristic algorithms were developed for solving VRPTW: Simulated Annealing and Iterated Local Search. Both algorithms generate initial feasible solution using constructive heuristics and use operators and various strategies for an iterative improvement. The algorithms were tested on Solomon's benchmark problems and real world vehicle routing problems with time windows. In total, 44 real world problems were optimized in the case study using described algorithms. Obtained results showed that the same distribution task can be accomplished with savings up to 40% in the total travelled distance and that manually constructed routes are very ineffective.

KEY WORDS

vehicle routing problem with time windows, simulated annealing, iterated local search, metaheuristcs

1. INTRODUCTION

Vehicle routing problem (VRP) is interesting to the scientific community because of the possibility of greatly reducing transportation costs by successfully solving the problem. Optimization of delivery routes is important because transport costs influence the overall cost of the delivered goods. In this field of combinatorial optimization many scientific papers and surveys were published [1, 2, 3, 4, 5]. Algorithms for VRP optimization can help to solve many different logistics problems in distribution. Manually constructed routes are very inefficient and can be significantly improved by algorithms for solving VRP. This is done by reducing overall travelled distance, number of vehicles used and total waiting time. When planning delivery routes for their products, logistic companies are facing a problem that can be classified as one of the VRP instances, depending on the given constraints. Most common constraints are time windows and limited capacities of vehicles which define Vehicle Routing Problem with Time Windows (VRPTW). In solving VRPTW exact methods evaluate every possible solution, but those methods are extremely time-consuming, even when using high-end computers. Metaheuristic methods can produce reasonably good solutions in relatively short time. Two metaheuristic strategies for solving VPRTW were applied: Simulated Annealing and Iterative Local Search. Algorithm parameters were tuned to specific set of problems. Both strategies were tested on Solomon's benchmark problems, as well as on 44 real world problems as case study.

2. VEHICLE ROUTING PROBLEM

Vehicle routing problem is NP-hard problem defined by the task of determining the optimal set of routes to be performed by a fleet of vehicles to serve a given set of customers [6]. The solution of the VRP is a set of routes which all begin and end in the depot and where all customers are served only once [4]. The VRP is combinatorial optimization problem on the graph. Let $G = (V, A)$ is fully connected graph with a set of vertices $V = \{0, \ldots, n\}$. Vertex with index 0 is depot, while other vertices have indices $i = 1, 2, \ldots, n$ and represent customers. Every arc (i, j) from the set of arcs A has non-negative transportation cost c_{ij} which can represent the cost of fuel or distance between customers or some other combination of costs. In a case when one way streets exists in a street topology, the asymmetric graphs are used where costs in opposite direction are not same $c_{ij} \neq c_{ji}$. The homogeneous fleet of K vehicles is available in the depot and each vehicle is used only for one route which begins and ends in the depot. Let route of single vehicle R is sequence of customers starting and ending in the depot. If there are only capacity constrains in the problem it can be treated as Capacitated Vehicle Routing problem (CVRP). Each customer has non-negative demand m_i while each vehicle k_i is limited by maximal capacity q_i. Basic variant of problem can be extended by additional constraints. *Figure 1* shows different variants of VRP problem and relations between them [1].

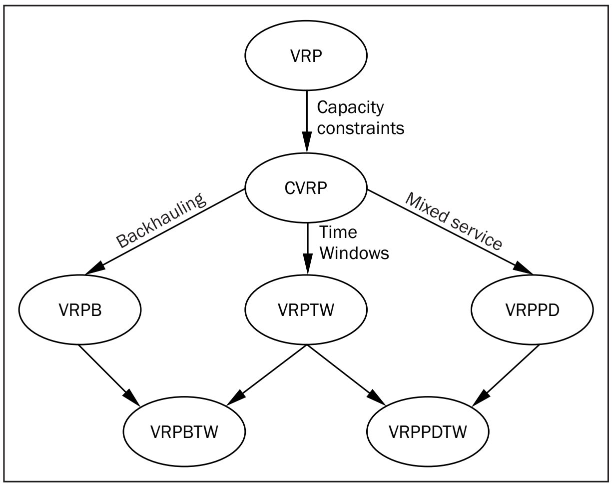

Figure 1 - VRP variants and their interconnections

Source: prepared by the author on the basis of [1]

If customers request delivery within a specific time window, the problem can be modelled as vehicle routing problem with time windows (VRPTW). In case of simultaneous pickup and delivery of goods, the problem can be modelled as vehicle routing problem with pickup and delivery (VRPPD). In case of vehicle routing problem with backhauls (VRPB), a vehicle needs to collect goods after it completes delivery. Last two problems can also be extended with time constraints (VRPBTW and VRPPDTW). In this paper the practical problem of postal delivery company is modelled as the VRPTW problem. Service of each customer has to be done inside time window $[e_i, l_i]$, where e_i is earliest and l_i latest possible time when service must occur. If vehicle arrives before opening of time window it has to wait. It is forbidden to start service after closing of time window l_i because it would produce an unfeasible solution. The depot has also time window that represent its opening and closing time. The primary objective of VRPTW optimization is to find the minimal number of vehicles that can accomplish delivery task in a way that each route satisfies all time and capacity constraints and that each customer is serviced only once. The secondary objective is to minimize overall travelled distance or time.

3. INITIAL SOLUTION

In order to effectively solve VRPTW problem, it is necessary to obtain a feasible initial solution in which all constraints are satisfied. Due to the fact that finding of a feasible solution for VRPTW problem with minimal number of vehicles is a very complex task, constructive heuristic algorithms often produce solutions of bad quality which serve only as a starting point for further optimization. Best known constructive algorithm for VRPTW is Solomon I1 heuristic. Simplified description is presented in *Algorithm 1* and more details can be found in original paper [7]. Function *FindSeed()* selects the first customer which will be inserted into the vehicle route. Criteria for its selection are the measure of how far away it is from the depot and how close is its time window opening. Inside main loop function *FindNewCustomer(v)* searches for un-served customer that can be inserted into a route of current vehicle v without violation of time and capacity constraints. If it fails, a new vehicle is engaged and a new seed customer is added to the route. Function *Terminate()* checks if there are more un-routed customers.

Algorithm 1. *Solomon's I1 insertation heuristic*

```
procedure SolomonI1()
    c := FindSeed()
    v :=NewVehicle()
    AddToRoute(c, v)
    while not Terminate() do
        c := FindNewCustomer(v)
        if c != null then
            InsertCustomer(c, v)
        else
            c:=FindSeed()
            v :=NewVehicle()
            AddToRoute(c, v)
        endif
    endwhile
    return CurrentSolution()
end
```

4. IMPROVEMENT OPERATORS

Initial solution can be significantly improved by simple operations such as relocation, exchange and reposition of customers in/between routes of vehicles. Simplest improvement operators are presented bellow and more complex ones can be found in [8]. *Intra Route* operators relocate one or more customers from one position in route to another position in same route. *Inter Route* operators relocate and exchange customers between two different routes. *Intra Relocate* operator shown in *Figure 2* removes one customer from a route and inserts it to the other position inside the same route. Such operation will cause shifting of one

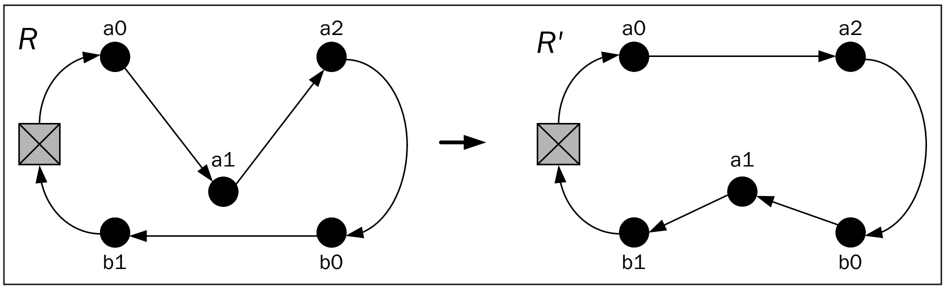

Figure 2 - Intra Relocate operator

Source: prepared by the author on the basis of [13]

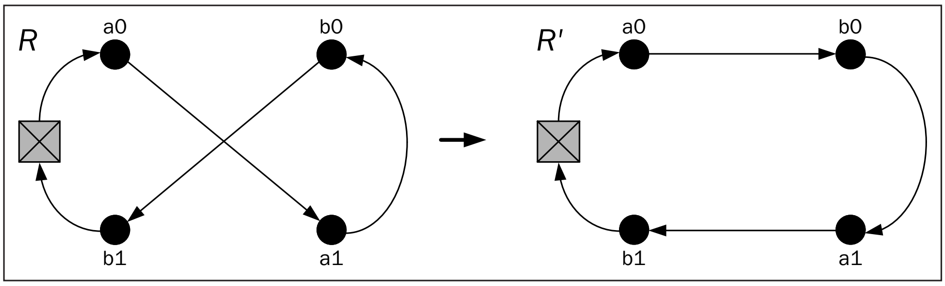

Figure 3 - Intra TwoOpt operator

Source: prepared by the author on the basis of [13]

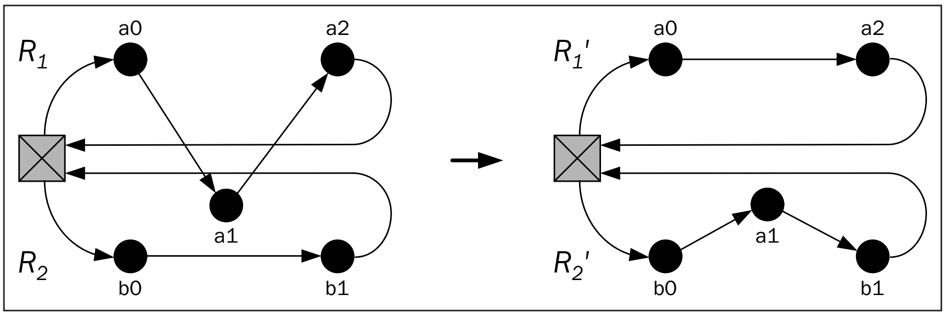

Figure 4 - Inter Relocate operator

Source: prepared by the author on the basis of [13]

or more customers as well. The customer a_1 is inserted between customers b_0 and b_1 if there is a positive saving in the new route R'. Saving u is expressed as the following sum:

$$u = (c(a_0,a_1) + c(a_1,a_2) + c(b_0,b_1)) - \\ -(c(a_0,a_2) + c(b_0,a_1) + c(a_1,b_1)) \quad (1)$$

Intra TwoOpt operator shown in *Figure 3* modifies route in such a way that it removes crossings in the route R and reverses travel directions between customers b_0 and a_1. Savings u produced by a new route R' have to be positive and they are expressed as the following sum:

$$u = (c(a_0,a_1) + c(b_0,b_1)) - (c(a_0,b_0) + c(a_1,b_1)) \quad (2)$$

Beside savings calculation in *Intra Route* operators it is also necessary to check for feasibility of the time window constraints for new route R'. Capacity constraints do not need to be checked because they are already examined in route R. *Inter Relocate* operator shown in *Figure 4* removes customer a_1 from route R_1 and inserts it between customers b_0 and b_1 in new route R_2' if there is positive saving expressed as the following sum:

$$u = (c(a_0,a_1) + c(a_1,a_2) + c(b_0,b_1)) - \\ -(c(a_0,a_2) + c(b_0,a_1) + c(a_1,b_1)) \quad (3)$$

Route R_1' will keep all constraints fulfilled after removal of a_1 but for route R_2' time and capacity feasibility has to be checked because insertion of a_1 can produce overload and delay to customers. *Inter Exchange* operator shown in *Figure 5* swaps two customers from different routes. Customer a_1 from route R_1 is exchanged with customer b_1 from route R_2. It is necessary to check the feasibility of both routes and reject exchange combination $\{a_1,b_1\}$ if it will produce overload or delay in any of two routes. Savings of exchange move are expressed as the following sum:

$$u = (c(a_0,a_1) + c(a_1,a_2) + c(b_0,b_1) + c(b_1,b_2)) - \\ -(c(a_0,b_1) + c(b_1,a_2) + c(b_0,a_1) + c(a_1,b_2)) \quad (4)$$

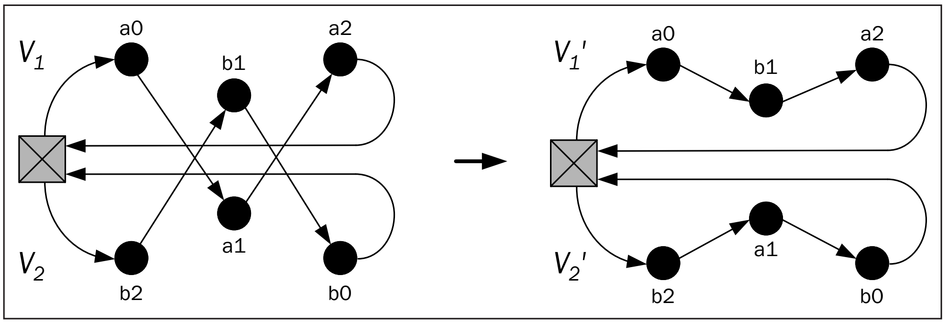

Figure 5 - Inter Exchange operator

Source: prepared by the author on the basis of [13]

It is important to note that each improvement operator among all feasible moves selects the one that will produce maximal saving u.

5. LOCAL SEARCH AND METAHEURISTICS

Improvement operators are iteratively applied in the local search procedure in order to improve solution as much as possible. Each operator searches for narrow neighbourhood of the current solution and tries to find a better one. Eventually, local search will stop in a local optimum or suboptimal solution if there are no feasible solutions that can be found using the same improvement operators. In order to continue improving the obtained solution it is necessary to move to another area of solution space by temporarily accepting worse solutions and applying the local search again.

Metaheuristic methods are developed in order to reduce the solution search space and to consider evaluating only areas which have high probability to contain good solutions. Moving from one neighbourhood to another by accepting worse solutions can lead to global optima in successive iterations of local search.

5.1 Simulated annealing

Methodology of SA is analogous to the physical process of annealing in metallurgy [9, 10, 11]. In order to obtain perfect crystal structures of metal without irregularities, solid metals are melted and then cooled down slowly. Heat enables atoms to become unstuck from their initial positions which correspond to local optimum of minimal energy and wander randomly through states of higher energy. In this analogy, the different energy levels represent candidate solutions, and evaluation function represents the internal energy of the solid [12]. Cooling needs to be done slowly in order to increase the chance of getting a configuration with minimal internal energy (global optimum). Simulated annealing (*Algorithm 2*) starts from the initial feasible solution obtained by Solomon's I1 insertion heuristic described in the previous section (*Algorithm 1*). In each iteration of the main loop function *Escape*(s) generates new solution s' using modified variants of *Intra Relocate*, *Inter Relocate* and *Inter Exchange* operators in which negative savings are allowed. In the next step local search procedure is applied in order to produce solution s'' which represents local optimum [13]. Solution s'' is automatically accepted as starting point for next iteration if objective function yields better value than for solution s. Otherwise, acceptance criterion for solution s'' follows the probability function known as Metropolis condition [11]:

$$p_t = e^{-\frac{f(s'')-f(s)}{T}} \tag{5}$$

Temperature T determines how likely solution s'' will be accepted. As temperature drops throughout the process, the probability of accepting worse solution decreases. This is called cooling schedule and determines the number of iterations of the algorithm. The cooling schedule is a geometric function $T = T \cdot \alpha$, where $0 \leq \alpha < 1$. The maximal (initial) and the minimal temperature are empirically determined for each problem set. The objective function that evaluates the quality of the given solution is calculated as follows:

$$f(s) = v_s \cdot d_s \tag{6}$$

where v_s is the number of vehicles used in the solution and d_s is the sum of route distances. Solution *best* is updated in case of improvement of the evaluation function (6). The first factor of this function greatly increases the value of the function thus forcing the algorithm to accept solutions with less used vehicles (routes). The second factor is needed to reduce the overall distance if the number of the used vehicles is maintained. Function *Terminate()* is responsible for stopping the algorithm after temperature T reaches its allowed minimum. Additionally, the temperature is reset to the initial value a given number of times, to repeat the cooling process again starting from the best solution found at the time.

Algorithm 2. *Simulated Annealing*

procedure SA()
T := InitialTemperature()
init := SolomonI1()
s := LocalSearch(init)

```
    best := s
    while not Terminate() do
        s' := Escape(s)
        s" := LocalSearch(s')
        if (f (s'') <f (s)) then
            s := s''
        else
            j := rnd(0, 1)
            k := -((f(s'')-f(best))/T
            if j <exp(k) then
                s := s"
            endif
        endif
        if (f(s) <f(best)) then
        best := s
        endif
        T := CoolingSchedule (T)
    endwhile
    return best
end
```

5.2 Iterative Local Search

Iterative Local Search (*Algorithm 3*) is a metaheuristic method for guiding local search [12]. Initial solution s is obtained by Solomon's I1 insertion heuristic and further improved with local search procedure. In each iteration of the main loop, solution s' is randomly chosen from the neighbourhood of the best solution using *Escape(best)* procedure and by applying local search operators it converges to a local optimum producing solution s''. The goal is to broaden up search space and eventually converge to global optimum. Because of the stochastic nature of the algorithm there is no guarantee of finding the optimal solution, but producing results near optimum is possible. Unlike SA algorithm, where escaping local optima starts from current solution s in each iteration, ILS algorithm uses the best solution to randomly generate neighbouring solution s'. In case of improvement of evaluating function (6) after applying local search, solution *best* is updated. The algorithm will terminate when it reaches the maximal number of iterations.

Algorithm 3. *Iterative Local search*

```
procedure ILS()
    init := SolomonI1()
    best := LocalSearch(init)
    while not Terminate() do
        s' := Escape(best)
        s" := LocalSearch(s')
        if (f (s") <f (best)) then
            best := s"
        endif
    endwhile
    return best
end
```

6. BENCHMARK RESULTS

The efficiency of ILS and SA algorithms was tested on the standard Solomon's benchmark problem set that consists of 100 customers [14]. There are six sets of problems: R1, R2, C1, C2, RC1 and RC2. In sets R1 and R2 the customers' position is created randomly through a uniform distribution. In sets C1 and C2 the customers are positioned in groups. In sets RC1 and RC2, part of the customers is placed randomly and part is placed in groups. Besides, R1, C1 and RC1 problems have a short-term planning horizon, short service time, lighter capacity vehicles, and they allow only some customers in each route. Sets R2, C2 and RC2 have got a long-term planning horizon, longer

Table 1 - Comparison of the results for the number of vehicles and distances obtained by ILS and SA to the best proposed results for Solomon's VRPTW problems. CM = cumulative values. HG = (Homberger & Gehring, 2005) [15], BC = (Le Bouthillier & Crainic, 2005)[16], PR = (Pisinger &Röpke, 2005)[17], MBD = (Mester et al., 2007)[18]

	R1	R2	C1	C2	RC1	RC2	CM
HG	12.08	2.82	10.00	3.00	11.50	3.25	408.00
	1,211.67	950.72	828.45	589.96	1,395.93	1,135.09	57,422.00
BC	12.08	2.73	10.00	3.00	11.50	3.25	407.00
	1,209.19	936.62	828.38	589.86	1,389.22	1,143.70	57,412.00
PR	11.92	2.73	10.00	3.00	11.50	3.25	406.00
	1,212.39	957.72	828.38	589.12	1,387.12	1,123.49	57,332.00
MBD	12.00	2.73	10.00	3.00	11.50	3.25	406.00
	1,208.18	954.09	828.38	589.12	1,387.12	1,119.70	46,812.00
SA	13	3.09	10	3	12.62	3.62	434
	1,206.07	979.81	838.70	590.69	1,378.70	1,131.91	57,609.57
ILS	12.67	3.09	10	3	12.5	3.62	429
	1,199.98	955.99	832.85	590.12	1,371.97	1,125.08	57,108.85

service time, and since they have got higher capacity vehicles, they are able to supply more than 10 customers per route. In each set of problems, the customers' geographical distributions, the demand and the service time do not change. *Table 1* shows the comparison of results obtained by SA and ILS algorithms implemented here with results of authors that used different methods for solving VRPTW.

Compared to the results published by other relevant authors, SA and ILS algorithms implemented here produce solutions of less quality for standard benchmark problems. Due to the fact that minimization of vehicle number is the primary optimization criteria in solving VRPTW, better or similar average travelled distance on some instances should not be taken into consideration while comparing the results. The main reason for obtaining solutions with greater number of vehicles on the average is the lack of specialized strategy or phase for vehicle number minimization. Another possible reason is the absence of more complex improvement operators used in other papers. ILS produced better overall results than SA because of more intensive search of solution space around best solutions found during the optimization. Tuning of SA algorithm for specific problem instances is harder because temperature parameters used for controlling diversification and intensification ratio depend on the customers' spatial and/or temporal configuration.

7. OPTIMIZATION OF PROBLEMS IN CASE STUDY

7.1 Data collection and preparation

Historical data were obtained from the postal service provider for the period of 15 working days and prepared for the optimization using the described algorithms. The collected data include 1,200 served customers located in the area of the cities of Vinkovci, Vukovar and Županja in the Republic of Croatia. In total, 44 real-world VRPTW problems are considered for optimization. Each customer has a certain demand and defined period in which the ordered products have to be delivered. The serving and unloading time depends on the amount of cargo and ranges from 2 minutes to 12 minutes per customer. Each vehicle has the same predefined time when it must leave the depot.

Although available vehicles have limited freight capacity, these restrictions were not taken into consideration. The nature of the cargo (relatively small mass and dimensions) relaxed the problem because the vehicle capacity was never a limiting factor. Major problem were the restrictive time windows. The collected data show that 59 of 1,200 customers were not served on time.

In order to solve the practical vehicle routing problem it is necessary to operate with real-world distances and travel times based on the street topology and traffic regulations. The geographical locations of customers determined by street name and house number were geocoded in the geographic information system with a digital map of the region. In order to solve the time-constrained problems, as in the case of VRPTW, an additional matrix containing forecast travel times data between each pair of customers has to be available. Therefore, the problems are defined by two traffic matrices: the distance asymmetric look-up matrix and the related forecast travel time matrix. The calculation of the travel time matrix is based on the average speed specified for different types of roads and calculated using the Geographic Information System (GIS) and navigation map of Croatia.

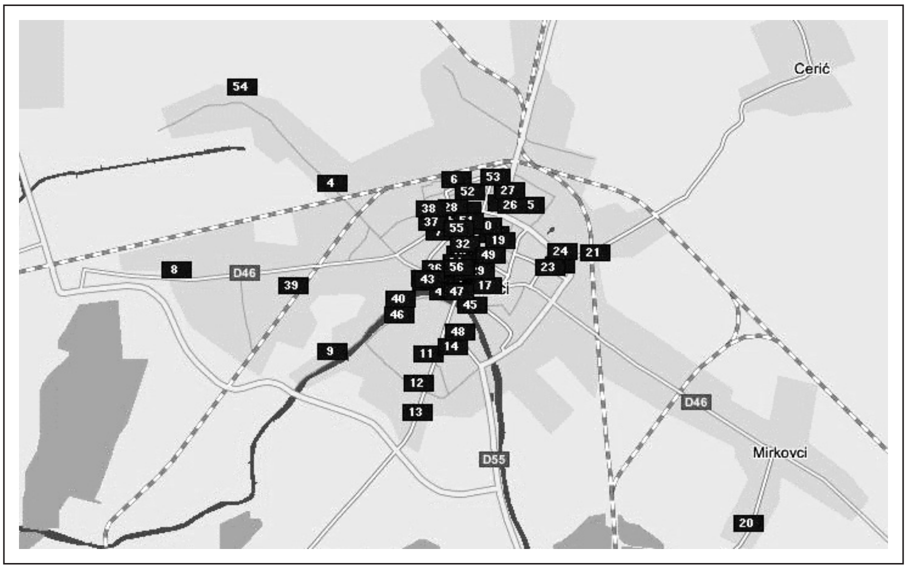

Figure 6 – Delivery problem in the City of Vinkovci

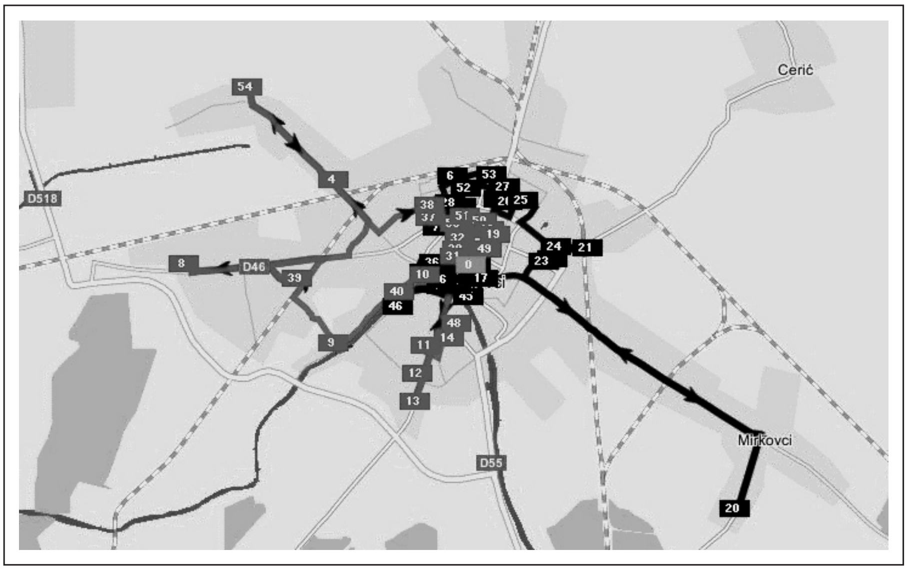

Figure 7 – Solution for delivery problem in the City of Vinkovci

The road networks are spread within three small towns and their rural parts. The number of customers for each problem is relatively small (27.3 on average). There are three time widows: from 8:00 a.m. till 10:00 a.m., from 8:00 a.m. till 3:00 p.m. and from 8:00 a.m. till 7:00 p.m. The geographical distribution of customers for one day in the City of Vinkovci is shown in *Figure 6*.

7.2 Case study results

The results show improvement of up to 40% for overall travelled distance. In some cases algorithms produced solutions with increased number of used vehicles in order to service all customers in the designated time. This is not a problem considering the fact that the number of available vehicles in this case study exceeds the number of the used ones. That strategy is much more effective because of the time constraints imposed by the customers. For every time window violation, even the smallest one, the delivery company pays a certain money penalty. The solution routes for one day of delivery in the City of Vinkovci are shown in *Figure 7*.

Due to very inefficient manually constructed route plans it is possible to obtain much better results even with algorithms that do not perform well on benchmark problems. This is because of the following characteristics of the real world problems: short-term planning horizon, short service time, and wide time windows for large number of customers. Since there are no capacity constraints, vehicles are able to supply more customers per route. The difference between overall travelled distance before and after optimization of all 44 problems in case study is shown in *Figure 8*.

8. CONCLUSION

The Vehicle Routing Problem is a combinatorial optimization task that occurs during organizing delivery

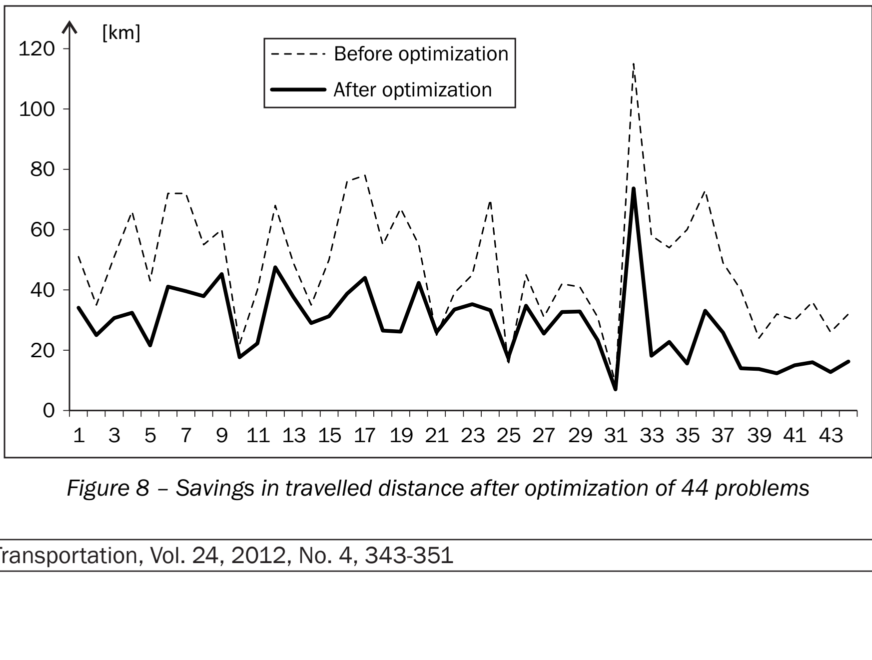

Figure 8 – Savings in travelled distance after optimization of 44 problems

and/or collection of goods in a certain area. Successful solving of VRP requires knowledge of mathematical models that represent real world problems. One of these models is the Vehicle Routing problem with Time Windows, the solving of which is done with metaheuristic algorithms which produce relatively good results in a reasonable time. Two metaheuristic strategies with tuned parameters were developed for this case study and their efficiency was tested on the standard Solomon's benchmark problems. Optimization approach with GIS tools and optimization algorithms requires minimal investment and almost no change in the company's technological system, but produces significant results. The delivery companies usually have practice of manual construction of delivery routes which are often inefficient and produce extra transport costs. Routes of a company that is specialized in package delivery were analyzed. In total, 44 separate delivery plans were optimized using the proposed algorithms and the average distance reduction per process is 40%. Because of the restrictive time windows imposed by the customers it was necessary to increase the number of vehicles for some of the problems. This strategy is more effective because of money penalty for the company in case of time window violation. The optimization of manually constructed routes on real world vehicle routing problems using metaheuristic algorithms shows two achievements: reduction in total travelled distance and successful solving of time window violation problem.

FILIP TANER, dipl. ing.
E-mail: filip.taner@gmail.com
ANTE GALIĆ, dipl. ing.
E-mail: ante.galic@fpz.hr
TONČI CARIĆ, Ph. D.
E-mail: tonci.caric@fpz.hr
Sveučilište u Zagrebu, Fakultet prometnih znanosti
Vukelićeva 4, 10000 Zagreb, Hrvatska

SAŽETAK

RJEŠAVANJE PRAKTIČNOG PROBLEMA USMJERAVANJA VOZILA S VREMENSKIM OGRANIČENJIMA UPORABOM METAHEURISTIČKIH ALGORITAMA

U ovome članku analiziran je problem usmjeravanja vozila s vremenskim ograničenjima. Primjenom algoritama za rješavanje različitih klasa VRP problema mogu se bitno smanjiti troškovi koji se pojavljuju kod distribucije dobara. Za rješavanje primijenjena su dva algoritma: Simulirano kaljenje i Iterativna lokalna pretraga. Oba algoritma generiraju početno rješenje pomoću konstruktivnih heuristika i rabe operatore i različite strategije za iterativno poboljšanje rješenja. Algoritmi su testirani na testnim (benchmark) kao i na stvarnim problemima usmjeravanja vozila s vremenskim prozorima. Ukupno 44 stvarna distribucijska procesa optimizirana su navedenim algoritmima. Dobiveni rezultati pokazali su da se isti posao distribucije u nekim slučajevima može obaviti i uz 40% manji ukupni prijeđeni put, te da su ručno rađeni planovi ruta vrlo neučinkoviti.

KLJUČNE RIJEČI

problem usmjeravanja vozila s vremenskim ograničenjima, simulirano kaljenje, iterativna lokalna pretraga, metaheuristike

LITERATURE

[1] **Toth, P.**, **Vigo, D.**: *Vehicle Routing Problem*, SIAM Publishing, monographs on discrete mathematics and applications, Philadelphia, 2002

[2] **Bräysy, O.**, **Gendreau, M.**: *Vehicle Routing Problem with Time Windows Part I: Route Construction and Local Search Algorithms*, Transportation Science, Vol. 39, No. 1, 2005, pp. 104-118

[3] **Bräysy, O.**, **Gendreau, M.**: *Vehicle Routing Problem with Time Windows, Part II, Metaheuristics*, Transportation Science, Vol. 39, No. 1, 2005, pp. 119-139.

[4] **Golden, B.**, **Raghavan, S.**, **Wasil, E.**: *The Vehicle Routing Problem, Latest Advances and New Challenges*, Springer, 2008.

[5] **Nagata, Y.**, **Bräysy, O.**, **Dullaert, W.**: *A penalty-based edge assembly memetic algorithm for the vehicle routing problem with time windows*, Computers and Operations Research, Vol. 37, 2010, pp. 724-737.

[6] **Cordeau, J.F.**, **Desaulniers G.**, **Desrosiers, J.**, **Solomon, M.**, **Soumis, F.**: *The VehicleRouting Problem with Time Windows*. Toth, P., Vigo, D. (eds.) The Vehicle Routing Problem,. SIAM Publishing, Philadelphia, 2002, pp. 157–193

[7] **Solomon, M.**: *Algorithms for the Vehicle Routing and Scheduling Problems with a Time Windows Constraints*. Operations Research. Vol. 35, 1987, pp. 254–265

[8] **Carić, T.**, **Galić A.**, **Fosin, J.**, **Gold, H.**, **Reinholz, A.**: *A Modelling and Optimization Framework for Real-World Vehicle Routing Problems*: Carić T., Gold H. (eds.) Vehicle Routing Problem, I-TECH, Vienna, 2008, pp. 15-34

[9] **Kirkpatrick, S.**, **Gelatt, C.D.**, **Vecchi, Jr. M.P.**: *Optimization by Simulated Annealing*, Science, 220, 1983, pp. 671–680

[10] **Cerny, V.**: *A Thermodynamical Approach to the Travelling Salesman Problem: An Efficient Simulation Algorithm*, Journal of Optimization Theory and Applications 45, 1985, pp. 41–51

[11] **Metropolis, N.**, **Rosenbluth, A.W.**, **Rosenbluth, M.N.**, **Teller, A.H.**: *Equations of State Calculations by Fast Computing Machines*. The Journals of Chemical Physics 21, 1953, pp. 1087–1092

[12] **Hoos, H.H.**, **Stutzle, T.**: *Stochastic local search, foundations and applications*, Elsevier, Morgran Kaufmann, 2005

[13] **Carić, T.**, **Fosin, J.**, **Galić, A.**, **Gold, H.**, **Reinholz, A.**: *Empirical Analysis of Two Different Metaheuristics for Real-World Vehicle Routing Problems*, Bartz-Beielstein T. et al. (Eds.): Hybrid Metaheuristics 2007, Lecture Notes in Computer Science (LNCS), Springer-Verlag, Berlin Heidelberg 2007 4771, pp. 31–44

[14] Solomon benchmark available from web page (last accessed on July 2011): http://www.sintef.no/Projectweb/TOP/Problems/VRPTW/Solomon-benchmark/

[15] **Homberger, J.**, **Gehring, H.**: *A two-phase hybrid metaheuristic for the vehicle routing problem with time windows*, European Journal of Operational Research, Vol. 162, No. 1, 2005 pp. 220–238

[16] **Le Bouthillier, A.**, **Crainic, T.G.**: *Cooperative parallel method for vehicle routing problems with time windows*. Computers and Operations Research, Vol. 32, No. 7, 2005, pp. 1685-1708

[17] **Pisinger, D.**, **Röpke, S.**: *A general heuristic for vehicle routing problems*. Technical Report, Department of Computer Science, University of Copenhagen, Copenhagen, Denmark, 2005

[18] **Mester, D.**, **Bräysy, O.**, **Dullaert, W**: *A multi-parametric evolution strategies algorithm for vehicle routing problems*. Expert Systems with Applications, Vol. 32, No. 2, 2007, pp. 508-517