Proceedings of the International Multiconference on
Computer Science and Information Technology pp. 921–925
ISBN 978-83-60810-14-9
ISSN 1896-7094

# Two stages optimization problem: New variant of Bin Packing Problem for decision making

Ahmad Shraideh
LAGIS Ecole Centrale de Lille
Avenue Paul Langevin
59651 Villeneuve d'Ascq
France
Email: ahmad.shraideh@ec-lille.fr

Hervé Camus
LAGIS Ecole Centrale de Lille
Avenue Paul Langevin
59651 Villeneuve d'Ascq
France
Email: herve.camus@ec-lille.fr

Pascal Yim
3Suisses International Group
Cité Numérique,245 rue Jean Jaurès
59650 Villeneuve d'Ascq
France
Email:pyim@3suisses.fr

***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). The GAPIFB will be used to supply decision makers with quantitative and qualitative indicators in order to optimize a business process. An algorithm based on the GAP problem model and on GAPIFB is proposed.**

## I. 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. Some contracts must be treated at the same day others can wait for some days. 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 decision makers can either come to the aid of temporary workers to treat the overloaded contracts or they can holdup the treatment of unimportant type of contracts. The objective is to find the best distribution of contracts that will best exploit collaborators capacities, and to determine if the daily load of contract exceeds current collaborators capacities. In this case decision makers need to know the exact number of temporary workers required to treat all the contracts. 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 method, contracts are distributed depending on previous experience, and not on approved optimal method. Bad distribution of contracts could lead to unnecessary call of temporary workers. Different distributions of contracts could result in different total time of treatment, this can be seen clearly when the current load of contracts is close to company optimal capacity. The second problem is to detect in advance overloaded situations and to decide which contracts to treat if decision makers don't prefer to hire temporary workers. The third problem is to determine the exact number of temporary workers when needed. 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.

## II. Problem formulation

A generic formulation and notation for the problem is as the following

- $NC$ : Number of all tasks (contracts),
- $N$ : Number of primary agents (company workers),

[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

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

The objective functions are

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

$$Min \sum_{j=1}^{NC} X_{ij} \times T_{ij} \tag{2}$$

Subject to,

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

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

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

Notice that for all *i* ∈{1, 2,.., N}, $U_i$ represents a primary agent and for all *i* ∈{N+1, N+2,.., N+M}, $U_i$ represents a secondary agent. The objective function (1) searches to minimize the number of secondary agents used to treat all tasks. We have chosen in objectives function (2) to minimize the total treatment time as example, other objectives can be designed by the decision makers. 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 agents are used. As we can see, our problem is a specific case of this generic case, where the contract treatment time is defined by its type and the agent treating it. For simplification, in our proposed solution, we will reformulate the generic problem and new notations will be used where the contracts of the same type assigned to one agent are presented by one integer variable in stead of a set of binary variables for each contract. In fact, we pass from linear multi-criteria problem with binary integer to linear multi-criteria problem with integer variables and this will reduce the number of variable to use. For example, an instance of the problem with 100 primary agents, 10 available secondary agents, 5 contract type and 3500 contracts, will need 385110 binary variables if we use binary representation (100+10 variable to present agents and 100+10*3500 to present the contracts). Whereas by grouping the contract of the same type assigned to an agent will require 660 variables (100+10 variable to present agents and 100+10*5 to present the assigned contracts)

## III. 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(BPP) is well known for being one of the combinatorial NP-hard problems [1]. Many researches were realized to find the optimal or an approximated solution for this problem [2] [3] [4]. 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 in order to pack all the objects in the list. 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 [5] [6] [7] [8] and three dimensional bin packing problem [9] [10], in which each object have either two dimensions (area) or even three dimensions (volume). Another variant and well studied BPP is the extendable bin packing problem [11] [12] , 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 [13]. 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 [14].

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 problem in that it search to optimize certain predefined objectives by decision makers. But unfortunately, it is unable to determine the minimum number of temporary workers needed to achieve these objectives in overloaded cases, as the number of bins (workers in our problem) in GAP problem must be defined in advance.

On 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 than using company workers. This is an important matter, as we must first verify company workers capability to treat all the contracts and to wait decision maker to decide whether to hire temporary workers or not. Before integrating temporary workers with company workers to find optimal solution.

## IV. Proposed Approach

To solve this multi-criteria problem, we propose to decompose it into two mono-criteria problems. Each mono-criteria problem is presented by a model, and an exact method is used to find the optimal solution for each of these models as the size of our problem is not large. 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. In GAPIFB, a set of tasks (objects) 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 find the minimal number of secondary agents to be used with the primary agent to treat all the received contracts. At the end of this stage, decision maker not only knows company situation if it is overloaded or under loaded but also he knows the exact number *EN* of secondary agents needed to treat all contracts in overload situation. Decision maker must also decide whether the company will hire secondary workers or not, and if the decision was to hire secondary workers, is it to hire the exact number of secondary workers needed to treat all the contracts or to hire certain number *L* of needed secondary workers?

In the second stage, a group of objectives function is available to decision maker which supply him with comprehensive vision of all possible decision scenarios that can be taken, and their effect on company contracts distribution process. The choice of this objective is left to decision makers. One objective can be to give high treatment priority to contracts that can not be delayed or to contracts considered as profitable to the company. Another objective could be to treat important contracts types uniquely by company collaborators as they have the best experience and skills. The fairness of collaborators loads can be significance objective from the social vision of point. To maximize the rate of profitability by collaborators can be an interesting objective from economic vision. For facility in this paper, an objective function which is to minimize the total treatment time of contracts was been chosen. Figure 1 demonstrates decision making process for our approach. Mathematical formulation for GAPIFB and GAP models are discussed in details in the next sections.

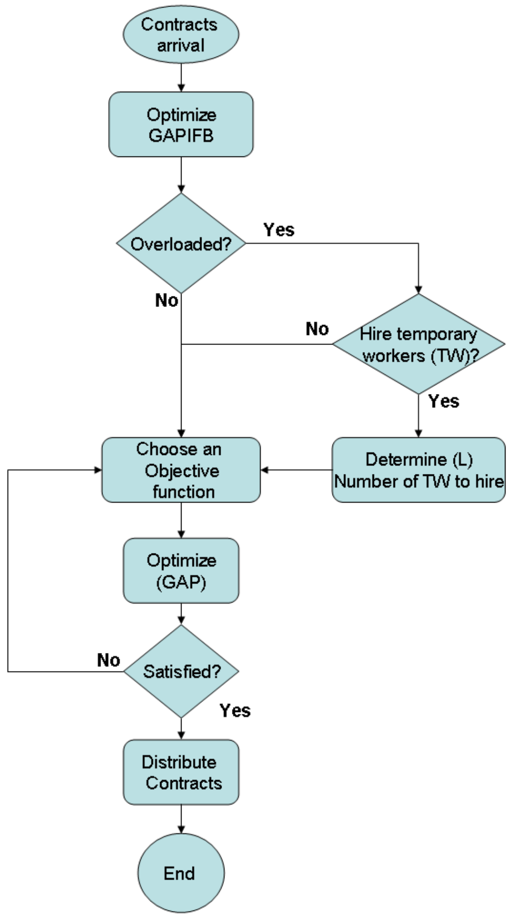


Fig. 1. Proposed decision making process

### A. Mathematical formulation

For the first stage, a linear formulation with integer variables is used to present GAPIFB. 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 $U_i$ to represent both primary agents and secondary agents, where for *i* ∈{1, 2,.., N}, $U_i$ represents a primary agent and for *i* ∈{N+1, N+2,.., N+M}, $U_i$ represents a secondary agent. $X_{ij}$ is used to indicate the number of tasks of type *j* attributed to primary or secondary agent *i*. Notice that in this stage, $X_{ij}$ is an integer variable and not binary variable, and $T_{ij}$ presents the treatment time for contract of type *j* by agent *i*. The new modified notations and a formulation to the first stage objective function with constraints are given as follows:

- *Z*: Number of tasks types,
- $QT_j$: Quantity of tasks of type *j*,

- $T_{ij}$: Needed time for primary or secondary agent *i* to treat a task of type *j*,
- $X_{ij}$: The number of tasks of type *j* assigned to primary or secondary agent *i*.

$$Min \sum_{i=1}^{N+M} U_i \qquad (6)$$

Subject to

$$\sum_{j=1}^{Z} X_{ij} \times T_{i,j} \leq CAP_i \times U_i, \qquad \forall i \in \{1, 2, .., N+M\} \quad (7)$$

$$\sum_{i=1}^{N+M} X_{ij} = QT_j, \qquad \forall j \in \{1, 2, .., Z\} \qquad (8)$$

$$\sum_{i=1}^{N} U_i = N, \qquad (9)$$

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 that the solver will search the optimal solution within the solutions that used all primary agents we added constraint (9), where $U_i$ presents a primary worker ∀ i ∈{1,2,...,N}. Without constraint (9), it is possible that the solver gives us a solution that excludes some primary agent if the use of secondary agents gives better results.

The second stage is formulated as classical GAP problem. Consider the following.

- *L* : Number of secondary agents to be hired (decision maker decision),
- $i \in \{1, 2, ..., L\}$ and $j \in \{1, 2, ..., Z\}$.

As we mention before we consider here only one objective function which is to minimise the total treatment time for all contracts. In using the new notation, the objective function (2) is replaced by the objective function (10) as the following:

$$Min \sum_{i=1}^{N+L} \sum_{j=1}^{Z} X_{ij} * T_{ij} \qquad (10)$$

Subject to

$$\sum_{j=1}^{Z} X_{ij} \times T_{ij} \leq CAP_i, \qquad \forall i \in \{1, 2, .., L\} \qquad (11)$$

$$\sum_{i=1}^{L} X_{ij} = QT_j, \qquad \forall j \in \{1, 2, .., Z\} \qquad (12)$$

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

### B. 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 were computed using Cplex9[6] solver, which uses an advanced mathematical programming and constraint-based optimization techniques. Many samples were generated randomly with different numbers of primary agents and secondary agents and with different competence matrix and different capacities for each sample. 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. Within the size of our problem, the execution time was ordered in milliseconds for bothe the first stage and in the second stage. This execution time was very satisfactory for company decision makers. When the number of contracts increases respectively with the number of agents (primary and secondary), execution time is increased to be in seconds. In fixing the number of primary agent and increasing the number of available secondary agent the execution time increase considerably. Table 1 shows the execution time for some realized samples.

## V. Conclusion

In this paper, we presented a new multi-criteria assignment problem for decision making 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 secondary agents can be used to treat all tasks. Each 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 first objective is to determine company situation (underloaded or overloaded) and to give the exact number of secondary agent in the overloaded cases. The second objective is to give a comprehensive vision of all possible scenarios for allocating tasks to decision maker. To solve this problem we divided it into two parts (stages) with mono-objective function for each. Exact methods were used to solve each stage. In the first part, we used the proposed GAPIFB. This model is able to distinguish between primary agent and secondary. Primary agents imperatively appear in the optimal solution, which is not the case in using simple form of BPP where the solver search the optimal solution whatever was the agent to use. This makes it possible to exclude some primary agents if the use of secondary agent gives best solution. In addition, simple BPP defines fixed treatment cost by task type, which is not the case in GAPIFB.

[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/

TABLE I
SIMULATION RESULT FOR THE FIRST STAGE—GAPIFB OPTIMISATION

| Primary agents | Secondary agents | Number of contracts | Execution time second |
|---|---|---|---|
| 100 | 15 | 3500 | 0.047 |
| 100 | 15 | 3500 | 0.031 |
| 100 | 15 | 3500 | 0.031 |
| 500 | 75 | 17500 | 0.078 |
| 500 | 75 | 17500 | 0.078 |
| 500 | 75 | 17500 | 0.073 |
| 1000 | 150 | 35000 | 0.141 |
| 1000 | 150 | 35000 | 0.125 |
| 1000 | 150 | 35000 | 0.14 |
| 5000 | 750 | 175000 | 1.00 |
| 5000 | 750 | 175000 | 1.031 |
| 5000 | 750 | 175000 | 1.016 |
| 10000 | 1500 | 350000 | 3.172 |
| 10000 | 1500 | 350000 | 3.14 |
| 10000 | 1500 | 350000 | 3.187 |
| 15000 | 2250 | 525000 | 6.375 |
| 15000 | 2250 | 525000 | 6.312 |
| 15000 | 2250 | 525000 | 6.297 |

In the second stage, a GAP model is used with different objective functions to give decision makers ideas about the suitable method to distribute the tasks. Others objectives can be used, this choice is left to company decision makers.

A formulation with integer variables was used instead of binary variable to implement GABIFB, which reduce the total number of variables. 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 time of execution was ordered in milliseconds. Future work can be conceived to extend our work in order to deal with and take in consideration contracts flow for long period e.g. one week flow. Other objective functions to distribute the contracts can be imagined in order to construct and supply new efficient Key Performance Indicators (KPI) for the decision makers.

## REFERENCES


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

[2] D. J. Brown, “A lower bound for on-line one-dimensional bin packing algorithms,” Illinois Univ at Urbana-Champaign applied computation theory group, Tech. Rep., 1979.

[3] J. Csirik and G. J. Woeginger, *On-line packing and covering problems*. Springer Berlin / Heidelberg, 1998, ch. 7, pp. 147 – 177.

[4] L. Epstein and R. van Stee, “Online bin packing with resource augmentation,” *Discrete Optimization*, vol. 4, pp. 322–333, 2007.

[5] F. R. K. Chung, M. R. Garey, and D. S. Johnson, “On packing two-dimensional bins,” *SIAM Journal of Algebraic and Discrete Methods*, vol. 3, pp. 66–76, 1982.

[6] J. O. Berkey and P. Y. Wang, “Two-dimensional finite bin-packing algorithms,” *The Journal of the Operational Research Society*, vol. 38, pp. 423–429, 1987.

[7] A. Lodi, S. Martello, and M. Monaci, “Two-dimensional packing problems: A survey,” *European Journal of Operational Research*, vol. 141, pp. 241–252, 2002.

[8] J. Puchinger and G. R. Raidla, “Models and algorithms for three-stage two-dimensional bin packing,” *European Journal of Operational Research*, vol. 183, pp. 1304–1327, 2007.

[9] S. Martello, D. Pisinger, and D. Vigo, “The three-dimensional bin packing problem,” *OPERATIONS RESEARCH*, vol. 48, pp. 256–267, 2000.

[10] F. Miyazawa and Y. Wakabayashi, “Two- and three-dimensional parametric packing,” *Computers & Operations Research*, vol. 34, pp. 2589–2603, 2007.

[11] P. Dell’Olmo, H. Kellerer, M. G. Speranzac, and Z. Tuza, “A 13/12 approximation algorithm for bin packing with extendable bins,” *Information Processing Letters*, vol. 65, pp. 229–233, 1998.

[12] E. Coffman and G. S. Lueker, “Approximation algorithms for extensible bin packing,” *Journal of Scheduling*, vol. 9, pp. 63–69, 2006.

[13] Silvano and P. T. Martello, *Knapsack problem algorithms and computer implementation*. John Wiley and Sons, 1990.

[14] D. G. Cattrysse and L. N. V. Wassenhove, “A survey of algorithms for the generalized assignment problem,” *European Journal of Operational Research*, vol. 60, pp. 260–272, 1992.