Possible Application of Solver Optimization Module for Solving Single-circuit Transport Problems

Josef Šedivý[1*], Jiří Čejka[2] and Mykola Guchenko[3]

[1]*University of Pardubice., Department of Transport Management, Marketing and Logistics, Studentská 95, 532 10 Pardubice, Czech Republic; Email: st61075@student.upce.cz*
[2]*Institute of Technology and Business in České Budějovice., Department of Informatics and Natural Sciences, Okružní 517/10, 370 01 České Budějovice, Czech Republic; Email: cejka@mail.vstecb.cz*
[3]*Kremenchuk Mykhailo Ostrogradskyi National University., Department of Computer and Information Systems, Pershotravneva st. 20, 39600 Kremenchuk, Ukraine; Email: mykolaguchenko@gmail.com*

*Corresponding Author: Josef Šedivý

Abstract: The article deals with the possible application of the Solver optimization module to solving the single-circuit transport problems. First, the article describes the single-circuit transport problems and the optimization module Solver itself. Using the specific model example of beer distribution, the author demonstrates the algorithm which may be applied to solving single-circuit transport problems by means of Solver. The travel route designed by Solver is then compared with the originally proposed route. The values being compared include the total length of travel routes created and the associated variable costs spent on serving customers and also route design time. Thus, using the practical example of beer distribution, the manuscript has demonstrated the algorithm which is used for addressing the single-circuit transport problems. Nonetheless, possible application of the Solver tool is not limited to seeking a solution to the travelling salesman problem only. It can also be implemented even to discussing the multi-circuit transport problems with various confinements.

1. Introduction

Vehicle routing problems belong to the problems typically solved within the operational analysis. They are designed to optimize transport services carried out by way of cyclical tours. In practice, they can be encountered in the situations where it is necessary to ensure transport services for a certain set of customers from one or a few initial points. The vehicle routing problems are solved using a number of methods which can be divided into exact, heuristic and metaheuristic ones as per

their computational complexity and quality of solution they offer. Exact methods allow us to find an optimal solution to the problem. Their drawback is that they are highly compute-intensive, especially in case of large problems. Hence, the other two groups of methods are frequently employed in practice. These methods seek to find a suboptimal solution to the problem, which merely gets closer to the optimal solution. The study by Linda and Volek [1] notes, these solutions most likely deviate from the optimal solution by 2 to 3 % only. Unlike exact ones, the other two methods provide even in case of more complex problems a good quality solution in an acceptable time frame.

Vehicle routing problems encountered in practice are mostly more complex types of problems which cannot be solved using manual methods which are very laborious, time consuming and prone to numerical errors. The problems need to be solved using the various software types. To this end, both simple cheap programs and more expensive professional products capable of solving the problems involving thousands of variables and constraints are available [2]. This article is aimed to introduce a time saving and effective possibility how to solve single-circuit transport problems using the generally available spreadsheet MS Excel and its add-in called Solver and apply research to specific case.

2. Characteristics of Single-circuit Transport Problem

Vehicle routing problems are a special type of distribution problems where the goods are distributed or collected by means of one or a few circular tours of vehicles [2]. The goal is to determine such a sequence in which individual points are to be visited and to sequence them in individual travel routes so that the transport intensity is kept to a minimum and in most cases each customer is visited exactly once. For example, the length of the travel route, time consumption or costs can be minimized [2]. At the same time, all requirements of customers and technological limits of the fleet deployed must be respected.

2.1 Single-circuit Transport Problem

A single-circuit transport problem, also known as the travelling salesman problem, represents a basic type of the vehicle routing problem. The principle of this problem consists in that we have n points which have to be cyclically served from a certain original point and then it is necessary to return back to this original point [3]. In doing so, the goal is to find the shortest possible circuit which begins and ends in the original point and includes all points which need to be served exactly once [4]. Apart from the above requirements, the problem has no other limitations. This allows all points to be served using a single circuit. The mathematical model of this problem is presented by Pelikán [5]:

Minimize

$$z = \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij} \qquad (1)$$

Subject to the constraints

$$\sum_{i=1}^{n} x_{ij} = 1, \quad i = 1,2,\dots,n \qquad (2)$$

$$\sum_{j=1}^{n} x_{ij} = 1, \quad j = 1,2,\dots,n \qquad (3)$$

$$u_i - u_j + n x_{ij} \leq n - 1, \quad i = 1,2,\dots,n, \quad j = 1,2,\dots,n \qquad (4)$$

$$x_{ij} \in \{0,1\}, \quad i,j = 1,2,\dots,n \qquad (5)$$

where: n is a number of points (including the original point) which need to be visited; c_{ij} is a distance between the points i and j in measurement units of length or time; x_{ij} is a binary variable which amounts to 1 where the vehicle travels from the point i to the point j and amounts to 0 in the opposite direction; u_i is an unknown real number allocated to the point i and u_j is an unknown real number allocated to the point j.

The constraints 2 and 3 ensure that each point on the travel route will be visited exactly once. The constraint 4 represents a system of Tucker conditions which prevents emergence of partial cycles. The constraint 5 ensures that the value of binary variable x_{ij} will be 0 or 1.

3. Method

In the article, there will be executed proposal of circular road using the Solver optimization module. The Solver optimization module is a freely available MS Excel add-in, which is designed to solve standard problems of mathematical programming and allows solving the problems containing up to 200 variables and 600 constraints [4]. The Solver principle is based on searching for an optimal value of one cell with respect to the values of other cells. To this end, Solver works with a group of cells marked as the variables. Values of variables are changing within the problem solution in such a way that the required value of the target cell (optimization criterion) is achieved and the constraints of the problem are met [6].

In solving the problems using Solver, it is first necessary to prepare the input data of a relevant mathematical model on the spreadsheet. It is necessary to express:

- coefficients,
- variables,
- constraints,
- and the optimization criterion of the mathematical model.

Optimization criterion must be written in form of a formula and located in one cell. The formula must ensure consistency of the optimization criterion with the values of variables so that their values directly affect the optimization criterion value.

After entering the input data, it is possible to run the dialogue box "Solver Parameters". In the Solver Parameters box all necessary parameters need to be entered. These parameters include:

- objective function,
- optimization type,
- module variables,
- constraints.

The reference to the cell containing the optimization criterion shall be entered as an objective function. It is also necessary to determine whether a value of this cell is to be maximized or minimized or if its specific value is to be searched for [7]. Relevant variables from the spreadsheet will be entered in the field "Module Variables", while limitations of the mathematical model will be entered in the dialogue box "Add Constraint". In this dialogue box, It is needed to select the specific type of constraint and the cells affected by this constraint.

After entering all necessary parameters, the solving method needs to be chosen, the simplex method, the gradient method or the evolutionary algorithm. The simplex method can be employed to solve linear problems only. The gradient method may also be used for solving smooth nonlinear problems and the evolutionary algorithm may be used even for solving non-smooth nonlinear problems. Once the solving method is chosen, the solution of the given problem may start by pressing the button "Solve". Once the solution is finished, Solver will provide a user with a choice of either maintaining the solution or restoring the original values. If the option to maintain the solution is chosen, the optimization criterion and variables on the spreadsheet will be adjusted according to the solution the optimization module has found.

4. Case Study

The possible application of Solver to solving single-circuit transport problems will be demonstrated using the specific model example, i.e. beer distribution. The selection of the example was based on requirement to optimize beer delivery routes, which was entered by an unnamed brewery. This distribution is carried out every business day upon the purchase orders placed by customers by way of circular tours. This means that an original travel route needs to be designed for every business day. These routes are currently planned upon the dispatcher´s intuition only. It is therefore likely that travel routes designed in this way would not be close to the optimal solution. In a model example, the route of distribution will be designed for one specific distribution day using Solver and the result obtained will be compared with the result of the intuitive method used so far.

When the vehicle routing problem is solved by means of Solver, it is necessary to first input the distance matrix into the MS Excel spreadsheet. This matrix displays distances among all points included in the travel route (see Fig. 1).

	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
2		V0	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	V14	V15	V16	V17	V18
3	V0	0	71	79	87	79	87	100	93	78	67	53	39,9	33,1	29,1	37,9	33,5	36,6	13,4	13,5
4	V1	71	0	8,2	16,3	16,7	21,1	33,4	42,1	27,7	23,2	29,2	32,1	56	68	69	84	36,6	60	57
5	V2	79	8,2	0	8,4	9,1	13,4	25,8	27,3	26	21,6	37	39,9	63	75	75	90	44,5	68	65
6	V3	87	16,3	8,4	0	17	8,9	21,2	27,6	34	29,6	45	48,4	71	83	83	98	53	76	74
7	V4	79	16,7	9,1	17	0	9,9	22,3	23,9	17	12,5	27,9	40,5	53	65	66	81	45,1	68	66
8	V5	87	21,1	13,4	8,9	9,9	0	12,3	18,7	24,8	20,4	35,8	48,4	61	73	74	88	53	76	74
9	V6	100	33,4	25,8	21,2	22,3	12,3	0	7,3	22,6	26,2	41,4	55	65	77	78	92	59	82	80
10	V7	93	42,1	27,3	27,6	23,9	18,7	7,3	0	15,3	18,9	34,1	47,3	58	70	71	85	52	75	72
11	V8	78	27,7	26	34	17	24,8	22,6	15,3	0	4,4	19,7	32,9	43,3	55	56	70	37,4	61	58
12	V9	67	23,2	21,6	29,6	12,5	20,4	26,2	18,9	4,4	0	15,8	28,4	41,4	53	54	69	33	56	54
13	V10	53	29,2	37	45	27,9	35,8	41,4	34,1	19,7	15,8	0	14,6	26,9	38,9	39,8	54	19,2	42,5	39,8
14	V11	39,9	32,1	39,9	48,4	40,5	48,4	55	47,3	32,9	28,4	14,6	0	29,8	36,6	41,5	52	5,7	29	26,3
15	V12	33,1	56	63	71	53	61	65	58	43,3	41,4	26,9	29,8	0	10,6	13,1	25,7	33,4	37,1	25,8
16	V13	29,1	68	75	83	65	73	77	70	55	53	38,9	36,6	10,6	0	9	15,1	39,7	39,5	29,5
17	V14	37,9	69	75	83	66	74	78	71	56	54	39,8	41,5	13,1	9	0	14,7	45,1	48,8	38,4
18	V15	33,5	84	90	98	81	88	92	85	70	69	54	52	25,7	15,1	14,7	0	55	44,7	42,1
19	V16	36,6	36,6	44,5	53	45,1	53	59	52	37,4	33	19,2	5,7	33,4	39,7	45,1	55	0	25,7	23
20	V17	13,4	60	68	76	68	76	82	75	61	56	42,5	29	37,1	39,5	48,8	44,7	25,7	0	15,1
21	V18	13,5	57	65	74	66	74	80	72	58	54	39,8	26,3	25,8	29,5	38,4	42,1	23	15,1	0

Fig. 1 Distance matrix. Source: authors

Figure 1 shows that the model example includes a total of 19 vertices. These are marked as vertices V0 – V18; V0 represents an original point from where the remaining vertices are served. It means that the proposed travel route must start and end in this vertex.

Another step of the solution is to prepare input data that will be entered directly into the optimization module as a optimization criterion and as variables. These data need to be interlinked using appropriate functions so that the optimization module could sequence vertices within a travel route on the basis of minimizing the optimization criterion value. For the purpose of preparing input data, the table which can be seen in Figure 2 has been created on the MS Excel spreadsheet. The table shows all formulas used to interlink necessary data.

Figure 2 shows that the table includes a total of 3 columns. The column Z with the name "Sequence" contains variable cells. Variables in these cells represent individual vertices taken from the distance matrix. The whole column shows the sequence of visits to individual vertices. Vertices are marked by numbers according to their sequence in the distance matrix, i.e. the vertex from the 1st line in the distance matrix is marked with 1, the vertex on the 2nd line with 2, etc.

The column AA with the name "Distance" sets the distance between the vertices indicated in the column Z. To this end, the INDEX function was used to find the distance between relevant vertices in the distance matrix. Using this function, the cells AA8:AA25 show the distance between vertices $Z(N)$ and $Z(N+1)$ in column Z. The cell AA8, for instance, shows the distance between vertices in cells Z7 and Z8. In the cell AA7, the INDEX function shows the distance between vertices present in cells Z25 and Z7, whereby the round trip is closed.

	Z	AA	AB
6	**Sequence**	**Distance (km)**	**Outcome**
7	1	=INDEX($F$7:$X$25;Z25;Z7)	=INDEX($E$7:$E$25;Z7)
8	2	=INDEX($F$7:$X$25;Z7;Z8)	=INDEX($E$7:$E$25;Z8)
9	3	=INDEX($F$7:$X$25;Z8;Z9)	=INDEX($E$7:$E$25;Z9)
10	4	=INDEX($F$7:$X$25;Z9;Z10)	=INDEX($E$7:$E$25;Z10)
11	5	=INDEX($F$7:$X$25;Z10;Z11)	=INDEX($E$7:$E$25;Z11)
12	6	=INDEX($F$7:$X$25;Z11;Z12)	=INDEX($E$7:$E$25;Z12)
13	7	=INDEX($F$7:$X$25;Z12;Z13)	=INDEX($E$7:$E$25;Z13)
14	8	=INDEX($F$7:$X$25;Z13;Z14)	=INDEX($E$7:$E$25;Z14)
15	9	=INDEX($F$7:$X$25;Z14;Z15)	=INDEX($E$7:$E$25;Z15)
16	10	=INDEX($F$7:$X$25;Z15;Z16)	=INDEX($E$7:$E$25;Z16)
17	11	=INDEX($F$7:$X$25;Z16;Z17)	=INDEX($E$7:$E$25;Z17)
18	12	=INDEX($F$7:$X$25;Z17;Z18)	=INDEX($E$7:$E$25;Z18)
19	13	=INDEX($F$7:$X$25;Z18;Z19)	=INDEX($E$7:$E$25;Z19)
20	14	=INDEX($F$7:$X$25;Z19;Z20)	=INDEX($E$7:$E$25;Z20)
21	15	=INDEX($F$7:$X$25;Z20;Z21)	=INDEX($E$7:$E$25;Z21)
22	16	=INDEX($F$7:$X$25;Z21;Z22)	=INDEX($E$7:$E$25;Z22)
23	17	=INDEX($F$7:$X$25;Z22;Z23)	=INDEX($E$7:$E$25;Z23)
24	18	=INDEX($F$7:$X$25;Z23;Z24)	=INDEX($E$7:$E$25;Z24)
25	19	=INDEX($F$7:$X$25;Z24;Z25)	=INDEX($E$7:$E$25;Z25)
26		**Total length**	=SUMA(AA7:AA25)

Fig. 2 Solving table. Source: authors

The column AB "Outcome" summarizes the travel route created. This column will contain the names of vertices taken from the distance matrix according to the sequence they are visited within the travel route designed. To this end, the INDEX function has been used in the cells AB7:AB25 to select the designation of a relevant vertex from the distance matrix.

The cell AB26 contains the SUM function which sums the values from the cells AA7:AA25. This cell shows the total length of a travel route and, therefore, represents an optimization criterion whose value needs to be minimized.

Now it is possible to enter the parameters in dialogue box "Solver Parameters". The cell AB26, i.e. the cell containing the optimization criterion, will be chosen as an objective function. Then it is necessary to search for the objective function minimum. Then, the variables of the optimization module (i.e. the cells Z7:Z25) need to be entered. In the end, the constraints must be entered as well. The constraints have to ensure that the variable cells obtain various values in a range of integers given by the number of vertices being served, i.e. numbers 1 to 19. This will make sure that each vertex will be visited only once and variables will obtain only the values which have been chosen for designating individual vertices. To enter the constraints, we will display the dialogue box "Add Constraint". We will enter the cells containing the variables and choose the option "various" to make sure that each cell will obtain various values from the interval set by the number of cells selected for variables.

Once the constraints are submitted, it is necessary to select the solving method in the dialogue box "Solver Parameters". Whereas the solving algorithm works with a nonlinear model, we will select the evolutionary algorithm to solve the problem.

Once all necessary parameters are set, the optimization module can be run by pressing the button "Solve". Solver will then start to search for the minimum value of the optimization criterion gradually in individual iterations. After the search is finished, the option "Maintain the Solver´s Solution" needs to be selected from the submitted options. The optimization criterion and the variables on the spreadsheet will be then adjusted accordingly. The outcomes obtained are shown in Figure 3.

	Z	AA	AB
6	**Sequence**	**Distance (km)**	**Outcome**
7	12	5,7	V11
8	2	32,1	V1
9	5	16,7	V4
10	3	9,1	V2
11	4	8,4	V3
12	6	8,9	V5
13	7	12,3	V6
14	8	7,3	V7
15	9	15,3	V8
16	10	4,4	V9
17	11	15,8	V10
18	13	26,9	V12
19	14	10,6	V13
20	15	9	V14
21	16	14,7	V15
22	1	33,5	V0
23	18	13,4	V17
24	19	15,1	V18
25	17	23	V16
26	**Total length**		**282,2**

Fig. 3 Final design of the travel route. Source: authors

The column AB in the table in Figure 3 shows the final sequence in which individual vertices within the designed travel route will be visited. The cell AB26 shows the entire calculated length of a travel route, i.e. 282.2 km.

5. Results and Discussion

Now we can compare the travel route designed by Solver and the travel route proposed using the originally employed intuitive method. A comparison of the results achieved using both methods is shown in Table 1.

Table 1 Method Comparison. Source: authors

Method	Route	Route length [km]	Solution time [s]
Intuitive	V0-V1-V2-V3-V4-V5-V6-V7-V8-V9-V10-V11-V12-V13-V14-V15-V16-V17-V18-V0	357.6	3,600
Solver	V0-V17-V18-V16-V11-V1-V4-V2-V3-V5-V6-V7-V8-V9-V10-V12-V13-V14-V15-V0	282.2	28

Table 1 shows proposals of the travel routes obtained using both methods. Based on these proposals, it was found that a 357.6 km route was designed using the originally employed intuitive method. A route of 282.2 km was designed using the Solver. Based on the results achieved, we can conclude that by applying the Solver optimization module to designing the travel route for beer distribution, a total saving of 75.4 kilometers that have to be travelled will be achieved as compared to the original design. Considering that the number of kilometers travelled is directly proportional to the costs of one kilometer travelled, the solution found by Solver will save 21 % of costs spent on serving the customers as opposed to the original proposal. Table 1 also shows that the use of the Solver has significantly reduced the time needed to design travel route. While using the intuitive method, the design time was 1 hour, the design time using the Solver was 28 s. In doing so, the time required for entering data in the "Solver Parameters" dialog box and the solution time itself, which was 1.5 s, are included in total time of the route designed using the Solver.

6. Conclusion

Using the practical example of beer distribution, the article has demonstrated the algorithm which is used by the Solver optimization module for solving the single-circuit transport problems. This algorithm makes it possible to solve the problems with up to 200 vertices among which the round trip needs to be created. The algorithm is, hence, suitable for solving more simple types of vehicle routing problems only. Within the Solver, the evolutionary algorithm, which belongs to the metaheuristic methods for solving optimization problems, has been chosen to solve the problem. This algorithm allows finding the suboptimal solution only. The quality of this suboptimal solution depends on the computation duration. In the Solver´s basic setup, the evolutionary algorithm is set to search for a solution that is better than the currently found solution in each iteration exactly for 30 seconds. In case of more complex problems, it is useful to extend this time-limit to find the solution which will converge as far as possible with the optimal one. As to the model example, the basic setup was sufficient for finding a better solution than the original one. The benefit of the Solver application can be seen not only in the quality of the solution found, but also in the speed of finding the solution that is much more faster than in case of employing manual procedures, as demonstrated by the model example.

Possible application of Solver is not limited to finding a solution to the travelling salesman problem only. It can also be applied to solving the multi-circuit transport problems with various constraints. Solving these problems requires making more complex algorithms whose solutions are very compute-intensive. Designing and applicability of these algorithms will be subject to further examination. In this context, it can be noted that also the Premium Solver optimization module may

be employed. It is a more powerful version of Solver, allowing solving the problems containing a bigger amount of variables.

References

[1] Linda, B. & Volek, J. (2011). Linear programming (4th ed.). Pardubice, Czech Republic: University of Pardubice.

[2] Holoubek, J. (2006). Economic-mathematical methods. Brno, Czech Republic: Mendel University of Agriculture and Forestry.

[3] Yang, J., Shi, X., Marchese, M. & Liang, Y. (2008). An ant colony optimization method for generalized TSP problem. Progress in Natural Science. 18(11), 1417-1422. DOI: 0.1016/j.pnsc.2008.03.028.

[4] Jablonský, J. (2007). Operations research: quantitative models for economic decision making (3rd ed.). Praha, Czech Republic: Professional Publishing.

[5] Pelikán, J. (2001). Discrete models in operational research. Brno, Czech Republic: Professional Publishing.

[6] Fylstra, D., Lasdon, L., Watson, J. & Waren, A. (1998). Design and use of the Microsoft Excel Solver. Interfaces. 28(5), 29-55. DOI: 10.1287/inte.28.5.29.

[7] Walsh, S. & Diamond, D. (1995). Non-linear curve fitting using Microsoft Excel Solver. Talanta. 42(4), 561-572. DOI: 10.1016/0039-9140(95)01446-i.

[8] Anbuudayasankar, S. P., Ganesh, K. & Mohapatra, S. (2016). Models for practical routing problems in logistics: design and practices. Cham, Germany: Springer.

[9] Desrochers, M., Desrosiers, J. & Solomon, M. (1992). A new optimization algorithm for the vehicle routing problem with time windows. Operations research. 40(2), 342-354. DOI: 10.1287/opre.40.2.342.

[10] Čejka, J. & Stopka, O. (2018). Effective Solutions to the Transport Distribution of Material by the Mayer Method. Advances in Science and Technology Research Journal. 12(4), 177-183. DOI: 10.12913/22998624/100364.

[11] Duan, C. J., Hu, J. & Garrott, C. (2016). Using Excel Solver to solve Braydon farms' truck routing problem: A case study. South Asian Journal of Management. 10(1), 38-47. DOI: 10.21621/sajms.2016101.04.

[12] Erdoğan, G. (2017). An open source spreadsheet solver for vehicle routing problems. Computers & operations research. 84, 62-72. DOI: 10.1016/j.cor.2017.02.022.

[13] Friebelová, J. (2008). Selected chapters from operational analysis. České Budějovice, Czech Republic: University of South Bohemia, Faculty of Economics.

[14] Patterson, M. C. & Harmel, B. (2003). An algorithm for using Excel Solver© for the traveling salesman problem. Journal of Education for Business. 78(6), 341-346. DOI: 10.1080/08832320309598624.

[15] Patterson, M. C. & Harmel, B. (2005). Solving the Traveling Salesman Problem using Premium Solver Platform Software. International Journal of Management. 22(4), 532.

[16] Stopka, O., Stopková, M. & Kampf, R. (2019). Application of the Operational Research Method to Determine the Optimum Transport Collection Cycle of Municipal Waste in a Predesignated Urban Area. Sustainability. 11(8), 2275. DOI: 10.3390/su11082275.

[17] Šedivý, J. (2019). Design of transport model of beer distribution in Samson company with support of mathematical methods. Unpublished thesis, Institute of Technology and Businesses in České Budějovice, České Budějovice, Czech Republic.

[18] Šubrt, T. (2000). Economic Mathematical Methods II: Application and Exercise. Praha, Czech Republic: Credit.

[19] Šubrt, T. (2015). Economical-mathematical methods (2nd ed.). Plzeň, Czech Republic: Publisher Aleš Čeněk.

[20] Získal, J. & J. HAVLÍČEK. (2010). Economic Mathematical Methods II: Study texts for distance study (2nd ed.). Praha, Czech Republic: Czech University of Life Sciences.