Methods for Solving Discrete Optimization Problems

D. Mocková*
Department of Logistic and Transportation Processes, Czech Technical University of Prague, Czech Republic
** Corresponding author: mockova@fd.cvut.cz*

DOI: 10.2478/v10158-012-0009-0

ABSTRACT: The article specifies a group of discrete optimization problems, such as location problems and tour problems, from the aspect of individual approaches (exact, heuristic, and metaheuristic) and seeks to explain all the approaches on specific problems.

KEYWORDS: Discrete optimization problems, heuristics, metaheuristics, P-median problem, P-hub median problem, genetic algorithms.

1 INTRODUCTION

In transportation and logistics systems (network problems) we distinguish two basic types of combinatorial problems. The first group consists of problems whose solution is a well-defined set of items, such as the spatial distribution of facilities (service centres), terminals, warehouses, etc.

The solution to the second group of combinatorial problems is defined as a sequence of visited nodes, or a sequence of customers served. A typical instance of the second group of problems is a VRP (vehicle routing problem). A travelling salesman is the simplest example of vehicle routing.

1. Exact approach

This approach allows an optimal solution to be found, if one exists, unless one can prove that there is no admissible solution to the problem. A great price is paid for precision – especially in computation time - where an extensive problem is concerned. Moreover, it is difficult to estimate the time required for computation purely from the size of the problem. The approach can, for example, prove that it is not possible to construct a Hamiltonian circle.

2. Simple heuristics

This is used to find a good and admissible solution to a problem, or improve existing solutions. The heuristic approach, though, does not guarantee that an optimal solution will be found. Nevertheless, heuristics are very useful for practitioners who need to solve in real time problems of great sizes in order for their solutions to be used quickly (in real time).

- Primal heuristic

A primal heuristic works from an admissible solution to a problem, and tries to improve the solution through permitted operations that transform the current solution to another admissible solution with a lower objective function value. The new solution then becomes the current solution for the next step in the improvement process. The primal heuristic ends once no further improvement can be achieved by way of the permitted operations.

- Dual heuristic

A dual heuristic works from an inadmissible solution and seeks to reduce the degree of inadmissibility by using permitted operations. Individual steps of the algorithm are performed with a view to minimising growth in the objective function value. The process continues until an admissible solution is found or until it is no longer possible to reduce the inadmissibility of the solution by means of permitted operations.

Primal and dual heuristics can be combined (e.g., a primal heuristic provides the initial admissible solution for a dual heuristic).

3. Metaheuristics

This removes certain shortcomings of simple heuristics in that it does not become trapped in a local optimum. It jumps in the set of all admissible solutions and, at one moment, it explores several places in the set of admissible solutions (producing a sequential series of solutions), the objective function jumps; its gradient does not decrease, as in the case of simple heuristics. It is able to escape from a local optimum and find the global optimum.

Approaches – methods for creating a sequence of solutions

- approach based on exploring surroundings – Simulated Annealing, Tabu Search
- approach based on the evolutionary process – Genetic Algorithms, Ant Colony

We will look at individual approaches to solutions to specific problems.

2 SOLUTION APPROACHES

2.1 Solving a P-median problem exactly

P-median problem (maximum distance problem)

We have a given set of customers and some facilities (service centres). The customer is satisfied if its distance from any facility is less than or equal to the constant D_{MAX}. This formulation of the problem corresponds to the real problem of decision-making, such as in the case of the placement of alarm sirens, public (municipal) loudspeakers, or location of healthcare centres. The optimality criterion is to cover all customers through a minimum number of localised facilities, depots, etc. A classic instance is for a set of customers $j \in J$ and a set of facilities $i \in I$, where we define the sets:

$$N_j = \{i \in I : d_{ij} \le D_{Max}\} \tag{1}$$

these are sets of possible locations of facilities from which it is possible to satisfy the need of a *j*-th customer.

The problem can easily be transformed into the known problem of covering the vertices of a graph (customers) using the minimum number of subsets from the given system of subsets. Note: a vertex is covered when its distance from a facility is $\le D_{MAX}$. Individual subsets contain customers whose distance from the given facility does not exceed D_{MAX}.

Instance of a specific problem:

Given facts: D_{MAX}, location of customers, set of candidates for the placement of facilities. We are looking for the minimum number of facilities that will cover the customers' requirements.

We compile the sets $N_j = \{i \in I : d_{ij} \le D_{MAX}\}$ for $j \in J, J = \{v_1, v_2, v_3, v_4, v_5\}$. The sets are expressed in a table where the rows are customers and the columns are facilities. The *j-th* row of the table corresponds to the *j-th* set N_j, a one in the *i-th* row indicates that the facility S_i belongs to the set.

Table 1: N$_j$ Initial admissible solution to the problem.

	S$_1$	S$_2$	S$_3$
v$_1$	1		
v$_2$	1	1	
v$_3$	1		1
v$_4$	1	1	
v$_5$			1

Formulation of the LP general problem:

We denote the elements of the table as a matrix $A = (a_{ij})_{i,j=1}^{3,5}$ (2)

Then we can formulate the model of the coverage problem: to minimize: $\sum_{i \in I} y_i$ (3)

Under the conditions: $\sum a_{ij} y_i \ge 1$ for $j \in J$, $y_i \in \{0,1\}$ for $i \in I$ (4)

We fill the model with data from the example: minimise the function: $y_1 + y_2 + y_3$ (5)

Under the conditions:

$$y_1 \ge 1, \quad y_1 + y_2 \ge 1, \quad y_1 + y_3 \ge 1, \quad y_1 + y_2 \ge 1, \quad y_3 \ge 1, \quad y_i \in \{0,1\} \qquad (6-10)$$

We will examine the entire set of solutions, and then choose the one which is admissible and has the smallest objective function value.

An admissible solution to the problem (5) – (10) is each triplet consisting of ones and zeros that satisfies the inequalities (6) – (10). It is obvious that the vector $y = (1,1,1)$ meets all the conditions and therefore belongs to the set of admissible solutions. Conversely, the triplets *(0,1,1)* and *(1,1,0)* fail to meet the conditions (6) and (10), and are therefore inadmissible solutions.

The number of admissible solutions to the illustrative example (5) – (10) is small and contains only two solutions, so it is easy to determine the optimal solution that minimizes the function (5). The first solution gives us a criterion value equal to 3, the second solution then gives us the value 2. Therefore, the second solution, with an objective function value equal to 2, is the optimal solution to our problem.

While this example is simple and easily solvable, we should not be misled into believing that for every problem it is easy to determine the admissibility of the solution and optimality. Quite the contrary, most real problems must be described by hundreds or thousands of variables and conditions, and the number of admissible solutions exceeds the number 10^{12}. In such circumstances the use of the brute-force method, i.e., an examination of all admissible solutions, is unrealistic and unmanageable in a reasonable space of time necessary for practical use.

The above points lead us to the use of various computer-oriented methods, allowing the calculation of the solution to be completed within a reasonable time.

2.2 Solving a P-median problem through a simple heuristic

P-median problem (maximum distance problem)
To illustrate a heuristic method we will use the example of a heuristic approach to solving the maximum distance problem already formulated above.

The solution to the maximum distance problem is determined by the set of located facilities. A solution is considered admissible if each customer has at least one facility available within the distance D_{MAX}. The default admissible solution can be obtained by placing a facility in all possible locations (the set of candidates for the depot, facility). Each solution can be represented by a list of located facilities. If we were to denote *n* number of facilities in the list, the following algorithm can describe the *primal heuristic*.

```
STEP 0:  We put $k = 1$.
STEP 1:  Examine, check each facility from the list of facilities currently valid. If removing
         a facility does not disturb admissibility, then remove that facility from the list.
STEP 2:  If $k < n$, put $k = k + 1$ and go to STEP 1., else end the calculation 3.
STEP 3:  End.
```

If we were to apply this heuristic to our example with the default list $\{S_1, S_2, S_3\}$ of facilities, $n = 3$, $k = 1$, we can easily determine that the facility S_1 cannot be removed because the customer v_1 cannot be satisfied by either facility S_2 or S_3. The facility S_2 is examined and we find that its removal does not violate the solution's admissibility (all customers have least one facility available at a distance of $d_{ij} \leq D_{MAX}$). The new solution is expressed by the set $\{S_1, S_3\}$, $k = 2, k < n$. By examining the facilities S_1 and S_3 we find that neither S_1 nor S_3 can be removed due to the requirement of v_5. The algorithm ends with the resulting solution $\{S_1, S_3\}$.

Even though in our simple demonstration example the heuristic enabled us to find the optimal solution, in general this does not hold true, since, particularly in the case of complex and extensive problems, heuristics do not guarantee optimal solutions. A primal heuristic, based on an admissible solution, can end the search process far from the optimum. Each primary heuristic merely ensures that one admissible solution is replaced by another, better solution.

2.3 Solving the P-hub median problem through metaheuristics by using genetic algorithms

Hub and Spoke principle in general

- consolidation of small consignments at terminals (hubs)
- transportation between terminals over long distances
- distribution of consignments and delivery

Advantage: transportation between terminals (hubs) is cheaper and more frequent, drawback: extension of the route

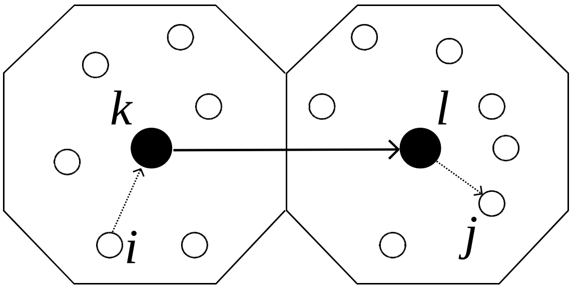

Figure 1: Hub and Spoke Principle.

Calculation of transportation costs – transportation costs between vertices *i*, *j*:

$$C_{ijkl} = c_{ik} + \alpha c_{kl} + c_{ij} \tag{11}$$

c_{ik} – cost of pick-up part
α – coefficient of savings in transportation between hubs *(0<* α *<1)*
c_{kl} – cost of transportation between the nodes *k*, *l*
c_{li} – cost of the delivery part

Variants of the problem:

- capacity limited / unlimited hubs
- known / unknown number of terminals
- same / different transportation costs in individual directions
- same / different α for individual routes between hubs
- same / different amounts of transportation flows between nodes
- simple / multiple allocation

Simple location – served objects are firmly assigned to a specific hub

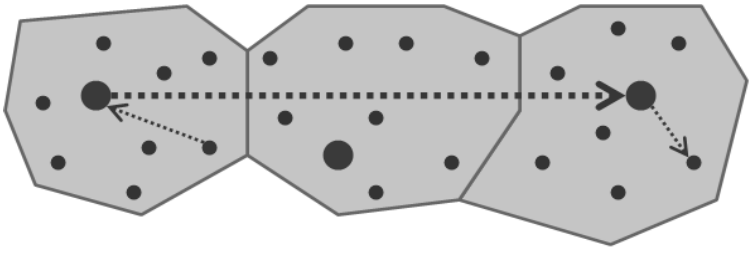

Figure 2: Simple location.

Fitness function:

$$\min \sum_{i} \sum_{j} b_{ij} \left(\sum_{k} c_{ik} h_{ik} + \sum_{k} \sum_{l} \alpha c_{kl} h_{ik} h_{jl} + \sum_{k} c_{il} h_{jl} \right) + \sum_{k} h_{kk} f_{k} \tag{12}$$

$$\sum_{k} h_{kk} = p \, , \sum_{k} h_{ik} \qquad \forall_{i} \in V$$

$$h_{ik} \le h_{kk} \quad \forall_{i}, k \in V \quad h_{ik} \in \{0,1\}$$

b_{ij} – transportation volume

c_{ik} – pick-up costs

αc_{kl} – transportation costs between hubs

c_{lj} – delivery costs

$\sum h_{kk} f_{k}$ - costs of building depots

Multiple location – objects can be assigned to attraction zones of several hubs

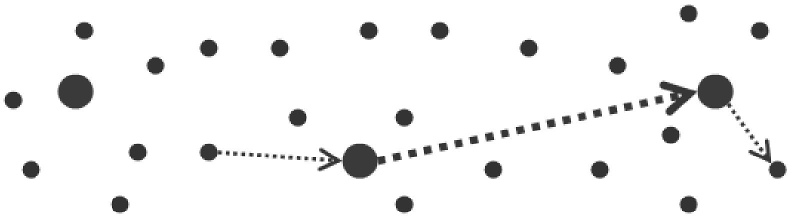

Figure 3: Multiple location.

- partial elimination of shipments in opposite directions
- pick-up (delivery) for a given point is not always performed by the same hub, the choice of the two hubs depends on the specific relation *i, j*

Fitness function:

$$\min \sum_{i} \sum_{j} \sum_{k} \sum_{l} b_{ij} (c_{ik} + \alpha c_{kl} + c_{lj}) X_{ijkl} + \sum_{k} h_{kk} f_{k} \tag{13}$$

b_{ij} – transportation volume

c_{ik} – pick-up costs

αc_{kl} – transportation costs between hubs

c_{lj} – delivery costs

$\sum h_{kk} f_{k}$ - costs of building depots

Solution using genetic algorithm

- encoding the problem's solution – chromosome
- creating the initial population
- selecting individuals for reproduction (selection)
- reproduction – crossing, mutation => new generation
- repeating iterations according to the requirements of the problem

Instance of a specific problem:

Find the optimal distribution of hubs on the network, where the number of a network's vertices is $n = 4$, the number of located hubs is $k = 2$, $\alpha=0.6$, C is the matrix of costs for unit shipments and B is the matrix of the quantities transported between individual vertices. This is a simple location, where the objects served are firmly assigned to one of the hubs.

$$B = \begin{pmatrix} 0 & 1 & 2 & 2 \\ 1 & 0 & 4 & 5 \\ 6 & 1 & 0 & 3 \\ 3 & 2 & 1 & 0 \end{pmatrix} \qquad C = \begin{pmatrix} 0 & 5 & 2 & 3 \\ 5 & 0 & 3 & 2 \\ 2 & 3 & 0 & 1 \\ 3 & 2 & 1 & 0 \end{pmatrix}$$

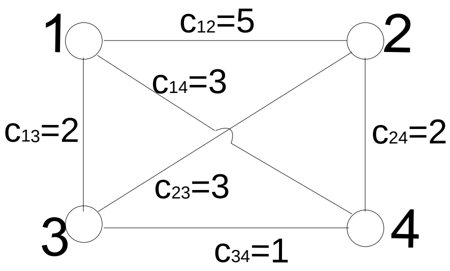

Figure 4: Graphical representation of the situation.

Options of encoding the problem's solution – chromosome

1) n parts each having 2 segments:

- is the given node a hub? (1-yes, 0-no)
- is the node assigned to its nearest hub? (0-yes, 1-no)

for example: (00|10|01|10) – hubs at vertices 2, 4; 1st in the attraction zone of the nearest hub, 4th in the attraction zone of the farthest hub.
(the specific network must be known)

2) n-component vector divided into 2 parts

- hubs – the length k, directly written numbers of hubs
- assignment of other vertices to hubs

for example: (24|21) – hubs at the vertices 2, 4; vertex 1 assigned to the 2nd hub in the order listed (i.e. to hub 4); vertex 3 assigned to the 1st hub in the order listed (i.e. to hub 2).

Encoding of the solution according to 1)

- Example encoding of one solution:
  k = 1, 2; A (1)={1,3}; A (2)={2,4}
- Solution: (10|10|00|00), hubs at the vertices 1, 2
  d(3,1)<d(3,2) *d(4,2)<d(4,1)*

Calculation of the fitness function

- according to the equation: (12) without costs for building the depot

example: (10|10|00|00), A (1) = {1,3}, A (2) = {2,4}, the sum of all shipments between *i* and *j*; $i, j \in \{1,2,3,4\}$

1. pick-up – non-zero for *i = 3, 4* => shipments 3-1; 3-2; 3-4; 4-1; 4-2; 4-3

2. shipment – non-zero: 1-2; 1-4; 2-1; 2-3; 3-2; 3-4; 4-1; 4-3

3. delivery – non-zero for *j = 3, 4* => shipments 1-3; 1-4; 2-3; 2-4; 3-2; 3-4

Table 2: Calculation of the fitness function of an individual.

hubs	**1**	**2**				
ass. vert's	1, 3	2, 4				
i, j	pick-up	shipment	delivery	total	b_{ij}	value
1 – 2	0	3	0	3	1	3
1 – 3	0	0	2	2	2	4
1 – 4	0	3	2	5	2	10
2 – 1	0	3	0	3	1	3
2 – 3	0	3	2	5	4	20
2 – 4	0	0	2	2	5	10
3 – 1	2	0	0	2	6	12
3 – 2	2	3	0	5	1	5
3 – 4	2	3	2	7	3	21
4 – 1	2	3	0	5	3	15
4 – 2	2	0	0	2	2	4
4 – 3	2	3	2	7	1	7
Total						**114**

The value of the fitness function of an individual is 114.

Initial population – 8 randomly selected individuals.

Table 3: Initial population.

number	configuration of the individual	fitness value
1	(10|10|00|00)	114,0
2	(00|10|10|01)	90,8
3	(01|01|10|10)	99,6
4	(10|00|00|10)	72,0
5	(10|10|01|01)	146,0
6	(00|10|00|10)	78,8
7	(10|00|10|00)	78,0
8	(10|01|00|10)	132,0

Selection – selection of individuals for reproduction using weighted roulette.

- the problem of genetic algorithms is that they work with a utility maximisation function, therefore we convert the utility fci is according to the equation

$$f_i = 1 + \frac{F_i - avg}{best - worst} \tag{14}$$

F_i – the value of fitness function for the given solution
avg – average fitness value fce in the generation
best – the best fitness value fce in the generation
worst – the worst fitness value fce in the generation

- probability of the individual passing into the next generation according to the equation

$$P_i = \frac{f_i}{\sum_{i=1}^{n} f_i} \tag{15}$$

Table 4: Selection.

numb er	configuration of the individual	fitness value F_i	converted fitness f_i	probability of passing P_i
1	(10|10|00|00)	114.0	0.830	10.4 %
2	(00|10|10|01)	90.8	1.143	14.3 %
3	(01|01|10|10)	99.6	1.024	12.8 %
4	(10|00|00|10)	72.0	1.397	17.5 %
5	(10|10|01|01)	146.0	0.397	5.0 %
6	(00|10|00|10)	78.8	1.305	16.3 %
7	(10|00|10|00)	78.0	1.316	16.5 %
8	(10|01|00|10)	132.0	0.586	7.3 %
Φ		101.4	1.000	

Using weighted roulette, the following pass into the next generation : f_4, f_2, f_6, f_3, f_7, f_1, f_4, f_8

Reproduction – Crossing

- creation of a combination of the genetic material of 2 individuals with the hope of gaining a better individual
- algorithm:

  1. browse the genetic code of individuals from right to left and look for the position *i* at which the 1st individual in the 1st gene segment has 1 and the second has 0. If found, replace the entire genes of both individuals at the position
  2. concurrently browse the genetic code of individuals from left to right and search for the position *j*, at which the 1st individual in the 1st gene segment has 0 and the other has 1. If you find it, replace the entire genes of both individuals at the position.
- both processes run until achieving $j \geq i$

Probability of crossing: $p_c = 0{,}75$, genes *f3´* and *f4´* proceed without crossing

Table 5: Crossing.

	original	**result of the crossing**
f1´=f4	(10|00|00|01)	(10|10|00|01)
f2´=f2	(00|10|10|01)	(00|00|10|10)
f3´=f6	(00|10|00|10)	(00|10|00|10)
f4´=f3	(01|01|10|10)	(01|01|10|10)
f5´=f7	(10|00|10|00)	(10|10|00|00)
f6´=f1	(10|10|00|00)	(10|00|10|00)
f7´=f4	(10|00|00|10)	(10|00|00|10)
f8´=f8	(10|01|00|10)	(10|01|00|10)

Reproduction – mutation

- operator’s purpose: to produce as yet unexplored or lost genetic material
- prevents premature convergence to a local optimum
- alteration of both segments with low probability
  - 1st segment: 2 substitutions in the generation (pm1 = 2/(8*4) = 6.3%)
  - 2nd segment: 1 change in the generation (pm2 = 1/16= 6.3%)
- in the case of the mutation of the 1st segment it is necessary to keep in mind the given number *k*, and, in the case of the 2nd segment, the occurrence of the nonsense positions of the type (…|11|…)
- mutation of the 1st segment
  - 6th individual – substitution of the 1st position for the 2nd
  - (10|00|10|00) → (**0**0|**1**0|10|00)
  - 5th individual – substitution of the 1st position for the 4th
  - (10|10|00|00) → (**0**0|10|00|**1**0)

- mutation of the 2nd segment
    - 3rd individual – substitution at the 3rd position
    - (00|10|00|10) → (00|10|0**1**|10)

The minimum of the function utility value after the first iteration is **67.6**; this corresponds to the minimum of the 2nd and 3rd iteration and also to the result of the exact solution.

Optimal solution: hubs at vertices 3 and 4. Vertex 1 is served from hub 3, and vertex 2 from hub 4.

Table 6: Mutation, result of the first iteration.

	genetic code	**fitness value**
f1	(10|10|00|01)	118.0
f2	(00|00|10|10)	67.6
f3	(00|10|01|10)	121.2
f4	(01|01|10|10)	99.6
f5	(00|10|00|10)	78.8
f6	(00|10|10|00)	71.2
f7	(10|00|00|10)	72.0
f8	(10|01|00|10)	132.0

3 CONCLUSION

Discrete optimization problems can generally be solved in different ways, ranging from exact, through simple heuristic methods, to metaheuristics. Simple heuristic methods, however, do not provide any guarantee of achieving an optimal solution, nor even an admissible solution. In general they are characterized by passing from one admissible or inadmissible solution to the next one, and by a local criterion with the help of which the resulting solution is selected from a set of possible subsequent solutions. Metaheuristics are heuristic approaches that under certain circumstances make it possible to depart from a local minimum, and, through a sequence of iterative steps, move into other parts of the set of admissible solutions where there is hope of finding a solution with a better objective function value than that of the local minimum. As with other heuristic methods, metaheuristics do not guarantee finding an optimal solution to the problem. The metaheuristic shown above using genetic algorithms does not create a sequence of solutions, but works concurrently with an entire set of solutions called a population. The work of a genetic algorithm begins with the creation of an initial population, the genes are then paired and crossed on the basis of benefit coefficients, and the newly-created individuals are mutated. Selection is made in the newly-created population, the best solution yet found is updated, and the whole process is repeated until either of the conditions for ending the optimization process is satisfied.

ACKNOWLEDGEMENT

This contribution is supported by MSM6840770043. The research was supported by the Czech Technical University, Faculty of Transport Science, Department of Logistic and Transportation Processes and by the EU - project FOCUS.

REFERENCES

Janáček, J., 2002. *Optimalizace na dopravních sítích*. Žilina: Žilinská universita v Žilině. ISBN 80-8070-031-1. (in Czech)

Janoš, V., Baudyš, K., Vichta, F., 2006. Financing the Public Transport in the Sphere of Regulated Competion. In *Proceedings Conference Euro Traffic 2006*. Praha (Czech Republic): ABF, a.s. ISBN 80-86905-15-2.

Lánská, M., 2006. Costs in Bus Transportation. In *Aktuálně problem v podnikání v cestnej doprave*. Žilina: Technical University of Žilina. ISBN 80-8070-610-7.

Campbell, A. M., Lowe, T. J., Zhang, L., 2005. The p-hub center allocation problem. *European Journal of Operational Research*. ISSN 0377-2217.