The Travelling Salesman Problem and its Application in Logistic Practice

EXNAR FILIP, MACHAČ OTAKAR
Department of economics and management of chemical and food industry
University of Pardubice
Studentská 95, 532 10 Pardubice
CZECH REPUBLIC
Filip.Exnar@upce.cz http://www.upce.cz

Abstract: - The article describes The Travelling Salesman Problem as a logistic transport task. The first part defines the TSP as a mathematical model and briefly describes main established methods of solving the problem. The second part brings experience with practical task solutions in a distribution company within specific conditions and other requirements of the transport management in the company.

Key-Words: - Travelling Salesman Problem, Genetic Algorithm, Objective Function, Constraints in Practice, Transport Management.

1 Introduction

The Travelling Salesman Problem (TSP) belongs in logistic management to the oldest and most frequently solved tasks for which new model solutions have been constantly sought. In a basic version it means a simple problem, easy to define, however, in a certain extent it becomes difficult to solve. The basic situation is given by existence of a certain starting point (basis, let us label it e.g. A_1) from which a product is supposed to be distributed to (n-1) other places (let us label them A_2, A_3, … A_n) so that every place is visited just once and the travelling sales person finishes their route back again at the basis A_1 and the objective value (e.g. travel distance in km) is minimal.

Theoretical solution of the basic model is easy. It is sufficient to explore all possible routes (i.e. sequence of visits to particular places) and find out which of them provides the minimum value of the objective function. If we are to visit *(n-1)* places in sequence, then the total number of all possible routes (closed circuits) is given by a permutation of the number (n-1)!. For a low-value *n* these routes can be explicitly defined, but with an increasing number of visited places the total number of all possible routes rises sharply. For example for (n-1) = 6 places, there are 720 possible circuits, for (n-1) = 12, there are 479 001 600 and for (n-1) = 18, there are already hardly imaginable $6{,}402 \cdot 10^{15}$ various routes.

In logistics numerous possibilities and practical examples of application of the travelling sales person model can be found. Typically, e.g. distribution of food products from producers to shops, distribution of fuel to petrol stations, distribution of various products from producers or distributors to customers, visits of doctors at patients´ homes, orbital inspection walks or check-ups starting at a certain point and leading through individual checking points etc.

2 Formulating the travelling salesman problem

Literature provides numerous differing methods and approaches in terms of the TSP solutions.

From a theoretical point of view the TSP is seen by some authors as a specific problem within the theory of graphs and networks, because visited places can be viewed as certain nodes and the network edges create transport links between them. ([1],[2] etc.) In case the nodes are particular places in a certain geographical territory, they can be perceived as destinations in a map and a contemporary navigation technology can be used for the solution.

From a methodological viewpoint the task is often solved as a problem of integer linear programming with specific conditions. There is a noticeable connection with assignment models used in solving certain problems of production planning.

Let us label the variable x_{ij} as a link between the nodes i and j and acquiring only two values:

$x_{ij} = 1$ if the transport is carried via the route (i,j) and

$x_{ij} = 0$ if the transport is not carried via the route (i,j)
and further let the c_{ij} = objective function rate on the route (i,j).
Then the TSP model can be formally expressed as follows:
Minimize objective function $z = \sum_i \sum_j x_{ij}c_{ij}$
Respecting constraints: $\sum_i x_{ij} = 1$ for i = 1,2, ... n
$\sum_j x_{ij} = 1$ for j = 1,2, ... n
$x_{i1,j1} + x_{i2,j2} + ... + x_{in,jn} = n$
for $j_1 = i_2, j_2 = i_3, ... j_n = i_1$
The last constraint introduces a requirement of a sequenced route, where each place is visited just once and all the route makes a closed circuit finishing in the starting point again. [3]
A basic model can be described also as a graph of nodes (places) and edges (connecting) between them and these connections can be put in a matrix of distances between nodes.

3 Methods of solving a TSP

Literature cites a whole range of TSP methods differing in various approaches to the solutions, efficiency of the procedures and also the outcomes. Let us quote brief characteristics of the most often used ones.

Method of total enumeration
In principle it is a combinatorial solution. The method rests in evaluation of all potential routes (sequences) in the total number of (n – 1)!. The advantage is that a global optimum is always found, however, it is not employable if higher numbers of visited places are considered. With every added element (node) the amount of possible solutions grows exponentially and not even nowadays do we have computers powerful enough for being able to provide optimum solution within reasonable time. [4]

Method of branches and bounds
This method belongs to the oldest ones and the most often used algorithms for the TSP solutions. The merit of the method rests in a gradual decomposition of a possible solution set into a number of mutually disjunctive subsets labelled as branches. In each step the following is estimated:

- The upper limit of the objective function that is most often the value of the objective function z_H without respecting limits and
- Maximum lower bound of an objective function z_D of acceptable solutions which are known to us within the step.

Both the estimates can be employed for seeking non-prospective directions of further procedures: if for any branch $z_H < z_D$, then the given direction can be excluded. However, this method is also, especially for higher *n*, too laborious and does not always guarantee an optimum solution at the first attempt. For more detail see e.g. [5]

Efficient algorithm of Clarke and Wright
A significant progress in TSP solutions was provided by the Clarke's and Wright's method. The initial situation assumes that each place is supplied individually and always a return to the starting base follows. The essential idea is based on the calculation of economies achieved through integrating other places into the circular route. An indisputable asset of this algorithm is its function to respect further restrictions often generated by the practice, e.g. the need to optimize more orbital routes, to use more vehicles while respecting their various capacities etc. (For detail see [4])
Guerra, Murino and Romano worked with this algorithm for optimize the routing phase in a Location-Routing Problem (LRP) in [6]. LRP can by assimilate to a Vehicle Routing Problem (VRP) and after that they combine and balance VRP with TSP. Both problems were solved with Clarke and Wright saving algorithm and the Branch and Bound model.

Computer Simulation
The development of simulation models, their program support and increasing computing power brought about attempts at use of simulation techniques for solving large TSPs. Their merit rests in random PC sampling in large scale that is later evaluated according to a selected objective function. Though the solution does not guarantee the global optimum, a sufficiently large number of simulations will issue in achieving the best possible solution, the value of which will be close to the optimum [7].

Ant Colony Optimization Algorithm (ACO)
ACO is one of the metaheuristic methods for solving TSP. Jalali, Afshar and Marino [8] described ACO as observation of real ants, and upon finding food return to their colony while laying down pheromone trails. If other ants find such a path, they are likely not to keep travelling at random, but to instead follow the trail, returning and reinforcing it if they eventually find food. There is a higher probability that the trail with a higher pheromone concentration will be chosen. The pheromone trail allows ants to find their way

back to the food source and in the opposite way. The trail is used by other ants to locate food source discovered by any ant. When a number of paths available from the nest to a food source, a colony of ants may by able to exploit the pheromone trail left by individual members of the colony to discover the shortest path from the nest to the food source and back. As more ants choose a path to follow, the pheromone on the path builds up, making it more attractive to be followed by other ants.
To solve TSP, we keep the strength of pheromone trail $\tau_{i,j}$ for each combination of two points. The role of each ant is to find a valid solution, thus possible routes. From the starting point, the ant gradually repeats a move when it chooses a place where it has not been yet while moving to it from its current location. Once there are no more vacancies left, the ant returns to the starting location. As a result, the ant keeps its path *T*. If the ant *k* is currently at the location *i*, then the probability that it goes to the city *j* is

$$P_{ij}^{k} = \frac{(\tau_{ij})^{\alpha}(\eta_{ij})^{\beta}}{\sum (\tau_{ij})^{\alpha}(\eta_{ij})^{\beta}}$$

Where τ^{k}_{ij} is the total pheromone deposited on path *ij*, η^{k}_{ij} is the heuristic value of path *ij* according to the measure of the objective function (a priori knowledge, typically $1/c_{ij}$, where c_{ij} is distance). α,β are parameters that control the relative importance of the pheromone trail versus heuristic value. When all the ants have completed a solution, the trails are updated by $\tau_{ij}^{k} = (1-\rho)\tau_{ij}^{k} + \Delta\tau_{ij}^{k}$
where ρ is the pheromone evaporation coefficient and $\Delta\tau^{k}_{ij}$ is the amount of pheromone deposited, for a TSP problem by Q/L_k if ant *k* uses way *ij* in its tour, where L_k is the cost of the *k*th ant's tour (typically length) and *Q* is a constant.
Ant colony optimization algorithms have been used to produce near-optimal solutions to the travelling salesman problem. The first ACO algorithm was aimed to solve the travelling salesman problem, in which the goal is to find the shortest round-trip to link a series of cities. It is able to find the global optimum in a finite time.
Rudeanu and Craus [9] presented Parallel implementation of ant colony optimization that is faster and more efficient. It's a framework based on the message-passing communication paradigm (MPI). MPI is a language-independent communications protocol used to program parallel computers. Another using of parallel algorithm using MPI was presented in [10].
ACO is one of the Swarm Intelligence systems which include many other algorithms such as Particle Swarm Optimization and River formation dynamics [11].

Particle Swarm Optimization Algorithm (PSO)
PSO proceed from the social behavior of organisms such as bird flocking and fishing schooling. Through cooperation between individuals, the group often can achieve their goal efficiently and effectively. PSO simulates this social behavior as an optimization tool to solve some optimization problems. Each particle flies in the search space with a velocity that is dynamically adjusted based on its own flying experience and its companions' flying experience. In other word, every particle will utilize both the present best position information of its own (*pbest*) and the global best position information (*gbest*) that swarm has searched up-to-now to change its velocity and thus arrives in the new position.
More about this algorithm and its mathematical description is in [12].

Genetic algorithms
In recent years there have been attempts to use so called genetic algorithms for TSP solutions. Simply stated, genetic algorithms transfer evolution principles in living organisms into intelligent searching and model optimization in other fields. Biological terminology is applied also to this very description of the algorithm. Genetic algorithm, as well as nature, works with *population* of individuals (P) defined by one or more mathematical *genes – chromosomes* (i.e. sequences of numbers in binary notation).
The genetic algorithm (GA) uses the following steps:
1. *Generate a population*. The GA randomly samples values of the changing cells between the lower and upper bounds to generate a set of (usually at least 50) chromosomes. The initial set of chromosomes is called the *population*.
2. *Create a new generation*. In the new generation, chromosomes with a smaller fitness function (in a minimization problem) have a greater chance of surviving to the next generation. Crossover and mutation are used to generate chromosomes for the next generation.
3. *Stopping conditions*. At each generation, the best value of the fitness function in the generation is recorded, and the algorithm repeats step 2. If no improvement in the best fitness value is observed after many consecutive generations, the GA terminates.(For detail see [1])
Technically, the GA is the sort of a simulation model. For solving of GA and other specific tasks

in Excel there was developed a new Evolutionary Solver. This Evolutionary Solver is available since Excel 2007 and later versions. The Evolutionary Solver is good for solving of "nonsmooth" problems with more local extremes or for solving of combinatorial models with few constrains. Respecting the simulation background of GA, the Evolutionary Solver finds usually a very good solution, but there is no guarantee that it will find the *bes*t solution. Evolutionary Solver doesn't handle constraints well and therefore it is usually better to penalize constraints violations and include the penalties in the objective. Because of the solution process is driven by random numbers, two different runs can lead to different solutions. In spite of these "weaknesses", the Evolutionary Solver is an effective tool for solving of GA and similar problems.
More authors propose new modifications and improvements of GA. For example, an effective parallel model was presented from Bai Xiaojuan and Zhou Liang in [13]. It is based on the traditional genetic algorithm, but a new operating mechanism of GA was improved means of adaptive crossover and mutation. It uses probability of GA, which can keep the solution space effective. Further, 2-opt neighbourhood search optimization techniques are imported, which can ensure the evolution process is not stagnation, and improve the efficiency of solving. Another improving proposals of Genetic Algorithm were presented in [14] [15]. The interest of many authors about GA proved to be the promising and effective solving method for the specific group of optimization models.

Solutions in geographically oriented databases
Significant advantage in solutions for orbital transport problems can be seen in software products using geographical databases or maps. For example, the MapInfo Professional software can find, inter alia, an optimum transport vehicle circuit in the map by means of its tool "Rote View", just on the basis of labelling the starting point and the places to be visited. After a selection has been done from various offered criteria (e.g. minimum travel distance or minimum travel time), the screen directly shows the optimum route. [5]
The asset of these software types subsists also in the capacity to reflect other limiting conditions which are inherent to particular tasks in the working practice. Their drawback consists in a relatively high price.

4 Application of the TSP in logistic practice

When solving real travelling salesman problems in logistic practice a lot of other constraints occur, modifying the described basic model and imposing new requirements on it. In some cases new limitations can be considered within the model, in other situations these requirements have to be solved outside the model.
When constructing a model for particular situations these varied conditions need to be respected to ensure a successful model implementation.
Above all, there is a need for distinguishing between situations when a circular route is always fixed to the same number of given places or when the circular routes vary for each distribution transport. In the first case, the task can be solved once forever, in the second case it is inevitable to count on planning and optimizing each distribution delivery all over again. In this case the early and correct demand forecasting and planning can contribute to planning improvement of products transport to clients. [16]
In his thesis [17], the author encountered a problem where a company was to distribute daily a variety of chemical products under orders to about 20 customers virtually all around the Czech Republic. This set of customers was different every day as some customers occurred therein more frequently while some only rarely or once. Optimization should therefore be carried out every day. However, the everyday application should be supported by a simple and suitable type of software.

4.1 Objective Function Selection Issue

The most commonly used objective function is minimization of the distance, i.e. the total number of miles driven. If we disregard the fact that the distances obtained from the GPS system typically do not copy exactly the distance according to the road network, the criterion of the minimum of the distances travelled does not always have to be satisfactory. In some cases, the carrier may prefer the criterion of minimum time or cost. In both the cases, however, it is much more difficult to obtain data to the input matrix of the coefficients.
The matrix of objective function rates between the starting point, marked with index 1, and all the customers may be found on the basis of the addresses of customers, for example using the route planner at www.mapy.cz. In this route planner we have two choices at first. We can choose between finding the shortest or fastest connections of the

places. In both cases, we obtain data on the connection length in km and, at the same time, the duration of the connection in minutes. In total, we can get up to 4 matrixes of the objective function rates: the matrix of distances in km found for the shortest connection, the matrix of time in minutes found for the shortest connection, the matrix of distances in km found for the fastest connection and the matrix of time in minutes found for the fastest connection. The matrices found for the same type of the objective function (by the route planner) express one connection along the same route in two different units (km and minutes). In contrast, when comparing two matrices with the same units (but found by to the route planner in a different way) we can easily find out that the connection two same points may take a different path (different distance and time for the same connection). Each of these matrices can therefore give a different optimal solution! For a sample solution of one selected day when 15 customers were to be served (Fig. 1), we found the following four matrices of the objective function rates (Table 1-4).

Fig. 1: Customers location (blue point represents starting point)

Table 1: Matrix of distances in km found for the shortest connection

i \ j	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	0	138,2	25,9	137,3	130,2	63,9	118,2	121,8	193,6	180,4	180,6	127,1	180,5	197,6	179,7	82,6
2	138,2	0	134,2	57,4	246,8	122,3	245,2	248,9	105,7	71,6	71,8	254	71,8	109,8	70,9	205,7
3	25,9	134,2	0	124,4	149,3	80,4	119,5	123,1	179,4	167,4	167,7	128,4	167,6	183,5	166,8	95,9
4	137,3	57,4	124,4	0	261,1	152,7	240,4	244	63,3	43	43,3	249,3	43,2	67,3	42,4	218,4
5	130,2	246,8	149,3	261,1	0	131,8	97,7	93,9	317,4	301,2	301,5	86,8	301,4	321,5	300,6	72,3
6	63,9	122,3	80,4	152,7	131,8	0	145	148,7	208,7	178,5	178,8	153,9	178,7	212,8	177,9	105,5
7	118,2	245,2	119,5	240,4	97,7	145	0	5,5	292	281,8	282,1	10,7	282	296	281,1	39,9
8	121,8	248,9	123,1	244	93,9	148,7	5,5	0	290	285	285,2	7,1	285,1	294	284,3	44,6
9	193,6	105,7	179,4	63,3	317,4	208,7	292	290	0	43,8	41,3	296,8	45,7	7	43,5	273,3
10	180,4	71,6	167,4	43	301,2	178,5	281,8	285	43,8	0	4,5	291,5	1,9	40,8	2,6	260,4
11	180,6	71,8	167,7	43,3	301,5	178,8	282,1	285,2	41,3	4,5	0	291,7	5	38,9	2,6	260,7
12	127,1	254	128,4	249,3	86,8	153,9	10,7	7,1	296,8	291,5	291,7	0	291	299,9	290,2	49,5
13	180,5	71,8	167,6	43,2	301,4	178,7	282	285,1	45,7	1,9	5	291	0	42,6	3,3	260,5
14	197,6	109,8	183,5	67,3	321,5	212,8	296	294	7	40,8	38,9	299,9	42,6	0	40,8	277,5
15	179,7	70,9	166,8	42,4	300,6	177,9	281,1	284,3	43,5	2,6	2,6	290,2	3,3	40,8	0	259,7
16	82,6	205,7	95,9	218,4	72,3	105,5	39,9	44,6	273,3	260,4	260,7	49,5	260,5	277,5	259,7	0

Table 2: Matrix of time in minutes found for the shortest connection

i \ j	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	0	161	31	155	124	73	137	140	206	206	207	148	207	207	205	90
2	161	0	137	61	240	137	222	232	99	88	88	234	88	100	87	193
3	31	137	0	132	136	90	130	134	161	183	184	142	184	163	182	94
4	155	61	132	0	263	156	251	255	62	51	52	262	52	63	50	250
5	124	240	136	263	0	131	89	86	314	286	286	76	286	315	285	85
6	73	137	90	156	131	0	125	125	195	181	182	137	182	196	180	95
7	137	222	130	251	89	125	0	8	208	299	300	16	300	219	298	30
8	140	232	134	255	86	125	8	0	203	211	211	10	211	205	210	37
9	206	99	161	62	314	195	208	203	0	49	43	199	52	6	47	248
10	206	88	183	51	286	181	299	211	49	0	7	204	3	45	4	237
11	207	88	184	52	286	182	300	211	43	7	0	204	8	44	5	237
12	148	234	142	262	76	137	16	10	199	204	204	0	208	201	206	43
13	207	88	184	52	286	182	300	211	52	3	8	208	0	47	6	237
14	207	100	163	63	315	196	219	205	6	45	44	201	47	0	45	248
15	205	87	182	50	285	180	298	210	47	4	5	206	6	45	0	236
16	90	193	94	250	85	95	30	37	248	237	237	43	237	248	236	0

Table 3: Matrix of distances in km found for the fastest connection

i \ j	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	0	149,8	26,2	160,3	133,1	66,2	129	126,2	215,6	207	205,9	133,2	206,5	219,6	205,2	89,5
2	149,8	0	141,7	58,4	255,9	139	310,5	307,7	110,9	80,8	80,6	314,7	80,3	115	79,4	212,4
3	26,2	141,7	0	134,5	151,5	82,2	145,2	142,3	189,7	181,2	180	149,4	180,7	193,8	179,4	96,3
4	160,3	58,4	134,5	0	280,6	171,7	268,4	265,5	67,1	44,9	54,4	272,6	44,4	71,2	43,5	237,1
5	133,1	255,9	151,5	280,6	0	146,9	101,4	97,3	393,1	329,4	328,2	90,2	328,9	397,1	327,6	81,5
6	66,2	139	82,2	171,7	146,9	0	145,9	150,1	229,6	221	219,9	155,3	220,6	233,7	219,2	106,4
7	129	310,5	145,2	268,4	101,4	145,9	0	5,5	300,9	297,5	296,3	11,6	297	305	295,7	40,3
8	126,2	307,7	142,3	265,5	97,3	150,1	5,5	0	299,4	296	294,9	7,9	295,5	303,5	294,2	45,6
9	215,6	110,9	189,7	67,1	393,1	229,6	300,9	299,4	0	51	42,8	305,3	47,9	7	45,2	294,8
10	207	80,8	181,2	44,9	329,4	221	297,5	296	51	0	4,6	301	1,9	56,1	2,6	285,4
11	205,9	80,6	180	54,4	328,2	219,9	296,3	294,9	42,8	4,6	0	300,4	5,1	46,8	2,8	284,9
12	133,2	314,7	149,4	272,6	90,2	155,3	11,6	7,9	305,3	301	300,4	0	301	308,9	299,6	50,7
13	205,5	80,3	180,7	44,4	328,9	220,6	297	295,5	47,9	1,9	5,1	301	0	53	3,5	285,7
14	216,6	115	193,8	71,2	397,1	233,7	305	303,5	7	56,1	46,8	308,9	53	0	49,2	298,9
15	205,2	79,4	179,4	43,5	327,6	219,2	295,7	294,2	45,2	2,6	2,8	299,6	3,5	49,2	0	284,1
16	89,5	212,4	96,3	237,1	81,5	106,4	40,3	45,6	294,8	285,4	284,9	50,7	285,7	298,9	284,1	0

The optimal solutions were found using the Solver Evolution, which works on the principle of genetic algorithm. Table 5 shows that we have three different "optimal" routes in the order of places visited, and even four different "optimal" routes by the actual route used. The "correct" optimal route then depends on the choice of the objective function and on the way of finding the rates matrix. Optimization by distance found using the shortest route results in the absolutely shortest route (814,4 km), but on the other hand, the most time-consuming route at the same time (857 minutes). In contrast, by optimizing the time, found using the fastest route, we get the absolutely fastest route (756 minutes). This route, however, is the longest one in terms of distance (944,9 km). The other two optimization methods are a kind of compromise between the shortest time and the shortest distance.

Table 4: Matrix of time in minutes found for the fastest connection

i \ j	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	0	123	30	113	106	67	101	97	136	135	131	99	135	137	133	74
2	123	0	120	60	207	180	184	179	91	76	76	181	75	92	75	175
3	30	120	0	110	130	86	101	96	133	132	129	99	132	135	131	93
4	113	60	110	0	207	187	221	216	53	48	49	159	47	55	46	176
5	103	207	130	207	0	123	70	65	223	228	225	58	228	225	226	73
6	67	180	86	187	123	0	122	127	149	148	145	131	148	150	146	93
7	101	184	101	221	70	122	0	8	172	178	175	13	178	173	177	30
8	97	179	96	216	65	127	8	0	168	174	171	8	174	169	173	36
9	136	91	133	53	223	149	172	168	0	37	31	170	38	6	35	199
10	135	76	132	48	228	148	178	174	37	0	7	177	3	39	4	197
11	131	76	129	49	225	145	175	171	31	7	0	176	8	33	5	196
12	99	181	99	159	58	131	13	8	170	177	176	0	176	171	175	40
13	135	75	132	47	228	148	178	174	38	3	8	176	0	40	6	197
14	137	92	135	55	225	150	173	169	6	39	33	171	40	0	37	200
15	133	75	131	46	226	146	177	173	35	4	5	175	6	37	0	196
16	74	175	93	176	73	93	30	36	199	197	196	40	197	200	196	0

Table 5: Solving by Solver Evolution

used matrices of different objective functions	optimal route	time [min]	distance [km]
distances found for the shortest connection	1 3 4 9 14 11 15 10 13 2 6 5 12 8 7 16 1	857	814,4
time found for the shortest connection	1 3 9 14 11 15 10 13 4 2 6 16 7 8 12 5 1	847	856,2
distances found for the fastest connection	1 3 4 9 14 11 15 10 13 2 6 5 12 8 7 16 1	800	888,5
time found for the fastest connection	1 3 2 4 13 10 15 11 9 14 6 16 7 8 12 5 1	756	944,9

4.2 Capacity Limitations of the Model in Practice

The solutions presented, however, proved unfeasible in practice due to capacity constraints. Above all, it was not possible to serve all customers in one day with one car, both in terms of the capacity of the car and capabilities to handle this circuit in one day.

In this case, there is a possibility to divide the entire transport request into two or more circuits.

First, there is therefore the task how to best create sub-sets of nodes (points) so that the distances between the nodes within the subsets were smaller than the distances between the nodes of different subsets. Similar tasks are addressed by, for instance, the cluster analysis (see [1], p. 444). If we have locations of customers on the map, we can usually create sub-circuits intuitively. (See fig. 1)

In more complicated cases, the application of the Generalized Travelling Salesman Problem (GTSP) can be employed. The GTSP is a generalization of the Travelling Salesman Problem (TSP), in which the set of nodes is dividend into mutually exclusive clusters. The objective of the GTSP consists in visiting each cluster exactly once in a tour, while minimizing the sum of the routing costs.

In [18] it was proposed a solution method for the Generalized Travelling Salesman Problem based on a memetic algorithm. Memetic algorithm is a genetic algorithm paired with local search techniques. The main contribution of the paper stands in the crossover operator based on the exploration of a large neighbourhood around the father and mother individuals.

Camelia M. Pintea at al presented in [19] an effective metaheuristic algorithm based on ant colony system in the case of the dynamic generalized travelling salesman problem. The dynamism is there at each moment when a cluster, determined with a given probability, is missing from the tour. In the real life frequently appear blocked ways due to poor weather, accidents, maintenance, etc. The objective is to find a minimum cost tour passing through exactly one node from each available cluster.

The company, where the thesis [17] was prepared, historically created two routs, which are, however, modified with each distribution according to customer requirements.

Tab. 6: Distance matrix in km

i \ j	1	9	10	11	13	14	15
1	0	193,6	180,4	180,6	180,5	197,6	179,7
9	193,6	0	43,8	41,3	45,7	7	43,5
10	180,4	43,8	0	4,5	1,9	40,8	2,6
11	180,6	41,3	4,5	0	5	38,9	2,6
13	180,5	45,7	1,9	5	0	42,6	3,3
14	197,6	7	40,8	38,9	42,6	0	40,8
15	179,7	43,5	2,6	2,6	3,3	40,8	0

For a given day, customers are classified into the following categories. One route consists of points 7, 8, 12, 16, with the starting point 1. The second route contains points 9, 10, 11, 13, 14, 15, again with the starting location 1. The remaining customers 2, 3, 4, 5 and 6 were not included in any of the routes. To demonstrate, we give an example of the optimization of the second rout. The starting point is the distance matrix (found for the shortest connection) between the company headquarters (i = 1) and six other customers, who are to be served that day (Table 6).
The optimization used a procedure combining the Hungarian method and the branch and bound method (see: Exnar [17]). It resulted in a recommended circuit 1 – 9 – 14– 11 – 15 – 10 – 13 – 1 which required travelling minimum of 427,1 km. The same result can be obtained by travelling the circuit in a reverse order. The found solution is optimum, as it was verified by means of Solver Evolution in Excel according to [1]. In our case there would not be a problem to verify either an optimum solution or total enumeration, as the total number of routes is merely 6!, i.e. 720 various routes. However, if a larger group of served customers was considered, a more sophisticated procedure integrating accessible suitable software would be necessary. (E.g. above mentioned Solver Evolution according to [1])
Further complications within the model solution can be caused by certain other limits and prerequisites, such as obligatory breaks for drivers or statutory requirements regarding transport of specific products (chemicals, explosives etc.). The model solution can also be impaired by specific requirements of some customers concerning the delivery just within a short and precisely defined time period. The efforts to fully meet the customer's requirements depends i.a. also on their importance for the supplier, i.e. for example on the volume and frequency of orders on regular basis.

4.3 Mix of internal and external transport services

Many production and distribution companies are to solve the strategic decision transport problem: to carry on the transport of products by own vehicles or to use services of external transport firms. Above all, it is an economic task, but not only one. The mix of both strategies is used often.
Our contribution to solving of the problem consists in suggestion and construction a comparing chart that enables, depending on the distance and the weight of the delivery, to decide, if the delivery should to be transported by own vehicles or if it should be better to use the service of a special transport firm.
When solving a diploma paper (Exnar [17]) we have come across a requirement of a company to find out conditions under which it pays to subcontract a forwarder or to integrate a delivery into a circular route ensured by their own delivery vehicle. The criterion should rest in forwarding costs. Ironically enough, it was much easier to find out the forwarding costs charged by a subcontracted company as it calculates the costs from fixed tariffs related to the weight of the delivery and the distance. The calculation of their own forwarding costs is complicated by the fact is consists of various parts: a part of costs is variable (fuel costs), a part is mixed (drivers´ salaries) and another part is fixed (depreciation of vehicles, maintenance and repairs). The author solved the company's requirement by creating a comparing chart which enables to decide when it is still advantageous to integrate the delivery into the circular route, or as the case may be, to forward the delivery individually by their own vehicle or to subcontract an external company. The table is created on the basis of a logical reasoning that for the same money that we would pay to an external company we are able to cover a certain number of kilometres with our own vehicle. The following sample is a part of the comparing chart (Fig 2)

Fig. 2: Comparing chart

Consignment weight in kg	Distance (in km) 50	100	200	300	Distance in km 50	100	200	300
0,1 - 5	36	39	45	54	18	20	22	27
5,1 - 10	52	58	69	72	26	29	35	36
10,1 - 20	68	75	91	95	34	38	45	48
20,1 - 30	81	90	113	117	40	45	56	58
30,1 - 50	113	124	154	165	56	62	77	83
50,1 - 75	139	152	189	205	69	76	95	103
75,1 - 100	161	180	231	251	81	90	115	126
100,1 - 150	202	232	302	329	101	116	151	164
150,1 - 200	228	263	360	371	114	131	180	186
200,1 - 300	291	336	446	492	146	168	223	246
300,1 - 400	345	395	532	588	172	198	266	294
400,1 - 500	390	449	609	677	195	225	304	339
500,1 - 700	457	541	746	832	228	271	373	416
700,1 - 1000	555	654	913	1035	278	327	456	517

Particular boxes in the left part of the chart give distance in km travelled by their own vehicle for variable costs of fuel, which equals tariffs of a subcontractor for forwarding under a given combination of weight and tariff zones. This part of a chart gives the distance which can extend the current circuit and be still economical for a company to carry the transport on their own.

The right part of the chart may serve for decision-making in case of selecting individual means of transport to a particular customer. For this purpose the chart is divided into three parts. The green colour highlights delivery orders with which the preferred transport is always more economical by a company's own vehicle. The yellow boxes show deliveries with which individual transport is economical only if the customer is located in a maximum distance stated in the chart. The white part of the chart shows the delivery orders that should never be served individually. On the basis of this right part of the table we can also specify what is the smallest possible weight of any customer's order (knowing his/her distance from our company) so that it paid out to serve the customer in a separate journey.

In the example from Chapter 4.2 there remained five unserved customers and now we examine what transportation service can be recommended to each customer using Figure 2. On the left side of the comparative table we find distances in kilometres, which can be used to maximally extend the existing circuit to make it still economically advantageous. This value can be found on the basis of the known weight of the shipment and the distance of the customer from our enterprise (the first line in Table 1 in Chapter 4.1) and subsequently we compare it with the distance between the newly incorporated customer and the two closest stops already included in the circuit (see the distance matrix in Table 1).

Practically, there may be two cases:

a) the distance between the new stop and the already included stop + the distance between the new and the second stop already included stop is *greater* than the distance between these two included stops + the value from the left side of the comparative table

b) the distance between the new stop and the already included stop + the distance between the new and the second stop already included stop is *smaller* than the distance between these two included stops + the value from the left side of the comparative table

In case *a)* it does not pay out to include the customer in this circuit, in case *b)* the inclusion of the customer in the circuit pays out because the cost of transportation using an external carrier would be greater than the cost of extending the existing route.

For illustration, let us give an example how to optimally solve the customer's order at point 2 who requires delivery of 50 kg. The best option here seems to be including the point at the end of the circuit between the connection 13 – 1. The distance 2 - 13 is 71,8 km, 2 - 1 is 138,2 km and 13 - 1 is 180,5 km. From the left side of the comparative table (Figure 2) we find out that the maximum possible extension of the relevant route is 154 km. The sum of the distances 2 - 13 and 2 -1 is *smaller* than the sum of 13 - 1 + 154 km (210,0 < 334,5), so it pays economically out to include the client 2 in this circuit. The resulting circuit will then have the following form - 9 – 14– 11 – 15 – 10 – 13 – 2 – 1.

As another example we will address inclusion of the client at the point 5 with a delivery of 4 kg weight again in the second circuit. The best option here seems to be inclusion at the end of the connection between 2 – 1. The distance 2 - 5 is 246,8 km, the distance 1 - 5 is 130,2 km and the distance 2 - 1 is 138,2 km. From the left side of the comparative table we find out that the maximum possible extension of the route is 45 km. The sum of the distances 2 - 5 and 1 - 5 is greater than 2 - 1 + 45 km (377 > 183,2), and therefore it does not pay out economically to include the client 5 in this circuit.

We demonstrate the use of the right side of the comparative table with the example of customer 6 and his/her shipment weighing 70 kg. The distance of this customer from our enterprise is 63,9 km (Table 1). On the right side of the comparative table we have a value of 76 km for the given contract. It applies to the yellow fields that it pays out to transport the given order separately if the customer is in the distance no longer than the one shown in the table. In our case, 63,9 < 76 and therefore it pays out economically to serve the customer 6 even separately without incorporating into the existing circuit.

In the same manner we find out whether it would be worthwhile to go individually to the above mentioned customer 5. Looking to the right side of the table we find a white field with a value of 22 km. It applies to the white fields that it does not pay out to serve the given orders individually. Now we will try to find out what minimum weight an order would have to have for the customer 5 that would make it worthwhile to go to him/her individually. The distance of the company from the customer is 130,2 km; we shall then use the third column (right part of Figure 2 and the fare distance up to 200 km) while looking for a field with the lowest value greater than 130,2. We find the value of 151 for an order weighing 100,1 - 150 kg. The minimum shipment weight for individual serving of the customer 5 is thus 100,1 kg.

For the final decision, however, it is important to take into account considerations other than just economic. When extending the circuit, the permitted loading capacity could, for example, be exceeded or the route could be disproportionately lengthened in terms of time, etc.

The comparing chart in fig. 2 was constructed for the particular situation in the particular company. However, the approach to solving this problem is universal and any company in similar situation can use it. Naturally, it must be adjusted to particular conditions in the concerned company.

5 Conclusion

In the article, we tried to prove that optimization of circular transport task is still topical in the logistic transport management and its use may bring significant benefits to the business, such as saving fuel, time and cost.

Practical applications withal require respecting a range of other specific conditions and requirements so that the solution was effective and usable in practice. Firstly, it is necessary to take into account various capacity constraints of the vehicles and their operators as well as other specific customer requirements. A special case is the option to choose between our own means of transport and external transport. We have proposed an analytical procedure of this problem and consequently we have constructed a comparing chart for decision support of this problem. A prerequisite is also a functional information system that can provide necessary input information and, ultimately, appropriate software to solve ODP, which can repeatedly and effectively carry out the optimization calculation.

[1] Albright, S. CH., Winston, W.L.: *Management Science Modelling*, South-Western Cengage Learning, 2009, ISBN-13:978-0-324-66346-4

[2] Pinney, W.E. and col.: *Quantitative Methods for Management*, Thomson Custom Publishing, 2009, 0-759-33812-4

[3] Jablonský, J.: *Operations research – Quantitative models for economic decision making*, Prague: Professional publishing, 2002, ISBN 80-86419-23-1 (In Czech)

[4] Kovarik V,: *Heuristic Algorithms for decision Support*, Dissertation, Prague: VŠCHT, 2008

[5] Gros, I.: *Quantitative Methods for managerial decision making*, Prague: Grada publishing a.s., 2003, ISBN 80-247-0421-8 (In Czech)

[6] L. Guerra, T. Murino, E. Romano, *The Location - Routing Problem: an innovative approach*, 6th WSEAS International Conference on System Science and Simulation in Engineering, Venice, Italy, November 21-23, 2007

[7] Wisniewski, M.: *Methods for Managerial Decision Making*, Prague: Grada publishing, 1996, ISBN 80-7169-089-9 (In Czech)

[8] Mohammad Reza Jalali, Abbas Afshar, Miguel A. Marino: *Ant Colony Optimization Algorithm (ACO); A new heuristic approach for engineering optimization*, Proceedings of the 6th WSEAS Int. Conf. on Evolutionary Computing, Lisbon, Portugal, June 16-18, 2005, pp. 188-192

[9] Laurentiu Rudeanu, Mitica Craus: *Parallel Implementation of Ant Colony Optimization for Travelling Salesman Problem*, Proceedings of the 4th WSEAS Int. Conf. on Soft Computing, Optimization, Simulation & Manufacturing Systems (SOSM 2004), Miami, Florida, April 21-23, 2004

[10] Izzatdin A. Aziz, Nazleeni Haron, Mazlina Mehat, Low Tan Jung, Aisyah Nabilah: *Solving Traveling Salesman Problem on High Performance Computing using Message Passing Interface*, Proc. of the 7th WSEAS Int. Conf. on Computational Intelligence, Man-Machine Systems and Cybernetics (CIMMACS '08), pp. 83-87, ISBN: 978-960-474-049-9

[11] Rabanal P., Rodríguez I., Rubio F.: *Using River Formation Dynamics to Design Heuristic Algorithms*, Springer, 2007, ISBN 978-3-540-73553-3

[12] Luping Fang, *Pan Chen, Shihua Liu: Particle Swarm Optimization with Simulated Annealing for TSP*, Proceedings of the 6th WSEAS Int. Conf. on Artificial Intelligence, Knowledge Engineering and Data Bases, Corfu Island, Greece, February 16-19, 2007, pp. 206-210

[13] Bai Xiaojuan, Zhou Liang: *A New Genetic Algorithm for solving Travelling Salesman Problem*, Proceedings of the 8th WSEAS International Conference on Applied Computer and Applied Computational Science, 20, pp. 451-455

[14] Fu Yiping, Chen Guangxi, *An improved genetic algorithm for solving TSP problem*, Journal of Guilin University of Electronic Technology, Vol.27, No.4, 2007, pp.287-290

[15] Han Lixia, Wang Yuping, *A Novel Genetic Algorithm for Travelling Salesman Problem*, systems Engineering-Theory & Practice, No.12,2007, pp.145-151

[16] Vlčková, V., Paták, M.: *Role of Demand Planning in Business Process Management*, 6th International Scientific Conference “Business and Management–2010” May 13-14, 2010, Vilnius, Lithuania, Selected papers. Vilnius, 2010, pp. 1119–1126, Volume II, ISSN 2020-4441, Vilnius Gediminas Technical University, 2010. And CD ISSN 2020-4441

[17] Exnar, F.: *Solving of transport problems in Logistics*, Diploma paper, Pardubice: University of Pardubice, 2010 (In Czech)

[18] Boris Bontoux, Christian Arzigues, Dominique Feillet: *A Memetic Algorithm with a large neighbourhood crossover operator for the Generalized Travelling Salesman Problem*, European Journal of Operational Research, Volume 59, Issue 3, 25 June 1992, pp. 345-358, Available online 18 May 2009.

[19] Camelia M. Pintea, Petric C. Pop, D. Dumitrescu: *An Ant-based Technique for the Dynamic Generalized Travelling Salesman Problem,* Proceedings of the 7th WSEAS International Conference on Systems Theory and Scientific Computation, Athens, Greece, August 24-26, 2007, pp. 255-259