UNIVERSITAS COMENIANA
ACTA MATHEMATICA UNIVERSITATIS COMENIANAE
LVI—LVII

A VEHICLE ROUTING PROBLEM IN NETWORK CLUSTERS

JAROSLAV JANÁČEK, Žilina

1 Introduction

To simplify the vehicle routing problem in large networks [6], a decomposition of the network into clusters is often used [3], [4]. The whole problem at the level of the clusters can be solved by some known method [5]. We shall study the routing problem inside the single cluster in this paper. Each cluster may contain both supply and demand nodes but the travel costs between nodes inside the cluster are supposed to be insignificant.

However, any visit at a node of the cluster claims a fixed cost. Having done the decomposition (see *Fig. 1*), we can solve the vehicle routing subproblem in each cluster and obtain either the supply or demand surrogate node from it.

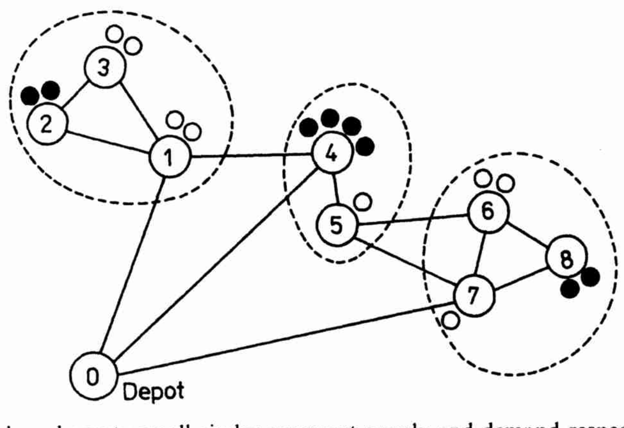

Fig. 1. Black and empty small circles represent supply and demand respectively

Thus we get a simpler surrogate network than the original one (see *Fig. 2*). We shall regard the subproblem — vehicle routing problem in network cluster in the rest of this paper.

If we add a subsidiary node to each cluster where the total supply is not equal to the total demand and define the supply or the demand of this node to get rid of the inequality, we can state the vehicle routing problem in a network cluster in the following way.

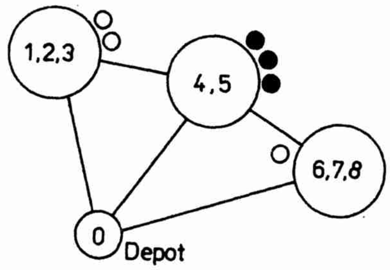

Fig. 2. Surrogate network

Given set of t sources and r sinks with integer supply a_i at each source $i = 1, \ldots, t$ and an integer demand b_j at each sink $j = 1, \ldots, r$.

Let

$$\sum_{i=1}^{t} a_i = \sum_{j=1}^{r} b_j$$

hold.

Find a route of a vehicle with capacity K to minimize the total number of visits at nodes and satisfy all demands in the cluster.

2 Other mathematical formulations

First, we shall show the way how to formulate the problem mentioned above as an integer linear programme. Let us create a substitutional network (see *Fig. 4*) for the given cluster (see *Fig. 3*), where each element of a source and each demanded element of a sink will be a node of the substitutional network.

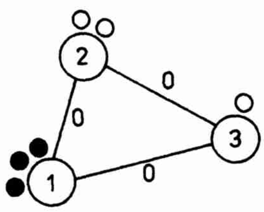

Fig. 3. A cluster with supplies (black circles) and demands (empty circles) and travel costs at the edges.

Thus we get the network where the original node of the cluster is represented by several new nodes. We add a depot to the network and denote it by 0. Let

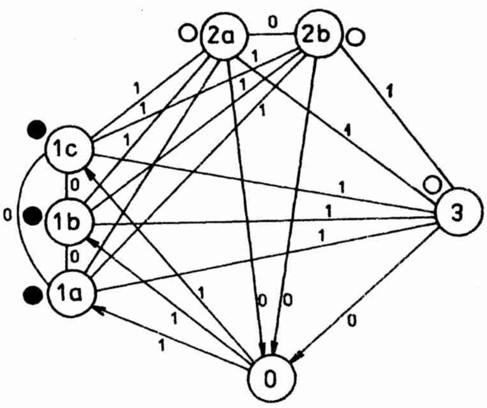

Fig. 4. The substitutional network

us denote N^+ set of sources, N^- set of sinks and $N = \{0\} \cup N^+ \cup N^-$ in the substitutional network. The following costs c_{ij} are assigned to the arcs:

$c_{i0} = \infty$ for all sources $i \in N^+$,

$c_{j0} = 0$ for all sinks $j \in N^-$,

$c_{0i} = 1$ for all sources $i \in N^+$,

$c_{0j} = \infty$ for all sinks $j \in N^-$,

$c_{ij} = 0$ if substitutional network nodes i, j were derived from the same original node,

$c_{ij} = 1$ if substitutional network nodes i, j were derived from different original nodes.

Furthermore, we shall use non-negative integer variables x_{ij} which get value 1 if the vehicle moves from node i to node j and they get value 0 in the opposite case. Non-negative integer variables r_{ij} will be equal to the number of elements transported on the vehicle along the arc $\langle i, j \rangle$. Besides variables x_{ij} and r_{ij}, auxiliary non-negative integer variables y_i will be used.

The problem can be formulated in this way [2]:
minimize

$$\sum_{j \in N} \sum_{i \in N} c_{ij} \cdot x_{ij} \tag{1}$$

subject to

$$\sum_{i \in N} x_{ij} = 1 \quad \text{for} \quad j \in N \tag{2}$$

$$\sum_{j \in N} x_{ij} = 1 \quad \text{for} \quad i \in N \tag{3}$$

$$r_{ij} \leqq K . x_{ij} \quad \text{for} \quad i, j \in N \tag{4}$$

$$r_{0i} = r_{i0} = 0 \quad \text{for} \quad i \in N \tag{5}$$

$$y_i - y_j + |N| x_{ij} \leqq |N| - 1 \quad \text{for} \quad i, j \in N^+ \cup N^- \tag{6}$$

$$\sum_{i \in N} r_{ij} - \sum_{i \in N} r_{ji} = 1 \quad \text{for} \quad j \in N^- \tag{7}$$

$$\sum_{i \in N} r_{ij} - \sum_{i \in N} r_{ji} = -1 \quad \text{for} \quad j \in N^+ \tag{8}$$

where $|N|$ denotes the number of elements from the set N.

Constraints (2) and (3) express that whenever a vehicle enters a node it leaves that node, and each node will be visited just once. Inequalities (4) ensure that the capacity of a vehicle will not be exceeded.

Equations (5) express that a vehicle will start and finish its route empty.

Inequalities (6) state that each circuit of the route will start and finish at the depot.

Equations (7) and (8) assert that all demands will be satisfied and all supplies will be used.

The optimal route in the cluster can be obtained from the optimal route in the substitutional network (see *Fig. 5a*) by contracting all the adjoining nodes which come from the same original node (see *Fig. 5b*).

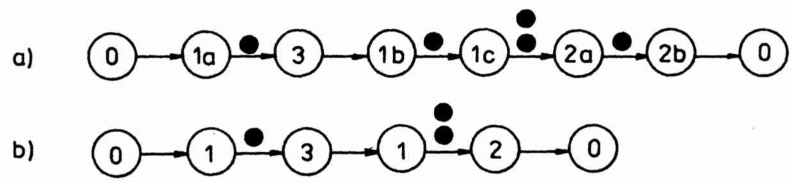

Fig. 5. Black small circles represent a load of the vehicle along the appropriate part of the route

The model (1)—(8) is usually very extensive due to great number of substitutional nodes which can be exponential with respect to the original number of cluster nodes.

Let us introduce another (nonlinear) model of the problem which is not so extensive and which may be useful when heuristics are developed. Let a number k_i, $i = 1, 2, \ldots, n$ be given for each of n original nodes.

We assume $k_i > 0$ to signify that i is a source and k_i is a supply and $k_i < 0$ to mean that i is a sink and $|k_i|$ is a demand.

It obviously holds $\sum_{i=1}^{n} k_i = 0$ for our problem.

Now we find the shortest sequences $y_1, y_2, \ldots, y_m$ and $x_1, x_2, \ldots, x_m$ where $y_i \in \{1, 2, \ldots, n\}$ and x_i is an integer subject to:

$$\operatorname{sign}(x_i) = \operatorname{sign}(k_{y_i}) \quad \text{for} \quad i = 1, \ldots, m \tag{9}$$

$$0 < |x_i| \leqq |k_{y_i}| \quad \text{for} \quad i = 1, \ldots, m \tag{10}$$

$$0 \leqq \sum_{i=1}^{s} x_i \leqq K \quad \text{for} \quad s = 1, 2, \ldots, m \tag{11}$$

$$\sum_{i \in I_j} x_i = k_j \quad \text{for} \quad j = 1, \ldots, n \tag{12}$$

where $I_j = \{i \mid y_i = j\}$.

In the second model the sequence $\{y_i\}$ determines the order in which nodes have to be visited and x_i determines the number of picked up ($x_i > 0$) or dropped ($x_i < 0$) elements at the node y_i during the i-th visit of the vehicle.

Equations (9) state that the elements will be picked up at sources and dropped at sinks. Inequalities (10) ensure that the supply of a source or the demand of a sink will not be exceeded. Inequalities (11) assert that the vehicle capacity will not be exceeded and equations (12) ensure that all demands will be met.

3 Reducibility of the problem

As far as the vehicle routing problem in a cluster is considered, the following theorems hold.

Theorem 1. If a cluster has only one sink and $k_i \leqq K$ holds for each i from N^+ then there is an optimal vehicle route which enters each source exactly once.

Proof. Assume that some optimal route of the vehicle enters source i with supply k_i r-times, where $r \geqq 2$. Removing all the visits at node i from the route, total number of visits will be reduced by r. The remaining k_i of the demands can be satisfied by adding one visit at source i and one at the sink (see *Fig. 6a, b*).

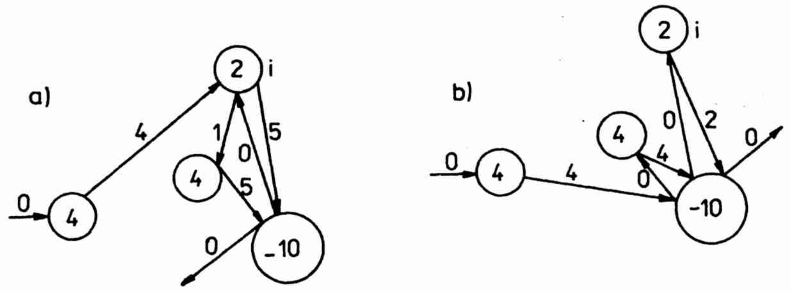

Fig. 6. Negative and positive numbers inside nodes represent original demands and supplies respectively. Numbers at the arcs give load of the vehicle whose capacity is 5

Thus the objective of the new feasible route will be reduced by $r - 2 \geqq 0$ and source i will be entered just once.

Repeating the procedure, an optimal route with claimed property will be obtained.

Theorem 2. If a cluster has only one sink and $k_i \geqq K$ holds for some source i with supply k_i then there is an optimal solution of the problem containing $\lfloor k_i/K \rfloor$ transports from node i to the sink with fully loaded vehicle. ($\lfloor x \rfloor$ designates integer part of x.)

Proof. Consider an optimal solution of the problem with m visits at the node i. On account of the solution being optimal, the visits must belong to m different circuits of the route (see *Fig. 7a*). Removing all the visits from the circuits and adding $\lfloor k_i/K \rfloor$ circuits of the type sink-source-sink, the total number of visits in the new route will increase by 2. $\lfloor k_i/K \rfloor - m$ and $k_i - K.\ \lfloor k_i/K \rfloor$ elements will remain at the source i (see *Fig. 7b*).

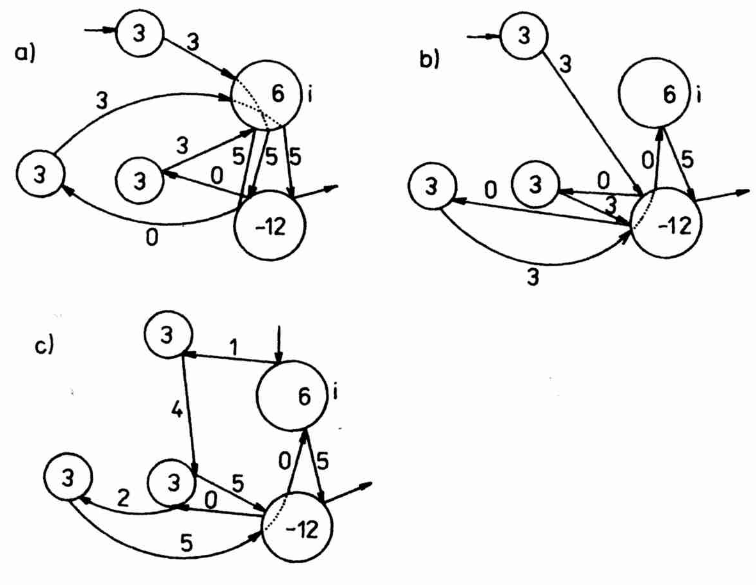

Fig. 7. The numbers in the figure have the same meaning as in *Fig. 6*

The number of the remaining elements at the node i and the elements transported by m original circuits mentioned above after removing will be less than or equal to $K\ (m - \lfloor k_i/K \rfloor)$.

To transport them to the sink, the $m - \lfloor k_i/K \rfloor$ circuits will do. Besides, transport of $k_i - K.\ \lfloor k_i/K \rfloor$ remaining elements from the source i can be carried out during the first circuit (see *Fig. 7c*).

Number of visits at sources including the first visit at the node i will increase at most by $m - \lfloor k_i/K \rfloor$ and number of visits at the sink will be reduced by $\lfloor k_i/K \rfloor$.

By these changes we obtain a feasible route whose number of visits differs from the original one at most by

$$2 \cdot \lfloor K_i/K \rfloor - m + m - \lfloor k_i/K \rfloor - \lfloor k_i/K \rfloor = 0.$$

Corollary. A vehicle routing problem in a network cluster with one sink is reducible to the problem where inequality $k_i \leqq K$ holds for each source i.

Note that a vehicle routing problem in a network cluster with more sinks and more sources is not generally reducible without loss of optimal solution.

Assuming the vehicle capacity K is equal to 13, the situation shown in *Fig. 8* is a good example of this. The arcs in the figure represent an optimal route with 19 visits.

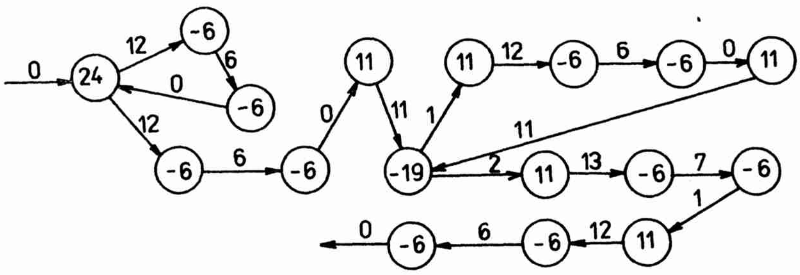

Fig. 8. The numbers in the figure have the same meaning as in *Fig. 6*

If 13 elements were picked up at the node with supply 24 and dropped at the node with demand of 19, two visits would be necessary and the reduced problem (see *Fig. 9*) would be obtained.

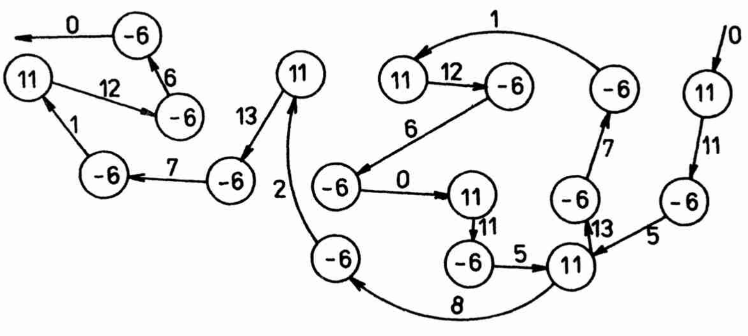

Fig. 9

An optimal solution of the reduced problem shown in the *Fig. 9* has 18 visits. Considering two former visits necessary for the reduction, an objective function of any feasible solution using reduction must be greater or equal to 20.

4 Problem completeness

We shall prove that the vehicle routing problem in a network cluster is NP complete. For this purpose it will be sufficient to prove that a bin packing problem whose NP completeness has been proved [1] is reducible to our problem.

Definition of "bin packing problem" [1]:

Given a finite set $U = 1, \ldots, t$ of items and a rational size $s(i) = k_i$ for each item $i \in U$, find a partition of U into disjoint subsets $U_1, \ldots, U_k$ such that the sum of the sizes of the items in each U_j is no more than K and such that k is as small as possible.

Theorem 3. The bin packing problem is equivalent to the vehicle routing problem in a network cluster with one sink and with $k_i \leqq K$ for each source $i \in N^+$.

Proof follows from the theorem 1.

5 Heuristics

For a practical solving of the vehicle routing problem in network cluster, six heuristics have been developed. Three of them are basic and the others have been derived from them.

Pick-up-Drop Algorithm (PDA)

Do through the arbitrary ordered sequence of the sources and pick up the maximum number of elements at each source until the vehicle capacity is fully utilized.

Then go through sequence of the sinks and drop maximum demanded number of elements at each sink until the vehicle is empty. Repeat the process until all demands are satisfied (see *Fig. 10a*).

Efficient-Picking-up Algorithm (EPA)

Find a sink with the largest demand.

Then go through the sequence of the sources towards their decreasing capacities and pick up the maximum number of elements until the demand is covered or until the vehicle is full. Satisfy the demand and repeat the process until all demands are satisfied (see *Fig. 10b*).

Efficient-Dropping Algorithm (EDA)

Go through the sequence of the sources towards their increasing capacities and pick up either whole capacity of the source or K elements.

If it is not possible to pick up anything yet, go through the sequence of the sinks towards increasing demands and drop maximum demanded number of elements at each visited sink until the vehicle is empty.

Repeat the process until all demands are met (see *Fig. 10c*).

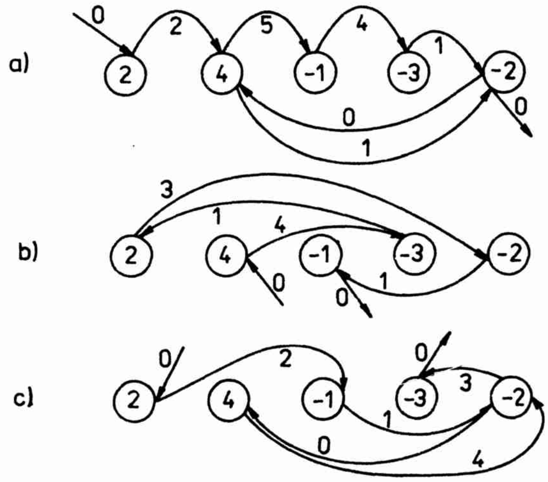

Fig. 10. The graphs a, b, c represent routes obtained by PDA, EPA, EDA respectively, when the vehicle capacity $K = 5$ and the original sequence of the $\{k_i\}$ $i = 1, \ldots, 5$ is given by numbers inside nodes in the order from the left to the right

The other three algorithms have been derived from the former by adding a phase of reducing (see Section 3) before each of the heuristics mentioned above.

Reducing Algorithm

Go through an arbitrary ordered sequence of the sources and find the first source with capacity greater or equal to K. Do the same for the sequence of the sinks.

Pick up K elements at the found source and drop them at the found sink.

Repeat it until either source capacity or sink demand is less than K. Then find another source or sink and repeat the procedure. If there is no source with supply greater of equal to K or no sink with demand greater or equal to K, stop the process.

The reducing algorithm causes that the original problem gets the form where

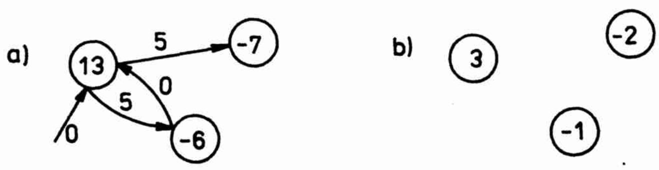

Fig. 11. The arrows in a) denote the part of the route which is constructed by phase of reducing for $K = 5$; b) shows the reduced problem

either all sources have capacities less than K or all sinks have their demands less than K (see *Fig. 11a, b*). The algorithms which have been derived from PDA, EPA and EDA thus are denoted RPDA, REPA and REDA respectively.

6 Computational results

We coded the heuristics mentioned in Section 5 and tested them on the randomly generated sets of problems. Differences between average results of particular algorithms and lower bound d of optimal solution which was obtained from (13) were compared.

$$d = \sum_{i=1}^{n} \lceil |k_i|/K \rceil \tag{13}$$

where $\lceil x \rceil$ denotes the smallest integer greater or equal to x. *Tables 1* and *2* show values v in percentage determined by (14)

$$v = \frac{\bar{f} - \bar{d}}{\bar{d}} \cdot 100 \tag{14}$$

where $\bar{f}$ is the average number of visits of the solutions obtained by the tested algorithm from the given set of problems and $\bar{d}$ is the average lower bound of the optimal solutions.

The effect of H/K ratio on the value v was studied in the first series of tests, where K was the vehicle capacity and uniform randomly generated source capacities and sink demands were in range $\langle 0, H \rangle$. The number of sources was approximatelly equal to the number of sinks and $H = 20$ for any problem of this series. The set of problems with n ranging from 10 to 100 by step 10 and with 20 problems for every n was generated for every $K = 5, 10, \ldots, 50$ (see *Table 1*).

The algorithm REPA proved to be the most effective. The effectiveness of the reduction algorithm was proved for the value $H/K > 1$ too.

Table 1

Alg. \ K	5	10	15	20	25	30	35	40	45	50
PDA	13.0	25.1	27.5	40.4	32.5	27.4	23.3	20.1	17.8	16.0
RPDA	8.8	22.5	24.5	40.4	32.5	27.4	23.3	20.1	17.8	16.0
EPA	8.9	11.4	11.4	11.8	6.5	3.8	2.1	1.5	1.3	0.9
REPA	2.5	6.1	6.8	11.8	6.5	3.8	2.1	1.5	1.3	0.9
EDA	12.1	21.2	23.7	25.1	22.6	18.1	14.5	11.7	10.8	9.2
REDA	5.8	14.7	16.1	25.1	22.6	18.1	14.5	11.7	10.8	9.2

In the second series of tests the effect on algorithm effectivity of the ratio (number of sources to the total number of nodes) was studied (see *Table 2*). The capacity was equal to 5 for any problem of this series and the same number of problems was generated for each value of the ratio as for the single capacity in the first series.

Table 2

Ratio / Alg.	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
PDA	7.7	9.9	10.7	12.7	13.2	12.5	11.0	8.9	7.5
RPDA	5.0	6.4	7.3	8.8	8.8	8.5	7.2	6.0	5.1
EPA	6.7	7.7	7.7	8.2	8.9	8.8	8.7	7.8	7.1
REPA	2.4	2.3	1.9	2.1	2.2	3.3	4.1	4.6	4.7
EDA	7.8	9.6	10.4	11.7	12.5	12.0	9.7	8.3	7.1
REDA	4.7	5.5	5.7	6.3	6.0	5.3	4.4	3.6	2.9

Even in this series the algorithm REPA proved to be the best one.

Computational results were obtained on a computer EC 1045 (approximately comparable to IBM 370) using FORTRAN H compiler. The *Table 3* presents the average CPU-time of one solution for particular methods and series in milliseconds.

Table 3

Algorithm	PDA	RPDA	EPA	REPA	EDA	REDA
SERIES 1	17	13	21	21	21	21
SERIES 2	67	24	27	24	27	25

REFERENCES

1. Garey, M. R.—Johnson, D. S.: Computers and Intractability. A Guide to the Theory of NP-Completteness. W. H. Freeman and Company, San Francisco, 1979.
2. Janáček, J.: Matematický model úlohy trasování v dopravní síti s mnoha zdroji a spotřebiteli. Práce a štúdie Vysokej školy dopravy a spojov v Žiline, séria kybernetická, zv. 6. ALFA, Bratislava 1988.
3. Janáček, J.—Szendreyová, A.—Varša, P.: Využití dekompozičních metod v operativním řízení nákladní dopravy. In: Výskumná správa k priebežnému oponentskému pokračovaniu III-8-9/01. Žilina, VŠDS. 1988, pp. 6—63.
4. Liberková, I.: Optimalizace rozvozních tras. NETTO, 1987, No 1, pp. 31—34.
5. Nelson, M. D.—Nygard, K. E.—Griffin, J. H.—Shreve, W. E.: Implementation Techniques for the Vehicle Routing Problem. Computation and Operations Research, Vol 12, No 3, pp. 273—283, 1985.

6. Eilon, S.—Watson-Gandy, C. D. T.—Christofides, N.: Distribution management: Mathematical modelling and practical analysis. Griffin, London 1971, p. 240.

Author's address:

Jaroslav Janáček
Katedra teoretickej kybernetiky
VŠDS
010 26 Žilina

Received: 17. 8. 1988

SÚHRN

ÚLOHA TRASOVANIA V MIKRORAJÓNOCH SIETE

JAROSLAV JANÁČEK, Žilina

V článku sa zaoberáme úlohou trasovania v sieti, kde sa množina uzlov skladá zo zdrojov a spotrebiteľov a kde dopravné náklady medzi uzlami sú zanedbateľné.

Úlohou je minimalizovať celkový počet návštev v uzloch. Je dokázaná NP-obtiažnosť úlohy a sú dokázané vety o redukovateľnosti úlohy.

Na dosiahnutie približného riešenia úlohy bolo navrhnutých niekoľko heuristík a bol skúmaný vplyv použitia heuristiky na efektívnosť riešenia.

РЕЗЮМЕ

ЗАДАЧИ МАРШРУТИЗАЦИИ В МИКРОРАЙОНАХ СЕТИ

ЯРОСЛАВ ЯНАЧЕК, Жилина

В статье формулируется задача маршрутизации в микрорайоне сети, где множество вершин состоит из источников и потребителей и где расстояния между вершинами запускаемые.

Минимализировать надо общее число посещений вершин. В статье дано доказательство NP-трудности задачи и доказательства теорем об редуцировании задачи.

Для приблизительного решения задачи были созданы приближенные алгоритмы и была определена зависимость между использованным алгоритмом и эффективностью решения.