Algorithm for generalized transportation problem with exclusionary constraints

Marcin Anholcer², Arkadiusz Kawa³

1. Introduction

Supply chain management (SCM) is a sequence of events in a goods flow within an inter-organizational network. In this flow the value is added to a specific good. The main objective of the supply chain is to provide the maximum value to customers at low costs and high speed (Chen, Daugherty and Roath 2009).

The literature on the subject and economic practice distinguish at least several types of supply chains depending mainly on the characteristics of the product or industry. Most frequently, the following are enumerated: an agile supply chain, a lean supply chain, a reverse supply chain, a green supply chain, a cold supply chain, a FMCG supply chain, a fashion supply chain.

Each of these chains has its character and importance for a given industry. In addition, more specialized flows of goods and information are created within their range. An example is the food supply chain (Marsden, Banks and Bristow 2000). It is part of the fast-moving consumer foods (FMCG) supply chain, which embraces quickly circulating products at relatively low prices. It includes not only food, but also certain pharmaceuticals, cosmetics and tobacco products.

The food supply chain is described as an integrated process in which raw materials are acquired, transformed into ready products and then delivered to the consumer; colloquially speaking, this is a flow “from field to plate” (Jiang, Zhao and Sun 2009).

The food supply chain is influenced by numerous market trends, such as: change of the retail sales structure (e.g. an increasing number of chain stores), reduction of delivery time and stock minimization, growing popularity of logistics outsourcing, dynamic development of electronic commerce (e.g. purchasing food products via the Internet), environment protection and sustainable development. Moreover, changes in consumer behavior directly affect food

---

¹ The paper was written with financial support from the National Center of Science (Narodowe Centrum Nauki) – the grant no. DEC-2014/13/B/HS4/01552.
² The Poznań University of Economics and Business, Faculty of Informatics and Electronic Economy, Department of Operations Research, Al. Niepodległości 10, 61-875 Poznań, Poland, m.anholcer@ue.poznan.pl
³ The Poznań University of Economics and Business, Faculty of Management, Department of Logistics and Transport, Al. Niepodległości 10, 61-875 Poznań, Poland, arkadiusz.kawa@ue.poznan.pl
supply chains. The most important ones concern an increase in: the demand for healthy food, the need for functional food (special purpose food, e.g. of adjusted nutrient content), the consumption of food from different cultural areas, the consumption of the so-called convenience food, the demand for food consumption outside home.

A characteristic feature of the food supply chain is the diversity of the entities representing it i.e., farmers, processing facilities, traders, wholesalers, distributors, logistics companies and retailers. Very large as well as medium and small enterprises are also among them, and their roles permeate – they may be competitors, suppliers or customers in a given inter-organizational network at the same time. All these links build an additional value by including their resources into the production and delivery of products (Porter 2008).

What is closely connected with the flow of goods among the entities of an inter-organizational network is transport. Transport is one of the key processes in a supply chain. It enables spatial configuration of separate organizations, which exchange goods between one another. It constitutes a significant part of logistics expenses, so it requires appropriate planning and effective control. The properties and possibilities of means of transport should be taken into account, just as many assumptions concerning transport organization, such as loading and unloading places, transport time and load weight.

At present, transport of food and farming products accounts for approximately 10% of the whole transport carried out by Polish enterprises (Trans 2016). Transport is often perceived as the most important element of the logistics system, requiring careful planning and effective monitoring. The Polish agricultural and food industry is characterized by relatively little concentration. Enterprises use various suppliers and sell their products to many buyers. This causes problems with delivery coordination, which affects the quality and costs of products. The lack of planning and making wrong decisions may cause excessive expenses, an inability to achieve the assumed goals and a loss of any chances to get a profit.

The key issue in transport are, then, its costs. They can be a significant part of a company’s overall logistics spending (Murray 2016). According to various estimates, transportation constitutes one-third of the logistics costs (Tseng, Yue and Taylor 2005; Ślusarczyk and Kot 2013, Michałowska 2013). All the costs are transferred to the customer who needs to pay more for the goods. That is why cutting transportation costs is the major target for companies. There is a number of strategies that can be applied by management to help improve performance. Only some of them use complex mathematical methods enabling cost optimization. The issue of transportation cost optimization is, however, treated as one of
the most difficult and, at the same time, most complex problems which transport enterprises deal with (Romanow 2013).

In the case of the food supply chain, the existence of exclusionary constraints is an additional difficulty in the optimization of transport processes. It may particularly apply in situations in which certain types of goods cannot be transported by the same means (e.g. meat and tobacco) or must be delivered on the same day (e.g. bread, pastry, some fruits and vegetables). The delivery time is an especially important issue because of the relatively short sell-by dates of some products. Delays may considerably lower the value of these products or make them useless.

2. Perishable goods

A very useful assumption is that the amounts of goods sent from the supply points are equal to those delivered to the destinations. It simplifies both models and algorithms and has been widely used in the research so far. However, it sometimes does not correspond with reality. There are several reasons causing the changes in amounts of goods during the flow through a logistic chain. Among others, in particular the following groups of causes may be distinguished:

– physical and chemical properties of transported goods,
– decrease in the value of transported goods caused by time,
– inappropriate transportation or storage conditions,
– not registered production rejects,
– accidents,
– crimes.

The framework that catches the phenomenon of perishable products are the generalized flow models, involving arc multipliers. For each arc $e = ij$, a multiplier $r_{ij}$ is defined, which represents the change in the amount of good transported through this arc. If the amount leaving the starting node $i$ equals to $x_{ij}$, then the amount entering the end node $j$ is $r_{ij}x_{ij}$. In the remainder of this paper we assume that all the arcs are lossy i.e., $0 < r_{ij} < 1$, which means that the amount of good decreases during the transportation, but the commodity does not vanish completely (in the latter case the respective arc may be removed and one needs to solve a reduced, simpler instance of the problem).

For more information on the perishable goods, see e.g. (Nagurney et al. 2013).
3. Exclusionary constraints

Exclusionary constraints in transport are the circumstances or conditions under which at least two commodities may not be delivered with the same mean of transport. They may be caused by the features of those commodities, in particular their physical and chemical characteristics. Another reasons are the legal issues, mutual agreements and other requirements imposed for example because of the rules of competition.

In practice we may distinguish at least over a dozen of common reasons that may cause the exclusions. The most frequent are:

− sensitivity for the time of delivery,
− sensitivity for transportation conditions (eg. items that are fragile, prone to damage),
− sensitivity for temperature,
− sensitivity for humidity,
− sensitivity for light,
− sensitivity for smells,
− perishability of chosen products,
− oversize of chosen products,
− competition,
− time windows,
− specific agreements,
− long distance from the loading or unloading sites,
− legal prohibitions.

All of those constraints make the problems to solve much more complicated – the classical network flow algorithms may not be applied here, since in those problems each configuration of the flows that satisfies the standard network balance and capacity constraints is feasible.

4. Mathematical model

Let us start with the presentation of the network topology. We assume that there are three sets of agents: suppliers, warehouses and final destinations. The direct transportation
between supply points and destination points is forbidden. This results in the topology of the supply chain as in Figure 1, see also (Anholcer and Kawa 2012, 2015).

Each node of the network in Figure 1 represents one agent (respectively, supplier, warehouse or destination). However, each arc may represent in fact a bond of arcs (we use a multigraph model), since the arcs represent not only particular connections, but also the means of transport. For example, if \( t \) cars are used to deliver goods between supplier \( S_i \) and warehouse \( W_j \), then \( t \) arcs connect nodes \( S_i \) and \( W_j \). We did not multiply the arcs in the figure for sake of clarity.

**Figure 1.** Supply chain in two-stage transportation process.

We assume that the goal is to minimize the total cost (that is, the real cost, not the amount of money payed e.g. due to the applied tariffs). We use the following notation, see also (Anholcer and Kawa 2012, 2015).

Let \( N = \{1, \ldots, n\} \) be the set of all the nodes \( i, i = 1, \ldots, n \). It consists of three mutually disjoint subsets, the set of suppliers \( N_S \), the set of warehouses \( N_W \) and the set of destinations \( N_D \) i.e., \( N = \bigcup_{i \in \{S,W,D\}} N_i \) and \( N_i \cap N_j = \emptyset, i, j \in \{S,W,D\}, i \neq j \). The set of commodities will be denoted with \( G \), and their indices by \( g, g \in G \). The volume of each commodity is known and equal to \( v(g) \). For each pair of nodes \( i_1 \) and \( i_2 \), the indices of the arcs that connect them will be denoted with \( j(i_1, i_2), j(i_1, i_2) \in J(i_1, i_2) \), where \( J(i_1, i_2) \) denotes the set of all
arcs connecting those nodes (possibly empty, if the nodes are not connected at all). The set of arcs beginning (ending) at \( i \) will be denoted by \( J(i, \cdot) \) (\( \cdot, i \)), respectively. The amount of commodity \( g \) accessible at node \( i \) will be denoted with \( \alpha(i, g) \) (this value will be positive for suppliers, equal to 0 for the warehouses and negative for the destinations, in the latter case the demand is equal to \(-\alpha(i, g)\)). Two variables are attached to each arc \( e \in E \) (where \( E = \bigcup_{i_1, i_2 \in N} J(i_1, i_2) \) is the set of all the arcs in the network) and product \( g \). A real variable \( x(e, g) \) is equal to the flow and a binary variable \( \alpha(e, g) \) is equal to 1 if and only if \( x(e, g) > 0 \). Also a binary variable
\[
\alpha^*(e) = \max_{g \in G} \alpha(e, g)
\] (1)
corresponds with each arc (i.e., it equals to 1 if and only if at least one commodity is transported using arc \( e \)). Unit variable costs \( c(e, g) \) and a fixed cost \( k(e) \) are attached to each arc. A capacity \( u(e) \) is defined for each arc. Also, a multiplier \( r(e, g) \) corresponds with each arc and commodity (it represents the losses during the transportation process).

Similarly, a real variable \( y(i, g) \) (flow) and a binary variable \( \beta(i, g) \), correspond with each node \( i \in N_W \) \( (\beta(i, g) = 1 \) if and only if \( y(i, g) > 0 \)). A variable cost \( d(i, g) \) and a fixed cost \( t(i) \) correspond with each node. Also a variable
\[
\beta^*(i) = \max_{i \in N_W} \beta(i, g)
\] (2)
and capacity \( w(i) \) are defined for each node \( i \in N_W \).

The objective (to be minimized) takes the form:
\[
f(x, y, \alpha, \alpha^*, \beta, \beta^*) = \sum_{g \in G} \sum_{e \in E} c(e, g)x(e, g) + \sum_{e \in E} k(e)\alpha^*(e)
\]
\[
+ \sum_{g \in G} \sum_{i \in N_W} d(i, g)y(i, g) + \sum_{i \in N_W} t(i)\beta^*(i)
\] (3)

Upper bounds and the relations between real and binary variables are described with the following inequalities:
\[
v(g)x(e, g) \leq u(e)\alpha(e, g), e \in E, g \in G,
\] (4)
\[
\sum_{g \in G} v(g)x(e, g) \leq u(e), e \in E,
\] (5)
\[
\sum_{g \in G} \alpha(e, g) \leq |G|\alpha^*(e), e \in E
\] (6)
\[
v(g)y(i, g) \leq w(i)\beta(i, g), i \in N_W, g \in G,
\] (7)
\[ \sum_{g \in G} \nu(g) y(i, g) \leq w(i), i \in N_W, \quad (8) \]

\[ \sum_{g \in G} \beta(i, g) \leq |G| \beta^*(i), i \in N_W. \quad (9) \]

The mass balance constraints are as follows:

\[ \sum_{e \in f(i, \cdot)} x(e, g) = y(i, g), i \in N_W, g \in G, \quad (10) \]

\[ \sum_{e \in j(\cdot, i)} r(e, g) x(e, g) = y(i, g), i \in N_W, g \in G, \quad (11) \]

\[ \sum_{e \in j(\cdot, i)} x(e, g) \leq a(i, g), i \in N_S, g \in G, \quad (12) \]

\[ \sum_{e \in j(\cdot, i)} r(e, g) x(e, g) \geq -a(i, g), i \in N_D, g \in G. \quad (13) \]

The nonnegativity constraints are:

\[ x(e, g) \geq 0, e \in E, g \in G, \quad (14) \]

\[ y(i, g) \geq 0, i \in N, g \in G. \quad (15) \]

If two products \( g_1 \) and \( g_2 \) may not be transported together through arc \( e \), the respective constraint takes the form:

\[ \alpha(e, g_1) + \alpha(e, g_2) \leq 1. \quad (16) \]

If the products \( g_1 \) and \( g_2 \) may not be stored together at node \( i \), we apply the inequality:

\[ \beta(i, g_1) + \beta(i, g_2) \leq 1. \quad (17) \]

More about the usage of binary variables defined above to model various kinds of exclusionary constraints may be found in (Anholcer and Kawa 2015).

In the next section we present an exact algorithm to solve the problem presented above.
5. The algorithm

5.1. Problem transformations

We are going to solve the problem with a branch-and-bound approach. We will subdivide the feasible region with respect to variables \( \alpha(e, g) \).

Note that for given choice of the values of the variables \( \alpha(e, g) = \alpha_{eg} \in \{0, 1\} \), the other binary variables in the model may be easily derived in a unique way. The variables \( \alpha^*(e) \) are defined with the formula (1). The variables \( \beta(i, g) \) are not related directly to \( \alpha(e, g) \), but one may easily observe, that the constraints (10) and (11) imply that for each \( i \in N_W \),

\[
\beta(i, g) = \max_{e \in j(l_i)} \{ \alpha(e, g) \} = \max_{e \in j(l_i)} \{ \alpha(e, g) \}.
\]  
(18)

Finally, the values of \( \beta^*(i) \) may be computed from (2).

For that reason, when the variables \( \alpha(e, g) \) are given the fixed values \( \alpha_{eg} \), the problem reduces to the form of two-stage multicommodity generalized transportation problem with the objective consisting of two parts: the one depending on the flows

\[
f_{\text{var}}(x, y) = \sum_{g \in G} \sum_{e \in E} c(e, g)x(e, g) + \sum_{g \in G} \sum_{i \in N_W} d(i, g)y(i, g)
\]

and the constant part

\[
f_{\text{const}} = \sum_{e \in E} k(e)\alpha^*_e + \sum_{i \in N_W} t(i)\beta^*_i,
\]

where \( \alpha^*_e \) and \( \beta^*_i \) are the fixed values of \( \alpha^*(e) \) and \( \beta^*(i) \), respectively. Moreover, all the arcs \( e \) for which \( \alpha^*(e) = 0 \) and all the nodes \( i \) for which \( \beta^*(i) = 0 \) are removed from the network. This implies that the reduced subproblem consists of only chosen constraints of the form

\[
\sum_{g \in G} v(g)x(e, g) \leq u(e), e \in E: \alpha^*_e = 1,
\]

(21)

\[
\sum_{g \in G} v(g)y(i, g) \leq w(i), i \in N_W: \beta^*_i = 1,
\]

(22)

\[
\sum_{e \in j(l_i)} x(e, g) = y(i, g), i \in N_W: \beta^*_i = 1, g \in G,
\]

(23)
Such a multicommodity generalized flow problem may be solved efficiently using the Lagrangian Relaxation approach, see (Ahuja, Magnanti and Orlin 1993, chapter 17). In this case, the relaxed constraints are in this case (21) and (22). The relaxed subproblems have the form of generalized multicommodity minimum cost flow problems with a very specific structure (two-stage generalized multicommodity transportation problems) and as so may be solved with a simplified version of the generalized network simplex method, see (Ahuja, Magnanti and Orlin 1993, section 15.6). Note that the mentioned method allows also to easily recognize the case when the problem is infeasible. To reassert – if the values of $\alpha(e,g)$ are fixed (equal to some $\alpha_{eg} \in \{0,1\}$), we are able to find easily the values of all the other variables or find out that the subproblem is infeasible.

The algorithm that we present below uses the branch and bound approach, where the choice of the branch of the search tree corresponds with setting the value of some variable $\alpha(e,g)$ to $\alpha_{eg}$ equal either to 0 (one branch) or to 1 (another). It is a common approach in all the branch and bound methods for problems with binary variables. However, in our algorithm, the variables are sorted dynamically. It which makes it significantly different e.g. from the Balas algorithm, see e.g. (Sysło, Deo and Kowalik 1983, section 1.5.2), where the order of the variables is fixed in the very beginning and so each layer of the vertices of the search tree corresponds with one chosen variable.

Each active leaf of the search tree corresponds with the situation where all the variables $\alpha(e,g)$ are fixed, so the complete solution may be found (see the considerations above).

Assume that we chose a node of the tree not being a leaf, and that the values of some of the variables were fixed while moving from the root of the tree. Then this node corresponds with the set of all the solutions, where all the non-fixed variables $\alpha(e,g)$ may take either of the values 0 or 1.
The significant parts of each branch and bound method are:

- the method of estimating from below the objective at the active nodes,
- the rule of choosing next node and next variable to analyze,
- the rules of reducing the subtrees.

We describe them with details in the three following subsections.

5.2. Estimation of the objective

For every active leaf\(^4\), the exact value of the objective is computed. Since when one knows all the values of \(\alpha(e, g)\), all the other binary values may be considered as fixed (see above) and so it is enough to solve an instance of the two-stage generalized multicommodity transportation problem (19-28). Eventually, if some of the variables \(x(e, g)\) equal to 0 in the optimal solution while \(\alpha(e, g) = 1\), we change the values of such \(\alpha(e, g)\) and respectively the values of the remaining binary variables. Since it affects only the constant part of the objective (20), the obtained solution remains optimal. If the problem is infeasible, we set the objective estimate equal to \(\infty\).

If the analyzed newly created node \(v\) is not a leaf, then the lower estimate of the objective is computed in the following way. First, based on the knowledge about which of the variables \(\alpha(e, g)\) were set to 0, the network corresponding with the respective product \(g\) is appropriately reduced by removing the arcs corresponding with the 0 variables. Then an instance of the relaxation of the problem (19)-(28) is solved, where the constraints (21)-(22) are skipped, all the variables \(\alpha^*(e)\) and \(\beta^*(i)\) for the arcs in the modified network are set to 1 and the objective consists only of the variable part (19). This means that we solve an instance of the Lagrangian relaxation of the problem (19-28) with the Lagrangian multiplier equal to 0. It may be easily observed, that this means solving \(|G|\) separate instances of the two-stage generalized transportation problem, one for each product \(g\) (we use the generalized network simplex method for this purpose). If all the subproblems are feasible, the lower estimate \(z_v\) equals to the sum of the optimal values of the objectives of the subproblems. Otherwise we set \(z_v = \infty\) for this node (any other branching may only remove new arcs from some of the networks corresponding with the products, so all the subproblems for the descendants are also infeasible and we may cut off the whole subtree rooted at the analyzed node). Finally, let us

\(^4\) I.e., the active nodes at which all the variables \(\alpha(e, g)\) have fixed values. Any other active node may finally become leaf or not. Closed leaves in turn, and the subtrees rooted at them are abandoned.
observe that we need to solve all the subproblems only in the case of root. In all other cases the new problems arise after a branching, where some variable $\alpha(e,g)$ were set to 0 or 1, all the other variables remaining unchanged. This means that only one new instance of the two-stage generalized transportation problem arises (the one with modified subnetwork corresponding with product $g$) if we choose $\alpha(e,g) = 0$. Since only one edge is removed from the subnetwork, we do not need, in fact, to solve the subproblem from the beginning. We rather check whether the optimal value of the variable $x(e,g)$ is positive. If it is, then the new solution may be derived starting from the current optimum by putting $c(e,g) = M$, where $M$ is a sufficiently large number. If in the new optimum $x(e,g)$ is still positive, then the subproblem is infeasible. Otherwise, the solution to the new subproblem is same as for the initial one, with the only difference lying in the fact that in the initial subproblem $x(e,g) = 0$, while in the new one that variable is not present at all. If in the new optimum $x(e,g)=0$, we do not need to modify the solution.

Of course, it is necessary to store the solutions (both primal and dual) as well as the additional information (e.g. the forest structure of the solution) for each solved subproblem corresponding with every product at each active node.

Note that the values computed in the way presented above are indeed the lower estimates of the objective, since at each node we consider a problem with some constraints removed and, moreover, we ignore the fixed charge. We may also observe, that the lower estimate does not decrease after a subdivision of the feasible region. To be more specific, if we choose the branch where some variable $\alpha(e,g)$ is set to 1, then it does not change (the set of removed edges does not change). On the other hand, if we choose the branch corresponding with the same variable equal to 0, it may happen that the new subproblem has the same solution (then the estimate $z$ does not change) or the solution changes (then the estimate may remain the same or increase, since the new subproblem has more constraints, since the constraint $x(e,g) = 0$ is added).

5.3. Choosing next node and next variable to analyze

The next active node $v$ of the search tree that is going to be analyzed is the one with the lowest estimate $z_v$ of the objective. The method of choosing the variable is a little bit more complex.
In the Balas method (Sysło, Deo and Kowalik 1983, section 1.5.2), as well as in some other branch and bound methods for problems with binary variables, these variables are ordered in the beginning and this order is not changed until the problem is solved. In particular it means that the order of variables at every branch of the search tree is exactly the same. We choose a dynamic method of ordering the variables.

Assume that \( v \) is the node of the search tree at which the branching is to be performed. Some of the variables have been already fixed, and some of them not. We define the effective fixed charge \( K_v(e, g) \) for each not fixed variable \( \alpha(e, g) \) as follows:

- if there is a variable \( \alpha(e, g') \), where \( g' \neq g \), that had been already set to 1, then
  \[
  K_v(e, g) = 0
  \]

- if there is no such variable \( \alpha(e, g') \), but there is a node \( i \) such that \( e \in J(:,i) \cup J(i,:) \) and based on the values of fixed variables it follows that \( \beta^*(i) = 1 \), then
  \[
  K_v(e, g) = k(e),
  \]

- if there is no such variable \( \alpha(e, g') \) and no such node, then
  \[
  K_v(e, g) = k(e) + t(i).
  \]

We choose unfixed variable \( \alpha(e, g) \) with minimum value of \( K_v(e, g) \). If there is more than one such variable, we choose the one with minimum unit cost \( c(e, g) \). If there is more than one such variable, we choose the one with lexicographically lowest indices.

There is one exceptions from the method of choosing the branching variable, described above. It follows from the specific rules of reduction and is presented in the next section.

5.4. The rules of reducing the subtrees

Each node corresponding with an empty subset (i.e., each node for which \( z = \infty \)) is closed and its child nodes will not be analyzed. Also, if a complete solution were already found for some leaf, all the nodes having the objective estimate greater or equal to the value of the objective for that solution should be closed. The following rules also allow to abandon significant parts of the search tree. For any node \( v \) of the tree, the node resulting with setting the next variable to 0 or 1 will be denoted with \( v_0 \) and \( v_1 \), respectively.

Rule 1: If for some node \( v \) of the search tree, where the branching variable is \( \alpha(e, g) \), all the subproblems of the form of two-stage generalized transportation problem corresponding with
$g$, defined for the nodes of the subtree rooted at $v_1$ are infeasible, then remove the whole subtree rooted at $v_0$ and set $z_{v_0} = \infty$.

This rule follows from the fact that if the respective subproblem is infeasible with $\alpha(e, g) = 1$, then it will be obviously infeasible if we remove another edge (here: $e$).

Rule 2: If the branching variable at $v$ is $\alpha(e, g)$ and there is another variable $\alpha(e, g')$ that has not been fixed yet, such that there is an exclusionary constraint $\alpha(e, g) + \alpha(e, g') \leq 1$, then perform immediately the branching with respect to all such variables $\alpha(e, g')$ and for each such branching set $z_{v_1} = \infty$.

Indeed, if the branching variable is equal to 1 and it is involved in an exclusionary constraint, then another variable in this constraint may not be equal to 1. This means that its value must be fixed as 0. When applying rule 2, we expand the search tree by attaching to $v$ a path consisting of the nodes $v_0$ corresponding with all the variables being in conflict with $\alpha(e, g)$. We also add all the respective nodes $v_1$, and immediately mark them as closed. Similar reasoning leads us to the following rule.

Rule 3: If the branching variable at $v$ is $\alpha(e, g)$ and there is another variable $\alpha(e', g')$ that has not been fixed yet, such that $\{e, e'\} \subseteq J(\cdot, i) \cup J(i, \cdot)$ for some node $i$, and there is an exclusionary constraint $\beta(i, g) + \beta(i, g') \leq 1$, then perform immediately the branching with respect to all such variables $\alpha(e', g')$ and for each such branching set $z_{v_1} = \infty$.

The last rule corresponds again with the network subproblems defined for chosen product $g$.

Rule 4: If the branching variable is $\alpha(e, g)$ and choosing $\alpha(e, g) = 0$ would make the sum of demand of any subset of destinations in the subnetwork corresponding with $g$ greater than the sum of the supplies of the suppliers connected with those destinations, then set $z_{v_0} = \infty$.

This rule follows from the fact that the removal of any arc in the network may not cut off the destinations from the suppliers. The rule eliminates most of such situations (unfortunately not all of them, because of the presence of the arc multipliers).
5.5. Algorithm – summary

The algorithm may be summarized as follows.

**Algorithm 1: Branch and bound method for the two-stage generalized multicommodity transportation problem with exclusionary constraints.**

**Step 1** (initialization): Assume that none of the variables has been fixed. Solve the subproblems corresponding with the root and store the estimate of the objective, as well as the solutions and tree structure for each subproblem. Go to step 2.

**Step 2** (optimality test): If all the nodes have been closed, then STOP: the best stored solution is the solution to the original problem. Otherwise, go to step 3.

**Step 3** (branching): Choose the branching node and branching variable according to the rules presented in section 5.3. Perform branching with possible expansion of the tree (rules 2 and 3 in section 5.4). Find the lower estimates of the objective using the formulae (29)-(31) or find the complete solution if a leaf were reached and store the solutions of the respective problems. Go to step 4.

**Step 4** (reducing the tree): Close all the nodes that may be closed using the rules presented in section 5.4 and go back to step 2.

**Conclusion**

After a brief introduction into the subjects of perishable products and exclusionary constraints in transportation and storage, we presented the two-stage generalized multicommodity transportation problem with exclusionary constraints. This is the model that corresponds with minimizing the cost in a two-layer supply chain of perishable products with additional exclusionary constraints.

The main part of our paper is dedicated to the branch and bound method that allows to solve such problems. The method is exact (which is a common feature of the branch and bound methods) and the applied rules of branching and bounding allow to reduce the search tree efficiently.

However, although the method is exact, it has the disadvantage common for all the branch and bound methods: the high computational complexity. It may be used only for the problems of moderate size, since the computational tests performed by the authors show that
the running times on a common PC start to be unacceptable already for the problems with several products and dozens of agents. The only exception are the problems with relatively small capacities and relatively many exclusionary constraints (which is not very surprising, since in such circumstances many branches of the search tree are quickly abandoned).

The authors do not know any other methods that allow to solve problems of this kind, so the comparison to other algorithms is impossible (it makes no sense to compare the method designed for a specific problem with the general algorithms for mixed integer linear programming problems).

In order to solve the problems of a larger size, it is necessary to design effective approximate algorithms. In our opinion, this should be the direction of the further research.

Bibliography


Murray M. (2016), *Reducing Transportation Costs*,
http://logistics.about.com/od/forsmallbusinesses/a/Reducing-Transportation-Costs.htm
Algorithm for generalized transportation problem with exclusionary constraints

Abstract

We consider the generalized two-stage multicommodity transportation problem, where the amount of transported products changes during the transportation process. Moreover, exclusionary constraints are present in the problem. We present a mathematical model of the problem and then an exact algorithm that allows to solve problems of this kind.

Algorytm dla uogólnionego zadania transportowego z warunkami wykluczającymi

Streszczenie

Rozważamy uogólnione dwuetapowe wieloasortymentowe zadanie transportowe, w którym ilość transportowanych dóbr zmienia się podczas transportu. Ponadto, w zadaniu występują warunki wykluczające. Przedstawiamy model matematyczny problemu, a następnie dokładny algorytm pozwalający rozwiązywać zadania tego typu.