A New Generalized Assignment Problem Model and Its Method

Sheng XIONG[1*], Yu-qi YANG[2] and Ming-fang NI[3]

[1]PLA University of Science and Technology, Nanjing, Jiangsu Province, China

[2]PLA University of Science and Technology, Nanjing, Jiangsu Province, China

[3] PLA University of Science and Technology, Nanjing, Jiangsu Province, China

[*]Email:1143950297@qq.com

Keywords: Generalized assignment problem, Nonlinear integer programming, Quadratic integer programming, Linear integer programming.

Abstract. The assignment problem generally assumes that one person undertakes one task and one task must be accomplished by only one person. However, a real task usually calls for more than one person. In some cases, those persons are required to work at the same time, if so, the classical assignment model will be inoperative to describe the problem. To solve this problem, a new generalized assignment model based on nonlinear integer programming is proposed in this paper. The new model is available for those problems in which a single task may demand more than one person to be engaged in at the same time, as a result, it makes up for the existing generalized assignment models' deficiency. Moreover, a method to convert the quadratic integer programming model to a linear one is also presented. Thus, the generalized assignment problem can be solved within a short time by using the existing mathematical software LINGO. The numerical results indicate that the generalized assignment model and method proposed are valid.

Introduction

The assignment problem requires the numbers of workers and tasks to be equal. Besides, it assumes that one person undertakes one task and one task must be accomplished by only one person. This type of problem is generally referred to as the balanced assignment problem, which is a special case of the transportation problem. The best method to solve the balanced assignment problem is the Hungarian Algorithm [1]. There are a lot of promotions of the balanced assignment problem which are generally referred to as the generalized assignment problem(GAP) [2]. In paper [3], a solution to the case where one task calls for one person while one person can undertake a number of tasks is presented. In paper [4], the case in which both the persons and tasks to be arranged are more than that of actual demands is discussed. The numbers of workers and tasks are usually unequal in actual personnel assignment problem, the existing research on GAP has already taken this fact into consideration, but it generally assumes that each person's work is independent and makes no difference to each other. Thus, the optimization model built from it is usually some linear integer programming(LIP)[5]. In general, the GAP can be solved by introducing a transformation method to translate it into a balanced assignment problem[6] . However, the actual personnel assignment problem will be more complicated, for instance a single task needs to be assigned with a number of persons to work at the same time. Consider the following problem.

In a joint exercise, there are a number of tasks to be completed for a technical department which consists of some technicians. The differences of technicians' specialty allow each of them to undertake different types of task. However, in order to achieve those tasks in high quality, each technician is allowed to undertake no more than one task. On the other hand, as the difficulty of each task may be different, some of those tasks require more than one technician to work at the same time. Once a task is assigned to one or more technicians, the corresponding time cost would be known. The problem to be solved is to find a proper assignment to complete all the tasks in the shortest time.

Manifestly, this kind of problem often appears in reality, but few of the existing surveys have taken the fact that a number of persons have to work together at the same time into account. As some tasks require several technicians to work at the same time, the optimization model built from the problem of above should be a nonlinear integer programming(NIP). Therefore, it's not only of important theoretical significance, but also of great application value to do some research on problems like this. To deal with the case where a number of persons have to work at the same time to complete a single task, a new type of generalized assignment model will be proposed in this paper. What's more, for the case of binary quadratic programming, a linearization method is introduced. The numerical results indicate that the generalized assignment model and method proposed are valid.

Generalized Assignment Model

The generalized assignment problem presented in this paper can be described as follows: *n* technicians are to be assigned to complete *m* tasks, each technician can undertake no more than one task, and each of the tasks requires at least one technicians to work at the same time.

Obviously, due to the reason that more than one technician must work at the same time for a single task, there should be a lot of collocation cases of personnel. That means, the number of variables that we must take into consideration is very great in quantity. For instance, in the case that a single task requires no more than 2 technicians to work at the same time. *s* and *t* respectively represent the numbers of tasks to be accomplished by one and two technicians, where *s*+*t*=*m*. Then the number of collocation cases of personnel for all the m tasks will be $(s \times n + t \times C_n^2)$, which will be of large value even if *m* and *n* are small. Thus, it's of great significance to present a brief and suitable optimization model for the proposed GAP.

We noticed that in a real-life situation, since the technicians are likely to be different in specialty, each technician is usually able to complete only part of but not all the tasks in cooperation with others. Then the GAP's model can be simplified. With this in mind, a new type of generalized assignment model will be presented in this paper. For the sake of the expedience of model description, several notations will be introduced as follows:

m and *n* are respectively the numbers of tasks and technicians to be arranged. *k* is the sequence number of the task. L_k denotes the number of technicians required to accomplish the *k*-th task. $\mathbf{I}_{k,L_k}$ denotes the set of technicians that are able to complete the *k*-th task in cooperation with other $(L_k - 1)$ technicians. N_k represents the number of technicians that are able to participate in the *k*-th task, which is the number of elements in set $\mathbf{I}_{k,L_k}$. $c_{i_1 i_2 \ldots i_{L_k},k}$ represents the time cost by the *k*-th task if it is completed by technicians chosen from the set $\mathbf{I}_{k,L_k}$ that are numbered as $i_1, i_2, ..., i_{L_k}$. For the *k*-th task, we can choose these technicians in $C_{N_k}^{L_k}$ different ways. x_{ik} is 1 if the *i*-th technician is assigned to participate in the *k*-th task, 0 otherwise. Apparently, x_{ik}=0 if $i \notin \mathbf{I}_{k,L_k}$.

As one technician is allowed to be involved in no more than one task, a constraint is formulated as

$$\sum_{k=1}^{m} x_{ik} \le 1, i = 1, 2, ..., n \tag{1}$$

Because the *k*-th task must be accomplished by L_k technicians, there is another constraint

$$\sum_{i \in \mathbf{I}_{k,L_k}} x_{ik} = L_k, k = 1, 2, ..., m \tag{2}$$

If the technicians chosen from the set $\mathbf{I}_{k,L_k}$ whose numbers are $i_1, i_2, ..., i_{L_k}$ are assigned to achieve the *k*-th task, the time consumption of it can be written as

$$\sum_{k=1}^{m} \sum_{i_1, i_2, ..., i_{L_k} \in \mathbf{I}_{k,L_k}} c_{i_1 i_2 ... i_{L_k}, k} x_{i_1 k} x_{i_2 k} ... x_{i_{L_k} k} \tag{3}$$

Then the total time consumption of all tasks will be

$$\sum_{k=1}^{m} \sum_{i_1, i_2, ..., i_{L_k} \in \mathbf{I}_{k,L_k}} c_{i_1 i_2 ... i_{L_k}, k} x_{i_1 k} x_{i_2 k} ... x_{i_{L_k} k} \tag{4}$$

The optimization objective of GAP is to complete all the tasks in the shortest time. Above all, its integer optimization model in a minimization sense can be formulated as

$$\left.\begin{array}{ll} \text{minimize} & \displaystyle\sum_{k=1}^{m} \sum_{i_1, i_2, ..., i_{L_k} \in \mathbf{I}_{k,L_k}} c_{i_1 i_2 ... i_{L_k}, k} x_{i_1 k} x_{i_2 k} ... x_{i_{L_k} k} \\ \text{subject to} & \displaystyle\sum_{k=1}^{m} x_{ik} \le 1 \quad i = 1, 2, ..., n \\ & \displaystyle\sum_{i \in \mathbf{I}_{k,L_k}} x_{ik} = L_k \quad k = 1, 2, ..., m \\ & x_{ik}, x_{i_1 k}, x_{i_2 k}, ..., x_{i_{L_k} k} \in \{0,1\}, i_1 \ne i_2 \ne ... \ne i_{L_k} \end{array}\right\} \tag{5}$$

The above model gives a unified description of the generalized assignment problem that one or more persons are required to work at the same time to complete a single task. If $\max\{L_1, L_2, ..., L_m\}$ =1, it appears to be the balanced assignment model. If $\max\{L_1, L_2, ..., L_m\}$ =2, it turns into a binary quadratic integer programming [7] [8] model. If $\max\{L_1, L_2, ..., L_m\}$ =*k*, the GAP model becomes a *k*-order integer programming.

As we know, the nonlinear integer programming [9] [10] is NP-hard, to optimally solve this problem within a limited time may not be possible for large-sized problems.

A general branch and bound algorithm [11] or heuristic algorithm [12] [13] is usually used to obtain the approximate optimal solution.

Note the following facts:

(1) The case that $\max\{L_1, L_2, ..., L_m\}$ =2 exists extensively in our real lives.

(2) If $\max\{L_1, L_2, ..., L_m\}$ > 2, the time consumption of a single task might be *k*-dimensioned, where *k* > 2. In this case, it will be hard to represent the constant $c_{i_1 i_2 ... i_{L_k}, k}$ in two-dimension plane accurately.

(3)Since the binary quadratic integer programming is too hard to be solved directly, it's of great significance to find an applicable method to handle it.

Taken these facts into consideration, in this paper, we will focus on finding a proper solution of the GAP proposed on the situation where $\max\{L_1, L_2, ..., L_m\}$ =2. In this case, the optimization model of the proposed GAP can be formulated as:

$$\left.\begin{array}{ll} \text{minimize} & \sum_{k=1}^{s} \sum_{i \in \mathbf{I}_{k,1}} c_{ik} x_{ik} + \sum_{k=s+1}^{m} \sum_{j_1, j_2 \in \mathbf{I}_{k,2}} c_{j_1 j_2, k} x_{j_1 k} x_{j_2 k} \\ \text{subject to} & \sum_{k=1}^{m} x_{ik} \le 1 \quad i = 1, 2, ..., n \\ & \sum_{i \in \mathbf{I}_{k,1}} x_{ik} = 1 \quad 1 \le k \le s \\ & \sum_{i \in \mathbf{I}_{k,2}} x_{ik} = 2 \quad s+1 \le k \le m \\ & x_{ik}, x_{j_1 k}, x_{j_2 k} \in \{0,1\}, \ j_1 \ne j_2 \end{array}\right\} \quad (6)$$

Solving Method of The Model

As we know, the quadratic integer programming is too hard to be directly solved with a limited time, especially when it is of large scale. Nevertheless, for the linear integer programming(LIP), we can either use some well-known algorithms such as branch-and-bound [14], cutting plane [15] or some mathematical softwares based on these algorithms to solve it precisely. Hence the linearization method will be a good way to solve the model.

Here we will import a new binary variable $y_{j_1 j_2, k}$, defined as $y_{j_1 j_2, k} = x_{j_1 k} x_{j_2 k}$, then the quadratic integer programming can be remodeled as the following linear integer programming

$$\left.\begin{array}{ll}\text{minimize} & \sum_{k=1}^{s}\sum_{i\in\mathbf{I}_{k,1}} c_{ik}x_{ik} + \sum_{k=s+1}^{m}\sum_{j_1,j_2\in\mathbf{I}_{k,2}} c_{j_1j_2,k}y_{j_1j_2,k} \\ \text{subject to} & \sum_{k=1}^{m} x_{ik}\le 1 \quad i=1,2,...,n \\ & \sum_{i\in\mathbf{I}_{k,1}} x_{ik}=1 \quad 1\le k\le s \\ & \sum_{i\in\mathbf{I}_{k,2}} x_{ik}=2 \quad s+1\le k\le m \\ & y_{j_1j_2,k}\ge x_{j_1k}+x_{j_2k}-1 \\ & y_{j_1j_2,k}\le x_{j_1k} \\ & y_{j_1j_2,k}\le x_{j_2k} \\ & x_{ik}, x_{j_1k}, x_{j_2k}, y_{j_1j_2,k}\in\{0,1\},\ j_1\ne j_2 \end{array}\right\} \quad (7)$$

Now we will prove that this LIP model is equivalent to the proposed binary quadratic integer programming model.

To prove the equivalence of these two models, we just need to prove that the equation " $y_{j_1j_2,k}=x_{j_1k}x_{j_2k}$ " is equivalent to the constraints "$y_{j_1j_2,k}\ge x_{j_1k}+x_{j_2k}-1, y_{j_1j_2,k}\le x_{j_1k}, y_{j_1j_2,k}\le x_{j_2k}, y_{j_1j_2,k}=0\ or\ 1; j_1\ne j_2$".

We first prove the sufficiency: since both x_{j_1k} and x_{j_2k} are binary, we can easily get that $(x_{j_1k}-1)(x_{j_2k}-1)\ge 0$, unfold this inequality we can get the result that $y_{j_1j_2,k}=x_{j_1k}x_{j_2k}\ge x_{j_1k}+x_{j_2k}-1$. Due to the fact that $x_{j_1k}\le 1, x_{j_2k}\le 1$, the following inequalities $y_{j_1j_2,k}\le x_{j_1k}, y_{j_1j_2,k}\le x_{j_2k}$ are knowable.

Then, the necessity will be proved: the necessity means that the relationship between $y_{j_1j_2,k}$ and x_{j_1k}, x_{j_2k} in constrains "$y_{j_1j_2,k}\ge x_{j_1k}+x_{j_2k}-1$, $y_{j_1j_2,k}\le x_{j_1k}$, $y_{j_1j_2,k}\le x_{j_2k}$, $y_{j_1j_2,k}=0\ or\ 1; j_1\ne j_2$ " are the same as that in the equality $y_{j_1j_2,k}=x_{j_1k}x_{j_2k}$. Here we will define a new two-dimensioned vector as (x_{j_1k}, x_{j_2k}). Since both x_{j_1k} and x_{j_2k} can only be valued at 0 or 1, there are 4 possible values for (x_{j_1k}, x_{j_2k}), they are (0,0),(0,1),(1,0),(1,1).

If (x_{j_1k}, x_{j_2k}) = (0,0), we can get $y_{j_1j_2,k}=0$ from the constraints "$y_{j_1j_2,k}\le x_{j_1k}, y_{j_1j_2,k}\le x_{j_2k}$ " and " $y_{j_1j_2,k}=0\ or\ 1$ ".

If (x_{j_1k}, x_{j_2k}) = (0,1) or (1,0), we can get $y_{j_1j_2,k}=0$ from the constraints "$y_{j_1j_2,k}\le x_{j_1k}, y_{j_1j_2,k}=0\ or\ 1$ " or " $y_{j_1j_2,k}\le x_{j_2k}$, $y_{j_1j_2,k}=0\ or\ 1$ ".

If (x_{j_1k}, x_{j_2k}) = (1,1), we can get $y_{j_1j_2,k}=1$ from the constraints "$y_{j_1j_2,k}\ge x_{j_1k}+x_{j_2k}-1$ " and "$y_{j_1j_2,k}=0\ or\ 1$ ".

Obviously, the relationship between $y_{j_1j_2,k}$ and x_{j_1k}, x_{j_2k} in these two different ways are the same.Q.E.D.

The transformation increases the number of constraints, that is, the scale of the problem. But for the existing software to solve large data, the expansion of the scale of the problem has little impact on the solution. Thus, the cost is acceptable.

Numerical Experiment

For the above linear integer programming model, there are some good mathematical softwares can be used to solve it, such as LINGO, LINDO, CPLEX, MATLAB, in which LINGO is the mainstream one of the domestic. LINGO is such a mathematical software based on the branch-and-bound algorithm, characterized by directly using the LINGO language to describe the model itself. The prominent advantages of LINGO are rapidity and accuracy, what's more, it can meet the requirements of the integer variable. The latest version of LINGO is able to precisely solve the LIP which involves hundreds of integral variables. The numerical experiment in this paper is run on a notebook equipped with CPU of Intel (R) Core TM 4710 MQ 2.50G HZ and 4.00GB memory. The solver is LINGO of version 11.1.

Since there is no actual data to be referred to, the values of the constants in the LIP model are assigned randomly in this paper, where $N_k \in [6,15]$, $c_{ik} \in [5,15](1 \le k \le s)$, $c_{j_1,j_2,k} \in [15,30](s+1 \le k \le m)$. The value of the scale of numerical instance($m \times n$) is set from (5×10) to (10×15), and the result is shown as TABLE 1.

Where "Original variable number" represents the number of variables we should take into account in the original GAP, while "Actual variable number" denotes the number of variables we need to care about in model (6).

Table 1. numerical results

$m\times n$	s	t	Original variable number	Actual variable number	Optimum solution	Computing time
5×10	4	1	85	60	46	1.41
	3	2	120	80	49	0.83
8×13	7	1	169	115	55	1.56
	6	2	234	150	67	1.45
	5	3	299	195	77	1.03
10×15	9	1	240	135	70	1.66
	8	2	330	170	79	1.63
	7	3	420	215	90	1.55

As shown in TABLE 1, we can draw some conclusions:

1. The dimension of the actual calculation is much smaller than the number of variables in the original generalized assignment problem. This indicates that the new model proposed in this paper is very effective to optimize the generalized assignment problem.

2. The computing time keeps very short as the scale of LIP grows. Owing to the simple structure of the linear integer programming model. On the other hand, it thanks to the fact that LINGO software is very powerful in solving integer programming.

As we can see, in the actual assignment problem, the optimal allocation scheme can be quickly obtained after the optimization model is established, which meets the requirements of the actual production management. So we can come to the conclusion that the generalized assignment model proposed in this paper and its solution method are effective in solving the GAP that more than one persons are required to work at the same time to complete a single task.

Conclusions

Generalized assignment problem is some promotion of the classical assignment problem. However, the existing research on the generalized assignment problem does not take into account a practical situation which often occurs in actual personnel assignment problems. That is the case in which more than one persons are required to work at the same time to complete a single task. To solve this kind of personnel assignment problem, a new generalized assignment model based on nonlinear integer programming is proposed in this paper. As a result of comprehensive consideration of some kinds of actual conditions, the main research of this paper focuses on the case where a single task demands at most two persons to be engaged in at the same time. The optimization model built from it is established for the multi-constrained binary quadratic programming, then a linearization method is introduced to solve it. The results of numerical experiments indicate that the generalized assignment model and its solution method proposed in this paper can meet the requirements of actual production management. Therefore, the GAP in cases where a single task demands more than two persons being engaged in at the same time exists extensively in our real lives as well. In that case, the optimization model will be more complex with multiple integer programming. Thus, future research will focus on finding the effective way to solve this kind of generalized assignment problem.

References

[1] CHANG T M,HAN Z G. Solution to a Class Optimization Problem By Utilizing the Hungary Calculate Way [J]. Journal of Information Engineering University, 2004, 5(01): 60-62.(in Chinese)

[2] Avella P, Boccia M and Vasilyev I. A Branch-and-Cut Algorithm for the Multilevel Generalized Assignment Problem [J]. Access IEEE, 2013, 1:475-479.

[3] FU X W, GUO Q and MA Q Q. A Multiple Objective Uncertainty Assignment Problem and the Study of Its Algorithm [J]. Operations Research and Management Science, 2013,22(06):34-38.(in Chinese)

[4] WANG L Z,LIU Y. Reverse Point Algorithm of Assignment Problem on Assignment Less Than Jobs and Persons [J]. Operations Research Transactions, 2011,15(03):124-128.(in Chinese)

[5] LI H, ZHANG L and JIAO Y C. Discrete differential evolution algorithm for integer linear bilevel programming problems[J]. Journal of Systems Engineering and Electronics, 2016, 27(4):912-919.

[6] YU Y Z,ZHANG Q. An Efficient Method for a Class of Generalized Assignment Problem[C]//Proceedings of the Seventh Academic Exchange Conference of the Operations Research Society of China. 2004:859-865.

[7] Gu S, Cui R. Polynomial time solvable algorithm to linearly constrained binary quadratic programming problems with Q being a five-diagonal matrix[C]// Fifth International Conference on Intelligent Control and Information Processing. IEEE press, 2014:366 - 372.

[8] Wang P, Shen C, Hengel A V D. Large-scale Binary Quadratic Optimization Using Semidefinite Relaxation and Applications [J]. Eprint Arxiv, 2014, 53(46):6302-6305.

[9] Chiang H D, Wang T. A Novel TRUST-TECH Guided Branch-and-Bound Method for Nonlinear Integer Programming [J]. IEEE Transactions on Systems Man and Cybernetics Systems, 2015, 45:1361 - 1372.

[10] Wang X, Wu W, Zhu D. An Approximation Algorithm for Nonlinear 0-1 Integer Programming Problems[M]. 2011.

[11] CHEN Z P,XI F. A New Branch-and-brand Algorithm for Solving Large Comlex Integer Convex Quadratic Programs[J]. Mathematica Numerica Sinica, 2004,26(04):445-458.(in Chinese)

[12] Zou G, Jia L, Zou J. Random-weight based genetic algorithm for multi-objective bilevel mixed linear integer programming[C]// Eighth International Conference on Natural Computation, May 29-31,2012, Chongqing. IEEE press, 2012:693-697.

[13] Chen H, Wang S, Wang H. Particle Swarm Optimization Based on Genetic Operators for Nonlinear Integer Programming[C]// International Conference on Intelligent Human-Machine Systems and Cybernetics. IEEE Computer Society, 2009:431-433.

[14] CHAN D Y, KU C Y, LI M C. A method to improve integer linear programming problem with branch-and-bound procedure[J]. Applied Mathematics and Computation, 2006, 179(2):484-493

[15] CHEN K D, WU Z, XIA Z J.A New Cutting Plane Algorithm for Integer Linear Programming[C]// International Conference on Computer Science and Service System, August 11-13, Nanjing. IEEE press, 2012:1591-1594.