

Jaroslav Janáček *

THE TRANSPORT-PRODUCTION COORDINATION PROBLEM

Vehicle route planning process, which follows a transportation problem solution, often provides a system of cyclic routes of vehicles. This system should cover transportation demands of individual network nodes within a middle term period. Due to non-uniform demand distribution over the period, a new problem emerges in particular parts of the period. This problem consists in balancing the sub-period demands and frequencies, with which the individual routes are performed. This new problem denoted as the transport-production coordination problem is studied in this contribution.

We described the problem by mathematical programming tools and proposed a decomposition method to enable the problem decomposition to irreducible sub-problems. The individual types of the problem were studied and particular solving algorithms were designed. To complete our approach to this NP-hard problem, we developed an exchange heuristic and explored its properties. The associated computational study was performed with several series of the irreducible sub-problems, which were solved by the exchange heuristic and, simultaneously, they were solved to optimality by the universal optimization environment Mosel. The results were used for evaluation of the heuristic quality and time consumption.

1. Introduction

The transport-facility coordination problem arose when a mass transport problem was solved to ensure a fluent process of highway building. The building of a highway is accompanied by the necessity to bring a huge amount of mass as stone or clay to fill cavities at different places of the building site. At the same time, demands for taking off some amounts of material can arise. These demands are connected with excavation works at some places of the site. If the associated transportation problems are solved, then cost optimal moves of material amounts are obtained (See Fig. 1).

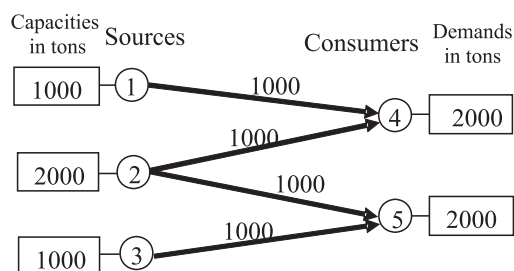


Fig. 1 Optimal flows of material

If corresponding vehicle capacities are assigned to the resulting moves, then a demand for free vehicle capacity emerges at beginning of each move and the end of move constitutes a source of free vehicle capacity. Then a problem of balancing the free and

demand vehicle capacities is to be solved. This solution results into a graph with flows of empty and loaded vehicles (See Fig. 2).

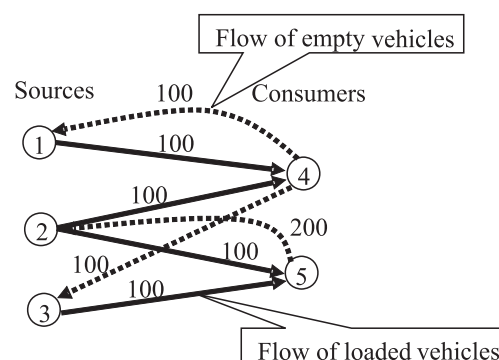


Fig. 2 Flows of vehicles having the capacity of 10 tons

These vehicle flows can be covered by numerous cyclic vehicle trips whose accomplishment usually takes several days or weeks (See Fig. 3).

After all trips are performed, all original demands for mass transport are satisfied.

The realization of above mentioned cyclic vehicle trips in the individual days brings another problem of vehicle coordination at some places, which constitute sources or consumers of the transported material. The necessity of coordination follows from the

* Jaroslav Janáček

Department of Transportation Networks, Faculty of Management and Informatics, University of Žilina, Slovakia, E-mail: Jaroslav.Janacek@fri.uniza.sk

fact that the material is excavated or stored at these places using assigned machines with a given productivity, which must be exploited as much as possible.

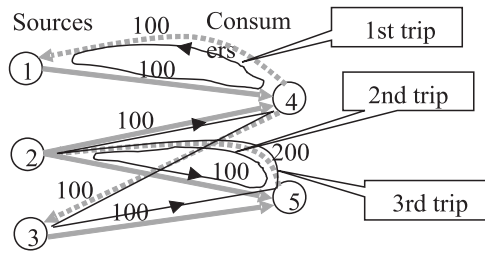


Fig. 3 Cyclic trips of vehicles covering the flows, when repeated hundred times

For this purpose a transport-production coordination problem must be solved daily. As shown below, the transport-production coordination problem constitutes a serious combinatorial problem, which is NP-hard. It follows from the fact that the knapsack problem is polynomially transformable to an instance of the transport-production coordination problem. Due to the complexity of problem and necessity to solve the individual instances of the problem quickly, we perform an analysis of the problem and suggest a set of quick heuristics for solving special cases of the problem together with a decomposition algorithm.

2. The transport-production coordination problem

Given the set J of customers, where each customer $j \in J$ produces the demand b_j with the rate q_j of demand units per a time unit. Each customer's demand can be reduced by the amount K by one visit of a vehicle. Let the set of customers be provided with the set I of given vehicle trips and let an incidental matrix $\{a_{ij}\}_{i \in I, j \in J}$ be indicated by the values one or zero of the element a_{ij} whether the trip i does or does not serve the customer j . Each performance of trip $i \in I$ takes a time t_i and must be performed exactly p_i times. After each trip i is performed p_i times, all customers' demands are completely satisfied. If only one vehicle performs the trip i , then the trip contributes to the satisfaction of j -th customer demand with the rate $K_i = K/t_i$.

Under assumption that $x_i \in \{1, \dots, p_i\}$ vehicles are assigned to the trip i and they perform regularly, the trip performance will satisfy the demand with the rate $x_i K/t_i = x_i K_i$.

Let us consider a sub-set $J' \subseteq J$, for which a lack of the total demand satisfaction rate causes an additional cost. The lack u at the customer j causes the additional cost $c_j u$. On the other hand, a surplus of the satisfaction rate may cause an undesirable delay of the involved vehicles and this way it may influence the satisfaction rate of other customers, which is not permitted.

The objective is to determine the numbers $x_i \in \{1, \dots, p_i\}$ $i \in I$ of assigned vehicles so that no total of demand satisfaction of the

customers from J' exceeds q_j and the total additional cost following the lack of satisfaction rate is minimal. It is assumed that if only one vehicle is assigned to each trip, then no satisfaction rate at any customer from J' exceeds the demand rate q_j .

3. An integer-programming model

Let us denote by $x_i \in \{1, \dots, p_i\}$ the variables which give the numbers of vehicles assigned to the trips $i \in I$. Then the demand at the j -th customer is q_j . The rate of demand satisfaction is $\sum_{i \in I} a_{ij} K_i x_i$.

The lack of demand satisfaction rate is $q_j - \sum_{i \in I} a_{ij} K_i x_i$. The total additional cost is then $\sum_{j \in J'} c_j [q_j - \sum_{i \in I} a_{ij} K_i x_i] = \sum_{j \in J'} c_j q_j - \sum_{j \in J'} \sum_{i \in I} c_j a_{ij} K_i x_i$.

As the first term of the total additional cost is a constant, the integer-programming model can be formulated as follows:

$$\text{Maximize} \quad \sum_{j \in J'} \sum_{i \in I} c_j a_{ij} K_i x_i \quad (1)$$

$$\text{Subject to} \quad \sum_{i \in I} a_{ij} K_i x_i \leq q_j \quad \text{for } j \in J' \quad (2)$$

$$x_i \leq p_i \quad \text{for } i \in I \quad (3)$$

$$x_i \geq 1 \quad \text{for } i \in I \quad (4)$$

$$x_i \in \mathbb{Z}^+ \quad \text{for } i \in I \quad (5)$$

It is necessary to take into account that the above model is formulated for one time unit only, e.g. for one working day. It follows some further assumptions of trip durations and upper bounds of variables x_i . In the first place, we suppose that trip duration is less or equal to the time unit. In the second place, we have to take into consideration the regularity of the trip repeating. Let us imagine that the upper bound p_i of number of the assigned vehicles to a trip i equals two and that the trip takes exactly half of the time unit. In this case it makes no sense to assign two vehicles to this trip because they will contribute to the demand satisfaction in this time unit with the same unit as one vehicle even if they make their work in the half of the time unit. From this it follows that inequality $p_i t_i / x_i > 1$ must hold and the constraints (3) can be replaced by the constraints $x_i \leq r_i$, where $r_i = \lfloor p_i t_i \rfloor$ or $r_i = p_i t_i - 1$, if $p_i t_i$ is integer.

The further rearrangement of the model follows from the fact that if each variable x_i is set to the value of one, then no rate q_j is exceeded. We can set $q_j = q_j - \sum_{i \in I} a_{ij} K_i$ for $j \in J'$. Then, the upper

bound of the variables x_i can be made more precise using the following formulation $r_i = \min\{r_i - 1, \min\{\lfloor q_j / K_i \rfloor : j \in J', a_{ij} = 1\}\}$.

After these rearrangements, the model (1)–(5) can be reformulated this way:

$$\text{Maximize} \quad \sum_{j \in J'} \sum_{i \in I} c_j a_{ij} K_i x_i \quad (6)$$

$$\text{Subject to} \quad \sum_{i \in I} a_{ij} K_i x_i \leq q_j \quad \text{for } j \in J' \quad (7)$$

$$x_i \in [0, \dots, r_j] \quad \text{for } i \in I \quad (8)$$

4. A decomposition algorithm for the transport-production coordination problem

The problem structure is given by the sub-matrix $\{a_{ij}\}_{i \in I, j \in J'}$. E.g. for the case illustrated in Fig. 3 and for $J' = \{1, 2, 3\}$ the incidental sub-matrix has the following form:

Incidental trip-customer matrix

Table 1

$I \setminus J'$	1	2	3
1	1		
2		1	
3		1	1

The original problem can be decomposed to smaller and simpler instances similarly as a bipartite graph with the node sets I and J' and with the edges given by an incidental matrix can be decomposed into maximal connected sub-graphs. The following labeling algorithm can do it:

0. [Initialization] Determine all the trips from I as unlabeled.
1. If each trip is labeled, then terminate. Otherwise choose an unlabeled trip. Provide the chosen trip with an unused label and insert it into a list of "labeled but unprocessed trips".
2. [Sub-problem determination] If the list of "labeled but unprocessed trips" is empty a new sub-problem has been labeled, go to the step 1, otherwise remove a trip from "labeled but unprocessed trips" and process it in the following way: Go over all incidental customers and for each of them inspect its incidental trips. If the incidental trip is unlabeled, give it the label of a removed trip and insert it into the "labeled but unprocessed trips". Repeat step 2.

5. Particular cases of the transport-production coordination problem

Even if the general transport-production coordination problem belongs to the family of NP-problems, there are some cases, which can be solved, exactly in polynomial time. We shall introduce several easily solvable cases together with the associated algorithms, which will be used to construct a general heuristic method [1] in the next section. In this section we assume that each discussed instance of the problem corresponds to a maximal connected sub-graph which means that it cannot be decomposed to smaller problems. The particular cases will be distinguished in accordance to

the sub-matrix $\{a_{ij}\}_{i \in I, j \in J'}$, where I is the set of trips and J' is the set of relevant (important) customers.

The simplest case arises if $|I| = 1$ holds, where the expression $|I|$ denotes the cardinality of the set I . For this case, where $I = \{i\}$, the sole variable x_i can be determined optimally and it can be done by the following formula: $x_i : \min x_i \lfloor \min\{q'_j : j \in J', a_{ij} = 1\} / K_i \rfloor$, where $\lfloor u \rfloor$ denotes the integer part of the number u and where q'_j is the current demand of the customer $j \in J'$. The condition $a_{ij} = 1$ can be omitted here because the assumption of connectivity caused that $a_{ij} = 1$ for each $j \in J'$. The next discussed case is distinguished from the others by the conditions $\sum_{j \in J} a_{ij} = 1$,

which hold for all the trips but one. Let this only one trip be denoted by u . The unique j for $i \in I - \{u\}$, for which $a_{ij} = 1$ can be denoted as $j(i)$. Now, let us consider that x_u is fixed at some value x . If we reduce the demands q_j by the values $a_{uj} K_u x$ to q'_j , then the associated best values of x_i for $i \in I - \{u\}$ can be obtained as $x_i := \min \{x_i, \lfloor \min\{q'_{j(i)} / K_i \rfloor\}$. Based on these preliminaries, an exact algorithm can be stated as follows:

0. Set $C^{best} := +\infty$
1. For $x_u := 0$ to x_u repeat
 - set $q'_j := q_j - a_{uj} K_u x_u$ for all $j \in J'$,
 - set $x_i := \min \{x_i, \lfloor \min\{q'_{j(i)} / K_i \rfloor\}$ for all $i \in I - \{u\}$,
 - set $C := \sum_{j \in J} c_j (q_j - \sum_{i \in I} a_{ij} K_i x_i)$
 - if $C < C^{best}$ then set $C := C^{best}$ and $x_i^{best} := x_i$ for $i \in I$.

The complexity of this algorithm is $O(|I| \cdot r)$, where r is the upper bound of x_i for $i \in I$.

The third exactly solvable case is distinguished from others by the fact that the condition of a unique trip-customer assignment holds for all the trips but two. Let the two trips be denoted by u and v , and then the following steps describe the exact algorithm:

0. Set $C^{best} := +\infty$
1. For $x_u := 0$ to x_u repeat step 2.
2. Set $q'_j := q_j - a_{uj} K_u x_u$ for all $j \in J'$,
- Set $x_{max} := \min \{x_v, \lfloor \min\{q'_j : j \in J' \mid a_{vj} = 1\} / K_v \rfloor\}$
- For $x_v := 0$ to x_{max} repeat
 - set $q'_{jv} := q_j - a_{uj} K_u x_u - a_{vj} K_v x_v$ for all $j \in J'$,
 - set $x_i := \min \{x_i, \lfloor \min\{q'_{j(i)} / K_i \rfloor\}$ for all $i \in I - \{u, v\}$,
 - set $C := \sum_{j \in J} c_j (q_j - \sum_{i \in I} a_{ij} K_i x_i)$
 - if $C < C^{best}$ then set $C := C^{best}$ and $x_i^{best} := x_i$ for $i \in I$.

The complexity of this algorithm is $O(|I| \cdot r^2)$. This exact algorithm can be generalized to heuristics, which can be used in

the cases when the unique trip-customer assignment condition does not hold for more than two trips. The generalization may consist in replacing the instruction $x_i := \min \{x_p, \lfloor \min\{q'_{j(i)}/K_i \rfloor\}$ for all $i \in I - \{u, v\}$, by the following set of instructions, which can be performed in an arbitrary order of trips i from the set $I - \{u, v\}$:
 set $x_i := \min\{x_p, \lfloor \min\{q'_j : j \in J' \mid a_{vj} = 1\} / K_i \rfloor\}$,
 update $q'_j := q'_j - a_{ij} K_i x_i$ for all $j \in J'$.

This algorithm will be denoted as $H(u, v, C^{best}, x^{best})$.

6. An exchange heuristics

The algorithm $H(u, v, C^{best}, x^{best})$ described in the previous section can be used in a construction of exchange heuristics. The proposed heuristics starts with an ordered sub-set I' of all the trips I . The ordering can be done, e.g. decreasingly in accordance to the number of incidental customers. The exchange heuristics will proceed all the unordered pairs $\{u, v\}$ of the trips from I' , whereas the embedded algorithm $H(u, v, C^{best}, x^{best})$ proceeds all the trips from I . Let us denote by \leq_o the considered ordering of the set I' , then the heuristics can be described by the following steps:

0. Set $C^{res} := +\infty$
1. For $u \in I'$ repeat step 2.
2. For all $v \in I', u \leq_o v$ repeat steps:
 Perform $H(u, v, C^{best}, x^{best})$.
 If $C^{best} < C^{res}$, then set $C^{res} := C^{best}$ and $x_i^{res} := x_i^{best}$ for all $i \in I$.

The complexity of this algorithm is $O(|I'|^2 \cdot |I| \cdot r^2)$, but it can be arbitrarily reduced by a restriction of the set I' .

7. Numerical experiments

In the following experiments, we have focused on investigation of the heuristic quality and time consumption. This matter was studied on six series of randomly generated instances where each series consists of ten instances. These series differ in an increasing size m of the problem which means the cardinality of the set I of trips and, as well, the cardinality of the set J' of relevant customers. The problem sizes start from ten and continue by ten to the size of sixty. The instances were generated so that no problem can be decomposed to smaller ones and that is why the decomposition heuristics was not involved into the experiments. All the problem coefficients c_j, t_i, r_i, q_j and a_{ij} were obtained this way:

- a_{ij} – randomly generated zero-one matrix where each row contains $m/3$ ones and the problem is irreducible,
- c_j – randomly generated real number from 10 to 80 with a uniform distribution,
- t_i – randomly generated real number from 0.05 to 0.4 with a uniform distribution,
- p_i – randomly generated real number from 50 to 80 with a uniform distribution,
- q_j – randomly generated real from $500 + \sum_{i \in I} a_{ij}(10/t_i)$ to $800 + \sum_{i \in I} a_{ij}(10/t_i)$,

$$K_i = 10/t_i.$$

To perform the numerical experiments, PC Pentium 4, 2.8 GHz, 512 MB was used. To be able to evaluate the quality of the proposed heuristics, we implemented it in the development environment Delphi 7 and solved all the generated instances. Furthermore, we made use of the mathematical programming software environment Mosel [3], [4], which integrates the modeller and solver, of previous optimization packages into one comprehensive tool, which is extended by a debugging tool. This optimization software environment embedded into graphical environment Xpress-IVE is successor of the famous optimization package XPRESS [2], [5], from which it inherited the excellent optimization procedures. This solver was able to provide us with optimal solution of the problem instances. The comprehensive results of experiments are plotted in table 2, where individual columns contain average result attributes of particular series. The size m of instances from a particular series is reported in the first row of the table. The next rows of table have the following meaning:

Avg.Obj.Heur. – average of objective function values of the resulting solutions obtained by heuristics on all instances of the given series,

Avg.Obj.Xpress – average of objective function values of the optimal solutions obtained by the environment Xpress-IVE on all instances of the given series,

Avg.Obj.Diff. – average of differences of objective function values of optimal solutions obtained by the environment Xpress-IVE and resulting solutions obtained by heuristics on all instances of the given series,

Std.Obj.Diff. – standard deviation of differences of objective function values of optimal solutions obtained by the environment Xpress-IVE and resulting solutions obtained by heuristics on all instances of the given series,

Avg.Obj.Diff.% – average of differences of objective function values of optimal solutions obtained by the environment Xpress-IVE and resulting solutions obtained by heuristics given in percentage of the objective function value of optimal solutions,

Avg.Time.Heur. – average of computational times in seconds consumed by heuristics on all instances of the given series,

Std.Time.Heur. – standard deviation of computational times consumed by heuristics on all instances of the given series,

Avg.Time.Xpress. – average of computational times in seconds consumed by the environment Xpress-IVE on all instances of the given series,

Std.Time.Xpress – standard deviation of computational times consumed by the environment Xpress-IVE on all instances of the given series.

Average objective function values and computational times for the particular series

Table 2

m	10	20	30	40	50	60
Avg.Obj.Heur.	113919	240658	346992	443151	522410	625862
Avg.Obj.Xpress	115144	251109	373154	484381	596138	718652
Avg.Obj.Diff.	1225	10451	26161	41230	73727	92790
Std.Obj.Diff.	1531	6087	10707	11272	13353	15138
Avg.Obj.Diff.%	1	4	7	9	12	13
Avg.Time.Heur.	0.00	0.05	0.21	0.68	1.45	2.72
Std.Time.Heur.	0.01	0.01	0.03	0.24	0.23	0.31
Avg.Time.Xpress	0.09	0.20	0.82	5.50	48.70	236.75
Std.Time.Xpress	0.13	0.10	0.64	4.25	57.96	247.05

6. Conclusion

We introduced a new problem which originated in civil engineering practice and we called it “The Transport-Production Coordination Problem”. This problem was formulated and modelled using means of integer programming and its computational complexity was studied theoretically and experimentally as well in connection with the usage of professional universal optimization environment Mosel. Our results confirmed that the problem is not easily solvable to exact optimality by a general solver due to increasing computational time (see row Avg.Time.Xpress in table 2). The

designed exchange heuristics proved to be able to obtain a good solution in very short time in comparison with the general IP solver. The results of heuristics differ from the optimal ones on average by 13% when the largest studied instances of the problem were solved. Taking into consideration an uncertainty of the input data in practical situations, e.g. the time t_i of trip i performance, it can be claimed that the proposed heuristic approach is satisfactory for embedding into decision support tools for such problems.

Acknowledgment: This work was supported by the grant VEGA 1/3775/06.

References

- [1] JANÁČEK, J.: *Optimization on Transportation Networks (in Czech)*, Žilinská univerzita, Žilina, 2002, p. 248.
- [2] LAUBER, J., JABLONSKÝ, J.: *Software for Mathematical Model Building (in Czech)*, Vysoká škola ekonomická, Praha, 1993, p. 316.
- [3] XPRESS-MP Manual “Getting Started”, Dash Associates, Blisworth, 2005, p. 105.
- [4] XPRESS-Mosel “User guide”, Dash Associates, Blisworth, 2005, p. 99.
- [5] XPRESS-MP Reference Manual, Dash Associates, Blisworth, 1991.