УДК 519.8

Research of Production Groups Formation Problem Subject to Logical Restrictions

Lubov D. Afanasyeva∗
Omsk State University,
Mira, 55a, Omsk, 644077,
Russia

Alexander A. Kolokolov†
Omsk Branch of Sobolev Institute of Mathematics, SB RAS,
Pevtsova, 13, Omsk, 644043,
Russia

Received 10.12.2012, received in revised form 10.01.2013, accepted 20.01.2013

This paper is devoted to the production groups formation problem subject to logical restrictions, reflecting interpersonal relations in a team. Mathematical models are developed and investigated using graph theory and linear integer programming, a number of algorithms of combinatorial type is presented, their theoretical and experimental analyses are conducted.

Keywords: operations research, discrete optimization, linear integer programming, production group, heuristics.

Introduction

At the present stage of society development a problem of recruiting qualified staff is becoming actual, which is affected by the opening of new companies, the development of trading networks, increasing requirements to specialists and other factors. The problems arising from it are quite complex and they require the usage of models and methods of discrete optimization. A number of statements of production groups formation problem based on well-known assignment problems are presented in papers [1–3]. The papers [4, 5] suggested their generalizations subject to interpersonal, hierarchical and other types of relations. The paper [6] includes issues of designing production teams related to sets coverings and plant locations.

In this paper we research production groups formation problem with maximized level of comfort between specialists and without strained relations. Graph-theoretic statements and linear integer programming models are proposed and analysed, NP-hardness of this problem is proved as well as its polynomial reduction to the maximal independent set problem [7]. Furthermore polynomially solvable cases are found. Algorithms of combinatorial type for exact and approximate solutions are developed and examined. Computational experiment which showed their practical importance is carried out.

1. Development and analysis of mathematical models

Consider the following statement of the production groups formation problem (denote it by Π). Suppose that there is a number of specialists to be included into the production group. There are two types of binary relations between them: comfortable and strained. It is possible

∗lubovshulepova@gmail.com
†kolo@ofim.oscsbras.ru
© Siberian Federal University. All rights reserved

– 145 –
that relationship between some of the specialists are not defined. The goal is to form the indicated
group so that the number of comfortable relationships between its members is maximal and
strained relations are absent.

Let us describe a graph-theoretic model of the problem. Define a graph $G$ with the vertex
set $V = \{1, \ldots, n\}$ and the edge set $E \subseteq \{(i, j) | \ i, j \in V, i \neq j\}$.

The vertices of the graph $G$ represent candidates for inclusion into the production group and
the edges represent the relations between them with $E = E_1 \cup E_2$, where

1) $E_1$ is the set of edges that reflect comfortable relations between specialists,
2) $E_2$ is the set of edges corresponding to strained relations.

For convenience in exposition, we will color the edges of the graph $G$ with different colors: the
edges from $E_1$ with green, and from $E_2$ with red. The graph $G$ is not necessarily connected. For
example, there are the following extreme cases: the set of edges is empty, it consists of only red
or only green edges. The aim of the optimization problem is to find $V' \subseteq V$ such that a subgraph
$G'$ of the graph $G$ generated by a subset $V'$ does not contain edges from $E_2$ and includes the
maximal number of edges from $E_1$.

We describe now a model of linear integer programming (LIP) for the considered problem.
Introduce the following boolean variables:

$x_j = 1$, if specialist $j$ is included in the group being formed, $x_j = 0$ otherwise, $j = 1, \ldots, n$;
$y_{ij} = 1$, if specialists $i$ and $j$ are included in the team, $y_{ij} = 0$ otherwise, $(i, j) \in E_1$.

The LIP model is as follows:

$$
\sum_{(i,j) \in E_1} y_{ij} \rightarrow \max
$$
subject to

$$
x_i + x_j \leq 1, \ (i, j) \in E_2, \quad (1)
$$
$$
x_i + x_j \geq 2y_{ij}, \ (i, j) \in E_1, \quad (2)
$$
$$
x_j \in \{0, 1\}, \ j = 1, \ldots, n, \quad (3)
$$
$$
y_{ij} \in \{0, 1\}, \ (i, j) \in E_1. \quad (4)
$$

The objective function (1) maximizes the number of green edges of the subgraph $G'$. Constraints (2) correspond to the condition of absence of red edges in $G'$. Constraints (3) ensure the fulfillment of the following condition: the variable $y_{ij}$ can take a value equal to one, only if both
vertices $i$ and $j$, which are incident to one green edge, belong to a feasible solution.

It is important to note that this problem always has a feasible solution. Indeed, even if all
edges from $G$ are red, some independent set of vertices from $V$ will be a feasible solution.

As shown below the problem (1)–(5) is NP-hard. To prove this, we construct a special graph
$G$. An arbitrary subgraph $G_1$ of $G$, in which all the edges are red, and a subgraph $G_2$ consisting
of a single vertex are given. Connect each vertex of $G_1$ and $G_2$ with green edge. It is easy to
notice that the vertices of $G$ which are not incident to red edges belong to the optimal solution
of (1)–(5). It remains to consider the subgraph $G_1$ all edges of whose are colored red. We need
to choose the vertices from $G_1$ to be included in the optimal solution satisfying the following
conditions: no two vertices are incident to red edge; the number of vertices should be maximal, as
in this case the largest number of green edges is in the optimum solution. Obviously, the indicated
formulation corresponds to the maximal independent set problem which is NP-hard [8].

Next, we will establish a connection of the problem (1)–(5) with the well-known maximal
independent set problem (denote it by $IS$) to apply $IS$ results to results of $\Pi$. Define the
function $F$ for the polynomial reduction $\Pi$ to the $IS$ problem [8]. Let $G = (V, E)$ be a graph of
the problem (1)–(5). We describe the construction of the graph $G_d = (V_d, E_d)$ for the corresponding
$IS$ problem. Green edges of $G$ are transformed into vertices $G_d$, namely $F(e) \in V_d$ if and only
if $e \in E_1$. An edge $(F(e_1), F(e_2)) \in E_d$ if and only if $e_1, e_2 \in E_1$ and there are vertices $i, j \in V$
incident to $e_1, e_2$, respectively, such that the edge $(i, j) \in E_2$. 

\[ \text{Page 146} \]
Theorem 1. The problem (1)–(5) is polynomially reduced to the maximal independent set problem.

Proof. We use the function \( F \) for the given reduction. Obviously, it is constructed in polynomial time. We will show that \( G^* \) is an optimal solution of the problem (1)–(5) if and only if \( F(G^*) \) is a maximal independent set.

Suppose, to begin with, that a vertex-generated subgraph \( G^* \) of the graph \( G \) is an optimal solution of \( \Pi \). In this case, the graph \( F(G^*) \) consists of an independent set of vertices by construction because \( G^* \) does not include red edges. Now we will prove that \( F(G^*) \) also contains the maximal number of such vertices. Suppose that this is not so, and some other graph \( F(G_{opt}) \) is the optimal solution of the IS, and the number of vertices in \( F(G_{opt}) \) is more than the number of vertices in \( F(G^*) \). Then the subgraph \( G_{opt} \) of the graph \( G \) is a feasible solution for the problem (1)–(5) in which the number of green edges is more than in the subgraph of the optimal solution \( G^* \). We have come to a contradiction.

Suppose we got the optimal solution of the maximal independent set problem. Then in the corresponding problem \( \Pi \) we choose the maximal number of green edges, and red edges are absent because of the independence of the set. It is proved by the argument similar to the given above.

The proposed reduction can be useful in solving some particular cases of \( \Pi \) for which the respective maximal independent set problem is polynomially solvable. For example, the problems (1)–(5) for the types of graphs like a line, a simple chain, a tree are polynomially solvable because the corresponding maximal independent set problems belong to the class \( P \) [7, 9].

2. Development and analysis of algorithms

This section provides a description of combinatorial algorithms for the solution of the problem (1)–(5). Consider a version of the algorithm in which solution of the original problem is reduced to solving the sequence of problems of choosing a set of green edges in the graph (denote it by \( BB \)).

In proposed algorithm the branching procedure and estimates of the value of the objective function for current problems are being used. The branching process is a successive partition of the set of feasible solutions into subsets followed by removal of those sets which do not contain the optimal solution. This operation can be recursively applied to the subsets that are conventionally called nodes of a search tree. Each subset in this partition is presented as a child of a vertex-parent and the original problem is a root of the tree.

In the \( BB \) algorithm at each node of the search tree we select a red edge of the graph \( G \) and construct two branches:

- the first vertex of the red edge is not included into the solution;
- the second vertex of the red edge is not included into the solution.

We introduce the following notation: \( rec \) is the best value of the objective function at the current iteration, \( f(G') \) is the number of green edges included in the graph \( G' \). The \( BB \) algorithm can be described both in terms of graph theory and LIP model. We begin with the first version of the description of the algorithm.

Step 0. \( rec = 0, \ G' := G \).

Step 1. Check the graph \( G' \) for the occurrence of red edges. If there are no red edges then go to step 4. Otherwise choose one of the red edges, whose vertices’ branching will be done, and construct two branches as mentioned above.

Step 2. Move along the branch which has not yet been examined. If there are no such branches then go to step 3. Associate the next node of the search tree with the graph \( G' \) that
consists of all the vertices that are not prohibited on the branch. Compute the value of the function \( f(G') \) and compare it with the value of \( \text{rec} \). If \( \text{rec} > f(G') \) then go to step 3, otherwise go to step 1.

Step 3. Go back to the parent node. If this node is the root of the search tree and all branches are examined then go to step 5. Otherwise go to step 2.

Step 4. Compare the value of the function \( f \) at this node of the search tree with the value of \( \text{rec} \). If \( f(G') > \text{rec} \) then \( \text{rec} := f(G') \). Go to step 3.

Step 5. The value of \( \text{rec} \) is the optimal value of the objective function of the original problem, and the set \( V' \subseteq V \) of the graph \( G' \) on which \( \text{rec} \) is attained is the optimal solution. The algorithm is complete.

The main difference between the description of the \( BB \) algorithm in terms of the LIP from the previous version is that in the initial moment we set all \( x_j = 1 \), \( j = 1, ..., n \) and then check the validity of conditions (2). The branching occurs when some restrictions (2) are not satisfied. It should be emphasized that the LIP model in this paper is used to compare the \( BB \) algorithm with the package IBM ILOG CPLEX. In addition, it may be useful for further study of the problem.

It is easy to show that the \( BB \) algorithm in a finite number of iterations finds the optimal solution to the problem \( II \) and its complexity is less than \( O( p(|E|) \cdot 2^{|E_1|}) \), where \( p(|E|) \) is a polynomial in the number of edges in the graph \( G \). Now we will show that the upper bound of complexity is achieved. To do it we consider an auxiliary graph for (1)–(5) consisting of two parts; each of them has \( 2h \) vertices numbered from 1 to \( 2h \). The vertex \( i \) of the first part will be connected with the vertex \( i \) of the second part by a green edge, \( i = 1, ..., 2h \); the vertex \( i \) of the first part will be connected with the vertex \( i + 1 \) of the same part by a red edge, \( i = 1, 3, 5, ..., 2h - 1 \). In this case the complexity of the \( BB \) algorithm is equal to \( O(p(3h) \cdot 2^h) \).

To reduce the running time of the \( BB \) algorithm we have proposed a number of heuristics which have been used in order to find an original feasible solution. For example, the idea of one of them (call it \( HA \)), which proved to be quite effective, follows. First, all vertices of the original graph \( G \) are included in the optimal solution, then we choose the node with the largest number of red edges incident to it and remove this vertex. This operation is repeated until all red edge are excluded from the solution. The complexity of the heuristics \( HA \) is \( O(n^3) \). The heuristics \( HA \) can also be used to find an approximate solution of \( II \).

To conduct experiments as well as to develop convenient tool for specialists involved in the formation of production groups, the mentioned above algorithms are realized in C++ as a software application. A computational experiment was performed to analyze the developed models and algorithms, in particular, their efficiency was compared with integer programming software package IBM ILOG CPLEX. Testing was carried out on problems with random input data using varied values of the parameters responsible for a number of vertices, red and green edges. Calculations showed that the developed algorithms are suitable for solving stated production groups formation problems. On some series of test problems the \( BB \) algorithm performed faster than this package. For example, the running time on the tests with about 100 vertices was on average 30 seconds while the package for solving the same problems required from 10 to 15 minutes. The usage of the \( HA \) heuristics allowed to reduce the operation time of the \( BB \) algorithm by about 20%. Mathematical models constructed by us were applied in an Omsk firm for the formation of a production group.

**Conclusion**

In the article we have studied the production groups formation problem subject to logical restrictions. Mathematical models, with the maximized level of comfort relations between specialists and without strained relations, have been constructed.
NP-hardness of this problem is proved as well as its polynomial reduction to the maximal independent set problem. To solve the problem, algorithms are proposed and realized, a computational experiment has been carried out. It has shown the prospectivity of application of the developed models and algorithms in decisions making support systems.

References


Исследование одной задачи формирования производственных групп с учетом логических ограничений

Любовь Д. Афанасьева
Александр А. Колоколов