Bin Packing Problem with priorities and incompatibilities using PSO: application in a health care community

Arnaud Laurent * **Nathalie Klement** **

* *Parc scientifique de la Haute Borne, 40 av. Halley 59650 Villeneuve d'Ascq - France (e-mail: arnaud.laurent@inria.fr)*
** *LISPEN, Arts et Métiers, HeSam. École Nationale Supérieure d'Arts et Métiers. 8, boulevard Louis XIV, 59046 Lille, France (e-mail: nathalie.klement@ensam.eu)*

Abstract: The present work deals with the Hospital Group of Territory problem. The objective of the cooperation between these health institutions is to provide a better treatment offer. To do so, these entities pool their means together. Our goal is to propose efficient methods to assign the different operations to the periods and resources, considering resources compatibilities and due dates. We consider this problem as an extension of the classical Bin Packing Problem. We propose a Particle Swarm Optimization to solve this problem using a hybridization proposed by Klement et al. (2017). The results show the interest of the proposed PSO for this kind of problem.

Keywords: Bin Packing, Metaheuristic, Particle Swarm Optimization, Hospital Community.

1. INTRODUCTION

In France, in 2014, Hospital Group of Territory (HGT) has been introduced. It is an evolution of Hospital Community of Territory (HCT), previously defined in 2008. It is a group of distinct places which aim at improving their efficiency by putting together means from different places (Gourgand et al., 2014a). A pool of human resources is shared on several distant hospitals belonging to the same group. The involved problem is to find a hospital assignment for the patients and their operations and, for each operation, to assign the needed resources. The hospitals are distant, so the patients and human resources have to take into account transportation times. The goal is to improve the productivity by pooling human resources and patients within the community. Other applications could be imagined (production sites with shared machines, multi-site time table, ...), rising yet the interest of taking into account the resource transport in a project scheduling context (Laurent et al., 2017).

In this paper, the medical imaging case is considered. Operations are exams. Exams have to be assigned to material resources. Exams can be either an X-ray, a scanner or a MRI. They may be done on a specific material resource: an X-ray can be done by any material resources, but an MRI can only be done by a material resource that can perform an MRI. Incompatibilities between exams and material resources are defined. Each exam has to be assigned to a period. Moreover, each exam has a due date: the period before which the exam should be done. For example, a surgery may be planned on that specific period and the surgeon needs the results of the exam to perform the surgery. This paper intends to solve a planning and assignment problem.

To do so, this problem has been identified as a Bin Packing Problem (BPP). The analogy is presented in Section 2. Two extensions of BPP are considered in Section 3. Section 4 presents our methodology to solve this problem: a hybridization between a metaheuristic and a list algorithm. More specifically, the used metaheuristic is a Particle Swarm Optimization (PSO). Experimentations and results are summarized in Section 5. We compare ourselves to a previous paper which used Simulated Annealing (SA) (Klement et al., 2017). The current paper ends with a conclusion and some perspectives in Section 6.

2. ANALOGY WITH BIN PACKING PROBLEMS

This problem can be seen as a Bin Packing Problem (Gourgand et al., 2014a). BPP considers N items, with a given size, and some bins with the same capacity. The aim is to pack all the items in a minimum number of bins. The size of the packed items has to respect the capacity of the bins. Each item has to be assigned once and only once.

In this paper, the assignment to the material resource is considered. The aim is to assign exams to a material resource during a period. The planning horizon is made by couples period/resource. The objective is to assign exams to couples period/resource. Exams have to be done as soon as possible: the aim is to minimize the number of couples, (= the number of bins). If bins are sorted by period and filled in this order, by using less bins, we use less periods. An example is given by Fig. 1, where the assignment of exams to material resources MR_1, MR_2 and MR_3 is considered. Table 1 summarizes analogies between BPP and the current problem: exams planning with resources assignment.

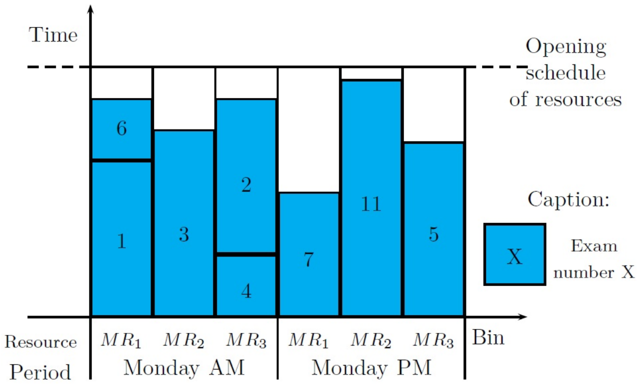

Fig. 1. Representation of HGT problem as a BPP

Table 1. Analogies between BPP and HGT problem

	Bin Packing Problem	Problem of exams planning with resources assignment
Data	Item	Exam
	Bin	Couple period/resource
	Size of an item	Processing time of an exam
	Capacity of a bin	Opening schedule of resources
	-	Due date
Problem	Assign items to one bin	Assign exams to one couple period/resource
Constraints	Capacity of bins	Opening schedule of resources
	-	Compatibility exam - resource
Criteria	Minimize the number of used bins	Minimize the number of used couples, so periods
	-	Minimize the number of non-respected due dates

3. BIN PACKING PROBLEM WITH CONSTRAINTS

Two extensions of BPP should be considered: BPP with item-bin conflicts, to represent the incompatibility between an item and a bin; and BPP with priority, to represent some items which have to be assigned to the first bins.

In the literature, BPP with conflicts mostly describes conflicts between items: some items can not be packed together in the same bin. A few papers consider this extension of BPP. In 2004, (Gendreau et al., 2004) developed a heuristic and defined a lower bound; in 2010, (Khanafer et al., 2010) improved some known lower bounds; still in 2010, (Muritiba et al., 2010) used approximate methods to build an initial solution to initialize a Branch & Price.

Few works take into account item-bin conflicts. In 2008, (Gupta et al., 2008) considered a consolidation of multiple underutilized servers into a fewer number of non-dedicated servers that can host multiple applications. The bin-item conflicts represented server-application conflict, when for example 64-bits applications can not be located on 32-bits servers. (Gupta et al., 2008) proposed a two stage heuristic algorithm and, in 2009, (Agrawal et al., 2009) proposed a grouping genetic algorithm to solve the same problem.

In our problem, the item-bin conflicts represent the exams (for example a MRI exam) that can not be assigned to all resources (for example an X-Ray machine).

The second point of the studied problem is the due date of the exams. A bin represents a couple period/resource. If an item has a due date of 3, it must be assigned to a bin with a period of 3 or less.

In 2003, (Guinet and Chaabane, 2003) considered a planning problem for an operating theatre with deadline on operations. To solve this problem, the authors proposed a primal dual heuristic. In 2006, (Jebali et al., 2006) proposed a two step heuristic to solve a similar problem. The first step consists of assigning surgical operations to operating rooms. The second step consists of sequencing the assigned operations with the objective of improving operating room use. In 2013, (Vijayakumar et al., 2013) proposed an extension of BPP with priority cases. The problem is modeled as a dual bin packing.

In our work, we propose to model both aspects of the problem, the item-bin conflicts and the priority level of items. A mathematical model of this extension has been proposed by (Gourgand et al., 2015).

4. METHOD

The proposed tool, illustrated by Fig. 2, uses a hybridization of a metaheuristic and a heuristic, more precisely a list algorithm. A single solution based metaheuristic or a population based metaheuristic can be used. The encoding used by the metaheuristic is a list Y of items. The metaheuristic browses the set of lists Y. List algorithm L considers the items according to their order in list Y to assign them to the required bins, considering the problem constraints. This builds solution X. Objective function H evaluates solution X. According to this evaluation, the solution is chosen or not by the metaheuristic. At the end of the running, the solution given by the hybridization is best list Y^* of items: the one which optimizes the objective function by applying the list algorithm. This hybridization can be used to solve many problems: the specificity of a given problem is only considered in the list algorithm.

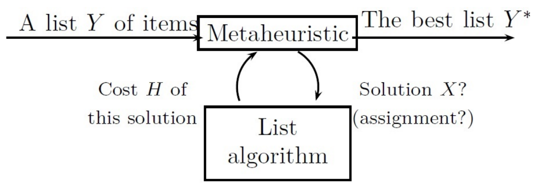

Fig. 2. Hybridization metaheuristic - List algorithm

4.1 List Y of items

The general scheme of the encoding is given by Equation (1). Set Ω is the set of all permutations of items. Cardinal of Ω is $N!$ with N the number of items. One solution $Y \in \Omega$ is a list of items. S is the set of all admissible solutions X built by list algorithm L. A solution is admissible if it respects all constraints of the problem. Solution X is evaluated thanks to criterion or objective function H. More details about the encoding are given in (Gourgand et al., 2014b).

$$Y \in \Omega \underset{Heuristic\ L}{\longrightarrow} L(Y) = X \in S \underset{Criterion\ H}{\longrightarrow} H(X) \quad (1)$$

To initialize a solution for the HGT problem, we propose to start from a good solution. Indeed, a good solution for this problem can be found by sorting the items by their due date. By doing this, the exam with the earliest due date will be packed in the first bin. Given that, a lot of items having the same due date, there are a lot of initial solutions which respect this rule.

4.2 List Algorithm L

The used list algorithm is adapted from Best Fit algorithm, developed by (Johnson, 1974) to build a solution to BPP. If this list algorithm is used to solve a HGT instance, the algorithm must check the compatibility between the exam and the resource before assigning the exam to the resource. The bins are sorted by increasing order of their period number.

4.3 Objective function H

To solve BPP, the classical objective function is the number of used bins. Many solutions may have the same value of this objective function. Thus, the searching space is reduced to a set of solutions whom criterion is one unit over the optimal. Guiding capacity of the searching algorithm is lost. To face this problem, (Falkenauer and Delchambre, 1992) introduced a new objective function. It characterizes the filling level of used bins. For a given number of bins, this objective function prefers solutions in which bins are full and bins are almost empty to solutions in which all bins are filled in a homogeneous way, where the lost space in each bin is equivalent. Thus, it is easier to converge on a solution with an empty bin. This function is defined by Equation (2). The objective of BPP is to maximize this function.

$$f_{BPP}(X) = \frac{\sum_{j=1}^{M(X)} (F_j(X)/C)^k}{M(X)} \quad (2)$$

With:

- $M(X)$ number of used bins in current solution X,
- F_j sum of sizes w_i of items i packed in bin j, such as defined in Equation (3), with $\delta_{j,X_i} = 1$ if $X_i = j$, *i.e.* item i is assigned to bin j, 0 otherwise.

$$F_j(X) = \sum_{i=1}^{N} w_i.\delta_{j,X_i}, \forall j \in \{1, N\} \quad (3)$$

- C capacity of a bin,
- $k > 1$ constant.

If k gets bigger, well-filled bins will be favored rather than homogeneous-filled bins. (Falkenauer, 1996) proved that for k bigger than 2, a solution with $M + 1$ bins with M_F full bins could have an objective function bigger than a solution with M bins homogeneously filled. Values of k bigger than 2 bring a too fast convergence of the current solution to a local optimum. Thus, the best value of k is 2.

For the HGT problem, the used objective function in the hybridization is adapted from Equation (2), but the criteria after the computation will be:

- The Multiplicative Inverse of the Falkenauer Criterion (MIFC= $1/eq.(2)$);
- The number of Non-Respected Due Dates (NRDD).

The goal is to find an optimal solution minimizing the sum defined by Equation (4), with M a number bigger than the maximal value of the Falkenauer criterion.

$$(NRDD) \times M + (MIFC) \quad (4)$$

By minimizing this sum, the objective is to find a solution with the minimum of Non-Respected Due Dates (NRDD), and then a solution with the same NRDD and the fewest used bins.

4.4 Metaheuristic: Particle Swarm Optimization

State of the art Particle Swarm Optimization (PSO) is a population based metaheuristic introduced by James Kennedy, a social psychologist, and Russel Eberhart, an electrical engineer (Eberhart and Kennedy, 1995). This metaheuristic considers a swarm of solutions. Each solution is called a particle. Each particle can move following its own trajectory, its previous one, or the trajectory of the swarm. It represents the moves of a swarm of birds. In continuous optimization, this combination is modeled by the System of Equations (5).

$$\begin{cases} X_{p,t+1} &= X_{p,t} + V_{p,t+1} \\ V_{p,t+1} &= c_1.V_{p,t} + c_2.r_2.(P_{p,t} - X_{p,t}) + c_3.r_3.(G_t - X_{p,t}) \end{cases} \quad (5)$$

With:

- p a particle,
- $V_{p,t}$ speed of particle p at time t,
- $X_{p,t}$ position of particle p at time t,
- $P_{p,t}$ best own position of particle p known at time t,
- G_t best position of the swarm known at time t,
- $c_1, c_2, c_3 \in [0, 2[$ socio-cognitive coefficients: c_1 to traduce the inertia, c_2 the influence of the best own position, c_3 the influence of the best position of the swarm,
- r_2, r_3 numbers randomly and uniformly generated $\in [0, 1[$.

The operators (plus +, minus -, times .) have to be translated in combinatorial optimization. (Kennedy and Eberhart, 1997) proposed the first translation to use PSO in combinatorial optimization.

Improvements Using our hybridization, a position of a particle p is a list $Y_{p,t}$ of items at time t. When particle p is moving, each item of the list is influenced by a combination between:

- **The best position of the particle**: the best solution browsed by the current particle since the last initialization of the swarm.
- **The best known position of the swarm**: the best solution browsed by the swarm since the last initialization of the swarm.
- **The best known position of the historical swarm**: the best solution browsed by the swarm since the first initialization of the swarm.

- **A random mutation** which moves an item of list $Y_{p,t}$ to a random place in the list.

To avoid the particle to get stuck into local minimums, we propose two mechanisms that do not exist in the classical PSO. The first one is the random mutation. Depending on a defined coefficient, a part of the solution will be influenced in a random way.

In addition to be subject of mutations, the swarm may be dispersed. Dispersion of a the swarm is inspired by tests of NoHope and factor of Rehope defined by (Clerc, 1999). After a given number of iterations without any improvements of the current solution, evolution of the particles is guided. In particular, last point of NoHope test says that after a predetermined number of iterations, PSO algorithm can not improve the current solution anymore. A part of the swarm is randomly re-initialized.

This re-initialization will lead to consider two "best known solutions" of the swarm, the actual best and the historical best. The first one considers only the best result found since the last initialization whereas the historical best considers the best ever found solution.

Our proposal to move a particle To light the computation, we propose a new move of particles. It does not use the notion of speed. According to the value of the coefficient c_2, c_3, c_4, c_5 and to a number randomly and uniformly generated $r \in [0, 1[$, each item of the list will use: either the used place in the current position, either the used place in the best own position, either the used place in the best position of the actual or historical swarm:

- If $r < c_2$, the current item will use the place it is using in the best position of the considered particle,
- If $c_2 \leq r < c_2 + c_3$, the current item will use the place it is using in the best position of the actual swarm,
- If $c_2 + c_3 \leq r < c_2 + c_3 + c_4$, the current item will use the place it is using in the best historical position of the swarm,
- If $c_2 + c_3 + c_4 \leq r < c_2 + c_3 + c_4 + c_5$, the current item will be moved randomly in the particle,
- Otherwise, the current item is not influenced, it stays where it is.

Using the same notations as previously, and with R_t the best position of the historical swarm known at time t, pseudo-code of the move of a particle with mutation is given by Algorithm 1. The sum of all coefficients must be lower than 1.

4.5 The best list Y^*

The proposed combinatorial PSO is used hybridized with a list algorithm. While stopping criteria is not reached, at each iteration, a move of particles is made: new position $Y_{p,t+1}$ of each particle of the swarm is determined as described by Algorithm 1. To determine the different best positions, list algorithm L defined in Section 4.2 is applied on each particle $Y_{p,t+1}$ to give solution $X_{p,t+1}$. Then, objective function H defined in Section 4.3 is used to compute the value of each particle. At the end of the computation, best particle Y^* is the particle Y with the minimum value of $H(L(Y))$. The principle algorithm is described by Algorithm 2.

Algorithm 1. Move of the particle

```
Require: Y_{p,t}, P_{p,t}, G_t, R_t, c2, c3, c4, c5, time t
  for all i ∈ {1, N} do
    Random and uniform computation of r ∈ [0, 1[
    Found := false, j := 0
    while Found = false do j := j + 1
      if r < c2 then
Item at place i in Y_{p,t} will be moved to its place in P_{p,t}
        if Y_{p,t}(i) = P_{p,t}(j) then Found := true
        end if
      else
        if c2 ≤ r < c2 + c3 then
Item at place i in Y_{p,t} will be moved to its place in G_t
          if Y_{p,t}(i) = G_t(j) then Found := true
          end if
        else
          if c2 + c3 ≤ r < c2 + c3 + c4 then
Item at place i in Y_{p,t} will be moved to its place in R_t
            if Y_{p,t}(i) = R_t(j) then Found := true
            end if
          else
            if c2 + c3 + c4 ≤ r < c2 + c3 + c4 + c5 then
Item at place i is moved to a random place j
              j := random ∈ {1, .., N}
            else
              j := i
            end if
          end if
        end if
      end if
    end while
    Item at place i is switched with item at position j
  end for
```

Algorithm 2. Principle Algorithm

```
Require: A number of iterations before a re-initialization
  Initialization of the swarm of particles, c := 0
  while Stopping criterion is non reached do
    if c < A then
      for All Particles do
        Y_{p,t+1}: Move of the particle
        Application of the list algorithm to the current position
        Determination of the best own position of the particle
      end for
      Determination of G_t the best position browsed by the swarm
      if R_t ≥ G_t then
        Recording of the best position R_t := G_t
      end if
      t := t + 1
      if G_{t+1} ≠ G_t then
        c := 0
      end if
    else
      Initialization of the swarm of particles, c := 0
    end if
    c := c + 1
  end while
```

5. EXPERIMENTATIONS AND RESULTS

Our method has been tested on two problems:

- The classical BPP
- BPP with item-bin conflicts and priority level on item

To solve these two different problems, only the list algorithm and the objective function will change (see Section 4.3). Indeed, for the extended BPP, the Best Fit algorithm has to consider the conflicts between items and bins to assign the items. The objective function will be a weighted sum of the two criteria such as the main objective is to minimize the number of non-respected due dates and the second objective is to minimize the number of used bins.

To initialize the swarm, several different solutions are needed. The rule for initial solution was given in Section 4.1. For classical BPP, there is no due date, so initial solutions are fully random.

To compare our results to the works of (Klement et al., 2017), its method has been re-implemented to use the same objective function and initial solutions. By doing this, we note that both changes improved its previous results.

5.1 Parameters

To compare both methods, the stopping criterion is fixed to a number of calls of objective function H, which is the same for both methods. This number is fixed to $1000 \times N$. To use the proposed PSO, we propose the following parameters: $c_2 = 0.6$; $c_3 = 0.07$; $c_4 = 0.08$; $c_5 = 0.01$.

We also need to fix the number of iterations without improvements which will induce a re-initialization of the swarm. We fix this number at $60 \times N$.

5.2 Dataset

The data are generated from bin packing instances, previously presented by (Klement et al., 2017). The characteristics and the size of the data represent real instances. The planning horizon is made by 8 to 40 periods. One period represents one half-day, thus the planning horizon is between 4 and 20 days. 4 to 8 resources are available. 50 to 500 exams need to be planned and assigned. Each exam has a priority representing the due date. Incompatibilities between resources and exams are set with a probability of 50%.

Two kinds of data were created:

- For instances 50A to 500A, each processing time is between 5 and 45, with 5 minutes steps. Each material resource has an opening schedule equal to 300 minutes.
- For instances 50B to 500B, each processing time is between 1 and 100. Each material resource has an opening schedule equal to 100.

5.3 Results

The results are presented in Tables 2 and 3. In both tables, Simulated Annealing (SA) and the proposed PSO are compared, both metaheuristics are used in the same conditions (same initialization, stopping criterion, list algorithm, number of evaluated solutions and objective function). The reported values are the best found solution and the worst solution if it is different from the best.

Table 2 presents results for Bin Packing instances, giving the number of used bins.

Table 3 gives results for the same instances adapted to the HGT context considering incompatibilities and priorities. It gives the number of non-respected due dates and the number of used bins (couples period/resource). This table also provides the original results obtained by (Klement et al., 2017), noted SA*. The second criterion used in the objective function by (Klement et al., 2017) is different from the one we use adapted from Falkenauer. In both cases, the first objective is to minimize the number of non-respected due date; so we report only this criterion in column SA*.

Table 2. BP Results

BPP	SA	PSO	Optimal
50A	5	5	5
50B	25	25	25
100A	8	8	8
100B	48	48	48
200A	16	16	16
200B	105	105	105
300A	26	26	26
400A	34	34	34
500A	42	42	42
500B	241	241-(242)	240
500C	262	262	262
500D	242	242-(243)	241
500E	248	248	246

Table 3. BP with constraints Results

BPP_inc_dd	SA*	SA	PSO
50A	(0,-)	(0,5)	(0,5)
50B	(1,-)	(1,26)	(1,26)
100A	(0,-)	(0,8)	(0,8)
100B	(0,-)	(0,49)	(0,49)
200A	(0,-)	(0,16)	(0,16)
200B	(0,-)	(0,106)	(0,106)-(0,107)
300A	(0,-)	(0,26)	(0,26)
400A	(0,-)	(0,34)	(0,34)
500A	(0,-)	(0,42)	(0,42)
500B	(18,-)	(7,252)-(11,250)	(2,244)-(3,247)
500C	-	(3,276)	(1,272)
500D	-	(1,253)	(1,246)
500E	-	(7,259)	(3,253)

The computational times are the same regardless the treated problem. For the 50 bins instances, the average computational time is 1 second, and for the largest ones, 30 minutes. According to Table 2, we note that the results for BPP are really closed for SA and PSO. This is due to the fact that nearly every instances are solved optimally. For the 500B instance, both methods give a solution 1 bin away from the optimal.

For the HGT problem, the obtained results are close except for the 500B instance. For this instance, which is the hardest one, the first thing that we can see is the big improvement from the work of (Klement et al., 2017). The previous best solution for the lowest number of non-respected due dates was 18, and, with the same method but with the new proposed objective function and initialization method, the best result is 7. For this instance, the PSO gives the best solutions. Due to the similarity between both methods, we create new instances based on the ones used by (Scholl et al., 1997). These instances are 500C, 500D and 500E. The results on these instances comfort the ones obtained previously.

Based on these results, we can conclude that PSO works better than SA on the largest instances. This is due to the principle of SA. At the beginning of the exploration of the searching space, SA principle is to accept nearly all solutions, which leads to lose the good properties of the initials solutions. With the proposed PSO, even if the particles lose the good properties of the initial solutions, the particle will still be influenced by the best known solution. In further work, we would like to use our proposal to solve other hard Bin Packing or HGT instances.

6. CONCLUSION

For several years, the public health care system has tried to improve the performance of the system. To do so, Hospital Group of Territory pools a part of its resources or pools a set of operations which have to be performed. The decision on where and when each operation will be executed has to consider several criteria and constraints on the operations. The goal of this paper is to propose an efficient method to assign operations to resources and periods, considering time constraints, due dates and resources incompatibilities.

This problem is an extension of the classical Bin Packing Problem, taking into account item-bin incompatibilities and non-respected due dates. To solve this problem, we propose a new Particle Swarm Optimization method. In this method, each particle represents a solution. To explore the searching space, a particle is influenced by its personnal best position, the best position of the swarm, the best known solution and a random influence. The results given by this method are compared to the ones obtained by Simulated Annealing proposed by (Klement et al., 2017). We also improve its work by using initial solutions with good properties and by using the Falkenauer criterion which helps to progress in the search space.

In future works, we want to solve largest instances to challenge our method. We also want to propose multi-criteria optimization to propose more than one solution where a criterion is favored to another one. The objective would be to obtained the Pareto front with all non-dominated solutions (all the solutions in which it is impossible to improve a criterion without downgrading another one). The current work provides only one point of the Pareto front, the solution where the number of non-respected due dates is the lowest.

REFERENCES

Agrawal, S., Bose, S.K., and Sundarrajan, S. (2009). Grouping genetic algorithm for solving the serverconsolidation problem with conflicts. In *Proceedings of the First ACM/SIGEVO Summit on Genetic and Evolutionary Computation*, GEC '09, 1–8.

Clerc, M. (1999). The swarm and the queen: towards a deterministic and adaptive particle swarm optimization. In *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on*, volume 3. IEEE.

Eberhart, R. and Kennedy, J. (1995). A new optimizer using particle swarm theory. In *Micro Machine and Human Science, 1995. MHS'95., Proceedings of the Sixth International Symposium on*, 39–43. IEEE.

Falkenauer, E. (1996). A hybrid grouping genetic algorithm for bin packing. *Journal of heuristics*, 2(1), 5–30.

Falkenauer, E. and Delchambre, A. (1992). A genetic algorithm for bin packing and line balancing. In *Robotics and Automation*, 1186–1192. IEEE.

Gendreau, M., Laporte, G., and Semet, F. (2004). Heuristics and lower bounds for the bin packing problem with conflicts. *Computers & Operations Research*, 31(3), 347–358.

Gourgand, M., Grangeon, N., and Klement, N. (2014a). Activities planning and resource assignment on multiplace hospital system: Exact and approach methods adapted from the bin packing problem. In *7th International Conference on Health Informatics, Angers, France*, 117–124.

Gourgand, M., Grangeon, N., and Klement, N. (2014b). An analogy between bin packing problem and permutation problem: A new encoding scheme. In *Advances in Production Management Systems. Innovative and Knowledge-Based Production Management in a Global-Local World*, volume 438, 572–579. Springer Berlin Heidelberg.

Gourgand, M., Grangeon, N., and Klement, N. (2015). Activities planning and resources assignment on distinct places: a mathematical model. *RAIRO-Oper. Res.*, 49(1), 79–98.

Guinet, A. and Chaabane, S. (2003). Operating theatre planning. *International Journal of Production Economics*, 85(1), 69 – 81. Planning and Control of Productive Systems.

Gupta, R., Bose, S., Sundarrajan, S., Chebiyam, M., and Chakrabarti, A. (2008). A two stage heuristic algorithm for solving the server consolidation problem with item-item and bin-item incompatibility constraints.

Jebali, A., Alouane, A.B.H., and Ladet, P. (2006). Operating rooms scheduling. *International Journal of Production Economics*, 99(1), 52 – 62. Control and Management of Productive Systems.

Johnson, D.S. (1974). Fast algorithms for bin packing. *Journal of Computer and System Sciences*, 8(3), 272–314.

Kennedy, J. and Eberhart, R.C. (1997). A discrete binary version of the particle swarm algorithm. In *International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation*, volume 5, 4104–4108. IEEE.

Khanafer, A., Clautiaux, F., and Talbi, E.G. (2010). New lower bounds for bin packing problems with conflicts. *European journal of operational research*, 206(2), 281–288.

Klement, N., Gourgand, M., and Grangeon, N. (2017). Medical imaging: Exams planning and resource assignment: Hybridization of a metaheuristic and a list algorithm. In *10th International Conference on Health Informatics, Porto, Portugal*.

Laurent, A., Deroussi, L., Grangeon, N., and Norre, S. (2017). A new extension of the RCPSP in a multisite context: Mathematical model and metaheuristics. *Computers & Industrial Engineering*, 112, 634–644.

Muritiba, A.E.F., Iori, M., Malaguti, E., and Toth, P. (2010). Algorithms for the bin packing problem with conflicts. *Informs Journal on computing*, 22(3), 401–415.

Scholl, A., Klein, R., and Jrgens, C. (1997). Bison: A fast hybrid procedure for exactly solving the one-dimensional bin packing problem. *Computers & Operations Research*, 24(7), 627 – 645.

Vijayakumar, B., Parikh, P.J., Scott, R., Barnes, A., and Gallimore, J. (2013). A dual bin-packing approach to scheduling surgical cases at a publicly-funded hospital. *European Journal of Operational Research*, 224(3), 583 – 591.