A heuristic for a special case of the generalized assignment problem with additional conditions

Lidia Zaozerskaya
Sobolev Institute of Mathematics, Omsk, Russia
E-mail: zaozer@ofim.oscsbras.ru

Abstract. We consider the following variant of the generalized assignment problem (GAP). There are a set of agents and a set of jobs for which a single resource use. Each job is to assign to one and only one agent subject to the constraints on the capacity and loading of agents. The resource expense for executing any job is independent of the agents choice unlike the profit from the job. Each job has a certain type (or color). For every agent, the maximum possible number of the job types given. It is necessary to find a feasible assignment of agents to jobs so that all jobs were completed and total profit was maximized. Finding a feasible solution to this problem is NP-hard. We present a heuristic algorithm based on the ideas of random search and local improvement of solutions. Used the mixed-integer programming (MIP) relaxation and variables fixing, we construct a set of integer linear programming (ILP) subproblems similar to the original problem and solve them by the general MIP solver. The results of a computational experiment for tasks with random initial data are presented.

1. Introduction
The GAP is the well-known NP-hard discrete optimization problem. One of the interpretations of the problem is as follows. There are the agents set and the jobs set. All agents have resource reserves for executing jobs. Each job can execute only by one agent. For any pair of agent-job, resource consumption to complete the job, as well as performed job profit or the execution job cost given. It is necessary to find an assignment of agents on jobs that all jobs be completed and the total profit is maximized or the total cost is minimized. The GAP generalizes the classical assignment problem and has many real-world applications such as vehicle routing, grouping, loading for flexible manufacturing systems, assigning ships to overhaul, assigning jobs to computers in computer networks, optimizing the supply chain, etc [5].

There is a variant of the GAP in which a subset of jobs with the maximum total profit need find to subject the capacity of agents. For this case, many results were obtained for constructing approximation solutions [1, 6]. An important special case of this GAP is the multiple knapsack problem in which the quantity of the resource for performing any job and profit of the job does not depend on the choice of agent.

Currently, the formulations of the GAP with additional conditions are considered, for example, multiple-resource GAP, multilevel GAP, dynamic GAP, the task bottleneck GAP [5].

In [2, 8], another special case of GAP was considered. Here, agents differ only in capacity and profit ratios. The resource consumption to complete the jobs does not depend on the choice of agent. It is shown that an optimal solution to this problem is characterized as a basic feasible solution to a slightly modified transportation problem.
We consider the variant of the GAP from [2] with additional restrictions on the agents loading and with color conditions. The GAP with color conditions was discussed earlier, for example, in [6]. In this problem, each job additionally has a type (or color). The maximum possible number of job types is specified for each agent. In the GAP variant, we are considering, the sought distribution of agents by jobs must ensure the know minimum loadings of agents also. This GAP arises when solving the bicriteria problems of the distribution academic load problem [9, 11] or the supply management with discrete batch sizes [10].

Many exact solution methods and heuristics, including metaheuristics such as genetic algorithms, tabu search algorithms, simulated annealing, developed to solve the GAP and its variants [5, 7]. On the other hand, there are papers devoted to general-purpose IP and MIP algorithms. For example, a heuristic based on the idea of searching for good feasible solutions located near the LP optimal solution was developed in [4]. In [3], a new solving process is proposed for general MIPs. It based on a clever exploration of the solution neighborhoods defined through constraints of local branching and on using the capabilities of MIP solvers was proposed.

In [10], for the problem under consideration, we applied a heuristic in which a set of reduced ILP problems is constructed based on information about a feasible (or optimal) solution to MIP-relaxation. These problems are obtained from the original problem by randomly fixing a part of the Boolean variables and then decided by the CPLEX solver. It was shown that such a heuristic applies to medium-size problems.

In this paper, we develop such a way of constructing approximate solutions. At the beginning of the process, we use a random search to find a feasible solution similar to the obtained MIP relaxation solution. We fix a part of the Boolean variables of the original ILP problem and trying to find an optimal or approximation solution for the constructed subproblem by solver CPLEX. If a feasible solution is found, then the next solution is sought similar to it, etc. A local improvement procedure is applied to obtained feasible solutions. The algorithm tested on problems with random initial data.

2. Problem Formulation

Let $I$ and $J$ be a set of agents and a set of jobs. A single resource is used for all jobs. We denote by $a_j$ the amount of this resource needed to complete the job $j$. Let $c_{ij}$ be the profit from the job $j$ performed by agent $i$, $i \in I$, $j \in J$. Each job has a certain type and the total number of types is $n$. Let $J_k \subset J$ be a set of jobs of type $k$, $k \in K = \{1, ..., n\}$, and $\cup_{k \in K} J_k = J$. Each job can be assigned to only one agent. Each agent can perform no more than $T_i$ types of jobs and use no more than $b_i$ units of a resource, i.e., $b_i$ is the capacity of agent $i$. Moreover, the total amount of the used resource cannot be less than $d_i$ for agent $i$, where $d_i \leq b_i$. The last condition is aimed at obtaining a fair loading of agents. It is required to find the appointment of agents for jobs so as to ensure that all jobs are completed and the total profit is maximum. Further, we assume that the total amount of resources required to complete all jobs does not exceed the total capacity of all agents and is not less than the sum of the lower bounds for all agents.

We introduce boolean variables: $x_{ij} = 1$ if agent $i$ is assigned to job $j$, otherwise $x_{ij} = 0$; $y_{ik} = 1$, if the agent $i$ is assigned to job of type $k$, otherwise $y_{ik} = 0$. Let $X$ be the set of all variables $x_{ij}$. Now, the ILP problem is written as follows

$$f(X) = \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} \rightarrow \text{max}$$

(1)

to subject

$$\sum_{j \in J} a_j x_{ij} \leq b_i, \quad i \in I,$$

(2)
\[
\sum_{j \in J} a_j x_{ij} \geq d_i, \quad i \in I, \quad (3)
\]

\[
\sum_{i \in I} x_{ij} = 1, \quad j \in J, \quad (4)
\]

\[
\sum_{j \in J_k} x_{ij} \leq |J_k| y_{ik}, \quad i \in I, \quad k \in K, \quad (5)
\]

\[
\sum_{k \in K} y_{ik} \leq T_i, \quad i \in I, \quad (6)
\]

\[
x_{ij}, y_{ik} \in \{0, 1\}, \quad i \in I, \quad j \in J, \quad k \in K. \quad (7)
\]

Constraints (2) and (3) set the upper and lower bounds on the total load of agents. Conditions (4) describe the requirement that each job must be performed by only one agent. It follows from inequalities (5) that \(y_{ik} = 1\) if agent \(i\) does at least one job of type \(k\). Constraints (6) set upper bounds on the number of different types of jobs assigned to each agent. The conditions (7) describe the variables of the model. The objective function (1) is the total profit, and it is maximized.

Clearly, (1), (2), (4), (7) is the GAP. The problem under consideration is a new variant of the GAP with color conditions and with lower bounds on agents loading (GAPCCLB). The GAPCCLB is NP-hard as a generalization of the GAP. We weaken the Boolean conditions for the variables \(x_{ij}\) replacing (7) with the following restrictions

\[
0 \leq x_{ij} \leq 1, \quad y_{ik} \in \{0, 1\}, \quad i \in I, \quad j \in J, \quad k \in K. \quad (8)
\]

We define the mixed integer programming (MIP) relaxation of the considered GAP variant as problem (1)–(5), (8) and the linear programming (LP) relaxation as problem (1)–(5) to subject

\[
0 \leq x_{ij}, \quad y_{ik} \leq 1, \quad i \in I, \quad j \in J, \quad k \in K. \quad (9)
\]

### 3. Heuristic

We propose a simple heuristic probabilistic algorithm using the ideas of random search near the MIP relaxation solution, local lifting, and local improvements of the received feasible solutions. It creates a series of reduced subproblems for which the exact or approximation solutions are found by exact methods such as the branch and bound algorithm. Here, in contrast to [10], we used a constant probability of fixing Boolean variables and added procedures for local improvement of the received feasible solutions.

The solution to the problem under consideration is defined completely by the matrix \(X\), therefore we will use this denotation for simplicity. If the first stage (i.e., finding a feasible solution) was unsuccessful, the algorithm stops. Otherwise, we have the best possible solution that we will denote as \(X_{\text{best}}\). In the second stage, we will tried to improve this solution. With a certain probability, we save the agents assignment from \(X_{\text{best}}\) and solve the reduced problem (1)–(7) with additional restriction

\[
f(X) \geq f_{\text{best}} + 1. \quad (10)
\]

For this, the CPLEX solver use within a given time. If the feasible or optimal solution is obtained and it improves the value \(f_{\text{best}}\) then the solution \(X_{\text{best}}\) is updated.

The local improvement procedure is applied to the obtained feasible solutions.

Input parameters:

- \(N\) is the number of iterations (the constructed ILP subproblems number);
- \(N_1\) is the number of iterations of the first stage;
\( p \in (0, 1) \) is the probability of fixing the single MIP-solution components;
\( t_0 \) is the time of solving the MIP or LP relaxation;
\( t_1 \) is the time of decide by CPLEX solver the ILP subproblem;
\( X_{\text{best}}, f_{\text{best}} \) be the best found solution and its value of the objective function.
We put \( f_{\text{rec}} = 0 \).

Algorithm \( A \)

**Iteration 0.**
The MIP relaxation of problem (1)–(7) is decide by CPLEX solver within time \( t_0 \). Let \( \bar{X} \) be optimal or feasible relaxation solution, and \( \mathcal{P} = \{(i, j) \mid \bar{x}_{ij} = 1\} \). If \( \bar{X} \) is not obtained within time \( t_0 \), we use the LP relaxation solution obtained by CPLEX solver within the same time.

**Iteration \( h \) (\( h \leq N \)).**
In problem (1)–(7), (10), we put \( x_{ij} = 1 \) with probability \( p \) for \((i, j) \in \mathcal{P}\) when the total assigned load for agent \( i \) does not exceed \( b_i - 2a_{j_0} \) where \( j_0 \) is index of job with minimum resource costs among unassigned jobs.

The constructed ILP subproblem decide by CPLEX solver within time \( t_1 \). If a feasible solution is obtained, and its objective function value exceeds \( f_{\text{rec}} \), then \( f_{\text{rec}} \) and \( X_{\text{rec}} \) are updated.
If \( h = N_1 \) and the feasible solution \( X_{\text{best}} \) is found, then put \( \mathcal{P} = \{(i, j) \mid (x_{\text{rec}})_{ij} = 1\} \). The procedure EPA is applied to \( X_{\text{best}} \).
If \( h > N_1 \) and \( f_{\text{best}} \) has changed, then the set \( \mathcal{P} \) is updated according to the received \( X_{\text{best}} \).
The procedure EPA is applied to \( X_{\text{best}} \).

We take the best feasible solution found by the algorithm as the approximate solution to the problem.

**Procedure EPA (Enumeration of pairs of agents)** is applied to each updated feasible solutions beginning with iteration \( N_1 \). Procedure EPA tries to improve the current found solution \( X_{\text{best}} \). To do this, we consider all pairs of agents. For select pair \( i, j \), we solve a subproblem of the form (1)–(7), (10) to redistribute jobs for this pair to increase of the objective function value. The rest of the agents save the assignments of jobs according to solution \( X_{\text{best}} \). If the reassignment improves the objective function value, then it is performed and \( X_{\text{best}} \) update.

When testing, we considered the algorithm modification \( A_m \) when the procedure of the enumeration of agent triples (ETA) is applied to the best solution found by Algorithm \( A \).

### 4. Computational Experiment
The proposed algorithm was tested on two series on 10 tasks that were used earlier in [10]. The tasks haven \( |J| = 22, |K| = 45 \). Other parameters were generated randomly from following ranges \( c_{ij} \in [1, 10], a_j \in [40, 120], b_i \in [300, 1400], |J_k| \in [1, 9] \) for all \( i, j, k \). We put \( d_i = [0.95b_i + 0.5] \). The tasks of series S6 and S9 differ only by the number of maximum possible job types for agents. These parameters were calculated using the formula \([b_i/b_{\text{max}}Q + 0.5]\) where \( b_{\text{max}} \) is the maximum capacity of agents. The value of \( Q \) equal to 9 for series S9, and to 6 for S6. In the obtained tasks, the average number of jobs was equal to 225. The average size of corresponding ILP tasks is 5900 variables and 2300 restrictions.

The calculations were performed on a PC with Intel(R) Xeon(R) processor 2.50 GHz. We used the CPLEX solver to find the exact solution in the mode "Emphasize feasibility over optimality". Computational experiments show that the tasks are hard for CPLEX solver for proof of optimality. The optimal solutions were found only for the 3rd ad 4th tasks of S9. The feasible solutions did not find for 2nd task from S6 and 4th, 5th tasks of both series. Note, the tasks have a slight duality gap increases as \( Q \) decreases.
Table 1. Results for series $S_9$.

| name | $\Delta_1$ | $t_1$, sec | $\Delta_{HT}$ | $\Delta_{best}$ | $t_{cplex}$, sec | $\Delta_{aver}$ | $\Delta_{max}$ | $t_A$, sec |
|------|------------|-------------|----------------|-----------------|-----------------|----------------|---------------|-------------|
| $S_1$ | 20.75      | 2           | 1.19           | 0.20            | 16300           | 0.92           | 1.09          | 470         |
| $S_2$ | 6.05       | 5424        | 100.00         | 2.48            | 45000           | 1.22           | 1.42          | 1758        |
| $S_3$ | 4.18       | 13          | 0.19           | 0.05            | 15300           | 0.66           | 0.77          | 364         |
| $S_4$ | 9.18       | 11          | 0.18           | 0.05            | 15500           | 0.39           | 0.45          | 344         |
| $S_5$ | 100.00     | –           | 100.00         | 100.00          | 42500           | 1.71           | 2.49          | 2090        |
| $S_6$ | 4.81       | 745         | 1.25           | 0.22            | 12450           | 0.63           | 0.89          | 969         |
| $S_7$ | 11.98      | 125         | 1.27           | 0.38            | 10000           | 0.92           | 1.04          | 380         |
| $S_8$ | 3.85       | 569         | 2.10           | 0.44            | 11400           | 1.01           | 1.36          | 998         |
| $S_9$ | 100.00     | –           | 100.00         | 100.00          | 42500           | 1.77           | 2.25          | 1297        |
| $S_{10}$ | 6.42  | 347         | 100.00         | 0.41            | 12000           | 0.70           | 0.78          | 301         |

Tables 1 and 2 contain the received results of the solver ILOG CPLEX 12.1.0 and heuristic algorithm $A$ for both series. For CPLEX solver, we indicate the time executing ($t_{cplex}$) and time of first feasible solution obtaining ($t_1$); percent deviation of the objective function value of the first feasible solution found from the known optimal value or the best upper estimate ($\Delta_1$). Columns $\Delta_{HT}$ and $\Delta_{best}$ specified similar values for the solution found during a time equal to the average time of the heuristic executing and for the best solution obtained during all time of running the CPLEX solver. Heuristic $A$ was running 5 times for each task. Column $\Delta_{aver}$ reports the average percentage gap that computed as $100 \times \frac{\text{heuristic value} - \text{best upper bound}}{\text{best upper bound}}$; $\Delta_{max}$ indicates the maximum gap and $t_A$ is an average executing time.

We should note that the number of fractional components of the solution of the MIP relaxation does not exceed $2m$ for all considered tasks. This prompted us to look for an approximate solution to the GAPCCLB near the relaxation solution.

Table 2. Results for series $S_6$.

| name | $\Delta_1$ | $t_1$, sec | $\Delta_{HT}$ | $\Delta_{best}$ | $t_{cplex}$, sec | $\Delta_{aver}$ | $\Delta_{max}$ | $t_A$, sec |
|------|------------|-------------|----------------|-----------------|-----------------|----------------|---------------|-------------|
| $S_1$ | 13.10      | 27          | 4.05           | 3.26            | 23025           | 2.60           | 3.15          | 931         |
| $S_2$ | 100.00     | –           | 100.00         | 100.00          | 15020           | 3.31           | 3.97          | 1181        |
| $S_3$ | 11.26      | 121         | 2.45           | 0.92            | 10330           | 2.02           | 2.70          | 614         |
| $S_4$ | 9.94       | 84          | 2.78           | 1.13            | 13196           | 1.79           | 2.07          | 707         |
| $S_5$ | 100.00     | –           | 100.00         | 100.00          | 3100            | 5.61           | 5.98          | 734         |
| $S_6$ | 10.61      | 137         | 3.11           | 1.60            | 13883           | 1.58           | 2.03          | 840         |
| $S_7$ | 13.28      | 114         | 7.21           | 2.34            | 15860           | 1.87           | 2.24          | 530         |
| $S_8$ | 9.53       | 2633        | 3.87           | 3.18            | 11983           | 2.77           | 3.04          | 1385        |
| $S_9$ | 100.00     | –           | 100.00         | 100.00          | 30027           | 4.82           | 5.68          | 895         |
| $S_{10}$ | 13.22 | 405         | 5.10           | 3.22            | 11572           | 2.11           | 2.69          | 657         |
We tested heuristics $A$ and $A_m$ with $p = 0.7$ and $p = 0.9$. For series $S9$, we have received an approximation solution for all tasks on all starts. The value $\Delta_{\text{aver}}$ is not over 1.77 percent, and maximum $\Delta_{\text{max}}$ is 2.49. For $p = 0.9$, the results are at the results are slightly worse in particular $\Delta_{\text{aver}}$ is not over 2.64. For the 2nd, 5st, 9st, and 10th tasks, the CPLEX solver didn’t find feasible solutions during the average time of one running the heuristic. The heuristic is slightly inferior to the solver for problems 3 and 4 for a similar time. Application of the ETA procedure results in a decrease in index $\Delta_{\text{aver}}$ an average on 0.4 but increases the average solving time by 2.25 times.

The series $S6$ is more complicated for CPLEX solver and heuristic $A$. For all tasks, the average percentage gap increased. For task 5, this index equal to 5.61 for 3 successful launches. Problems 5 and 9 turned out to be the most difficult for heuristics. We increased the number of launches to 15. The success rate was 7 in both cases.

5. Conclusion
We considered the special variant of the GAP with color conditions. In this problem, lower bounds on the agents’ load are given. Besides, resource costs for the execution of jobs do not depend on the choice of the agent. This problem is NP-hard.

We proposed simple heuristic probabilistic algorithms for finding approximate solutions to the problem. The algorithm searches an approximation solution near the optimal or feasible solution of MIP relaxation. On each algorithm iteration, the reduced ILP problem is solved by the MIP solver. A comparison of the computational experiment results for problems with random initial data showed that the proposed heuristic applied to solve some real-world problems, for example, distribution of the academic load of teachers. It seems to us interesting in the future to use the capabilities of general MIP solvers together with the ideas of well-known heuristics.

Acknowledgments
The work was supported by the program of fundamental scientific researches of the SB RAS I.5.1, project 0314-2019-0019.

References
[1] Cohen R, Katzir L and Raz D 2006 An efficient approximation for the generalized assignment problem Inform. Process. Lett. 100 pp 162–166
[2] De Maio A and Roveda C 1971 An all zero-one algorithm for a certain class of transportation problems Oper. Res. 19 6 pp 1406–1418
[3] Fischetti M and Lodi A 2003 Local branching Math. Program. B 98 pp 23-47
[4] Ibaraki T, Ohashi T and Mine H 1974 A heuristic algorithm for mixed-integer programming problems Mathematical Programming Study 2 pp 115-136
[5] Kundakcioglu O E and Alizamir S 2008 Generalized assignment problem In: Floudas C., Pardalos P. (eds) Encyclopedia of Optimization Springer, Boston, MA pp 1153–1162
[6] Nutov Z, Beniaminy I and Yusterc R 2006 A (1–1/e)-approximation algorithm for the generalized assignment problem Oper. Res. Lett. 34 3 pp 283–288
[7] Raja Balachandar S and Kannan K 2009 A new heuristic approach for the large-scale generalized assignment problem International Journal of Computational and Mathematical Sciences 311 pp 969–974
[8] Srinivasan V and Thompson G L 1973 An algorithm for assigning uses to sources in a special class of transportation problems Oper. Res. 21 1 pp 284–295
[9] Zaozerskaya L Analysis of integer programming model of academic load distribution 2019 In: Bykadorov I., Strusevich V., Tchemisova T. (eds) Mathematical Optimization Theory and Operations Research. MOTOR 2019. Communications in Computer and Information Science Vol 1090. Springer, Cham, pp 266–279.
[10] Zaozerskaya L A and Plankova V A 2019 Researching and solving a bicriteria supply management problem with the given volumes of batches J. Phys.: Conf. Ser. 1210 012164
[11] Zaozerskaya L A, Plankova V A and Devyaterikova M V 2018 Modeling and solving academic load distribution problem Proc. Int. Conf. OPTA-SCL (Omsk) Vol 2098 (CEUR-WS) pp 438–445