New generalized assignment problem with identified first-use bins

Ahmad H. Shraideh[a], **Hervé G. Camus**[a] **and Pascal G.M. Yim**[b]

[a]*LAGIS Ecole Centrale de Lille, Avenue Paul Langevin, BP 48, 59651 Villeneuve d'Ascq,*
{ahmad.shraideh,herve.camus}@ec-lille.fr
[b]*3 Suisses International Group, Cité Numérique, 245 rue Jean Jaurès, 59650 Villeneuve d'Ascq,*
pyim@3suisses.fr

Keywords: Combinatorial optimization, Generalized assignment problem (GAP), Bin packing problem (BPP), Integer Programming.

Abstract. In this paper, we present a new multi-criteria assignment problem that groups characteristics from the well known Bin Packing Problem (BPP) and Generalized Assignment Problem (GAP). Similarities and differences between these problems are discussed, and a new variant of BPP is presented. The new variant will be called generalized assignment problem with identified first-use bins (GAPIFB). An algorithm based on the resolution method used for GAP problem and on GAPIFB is proposed to solve the new assignment problem.

1 Introduction

Within the context of the French competitive cluster[1] "Industrie du commerce", with the collaboration of COFIDIS[2] and ALFEA consulting[3]. The project GOCD[4] aims to set up a new dematerialized workflow system, to treat the received contracts at COFIDIS. Our participation was to install a new optimization and decision-making tool for the new system with the necessary key performance indicators . Every day, COFIDIS receives from the post office thousands of contracts and credit demands of different types (for facility we will use the terms contracts for contracts and credit demands). The quantities and the types of contracts are known in the morning and can change from one day to another. We suppose here that these contracts must be treated at the same day. The treatment time for a contract by a collaborator is defined by a matrix of competence, as each collaborator has different skills and experiences with respect to contract type. The contracts are distributed currently to company collaborators according to past acquired experience in heuristic method. This distribution is not optimal, but hoped to be approximated to the optimal one. The daily work hours for the collaborators are not equal, in reason of human resources management considerations. If the capacity of the collaborators is overloaded, the company comes to the aid of temporary workers to treat the overloaded contracts. The objective is to find the best distribution of contracts that will best exploit collaborators capacities i.g. minimize the total time of contracts treatment, and to determine the exact number of temporary workers if needed. This problem can be seen as a multi-criteria optimization problem with discrete variables.

In examining the actual used heuristic, we can identify three problems. The first one concerns the assignment method used to distribute contracts. In this approach , contracts are distributed depending on previous experience, and not on approved optimal method. Bad distribution of

[1]A competitive cluster is an initiative that brings together companies, research centers and educational institutions in order to develop synergies and cooperative efforts. http://www.industrie.gouv.fr/poles-competitivite
[2]French consumer credit company. http://www.cofidis.com
[3]French information system consulting company. http://www.alfea-consulting.com
[4]GOCD : French acronym for Management and optimization of document life cycle

contracts could lead to unnecessary call of temporary workers, this can be seen clearly when the current load of contracts is close to company optimal capacity. The second problem is to evaluate the total treatment time of daily received contracts since it depends on collaborators competence availability and the quantity of each type. Different distributions of contracts could result in different total time of treatment. The third problem is to determine the exact number of temporary workers needed when an overload exist. To the best of our knowledge, there is no model capable to represent completely these problems; rather we find models with partial solution for partial problem.

The paper is organized as follow. In the next section, a formulation of the problem is presented. In section three, we study two famous assignment problems, the BPP problem and GAP problem. The different variants of each problem are discussed and their weaknesses regarding our problem are clarified. We will demonstrate that neither of these problems can solitary gives a complete answer to the mentioned problem. In section four, we present our approach to solve the problem, followed by mathematical formulation and evaluation results. We terminate by our conclusions and future work.

2 Problem formulation

The notations used to formulate the problem are

- *NC* : Number of all tasks (contracts),
- *N* : Number of primary agents (company workers),
- *M* : Number of available secondary agents (temporary workers),
- *CAP*: Primary, secondary agents capacities (in hours), $CAP = \{CAP_1, CAP_2.., CAP_{N+M}\}$,
- T_{ij}: Needed time for primary, secondary agent *i* to treat task *j*,
- U_i: Boolean. 1 if primary or secondary agent *i* is used, otherwise 0, $U = \{U_1, U_2, .., U_{N+M}\}$,
- X_{ij}: Boolean. 1 if task *j* is assigned to primary or secondary agent *i*, otherwise 0.

The objective functions are

$$Min \sum_{i=1}^{N+M} U_i \tag{1}$$

$$Min \sum_{i=1}^{N+M} U_i \sum_{j=1}^{NC} X_{ij} \times T_{ij} \tag{2}$$

Subject to,

$$\sum_{i=1}^{N+M} X_{ij} = 1, \qquad \forall j \in \{1, 2, .., NC\} \tag{3}$$

$$\sum_{j=1}^{NC} X_{ij} \times T_{ij} \leq CAP_i \times U_i, \qquad \forall i \in \{1, 2, .., N+M\} \tag{4}$$

$$U_i = 1, \qquad \forall i \in \{1, 2, .., N\} \tag{5}$$

The objective function (1) searches to minimize the number of secondary agents used to treat all tasks. Objectives function (2) minimize the total treatment time. Constraint (3) indicates that all tasks must be distributed, and each task is given only to one agent. Constraint (4) explains that the capacity of each used agent must not be violated. Finally constraint (5) is used to be sure that all primary collaborators are used.

3 Bin packing and Generalized assignment problems

In literature, we find some assignment problems which are similar to our problem. These problems were widely studied and analyzed. The closest ones to our problem are Bin Packing Problem and Generalized Assignment problem.

Bin packing problem(BBP) is well known for being one of the combinatorial NP-hard problems (M Garey, 1979). Many researches were realized to find the optimal or an approximated solution for this problem (Brown, 1979); (Csirik and Woeginger, 1998); (Epsteina and van Steeb, 2007). In its simplest form, we have a set of bins of equal capacity and a list of objects, each object has an equivalent weight (costs of treatment) for all bins. The objective is to find the minimum number of bins to pack all the objects exist in the objects set. A bin packing problem can be either on-line or off-line. In on-line packing problem, we have information only about the current object in the list to be packed, and no objects can be repacked later. In off-line packing problem complete information about all objects are known in advance. Variants of BPP include, two dimensional bin packing (Chung et al., 1982) (Berkey and Wang, 1987); (Lodi et al., 2002); (Puchinger and Raidla, 2007) and three dimensional bin packing problem (Martello et al., 2000); (Miyazawaa and Wakabayashib, 2007), in which each object have either two dimension (area) or even three dimension (volume). Another variant and well studied BPP is the extendable bin packing problem (Dell'Olmo et al., 1998); (Coffman and Lueker, 2006) , where the sizes of the bins are extendable when necessary to answer work needs.

Another famous problem is the generalized assignment problem (GAP), a generalization of Multi-knapsack problem (Silvano and Martello, 1990). In GAP problem, a set of objects with cost and profit, are assigned to a set of agents. Each object can be allocated to any but only one agent, and the treatment of an object needs resources which change, depending on the object and the agent treating it, each agent can have different capacity. The objective is to maximize the profit without exceeding agent's capacities. A survey on the algorithms used to solve this problem can be found in (Cattrysse and Wassenhove, 1992).

It is clear that the two models have different objectives and different formulation. In the BPP problem, we search to minimize the number of used bins to pack all the objects without any consideration to profit. Whereas in the GAP problem, profit is considered but the allocation of all objects is not important, which means the possibility to have an optimal solution without distributing all objects. More over, in BPP the objects have an equal value whatever was the bin used to pack them, which is not the case in the GAP problem, where the profit of an object depends on the object and on the agent. From the previous description for BBP and GAP problem, we see that our assignment problem corresponds to GAP in searching optimal treatment time for company workers. But unfortunately, it is unable to determine the minimum number of temporary workers needed in overloaded cases, as the number of bins (workers in our problem) must be predefined in advance.

In the other hand, BPP model can find the minimum number of workers to treat all the contracts. Still neither classical form nor its variants are capable to distinct between company collaborators and temporary workers in there solution. This can lead to solutions which exclude some company workers if the utilization of temporary workers gives better solutions. This is an important matter, as we are forced to use company workers at first, and to include them in every proposed distribution before calling a minimum number of temporary workers to treat the overload.

4 Proposed Approach

To solve this multi-criteria problem, we propose to decompose it into two mono-criteria problems. For each mono-criteria problem, an exact method is used to find the optimal solution. We can imagine this solution in two stages. In the first stage, we use GAPIFB model to search the minimum number of secondary agents needed to treat all contracts. This is a major concern to enterprise managers as it is considered the most enterprise financial resources consumers. At the end of this stage, we are sure about company situation (under loaded or overloaded), and the minimum number needed of secondary agents is exactly known. Still, we do not know what tasks distribution to use in order to realize minimum time of treatment, so we pass to the second stage. In this stage, we add the computed secondary agents in the first stage to the primary agents, and classical GAP representation is used to find the optimal solution. In our problem, we choose to minimize the total time of treatment; still decision makers can choose others objectives. Later, we detail the mathematical formulation for the two stages.

4.1 Mathematical formulation

For the first stage, a binary programming formulation for GAPIFB is used. In GAPIFB, a set of tasks(contracts) must be assigned to a set of agents(bins). The size of each task can vary from one agent to another, agent's capacities are not equal, and the set of agents includes two types of agents, primary agents and secondary agents. The use of secondary agents is allowed only when the primary agents are not capable to treat all the tasks. The objective function is to minimize the number of secondary agents used with primary agents to treat the whole quantity of received tasks. In this formulation we used binary variable to represent both primary agents and secondary agents U_i. Another set of binary variables X_{ij} is used to indicate if task *j* is attributed to primary or secondary agent *i*. Notice in this stage, we define (N+M) variable for each task, where *N* represent the number primary agents and *M* is the number of available secondary agents. The formulation to the first stages objective function and constraints is given as follows:

$$Min \sum_{i=1}^{N+M} U_i \tag{6}$$

Subject to

$$\sum_{j=1}^{NC} X_{i,j} \times T_{i,j} \leq CAP_i \times U_i, \qquad \forall i \in \{1, 2, .., N + M\} \tag{7}$$

$$\sum_{i=1}^{N+M} X_{ij} = 1, \qquad \forall j \in \{1, 2, .., NC\} \tag{8}$$

$$\sum_{i=1}^{N} U_i = N, \tag{9}$$

Constraint (7) ensures that the capacity of agents is not violated. Constraint (8) ensures that all tasks are allocated and each task is assigned to only one agent. To ensure the use of all primary agents, we added constraint (9). This constraint forces the solver to search solution where U_i equal 1, $\forall$ i $\in${1,2,...,N}, this set contains only primary agents and *N* is the number of company primary agents.

The second stage is formulated as GAP problem. The result of objective (1) is used to indicate the new number of agents *L*. Consider the following

- L : Minimum number of primary and secondary agents needed to treat all the tasks,
- Z : Number of tasks types,
- Y_{ik}: Number of task of type *j* to be assigned to agent *i*,
- QT_k : Quantity of tasks of type *k* , $QT_k \in \{QT_1, QT_2, ..., QT_Z\}$,
- TT_{ik}: Time per task, needed time for primary, secondary agent *i* to treat a task of type *k*,
- $i \in \{1, 2, ..., L\}$ and $k \in \{1, 2, ..., Z\}$.

The objective is to minimize the total treatment time for all agents and is formulated as follow

$$Min \sum_{i=1}^{L} \sum_{k=1}^{Z} Y_{ik} * TT_{ik} \tag{10}$$

$$Subject\ to \quad \sum_{j=1}^{Z} Y_{ik} \times TT_{ik} \leq CAP_i, \qquad \forall i \in \{1, 2, .., L\} \tag{11}$$

$$\sum_{i=1}^{L} Y_{ik} = QT_k, \qquad \forall k \in \{1, 2, .., Z\} \tag{12}$$

Constraint (11) ensures agent capacity to not be violated. Constraint (12) ensure that the distributed quantity of tasks type *j* is less than the received quantity of that type.

4.2 Evaluation and test results

In order to evaluate our approach, a formulation of the problem as mixed integer program was realized[5]. The optimal solution in the two stages was computed using Cplex9[6] solver. Many samples of 100 primary agents, 20 secondary agents, and 5000 tasks, were generated randomly with different competence matrix and different capacities for company collaborators. The quantities and types of each tasks was generated to be near to the average of company capacity. It was found that the proposed approach is capable to detect under loaded situation and to find the optimal solution to distribute all tasks. In overloaded situation, the proposed GAPIFB model was able to find the minimum number of secondary agents needed to treat all the tasks, all company agents were included in every produced solution. The capacities were shown to be optimal for both agents kinds primary and secondary. Within these conditions, the execution time was ordered in seconds which is satisfactory for decision makers. When the number of contracts is increased respectively with the number of temporary workers, the time of execution is increased also. The only concern in the GAPIFB model was memory consumption, as we are obligated to define for each task (N+M) variables, and for each problem sample, we need to define (NC*(N+M))+(N+M) variables. We believe this problem is not crucial as memories become more and more cheap with larger capacities.

5 Conclusion

In this paper, we presented a new multi-criteria assignment problem and proposed a new exact approach to solve it. The problem consists of allocating a set of different type of tasks to a set of primary agents; in case of overload we use secondary agents to treat all tasks. Each

[5]Tests were held on Intel Dual Core T7200 2.00GHz machine, 2Go of RAM

[6]CPLEX : an optimization software package produced by ILOG.http://www.ilog.com/

agent has different capacity and different experience per task type according to matrix of competence. The treatment time of a task, as a result, will depend on the type of the task and the agent treating it. The objective is to determine the exact number of secondary agent if needed and to distribute the tasks in the best way. To solve this problem we divided it into two parts with mono-objective function for each. Exact methods were used to solve each part. In the first part, we used the proposed GAPIFB. The primary agents in this model appear imperatively in the optimal solution. In using simple form of BPP, the solver will search the optimal solution whatever was the agent to use, which make it possible to exclude some primary agents. In addition, simple BPP defines fixed treatment cost by task type, which is not the case in GAPIFB. In the second stage, a GAP model is used to determine the best distribution to tasks that minimizes total treatment time. Others objectives can be used, this choice is left to company decision makers. A formulation with binary variable was used to implement GABIFB. In this formulation, both primary and secondary agents are represented as binary variables. Many samples were generated and tested, and our approach proved to be capable to define the minimum number of needed secondary agent. The only worry in our approach was memory consumption in the first part. We believe this is not important problem especially with the enormous development in memories sizes and prices. Future work can be conceived to extend our work in order to treat and take in consideration contracts flow for long period e.g. one week flow. Other objective functions to distribute the contracts can be created in order to construct and supply new Key Performance Indicators (KPI) for the decision makers

References

Berkey, J. O. and Wang, P. Y.: 1987, Two-dimensional finite bin-packing algorithms, *The Journal of the Operational Research Society* **38**, 423–429.

Brown, D. J.: 1979, A lower bound for on-line one-dimensional bin packing algorithms, *Technical report*, Illinois Univ at Urbana-Champaign applied computation theory group.

Cattrysse, D. G. and Wassenhove, L. N. V.: 1992, A survey of algorithms for the generalized assignment problem, *European Journal of Operational Research* **60**, 260–272.

Chung, F. R. K., Garey, M. R. and Johnson, D. S.: 1982, On packing two-dimensional bins, *SIAM Journal on Matrix Analysis and Applications* **3**, 66–76.

Coffman, E. and Lueker, G. S.: 2006, Approximation algorithms for extensible bin packing, *Journal of Scheduling* **9**, 63–69.

Csirik, J. and Woeginger, G. J.: 1998, On-line packing and covering problems, *Lecture Notes in Computer Science* **1442**, 147–177.

Dell'Olmo, P., Kellerer, H., Speranzac, M. G. and Tuza, Z.: 1998, A 13/12 approximation algorithm for bin packing with extendable bins, *Information Processing Letters* **65**, 229–233.

Epsteina, L. and van Steeb, R.: 2007, Online bin packing with resource augmentation, *Discrete Optimization* **4**, 322–333.

Lodi, A., Martello, S. and Monaci, M.: 2002, Two-dimensional packing problems: A survey, *European Journal of Operational Research* **141**, 241–252.

M Garey, D. J.: 1979, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman, New York.

Martello, Pisinger, S. and D. Vigo, D.: 2000, The three-dimensional bin packing problem, *OPERATIONS RESEARCH BALTIMORE* **48**, 256–267.

Miyazawaa, F. and Wakabayashib, Y.: 2007, Two- and three-dimensional parametric packing, *Computers & Operations Research* **34**, 2589–2603.

Puchinger, J. and Raidla, G. R.: 2007, Models and algorithms for three-stage two-dimensional

bin packing, *European Journal of Operational Research* **183**, 1304–1327.
Silvano and Martello, P. T.: 1990, *Knapsack problem algorithms and computer implementation*.