Computer Science Journal of Moldova, vol.7, no.1(19), 1999

Decision support system for vehicle scheduling

C. Gaindric

Abstract

A decision support system (DSS) is described to form schedules of traffic from a central warehouse to a set of consumers by cyclic routes. The system may be used by dispatchers at transportation enterprises. The system structure, short description of modules, and algorithms solving the originating problems are presented.

1 Introduction

The important place among problems of effective transportation by trucks automatized organization is taken by the following problem: to determine optimum ring routes, if there is one central warehouse, many consumers, cargoes are uniform, and consumers' consignements are small comparatively with cars carrying capacity.

Consumers' requests are diverse, and a single general condition is that their needs q_i, $i = \overline{1,n}$ are much less than car carrying capacities P_k, $k = \overline{1,k}$.

Some consignees, e.g. food shops, require that their consignments delivering in the predefined time intervals (τ_i^1, τ_i^2), $i = \overline{1,n}$.

Therefore, car transportation enterprises can satisfy their contracts under prescribed consumers' requests with minimum costs only by the best possible organisation of operative dispatching service.

To determine routes with minimum costs became a complicated task if there are several hundreds of consuming points and several car types with different carrying capacities.

At the first stages of application of mathematical simulation methods large efforts were undertaken to formalize problems and to develop methods of their solution. Many more or less successful experiments were made but there exist no mathematical metods widely applied to find ring routes of small consignement transportation.

Which were the reasons?

It is known that in all cities thousands of shops, newspaper boothes etc. require maintenance of their orders. Firstly, there are already traditional routes which became customary but are far from optimum in many cases. Algorithms for solution of such problems are combinatorial in their nature and did not ensure the solving for the large number of points in acceptable time. The developed models are the rigid ones and various modifications of conditions required by the consumers imply essential modifications in models.

The emerging of the DSS concept was met with some interest but the financial and organizational problems do not allow small and average enterprises to order a specialized DSS. But the main obstacle in distribution of DSS for dispatchering ring routes forsmall-sized consignment transpotation to a large number of consumers is still the necessity to combine the big complexity of a solution of such combinatorial problems with heterogeneous requests depending on the consignment, conditions of entrance to a consumption point, time of delivery, conditions at car loading board before the trip, etc.

The problem of small-size consignment transportation on ring routes has its spource in a famous problem of the travelling salesman formulated in 1930 by K. Meneger and consisting in determination of the order of given n points to minimize the length of closed path.

This simple formulation has attracted attention of many researcher offering many diverse algorithms implementing both various heuristic approaches and exact methods of integer programming, and methods of branch and bound kind.

But the problems of practical value are more complicated. To each i-th point of consumption, a weight q_i and a salesman's visit interval (τ_i^1, τ_i^2) are assigned.

Another generalization of the travelling salesman problem is an m salesmen problem formulated as follows.

m travelling salesmen depart from a starting point. Each of them visits each point from a given subset selected from $n-1$ points, and then returns to the starting point. It is necessary to find such cyclic routes that each point was visited by only one of the salesmen and the total path of all salesmen was the shortest.

The transition from the problem of m travelling salesmen to the problem of small-sized consignment transportation on ring routes is naturally achieved by assignement to each point its weight q_i and to each salesman its weight P_k and by imposing a condition that the total weight of route points for each salesman does not exceed salesman's weight.

Hereafter a number of problems is considered which meet in daily work of the car transportation enterprise dispatcher. Methods of their solution are described and the DSS structure to help a dispatcher is offered.

We can mark several reasons attracting attention to DSS:

1. A certain experience of computer use in organizational control systems is already accumulated and reasons of failures in a decision making with the help of such systems are revealed.

2. Mathematical models developed for a decision making, being only partially supplied by a necessary information, do not correspond to imposed goals. It is obvious that it is impossible to receive all objective information necessary to operation with these models. This forces the managers (decision makers, DM) to use their own suppositions about the simulated process behaviour and the missing information. It, in turn, distorts initial premises entered by the developers of models and introduces a discord between supposed and real behaviour of the model and in some sence justifies DM's refusal to use these models in real conditions. Moreover, not all problems may be easy formalized.

3. The convincing results of psychological researches of limited decision maker's possibilities to accumulation and evaluation of base information to accept his solutions.

It shows that it is necessary to help a decision maker with systems which combine intellectual possibilities of the manager, its knowledge, skill, and ability with a power, memory and possibilities of computers. Such systems are called DSS and rationally organize the information obtaining and processing, and interaction with a DM.

There is many definitions of DSS, but that given in [1] seems more adequate:

> "DSS is a man-machine systems which permits decision maker to use data, knowledges, objective and subjective models for the analysis and solving poorly structured and unstructured problems."

2 Formulation of some scheduling problems

Scheduling problems have very wide area of application and, of course, many formulations. Distracting from details, it is possible to formulate some frame formulations, which in practice cover majority of needs. We can obtained formulation of any practical problem adding conditions reflecting specificity of enterprises or transported consignments.

2.1 Problem of m travelling salesmen

Given n points numbered by i, $i = \overline{1, n}$ and matrix of distances between them $C = (c_{ij})_{n \times n}$. m traveling salesmen ($1 \leq m \leq n - 1$) leave an initial point. They should bypass $n - 1$ remaining points so that each point is visited by one and only one salesman and each travelling salesman visits at least one point. It is necessary to find such m closed paths for travelling salesmen that the total path is minimum.

For $m = 1$ the problem become a usual problem of the travelling salesman. The first formulation of this problem was published in [2] in 1960. The problem of m travelling salesmen describs a practical situations when the expert should perform the periodic checking of some devices located at some distance one from other so the transition time from one device to another influences the total checking duration. Having many devices one person is physically not capable to inspect then during the fixed time. Therefore, there is a problem of path determination for object detour and survey by two or more experts.

2.2 Problem of m travelling salesmen with weights

This problem is formulated as follows: m traveling salesmen leave an initial point. A weight P_k, $k = \overline{1, m}$ is assigned to each salesman. They should visit $n-1$ points each of which also has its weight q_i, $i = \overline{2, n}$.

It is necessary to find such closed paths through the points for travelling salesmen that the total path is minimum and the following conditions fulfil:

- Each point is visited only by one travelling salesman.
- Each travelling salesman visits at least one point except of the starting.
- The total weight of points included in the route for a travelling salesman should not exceed salesman's weight.

If in a problem of device detour by experts we designate through q_i, $i = \overline{1, n}$ the time necessary to check devices, and through P_k, $k = \overline{1, m}$ the working time of experts, the result of solution of the m travelling salesmen problem with weights is the plan of technological paths for experts.

2.3 Problem of m travelling salesmen with fixed intervals of visiting

If during the detour and checkup of devices each of them should be examined in the fixed interval of time (τ_i^1, τ_i^2), this complicates the problem. The order of visits determined solving problem 2.2 can be impracticable because of fixing the visit intervals.

2.4 Problem of m travelling salesmen with weights and fixed intervals of visiting

This problem appears by combining the conditions of problems 2.2 and 2.3.

2.5 Minimization problem for the longest path of m travelling salesmen

The problem of minimization of the longest paths for m travelling salesmen describes real situations of delivery of the newspapers in boothes, when it is important that even the last point in a path of the salesman would receive the newspapers as early as possible. Naturally in this case as for the problem with the fixed visit intervals, the matrix $T = (t_{ij})_{n\times n}$ will be used instead of $C = (cij)_{n\times n}$. Here t_{ij} is the time to pass the path from i to j, $i, j = \overline{1, n}$.

3 DSS structure for truck dispatching

Starting from description of problems, which should be solved by a dispatcher, it is possible to propose the DSS structure presented on Fig. 1.

Denotations for the Fig. 1:
1 – interface "decision maker – DSS";
2 – problem recognition module;
3 – problem solution module;
4 – result presentation module;
5 – control system of the decision making knowledge base;
6 – decision making knowledge base;
7 – control system of a knowledge base for the controlled area of activity;
8 – model base for the controlled area of activity;
9 – algorithm base for problem solving
10 – new model construction module;
11 – data base management system for the controlled area of activity;

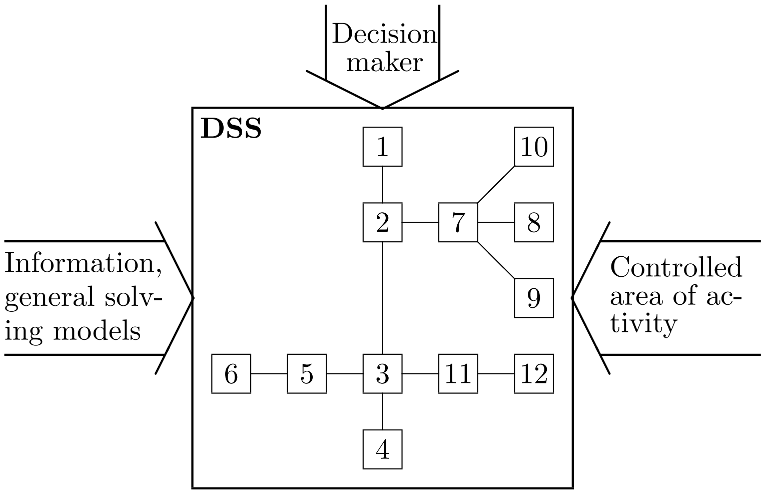

Figure 1. DSS structure for truck dispatching

12 – data base for the controlled area of activity.

DSS should reveal from the full scale of decision making situations such problem to be solved which is already familiar, i.e., there are its model and its solution algorithm in the model base (8) and algorithm base (9), and to distinguish situation met in the first time, if the user has the experience with the system.

The *user interface* ensures:

- user identification and providing access to a DSS session;
- user testing and adaptation of the dialog in the correspondence to its knowledges and working style;
- dialog initiation and helping to the user during a working session.

In the proposed system the special problem recognition module is separated which is included in the interface in other DSS.

The *problem recognition module* receives information on models from the model base through the control system for the area of activity, and ensures choice of an adequate model to solve the proposed problem. In the case when such model exists, its *solving module* starts.

This module, through a data base management system of controlled area, connects the model and the selected algorithm solving the problem to data from the data base of thecontrolled area.

Let's note that in the beginning of its session the DSS has already defined who is the user, which is his experience in interaction with the system, and has adapted itself to his decision making characteristics to his style.

The received results of the algorithm's run are transmitted to the *result presentation module* which represents them in the most convenient form for the decision maker.

This module includes tools which analyze the obtained solution and reveal vulnerable places in operation of controlled object or system.

Through the *control system of the knowledge base* the user can select from the model base of the controlled area that model which in the best way corresponds to features of the problem and his own way of thinking.

Using the *data base of controlled area* and the *base of solving algorithms* for consultation through the *control system of the knowledge base,* the user selects an algorithm for calculations.

In the case the algorithm has not enough data to run, the additional data are input in the base if it is possible, or another model is selected which is fully supplied with available data. When no model is found satisfying he user and having enough data, the *new model construction module* is called.

Infortunately, we do not know any successful implementation of such module, though research in this direction is very necessary.

4 Input and output data

Depending on their modification period, *input data* may be divided to

- relatively constant data;
- data specific to each request, which vary daily.

Information on positions of starting point (warehouse) and finishing points of consumption is to be present inependently of the problem formulation. Therefore, coordinates (addresses) of these points are kept in the data base. Taking into account that algorithms based on various variants of the *branch and bound* method are used to solve most of problems, we will use in the system square and, as a rule, asymmetric matrices $C_0 = (c_{ij})_{n\times n}$ of distances and $T_0 = (t_{ij})_{n\times n}$ of travel time between points i and j, $i,j = \overline{1,n}$ where consumers have contracts whith the warehouse. We do not permit repeated visitings of the same point except of initial, so main diagonal elements of matrices C_0 and T_0 will be $c_{ii} = \infty$, $t_{ii} = \infty$, $i = \overline{1,n}$.

Warehouse should daily execute applications of consumers and deliver to them their consignments. Therefore, the following information serves dispatchers as input data for the system:

1. list of consumers;
2. amount of the required consignment q_i.
3. time interval (τ_i^1, τ_i^2) in which the consignment should be delivered to the consumer.

Corresponding to these data, matrices C_1 and T_1 of distances and times are extracted from the data base which keeps matrices C_0 and T_0, to solve the problem of ring routes dispatching for those points whose applications need to be satisfied in the current day.

Other input data concern to transportation tools: the amount of trucks n_k, and their capacities P_k.

The *output data* are of two types: these for car drivers, and these for the dispatcher.

Data accompanying consignments are for the car driver and contain sequence visited points, calculated intervals of delivery for each point, and consignment quantity in the form standard for such documents.

For the dispatcher, documents are issued representing each ring route. They contain departure moments from the initial point, sequence of points to be visited, return time in the initial point, the rest time after the route, time of loading for the next route, the same data for the next route, total working time of of each driver, and other information specific for a transportation enterprise.

5 Functioning of the system

The problem solving session begins with input of the consignement delivery point list, amount of the ordered consignment for each point and intervals of delivery.

The system is tuned depending on the characteristics of the solved problem and input data. If the intervals of visiting are not given, i.e., the consumer is indifferent to the delivery time, the algorithm for solving of problem 2.2 is applied.

For problems of mail transportation, the algorithm for solving of problem 2.4 is used. But the most often problem is that from 2.3 with fixed delivery intervals. The algorithm for such problems is based on the solving method of problem 2.2 described in [3] with additions necessary to visit the given point in required time intervals.

At first let us found necessary conditions for the arc (i, j) to be included in a possible solution. We calculate for this purpose the quantities:

$$\eta_{ij} = \max[\tau_i^1, \tau_j^1 - t_{ij}] \tag{1}$$

which shows the earliest departure moment from the i-th point to arrive in the j-th point at the given interval. Here τ_i^1, τ_j^1 are beginning of delivery intervals respectively for points i, j, and t_{ij} is the time necessary to arrive from i to j. We calculate also

$$\delta_{ij} = \min[\tau_j^2 - t_{ij}, \tau_i^2] - \eta_{ij} \tag{2}$$

where τ_i^2, τ_j^2 are the latest moment of visiting the i-th and j-th point.

When $\delta_{ij} \geq 0$, $|\delta_{ij}|$ is the length of the time interval, during which the travelling salesman may depart from i and arrives in j in the given interval, i.e., the travelling salesman can depart from i at any moment of the interval $(\eta_{ij}, \eta_{ij} + \delta_{ij})$ and will arrive in j during the required interval.

When $\delta_{ij} < 0$, $|\delta_{ij}|$ is a maximum idle period to wait in i for to arrive in j at the moment τ_j^1.

If we do not permit idle periods, then we use the condition $\tau_j^1 - t_{ij} \leq \tau_i^2$, but this introduces too severe constraints which imply that, frequently, the problem do not have any solution.

In practice, the idle time $|\delta_{ij}|$ is usually limited by some quantity δ^*.

Let the arc (i, j) is included in the solution. Then the new visiting interval for the i-th point is

$$(\max[\eta_{ij} + \hat{\delta}, \tau_i^1]; \min[\eta_{ij} + \delta_{ij}, \tau_i^2]),$$

where

$$\hat{\delta} = \begin{cases} 0 & \text{for } \delta_{ij} \geq 0 \\ \delta^* & \text{for } \delta_{ij} < 0 \end{cases},$$

and for the j-th point

$$(\eta_{ij} + t_{ij}; \min[\eta_{ij} + t_{ij} + \delta_{ij} - \hat{\delta}, \tau_j^2])$$

Therefore, the necessary conditions to include (i, j) in the solution are:

$$\tau_j^2 - \tau_i^1 - t_{ij} \geq 0 \tag{3}$$

and

$$|\delta_{ij}| \leq \delta^* \qquad \text{for } \delta_{ij} < 0 \tag{4}$$

Following to [3], we insert in the matrix $T_0 = (t_{ij})_{n \times n}$ $m - 1$ rows and $m - 1$ columns corresponding to t_{1j}, $j = \overline{2, n}$ and to t_{i1}, $i = \overline{2, n}$ receiving the matrix $T = (t_{ij})_{(m+n-1)\times(m+n-1)}$. Elements of its main minor of the order m we consider as ∞.

Visiting intervals for the first m points (really, the initial point repeated m times) are be $(0, M)$ where $M > \tau_j^2 + t_{j1}$.

Taking into account visiting intervals, we order rows and columns of the matrix T so that the left boundaries of visiting intervals make a nondecreasing sequence. From two points at which the left boundaries

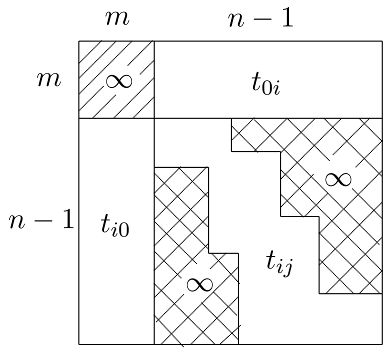

Figure 2. Block structure of the matrix T_1

of intervals coincide we include in the sequence firstly that point at which the right boundary of the interval is less.

To exclude from our consideration variants not conforming (3) and (4) we form the matrix $T' = (t'_{ij})_{(m+n-1)\times(m+n-1)}$ so that

$$\begin{array}{ll} t'_{ij} = t_{ij} & \text{if } \tau_j^2 - \tau_j^1 - t_{ij} > 0 \text{ and } |\delta_{ij}| \le \delta^* \text{ for } \delta_{ij} < 0 \\ t'_{ij} = \infty & \text{if } \tau_j^2 - \tau_i^1 - t_{ij} < 0 \text{ or } |\delta_{ij}| > \delta^* \text{ for } \delta_{ij} < 0 \end{array}$$

We receive thus the matrix T_1 which is used in the algorithm for ring routes dispatching and has a specific block-diagonal structure (Fig. 2).

As the result of the algorithm run, we receive m ring routes (ordered point sequences) $\mu_s = (1, i_1, \ldots, i_k, 1)$, $s = \overline{1,m}$, where number 1 corresponds to the initial point (the warehouse), the sequence of recalculated visiting intervals, and the sequence $\delta_{i_l i_r}$ ensuring the minimum total path.

Now it remains to find starting moment of each route (from the initial point) to ensure minimum idle time during the route.

Let $\mu = (1, i_1, \ldots, i_r, \ldots, i_k, 1)$ be the sequence of route points produced by the algorithm, $\delta_{1i_1}, \delta_{i_1 i_2}, \ldots, \delta_{i_k 1}$ be intervals and (τ_1^1, τ_1^2), $(\tau_{i_1}^1, \tau_{i_1}^2)$, ..., $(\tau_{i_k}^1, \tau_{i_k}^2)$ be the sequence where τ_1^2 is the latest starting moment for the initial point agreed with recalculated visiting intervals for route points $i_1, \ldots, i_r, \ldots, i_k$.

Using these data, let us estimate the total idle time during the route if the route starts at the moment t_0.

Let $\delta_0 = \min\limits_{(i,j)\in\mu} \delta_{ij}$ and $\delta_0^+ = \min\limits_{(i,j)\in\mu} (\delta_{ij}|\delta_{ij} \geq 0)$.

Let $\theta = \tau_1^2 - t_0$.

It follows from the construction of (2) that $\delta_{i_{k-1}i_k} \geq \delta i_k i_{k+1}$.

Let us consider two possible cases.

1. $\delta_0^+ = \min\limits_{(i,j)\in\mu} (\delta_{ij})$, and all $\delta_{ij} \geq 0$

Let γ_s be the idle time before leaving the s-th point. Then the total idle time during the route is $\gamma = \sum\limits_{s=i_1}^{i_k} \gamma_s$.

Let

$$p = \min\{l|\theta > \delta_{i_{l-1}i_l}\,,\ l = \} \tag{5}$$

For every point i_s, $s = \overline{1, p-1}$, $\gamma_s = 0$, but $\gamma_p = \theta - \delta_{i_{p-1}i_p}$.

Starting from the p-th point the travelling salesman arrives at the moments τ'_{i_r} $(p < r \leq k)$, and $\gamma_r = \delta_{i_{r-2}i_{r-1}} - \delta_{i_{r-1}i_r}$. Therefore,

$$\begin{aligned}
\gamma &= \sum_{s=1}^{p-1} \gamma_s + \gamma_p + \sum_{s=p+1}^{k} \gamma_s = \\
&= 0 + \theta - \delta_{i_{p-1}i_p} + \sum_{s=p+1}^{k} (\delta_{i_{s-2}i_{s-1}} - \delta_{i_{s-1}i_s}) \\
\gamma &= \theta - \delta_{i_{p-1}i_p} + \delta_{i_{p-1}i_p} - \delta_{i_{k-1}i_k} \\
\gamma &= \theta - \delta_{i_{k-1}i_k}
\end{aligned}$$

The sequence $\delta_{i_{s-1}i_s}$ is monotonously nondecreasing and all its members are nonnegative. Therefore $\delta_{i_{k-1}i_k} = \delta_0^+$ and

$$\gamma = \theta - \delta_0^+ \tag{6}$$

(6) implies that the condition (5) can not be satisfied for any point i_p and $\theta < \delta_0^+$. Then $\gamma_p = 0$ for each p, $p = \overline{1, k}$. Therefore the total idle time during the route is zero if the route starts from the initial point at any moment t_0, $\tau_1^2 - \delta_0^+ \leq t_0 \leq \tau_1^2$.

2. $\delta_0 = \min\limits_{(i,j)\in\mu} (\delta_{ij} \mid \exists \delta_{ij} < 0)$

In this case, there are idle times befor starting from some route points.

Let there is in the route Q maximum segments μ_q without idle times, $\mu = \{\mu_1, \mu_2, \ldots, \mu_Q\}$.

On each segment μ_q, all δ corresponding to its arcs are positive.

Let l_1^q and $l_{p(q)}^q$ be lengths of recalculated wisiting intervals for the first and last point of the segment μ_q, and $\delta_{q,q+1}$ be the minimum idle time between segments μ_q and μ_{q+1}.

The total idle time during the route

$$\gamma = \sum_{q=1}^{Q} \gamma_q + \sum_{q=1}^{Q} \gamma'_q,$$

where γ_q – the idle time during the segment μ_q;
γ'_q – the idle time before the segment μ_q.

$$\gamma_1 = \begin{cases} l'_1 - l'_{p(1)} & for \quad \theta \geq l'_1 \\ \theta - l'_{p(1)} & for \quad l'_1 > \theta \geq l'_{p(1)} \\ 0 & for \quad \theta < l'_{p(1)} \end{cases}$$

$$\gamma_q = l_1^q - l_{p(q)}^q, \ 1 < q \leq Q$$

$$\gamma'_2 = |\delta_{12}| + z,$$

where $z = \begin{cases} l'_{p(1)} & for \quad \theta > l'_{p(1)} \\ \theta & for \quad \theta \leq l'_{p(1)} \end{cases}$

For each $1 < q \leq Q-1$, $\gamma'_q = l_{p(q-1)}^{q-1} + |\delta_{q-1,q}|$.

Let us note that $\gamma'_1 + \gamma_1 + \gamma'_2 = \theta + |\delta_{12}|$ in all three possible cases.

2.1. $\theta = l'_1$,

$$\gamma'_1 + \gamma_1 + \gamma'_2 = \theta - l'_1 + l'_1 - l'_{p(1)} + |\delta_{12}| + l'_{p(1)} = \theta + |\delta_{12}|$$

2.2. $l'_1 > \theta \geq l'_{p(1)}$,

$$\gamma'_1 + \gamma_1 + \gamma'_2 = 0 + \theta - l'_{p(1)} + |\delta_{12}| + l'_{p(1)} = \theta + |\delta_{12}|$$

2.3. $\theta < l'_{p(1)}$,

$$\gamma'_1 + \gamma_1 + \gamma'_2 = 0 + 0 + |\delta_{12}| + \theta = \theta + |\delta_{12}|$$

Now

$$\begin{aligned}
\gamma &= \gamma'_1 + \gamma_1 + \gamma'_2 + \sum_{q=2}^{Q} \gamma_q + \sum_{q=3}^{Q} \gamma'_q \\
\gamma &= \theta + |\delta_{12}| + \sum_{q=2}^{Q} (l_1^q - l_{p(q)}^q) + \sum_{q=3}^{Q} (l_{p(q-1)}^{q-1} + |\delta_{q-1,q}|) \\
\gamma &= \theta + \sum_{q=2}^{Q} l_1^q - \sum_{q=2}^{Q} l_{p(q)}^q + \sum_{q=3}^{Q} l_{p(q-1)}^{q-1} + \sum_{q=2}^{Q} |\delta_{q-1,q}| \\
\gamma &= \theta + \sum_{q=2}^{Q} |\delta_{q-1,q}| + \sum_{q=2}^{Q} (l_1^q - l_{p(q)}^q) \\
\gamma &= \theta - l_{p(Q)}^Q + \sum_{q=2}^{Q} (l_1^q - |\delta_{q-1,q}|)
\end{aligned}$$

In the last expression all members except for θ are constant,

$$\gamma = \theta - K, \quad K = l_{p(Q)}^Q - \sum_{q=2}^{Q} (l_1^q - |\delta_{q-1,q}|), \quad \text{and } \theta = \tau_1^2 - t_0.$$

Changing the leaving time t_0 for the initial point, we can ensure minimum total idle time during the route, but before this it is necessary to co-ordinate the route loading interval with possibilities of loading frame at the warehouse, because the route scheduling is only the first part of the solution.

As a rule, the limited loading capacity greatly influences over the car loading queue and, therefore, over the route starting moments.

Therefore the next problem to be solved is to adjust route starting intervals taking into account the capacity of the loading board.

At first it is necessary to attach routes to loading boards, i.e., to find free "windows" in their work during which it is possible to load $\sum_{i\in\mu_s} q_i$ units of cargo with the loading speed ε.

First of all we select to load those routes for which $\delta_{0s}^{+} = 0$, i.e., it is impossibille to delay their start.

The next routes are attached to the boards in ascending order of their δ_{0s}^{+}.

If at the moment $\tau_s^1 - \varepsilon \sum_{i\in\mu_s} q_i$ the board is occupied, it is necessary to shift loading of the s-th route to the right (on later time when the board is free), but no more than on δ_{0s}^{+}.

If all routes are attached to the loading board, we consider the moment of loading termination as the moment of route start. For routes with idle times during them, the moment of route start $\tau_s' = t_0$ is calculated as in (6) to minimize the total idle time during the route.

At the last stage we have a set of routes of the same carrying capacity and it is necessary to generate routes for each car taking into account duration of drivers' shifts, obligatory breaks, etc.

This problem can also be considered as the problem of m travelling salesmen. If m is the amount of chosen cars of the same carrying capacity, we can consider routes as points (cities), their durations as weights of points, and duration of the driver shift as the weight of the travelling salesman.

The scheme of the solution for the problem of scheduling the route for each carrying capacity is as follows. We form the matrix $C = (c_{ij})_{k\times k}$, where k is the amount of routes of the same carrying capacity, and c_{ij} is the time interval between the termination of the i-th route and the start of the j-th route, corrected after attaching to loading boards.

$$c_{ij} = \begin{cases} t_j^1 - (t_i^2 - \varepsilon P_i) & for \quad t_j^1 \geq (t_i^2 - \varepsilon P_i) \\ \infty & for \quad t_j^1 < (t_i^2 - \varepsilon P_i) \end{cases}$$

t_i^2 is the calculated moment of the i-th route termination after the coordination with possibilities of the loading board and minimization of idle time, P_i is the total consignment for points of the i-th route, and ε is the loading time of one unit of cargo.

Routes are ordered in the matrix so that their starts form a nondecreasing sequence. For the first m routes (m is the amount of cars of the same carrying capacity), $c_{ij} = \infty$, $i, j = \overline{1, m}$.

Thus, the data matrix to run the solving algorithm for the problem 2.2 is generated.

As the result we obtain routes satisfying to restrictions on the duration drivers' work and minimizing total idle time between routes.

The problem is solved for each carrying capacity of cars.

Thus the target documents described in Sec. 4 are issued to the dispatcher.

The variant of attaching routes to boards is described only schematically. We had not considered cases when it is impossible to attach some routes because of the limited loading capacity of the board. Detailed consideration of a problem in full involves possible relaxation of visiting intervals and is out of frames of the present article.

The exact solution for the problem may be obtained by construction of a net [5] using the following rules.

Route μ_s is included in the queue at the M-position board at the moment $\tau'_s - t_s$ and needs the time $t_s = \varepsilon \sum_{i \in \mu_s} q_i$ to be served. We know the latest moment $\tau'_s + \delta^+_{s0}$ in which the car is to be loaded.

Let us suppose that τ'_s, t_s, and δ^+_{s0} are integer. Let us renumber time intervals starting since $t = 0$ as $1, 2, \ldots, \max\limits_s(\tau'_s + \delta^+_{s0})$.

Because the route μ_s is set to loading at the moment $\tau'_s - t_s$ and is to be loaded not later than at $\tau'_s + \delta^+_{s0}$, therefore its loading is possible at moments numbered $\tau'_s - t_s + 1, \ldots, (\tau'_s + \delta^+_{s0})$.

Let as form the net making x_0 its input, z its output and x_1, x_2, $x_{\max\limits_s(\tau'_s + \delta^+_{s0})}, y_1, y_2, \ldots, y_n$ its vertices. The vertex x_θ corresponds to the time interval θ, $\theta = \overline{\tau'_r - t_r + 1, \tau'_r + \delta^+_{0r}}$, the vertex y_s corresponds to the route μ_s.

Vertices x_θ are connected with vertices y_s by arcs of the throughput capacity $C(x_\theta, y_s) = 1$. The vertex x_0 is connected with vertices x_θ by arcs of the through-put capacity $C(x_0, x_\theta) = M$, $\theta = \overline{1, \max(\tau'_s + \delta^+_{s0})}$. Vertices y_s are connected with the output vertex z by arcs with through-put capacity $C(y_s, z) = t_s$, $s = \overline{1, n}$.

Every flow saturating the output arcs of the net defines a schedule which services all routes satisfying given terms. The inverse is also true, i.e., every schedule which services all routes satisfying given terms defines a flow saturating output arcs of the net.

6 Conclusion

The scheduling problem small-sized consignments transportation on ring routes continues to attract attention of researchers and practical workers. Only in [4], more than 80 works of the last years with algorithms of the problem solution are described. On my opinion, the real practical results for this problem can be received only after a DSS construction in which module 9 (the solving algorithm base) will be constantly supplemented with new algorithms, and in the module 7 (control system of the knowledge base of controlled area) the newest ideas and knowledge will be assimilated.

Then DM (in this case a dispatcher) can solve problems, selecting those models and algorithms which present the most adequate description of his problem and which ensure specific requests of consumers.

References

[1] Larichev O.I., Petrovskiy A.B. Decision support systems. State-of-the-art and development trends. // Itogi nauki i tehniki, ser. Technical cybernetics, **v.21.** – Moscow, “Nauka”, 1987. (Russian)

[2] Miller C., Tucker A., Zemlin R. Integer programming formulation of travelling salesman problem. Journal of ACM, **v.7,** No.4, 1960.

[3] Gaindric C.V., Zhitkov V.A. Algorithm to solve a simplified transportation problem. / In: Mathematical methods for economical problems. – Moscow, "Nauka", 1969. (Russian)

[4] Min H., Jayaraman V., Srivastava R. Combined location-routing problems: A synthesis and future research directions. European Journal of OR, **v.108,** 1998, pp.1–15.

[5] Tanayev V.S., Shkurba V.V. Introduction to schedule theory. – Moscow, "Nauka", 1975. (Russian)

Constantin Gaindric,
Institute of Mathematics
and Computer Science,
Academy of Sciences of Moldova
5, Academiei str., Kishinev,
MD028, Moldova
phone: (373+2) 72-50-12
fax: (373+2) 73-80-27
e-mail: *gaindr@math.md*

Received December 2, 1998