Computers & Operations Research 37 (2010) 640–648

Contents lists available at ScienceDirect

Computers & Operations Research

journal homepage: www.elsevier.com/locate/cor

Using a greedy random adaptative search procedure to solve the cover printing problem

D. Tuyttens*, A. Vandaele

Polytechnic Faculty of Mons, Laboratory of Mathematics and Operational Research, Belgium

ARTICLE INFO

Available online 21 June 2009

Keywords:
Combinatorial optimization problem
Metaheuristics
GRASP

ABSTRACT

In this paper, the cover printing problem, which consists in the grouping of book covers on offset plates in order to minimize the total production cost, is discussed. As the considered problem is hard, we discuss and propose a greedy random adaptative search procedure (GRASP) to solve the problem. The quality of the proposed procedure is tested on a set of reference instances, comparing the obtained results with those found in the literature. Our procedure improves the best known solutions for some of these instances. Results are also presented for larger, randomly generated problems.

1. Introduction

Combinatorial optimization problems [4,17] involve finding optimal solutions from a large but finite set of feasible solutions. Many of these problems cannot be solved to optimality in reasonable computation times, due to their inner nature or to their size. It is the case of the cover printing problem treated in this paper. The use of heuristic methods is then the natural choice for solving these problems. The goal of heuristics [18] is to quickly produce good approximate solutions, without necessarily providing any guarantee of solution quality. The effectiveness of these methods depends upon their ability to adapt to a particular realization, avoid entrapment at a local optima, and exploit the basic structure of the problem.

Metaheuristics are techniques that have widespread success in providing high-quality near-optimal solutions to many real-life complex optimization problems in diverse areas. Among them, we find simulated annealing, tabu search, greedy random adaptative search procedure (GRASP), genetic algorithms, scatter search, VNS, ant colonies, and others. We refer to [15,18] for an introduction, [16] for a bibliography and [11,12] for an overview. The success of metaheuristics [10,23] motivated us to design and implement the greedy random adaptative search procedure (GRASP) for the cover printing problem.

The remaining parts of the paper are organized as follows. In Section 2 we give a formal statement of the considered problem. Section 3 consists of the greedy random adaptative search procedure's presentation. In Section 4 we illustrate efficiency of the proposed method by comparing the obtained results to the best-known results available in the literature. Conclusions are drawn in Section 5.

2. Problem formulation

In this section, we describe the cover printing problem. This problem concerns the publishing industry and consists in finding the best assignment of a set of I book covers to offset plates for print. For each book cover, a given number N_i, $i=1,\dots,I$, of copies has to be printed. Each offset plate contains four compartments so that each printed sheet of paper contains four book covers, which may be similar or different. We assume that no compartments are left empty. After printing, the sheets of paper are cut into four parts. The different quantities to determine in this problem are:

- the number of offset plates to use,
- the assignment of a book cover to each compartment of each offset plate,
- the number of sheets of paper to print with each offset plate in order to produce the required copies for all the book covers.

Once a 4-tuple (i_1,i_2,i_3,i_4) of book covers $(i_1,i_2,i_3,i_4 \in \{1,\dots,I\})$ is assigned to an offset plate j, z_j sheets of paper are printed with this offset plate. Let us call "configuration" of an offset plate j, the couple (n_j,r_j), where $1 \le n_j \le 4$ is the number of different book covers assigned to the offset plate j and $1 \le r_j \le n_j$ is the number of book covers assigned to the offset plate j for which the required copies are produced after printing. For an offset plate j, 10 different configurations (n_j,r_j) are possible: $(4,4),(4,3),(4,2),(4,1),(3,3),(3,2),(3,1),(2,2),(2,1)$ and $(1,1)$.

Some unnecessary copies of book covers may be produced and this results in sheets of paper losses. Suppose that we have n_j

* Corresponding author.
E-mail addresses: daniel.tuyttens@fpms.ac.be (D. Tuyttens), arnaud.vandaele@gmail.com (A. Vandaele).

0305-0548/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cor.2009.06.006

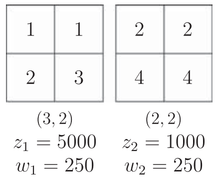

Fig. 1. A simple example.

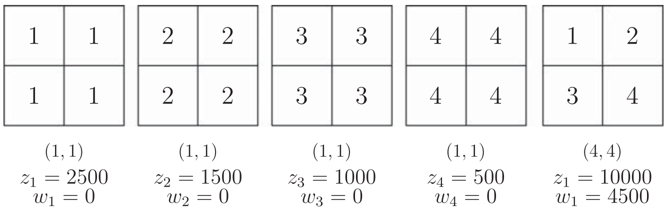

Fig. 2. Example of opposite solutions.

different book covers assigned to an offset plate j (noted $i_1, \ldots, i_{n_j}$), then the number of wasted sheets of papers can be easily calculated as follows:

$$w_j = \left[\sum_{k=1}^{n_j} \max(0, \eta_{i_k} z_j - \bar{N}_{i_k}) \right] \Big/ 4$$

$\bar{N}_{i_k}$ is the remaining number of copies required for the book cover i_k when considering the offset plate j. $\bar{N}_{i_k} = N_{i_k}$ if the offset plate j is the first offset plate wherein the book cover i_k is assigned. η_{i_k} is the multiplicity of book cover i_k on the offset plate j. We have that $\eta_{i_1} + \cdots + \eta_{i_{n_j}} = 4$.

This can be explained with a simple example. Let us consider four different book covers ($I = 4$) and the number of copies required for each book cover: $N_1 = 10\,000$, $N_2 = 6000$, $N_3 = 4000$, $N_4 = 2000$. A feasible solution to this problem is illustrated in Fig. 1. The 4-tuple $(1,1,2,3)$ is assigned to offset plate 1, $z_1 = 5000$ sheets of paper are printed and the configuration is $(n_1, r_1) = (3,2)$. One thousand unnecessary copies of book cover 3 are produced generating $w_1 = 250$ wasted sheets of paper. The 4-tuple $(2,2,4,4)$ is assigned to offset plate 2 with $\bar{N}_2 = N_2 - z_1 = 1000$ and $\bar{N}_4 = N_4 = 2000$. The configuration is $(n_2, r_2) = (2,2)$. One thousand unnecessary copies of book cover 2 are produced generating $w_2 = 250$ wasted sheets of paper.

The cost structure has two main components: C_f, the unit cost of an offset plate, and C_t, the unit cost of a sheet of paper whatever the offset plate used for printing it. The cost of one offset plate is usually very high, while the cost of each printed sheet of paper is normally low. The best assignment should thus minimize both the number of offset plates used and the total number of sheets of paper printed in order to minimize the total production cost. The problem, then, consists in finding a compromise between two opposite solutions: one solution using the maximal number of offset plates and producing the minimal number of printed sheets of paper (see left part of Fig. 2, four offset plates and 5500 sheets) or a solution using the minimum number of offset plates and producing a high number of printed sheets of paper (see right part of Fig. 2, one offset plate and 10 000 sheets). Fig. 1 presents a compromise between these two opposite solutions.

This problem has been treated in some previous published works. After the failure of the use of exact resolution methods [14], a simulated annealing combined with linear programming is proposed in [22], a tabu search metaheuristic is tested in [3] and two approaches of genetic algorithms with linear programming solver are presented in [5]. In [6], a branch and price algorithm is presented but the results are not quite satisfactory since only boundaries of the optimal solution are provided. In Section 4, we compare our results with the published results.

A non-linear formulation of the problem in both continuous and binary variables is presented in [22]. Let $i = 1, \ldots, I$ be the index of the different book covers to be produced in quantity N_i and $j = 1, \ldots, I$ be the index of the offset plates that can potentially be used. The following variables are defined:

$z_j \geq 0$ represents the number of printed sheets of paper from the offset plate j (I continuous variables).

$y_{ij} \in \{0,1,2,3,4\}$ represents the occurrence of book cover i in offset plate j (I^2 integer variables).

$$x_j = \begin{cases} 1 & \text{if offset plate } j \text{ is used} \\ 0 & \text{otherwise } (I \text{ binary variables}) \end{cases}$$

The problem is then formulated as follows:

$$\begin{cases} \min C_f \left(\sum_{j=1}^{I} x_j \right) + C_t \left(\sum_{j=1}^{I} z_j \right) \\ \sum_{j=1}^{I} z_j y_{ij} \geq N_i, \quad i = 1, \ldots, I \\ \sum_{i=1}^{I} y_{ij} = 4x_j, \quad j = 1, \ldots, I \\ z_j \leq M x_j, \quad j = 1, \ldots, I \end{cases}$$

where M is an upper bound of the number of copies of each offset plate to be produced, e.g., $M = \max_{i=1,\ldots,I} N_i$. This non-linear formulation of the problem involves $(2I + I^2)$ variables (I binary, I continuous and I^2 integer variables) and $3I$ constraints (I non-linear constraints and $2I$ linear constraints). However, a linear programming formulation is also presented in [22], but there are then $8I^2 + 2I$ variables ($4I^2 + I$ binary and $4I^2 + I$ continuous) and $8I^2 + 7I$ constraints. We used the powerful ILOG CPLEX 9.0 [13] software to solve the linear programming formulation of the problem and the maximum size that could be solved in a reasonable time is $I = 8$ (528 variables and 568 constraints). Since the size I of the number of book covers to be considered in a real production case is higher than $I = 8$, we turn to the use of heuristics. To be more specific, the size I of the number of book covers in a real production case is varying in general between 10 and 20 but sometimes this size can be much higher. The number of copies to be printed is varying from a few thousands to a few hundred thousands.

3. The greedy random adaptative search procedure

The greedy random adaptative search procedure is a multistart metaheuristic [7,8,19–21], in which each iteration returns a feasible solution to the problem and consists of two phases: a construction phase and a local search phase. The best solution over all iterations is kept as the final result. A generic GRASP pseudo-code is given in Fig. 3.

In the construction phase, a feasible solution is iteratively constructed, one element at a time. At each construction iteration, the choice of the next element to be added is determined by ordering all elements in a candidate list with respect to a greedy function. The method is adaptative because the greedy function takes into account previous decisions made in the construction. The probabilistic component of a GRASP is characterized by randomly choosing one of the best candidates in the list, but not necessarily the top candidate. The list of best candidates is called restricted candidate list (RCL). The solutions generated by a GRASP construction are not guaranteed to be locally optimal with respect to simple neighborhood definitions. Hence, it is almost always beneficial to apply a local search to attempt to improve each constructed solution. Successful applications can be found in [1,2,7,9,16].

```
procedure GRASP(MaxIterations,Seed)
  ReadInstance();
  Cost* ← ∞
  for k = 1,..., MaxIterations do
      Solution ← GreedyRandomConstruction(Seed);
      Solution ← LocalSearch(Solution);
      if Cost(Solution) < Cost* then do
          Cost* ← Cost(Solution);
          BestSolution ← Solution;
      end;
  end;
  return BestSolution;
end.
```

Fig. 3. A generic GRASP pseudo-code.

One of the major advantages of the GRASP metaheuristic is how easy this general scheme may be adapted to the solution of particular problems. In the next sections, we customize a GRASP metaheuristic for the cover printing problem.

3.1. The construction phase

A feasible solution of the cover printing problem is composed of a set of offset plates. Each offset plate is characterized by a 4-tuple (noted $i_{1234}=(i_1,i_2,i_3,i_4)$ with $i_1,i_2,i_3,i_4\in\{1,\dots,I\}$) of book covers assigned to the offset plate, which may be similar or different, and its configuration (n,r), where n is the number of different book covers in the 4-tuple and r is the number of book covers in the 4-tuple for which the required copies are produced after printing the offset plate. We have that $1\le n\le 4$ and $1\le r\le n$. One offset plate is now noted as $Op[i_{1234},n,r]$. The number of sheets of paper printed with the offset plate, noted $z(Op[i_{1234},n,r])$, is obtained based on the value of r and the number of wasted sheets of paper, noted $w(Op[i_{1234},n,r])$, is calculated with the formula given in Section 2. If we consider the simple example given in Fig. 1, the first offset plate is characterized by the 4-tuple $(1,1,2,3)$ and the configuration $(n,r)=(3,2)$. The number of sheets of paper printed with the first offset plate $z_1=z(Op[(1,1,2,3),3,2])$ is obtained as follows: to print all required copies for book cover 1 we need $2*z_1\ge\bar{N}_1=N_1=10\,000$, for book cover 2 we need $z_1\ge\bar{N}_2=N_2=6000$, for book cover 3 we need $z_1\ge\bar{N}_3=N_3=4000$ and to satisfy $r=2$, we need $z_1\ge 5000$ and we fix $z_1=5000$. We have that $2*z_1=\bar{N}_1, z_1<\bar{N}_2, z_1>\bar{N}_3$. $z_1-\bar{N}_3=1000$ unnecessary copies of book cover 3 are produced generating $w_1=w(Op[(1,1,2,3),3,2])=250$ wasted sheets of paper. The same reasoning can be done with the second offset plate.

Fig. 4 illustrates the pseudo-code of the construction phase of the GRASP metaheuristic. At each iteration of the greedy random construction phase, an offset plate is determined until a feasible solution is obtained. Initializations are performed in lines 1 and 2. $\bar{N}_i$ is the remaining number of copies required for the book cover i and $\bar{I}$ is the number of book covers with $\bar{N}_i>0$. $COp[n,r](c_{1234})$ represents the best candidate offset plate with configuration (n,r). Solution is computed from scratch and the loop in lines 3–22 is performed until a feasible solution is obtained. Each iteration starts by sorting out the book covers in the decreasing order of $\bar{N}_i$, so that the greedy function takes into account previous decisions made in the construction. In lines 5 and 6, we extract all 4-tuple from the m first book covers, where the parameter d_S defines a depth search. The best candidates offset plates for the 10 possible configurations (n,r) (i.e., $(4,4),(4,3),(4,2),(4,1),(3,3),(3,2),(3,1),(2,2),(2,1),(1,1)$) are selected in lines 6–16 and stored in $COp[n,r](c_{1234})$. In line 17, the RCL is made up of candidate offset plates with configuration (n,r) whose number of wasted sheets of paper is lower than a value depending on a parameter α $(0\le\alpha\le 1)$ and the values of the two costs C_f and C_t. Once the RCL is available, a candidate is then randomly

```
procedure GreedyRandomConstruction(Seed)
1  Ī = I;  N̄_i ← N_i i = 1,...,I;   Solution ← ∅;
2  z(COp[n,r](c_1234)) = 0,  w(COp[n,r](c_1234)) = ∞  ∀ n,r;
3  while Ī > 0 do
4    Let { l_1,...,l_Ī} s.t. N̄_{l_1} ≥ N̄_{l_2} ≥ ... ≥ N̄_{l_Ī};
5    m ← min{ Ī, d_S} ;
6    for each i_1234 = (i_1,i_2,i_3,i_4) with i_1,i_2,i_3,i_4 ? { l_1,...,l_m} do
7      for r = 1,...,n do
8        zOp = z(Op[i_1234,n,r]);          wOp = w(Op[i_1234,n,r]);
9        zCOp = z(COp[n,r](c_1234));   wCOp = w(COp[n,r](c_1234));
10       if (wOp < wCOp) or ( wOp = wCOp and zOp > zCOp ) do
11         COp[n,r](c_1234) ← Op[i_1234,n,r];
12       else if ( wOp = wCOp and zOp = zCOp ) do
13         COp[n,r](c_1234) ←(p) Op[i_1234,n,r] with probability p = 1/2;
14       end;
15     end;
16   end;
17   RCL= { COp[n,r](c_1234) | w(Cop[n,r](c_1234) ≤ α(C_f/C_t), ∀ n,r} ;
18   COp[n*,r*](c*_1234) = Random(RCL,Seed);
19   Solution ← Solution ∪ COp[n*,r*](c*_1234);
20   Ī ← Ī − r*;
21   N̄_{c*_1234} ← N̄_{c*_1234} − z(COp[n*,r*](c*_1234);
22 end;
23 return Solution;
end.
```

Fig. 4. Pseudo-code of the construction phase.

chosen from the restricted list in line 18 and is inserted in the solution under construction in line 19. In lines 20 and 21, we update the quantities $\bar{I}$ and $\bar{N}_i$.

3.2. The local search phase

Each solution built at the construction phase is the starting point for a local search procedure in which we try to improve the solution. In the construction phase, the RCL is made up of candidate offset plates whose number of wasted sheets of paper is lower than a given value (see line 17 in Fig. 4). This restriction minimizes the total sheets of paper printed but it may lead to the use of a high number of offset plates in the solution. The improvement phase basically consists of selecting the offset plates with small number of sheets of paper printed and then of changing the configuration of the other offset plates in order to reduce the total number of offset plates used. Fig. 5 illustrates the pseudo-code of the local search phase of the GRASP metaheuristic.

Initializations are done in lines 1 and 2. The loop in lines 3–22 is performed until all the offset plates, whose number of sheets of paper printed is lower than a given value (see line 5 in Fig. 5), have been examinated and no better solution has been found by removing one of these offset plates. One offset plate is removed by changing the configuration of the other offset plates. Variable *Improve* is a flag that indicates when a better solution is found. Fig. 6 illustrates the principle of the local search. The left part of Fig. 6 represents a feasible solution of the simple example defined in Section 2. This solution has been obtained with the construction phase and contains three offset plates. In this example, we examine the third offset plate (see line 5 in Fig. 5). The book cover 3 is already included in the first offset plate (see line 8 in Fig. 5) and we change the configuration of the first offset plate from $(2,1)$ to $(2,2)$. Now, all the required copies of book cover 3 can be obtained from the first offset plate. The book cover 4 is only included in the third offset plate but can be inserted in offset plate 1 or offset plate 2 (see line 10 in Fig. 5). In order to minimize the number of wasted sheets of paper, we insert book cover 4 in offset plate 2 and change the configuration from $(1,1)$ to $(2,2)$. All offset plates are updated and a new feasible solution is

```
procedure LocalSearch(Solution)
1  Let J be the number of offset plates used in Solution;
2  BestSolution ← Solution; Improve ← True;
3  while Improve = True do
4    NewSolution ← BestSolution; Improve ← False;
5    for each Op[(i_1234, n, r)] with z(Op[(i_1234, n, r)]) ≤ β(C_f/C_t) do
6      if Improve = False then
7        for each book cover i with i ∈ (i_1, i_2, i_3, i_4) do
8          if ∃ Op'[(i'_1234, n', r')] with i ∈ (i'_1, i'_2, i'_3, i'_4) then
9            Change the configuration of the offset plate;
10         else if ∃ Op"[(i"_1234, n", r")] with n" < 4 do
11           Insert i and change the configuration of the offset plate;
12         end;
13       end;
14       Update the configuration of the other offset plates, if needed;
15       Newsolution is built;
16       If Cost(NewSolution) < Cost(BestSolution) then
17         BestSolution ← NewSolution;
18         J ← J - 1; Improve ← True;
19       end;
20     end;
21   end;
22 end;
23 return BestSolution;
end.
```

Fig. 5. Pseudo-code of the local search phase.

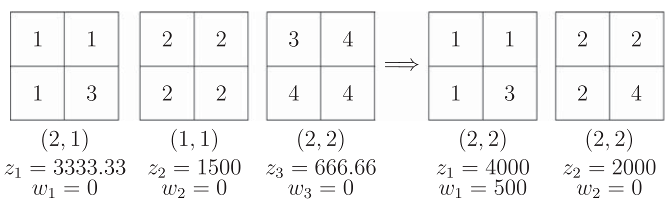

Fig. 6. Principle of the local search.

build (see lines 14 and 15 in Fig. 5 and the right part of Fig. 6). Based on the values of C_f and C_t, the procedure checks if a better solution is found (see lines 16–19 in Fig. 5).

4. Numerical results

In the first part of the numerical experiments, we have considered nine instances of the cover printing problem with 3, 4, 5, 8, 12, 15, 30, 40 and 50 book covers. All the test instances are described in Tables A1–A4 in Appendix A instance data. Instances (P1–4) are real instances and are reported from [5,22], instance (P5) was generated by [6] and used in [5], instance (P6) was generated in [5] and instances (P7–9) were constructed for the purpose of this experiment representing a larger number of book covers ($I = 30, 40, 50$). In order to be close to the experiments in [5,22], we use the same cost structure: the unit cost of an offset plate, $C_f = 18\,676$, and the unit cost of a sheet of paper, $C_t = 13.44$.

In a first step, we used the CPLEX 9.0 software [13] to solve exactly the linear mathematical programming formulation of the problem (see [22]). The results are presented in Table 1. Columns 1 and 2 indicate the name of the problem and the number of book covers. Column 3 reports the objective value of the optimal solution. Columns 4, 5 and 6 indicate the characteristics of the optimal solution: the number of offset plates used, the number of sheets of paper printed and the number of wasted sheets of paper produced. The last column indicates the CPU time computation required in seconds.

Table 1
Optimal solutions obtained with CPLEX.

Problem	I	Objective	#Op	#Sp	#Ws	CPU(s)
P1	3	136 472	2	7375	0	1
P2	4	247 916	2	15 667	292	2
P3	5	1 851 948	3	133 625	500	16
P4	8	264 348	3	15 500	750	705

Table 2
Best results obtained.

Problem	I	Optimal	SA [22]	GA [5]	GRASP
P1	3	136 472	136 472	136 472	136 472
P2	4	247 916	247 916	247 916	247 916
P3	5	1 851 948	1 855 872	1 855 872	**1 851 948**
P4	8	264 348	294 980	264 348	264 348
P5	12	–	–	276 304	**269 584**
P6	15	–	–	523 460	**515 256**

Table 3
Details on the best results obtained.

	GA [5]			GRASP			
	Objective	#Op	#Sp	Objective	#Op	#Sp	CPU(s)
P3	1 855 872	3	133 917	**1 851 948**	3	**133 625**	2
P4	264 348	3	15 500	264 348	3	15 500	4
P5	276 304	4	15 000	**269 584**	4	**14 500**	6
P6	523 460	5	32 000	**515 256**	**6**	**30 000**	9

The maximum size that could be solved in a reasonable time is I=8. For problem P5 (I=12) we stopped the computation after 2 h CPU time. The non-linear (quadratic) programming formulation of the problem presented in Section 2 has also been tested and produced similar results. Clearly the problem is difficult to solve exactly due to the exploding complexity. We now turn to the use of a GRASP heuristic to solve it.

The GRASP procedure was implemented in Visual C++ language version 6.3 and run on a Centrino Dual Core personal computer with 2 GHz and 2 Gb RAM. The control parameters of the GRASP heuristic have been fixed to: MaxIterations = 10 000, $d_S = 5$, $\alpha = 0.5$ and $\beta = 0.4$. The best results obtained with the instances P1–P6 are summarized in Table 2. This table includes the results obtained with the CPLEX software, with a simulated annealing combined with linear programming [22], with a genetic algorithm with linear programming solver [5], and with our GRASP heuristic.

The instances P1 and P2 are easy to solve and all the methods converge to the optimal solution. We will no longer consider these instances. However, the GRASP method shows better performance when dealing with the other instances. Table 3 gives more details on the best results obtained.

From Table 3 it can be seen that:

- The GRASP heuristic improves the solutions previously reported in the literature.
- For instances P3 and P5, the two methods find the same number of offset plates to use but not the same number of sheets of paper printed. In fact, the GRASP heuristic finds a better assignment of the book covers to the compartments of the offset plates. Clearly, finding good configurations for the offset plates is strategical and is more complex than finding the number of offset plates to use.
- For instance P6, the GRASP heuristic improves the best known solution. In this case, it is namely due to the construction phase of the method. The restriction of the RCL allows to minimize

Table 4
Sensitivity of parameter d_S.

	d_S	Best result			Worst result			CPU
		Objective	#Op	#Sp	Objective	#Op	#Sp	
P3	4	1 851 948	3	133 625	1 851 948	3	133 625	2
	5	1 851 948	3	133 625	1 851 948	3	133 625	2
P4	4	264 348	3	15 500	264 348	3	15 500	4
	5	264 348	3	15 500	264 348	3	15 500	4
P5	4	276 304	4	15 000	276 304	4	15 000	5
	5	**269 584**	**4**	**14 500**	269 584	4	14 500	6
P6	4	515 256	6	30 000	523 460	5	32 000	8
	5	515 256	6	30 000	521 976	6	30 500	9

Table 5
Testing larger instances.

	d_S	Best result			Worst result			CPU
		Objective	#Op	#Sp	Objective	#Op	#Sp	
P7	4	1 776 040	10	118 250	1 796 592	12	117 000	23
	5	**1 759 240**	**10**	**117 000**	1 781 276	11	117 250	32
	6	1 772 680	10	118 000	1 787 996	11	117 750	46
P8	4	2 602 040	14	174 150	2 640 736	16	174 250	33
	5	**2 585 912**	**14**	**172 950**	2 607 276	15	173 150	45
	6	2 597 672	14	173 825	2 601 368	14	174 100	64
P9	4	6 576 472	22	458 750	6 624 716	23	460 950	45
	5	**6 538 644**	**21**	**457 325**	6 603 100	19	464 900	63
	6	6 543 152	20	459 050	6 574 260	21	459 975	87

the total sheets of paper printed but it may lead to the use of a high number of offset plates in the solution. The solution proposed for instance P6 contains no wasted sheets of paper which corresponds to the minimal number of printed sheets of paper.

- The CPU times are small. The advantage of the GRASP heuristic is that it does not need the use of a linear programming solver to compute the number of sheets of paper to print for each offset plate, as in GA [5] and SA [22].

The choice of a good parameter setting is not always an easy task. The parameters of the GRASP heuristic have been fixed as follows:

- In GA [5], the CPU time is principally consumed by the linear programming solver to evaluate each generated solution and therefore the CPU time is evaluated by the number of fitness evaluations. Since 10 000 fitness function evaluations were considered, we fixed MaxIterations = 10 000 in order to compare the two methods.
- A large number of experiments was done with a large range of variation in the parameters α and β. The best results were obtained with $\alpha = 0.5$ and $\beta = 0.4$.
- In most instances, the GRASP heuristic with the parameter $d_S = 5$ is the best performer. For each instance, 10 runs were performed with $d_S = 4$ and 5. Table 4 summarizes the results obtained. The parameter d_S=6 was also tested but the solutions are not improved and the CPU times increase.

The GRASP heuristic is tested further by using the instances (P7–9) that were constructed for the purpose of this experiment representing a larger number of book covers ($I = 30, 40, 50$). These instances allow us to test more in depth the feasibility and efficiency of our GRASP heuristic. For each instance, 10 runs were performed with $d_S = 4$, 5 and 6. The results are summarized in Table 5.

Table 6
Comparison of the GRASP solution with two extreme solutions.

		min #Sp #Op $= \lceil I/4 \rceil$	GRASP	min #Op #Wp $= 0$
P7	#Op	8	10	15
	#Sp	124 000	117 000	114 375
	Objective	1 815 968	**1 759 240**	1 817 340
P8	#Op	10	14	23
	#Sp	188 550	172 950	169 588
	Objective	2 720 872	**2 585 912**	2 708 811
P9	#Op	13	21	29
	#Sp	490 700	457 325	455 388
	Objective	6 837 796	**6 538 644**	6 662 019

Turning to more speculative interrogation, one might wonder how far from the optimal solution our GRASP heuristic does leave us? We have no indication on what is the best value of the objective function but from Table 6 we can see that the best solution obtained with the GRASP heuristic is a good trade-off between two extreme solutions: the first one minimizes the total number of sheets of paper printed subject to the constraint that the number of offset plates used is minimal ($=\lceil I/4 \rceil$, the smallest integer number larger than $I/4$) and the second one minimizes the total number of offset plates used subject to the constraint that the minimum number of sheets of paper is printed (no wasted sheets of paper).

If we compare the first solution with the GRASP solution, we observe:

- for problem P 7: 2 additional offset plates reduce 72.7% of wasted sheets of paper. On an average, an additional offset plate saves 3500 sheets of paper and reduces the global cost of the solution of $C_f - 3500C_t = C_f - 2.52C_f = -1.52C_f$.
- for problem P 8: 4 additional offset plates reduce 82.3% of wasted sheets of paper. On an average, an additional offset plate saves 3900 sheets of paper and reduces the global cost of the solution of $-1.80C_f$.
- for problem P 9: 8 additional offset plates reduce 94.5% of wasted sheets of paper. On an average, an additional offset plate saves 4172 sheets of paper and reduces the global cost of the solution of $-2.00C_f$.

If we compare the GRASP solution with the second (best extreme) solution, we observe that not much improvement can be expected:

- for problem P 7: 2625 wasted sheets of paper (24.7%) remain in the GRASP solution. Since we have that $2625C_t$=$1.88C_f$, we can deduce that 11 is an upper bound for the number of offset plates and the number of offset plates can be bounded as follows: $9 \le \#\text{Op} \le 11$.
- for problem P 8: 3362 wasted sheets of paper (17.7%) remain in the GRASP solution. Since we have that $3362C_t = 2.42C_f$, the number of offset plates can be bounded as follows: $11 \le \#\text{Op} \le 16$.
- for problem P 9: 1937 wasted sheets of paper (5.5%) remain in the GRASP solution. Since we have that $1937C_t = 1.39C_f$, the number of offset plates can be bounded as follows: $14 \le \#\text{Op} \le 22$.

Clearly, the GRASP heuristic produces a good approximation of the optimal solution.

The numerical experiments described so far were made on the basis of a single instance of each size. We used also a unique cost structure: the unit cost of an offset plate, $C_f = 18\,676$, and the unit cost of a sheet of paper, $C_t = 13.44$. The reason of this choice was to be close to the experiments found in the literature, in order to compare the obtained results with other methods.

Table 7
Ratio C_f/C_t sensitivity on instances with 30 book covers.

P30-1	E#Op	C_f*4	C_f*2	C_f	$C_f/2$	$C_f/4$	E#Wp
#Op	8	8	8	12	12	13	18
OpUB		9	10	12	13	15	
#Sp	124 300	124 300	124 300	117 600	117 600	117 100	116 375
WpRD	7925	0%	0%	85%	85%	91%	0
P30-2	E#Op	C_f*4	C_f*2	C_f	$C_f/2$	$C_f/4$	E#Wp
#Op	8	8	9	10	12	12	17
OpUB		9	10	12	13	15	
#Sp	116 600	116 600	113 400	111 450	109 600	109 600	108 550
WpRD	8050	0%	40%	64%	87%	87%	0
P30-3	E#Op	C_f*4	C_f*2	C_f	$C_f/2$	$C_f/4$	E#Wp
#Op	8	8	8	11	13	14	17
OpUB		9	11	13	14	15	
#Sp	134 700	134 700	134 700	128 400	126 300	125 900	125 450
WpRD	9250	0%	0%	68%	91%	95%	0
P30-4	E#Op	C_f*4	C_f*2	C_f	$C_f/2$	$C_f/4$	E#Wp
#Op	8	8	9	12	13	14	16
OpUB		10	12	13	14	15	
#Sp	177 100	177 100	174 100	165 800	165 000	164 400	163 875
WpRD	13 225	0%	23%	85%	91%	96%	0
P30-5	E#Op	C_f*4	C_f*2	C_f	$C_f/2$	$C_f/4$	E#Wp
#Op	8	8	11	11	13	16	19
OpUB		10	12	14	15	17	
#Sp	187 400	187 400	175 450	175 450	172 900	171 600	171 200
WpRD	16 200	0%	74%	74%	90%	98%	0

Table 8
Ratio C_f/C_t sensitivity on instances with 40 book covers.

P40-1	E#Op	C_f*4	C_f*2	C_f	$C_f/2$	$C_f/4$	E#Wp
#Op	10	10	14	15	18	18	21
OpUB		13	15	17	18	19	
#Sp	313 200	313 200	301 100	299 000	296 700	296 700	296 050
WpRD	17 150	0%	71%	83%	96%	96%	0
P40-2	E#Op	C_f*4	C_f*2	C_f	$C_f/2$	$C_f/4$	E#Wp
#Op	10	10	14	15	17	21	21
OpUB		13	15	17	19	21	
#Sp	263 400	263 400	248 100	246 600	245 000	243 500	243 500
WpRD	19 900	0%	77%	84%	92%	100%	0
P40-3	E#Op	C_f*4	C_f*2	C_f	$C_f/2$	$C_f/4$	E#Wp
#Op	10	10	13	16	18	18	21
OpUB		14	16	17	18	19	
#Sp	311 800	311 800	296 300	289 000	287 600	287 600	287 000
WpRD	24 800	0%	63%	92%	98%	98%	0
P40-4	E#Op	C_f*4	C_f*2	C_f	$C_f/2$	$C_f/4$	E#Wp
#Op	10	10	15	17	18	19	22
OpUB		14	16	18	20	22	
#Sp	401 100	401 100	380 900	377 900	377 100	376 700	375 650
WpRD	25 450	0%	79%	91%	94%	96%	0
P40-5	E#Op	C_f*4	C_f*2	C_f	$C_f/2$	$C_f/4$	E#Wp
#Op	10	11	14	17	17	20	23
OpUB		14	17	18	19	21	
#Sp	375 400	369 600	357 100	35 000	350 000	348 700	348 050
WpRD	27 350	21%	67%	93%	93%	98%	0

In order to have a more complete computational analysis and to test the robustness of the GRASP procedure, several instances of different size will be considered. In particular, the ratio C_f/C_t is a very important parameter in the GRASP procedure, in the construction phase (see line 17 in Fig. 4) and in the local search phase (see line 5 in Fig. 5). Different ratios will also be considered. For these reasons, 15 new instances were generated randomly (five different instances of size $I=30$, 40 and 50). We denote by PI-k the k-th instance of size I (P30-2 is the second instance of size 30). The number of copies to be printed for each book cover are generated as follows:

$$N_1 = 1000$$

$$N_{i+1} = N_i + Random(0,1,\ldots,I) * Random(25,50,75,100,125)$$

We consider five different unit costs of an offset plate, C_f*4, C_f*2, C_f, $C_f/2$, $C_f/4$ and a unique cost of sheet of paper C_t, generating in this way five different ratios C_f/C_t.

For each instance and each unit cost of an offset plate, 10 runs were performed. The parameters of the GRASP procedure are the same as those used in Table 6: MaxIterations $=10\,000$, $d_S=5$, $\alpha=0.5$ and $\beta=0.4$. The best results obtained are summarized in Table 7 ($I=30$), Table 8 ($I=40$) and Table 9 ($I=50$). Column 1 indicates the name of the problem. Column 2 (E#Op) reports the extreme solution that minimizes the total number of sheets of paper printed subject to the constraint that the number of offset plates used is minimal ($=\lceil I/4 \rceil$ the smallest integer number larger than $I/4$). Columns 3–7 indicate the best solutions obtained corresponding to the different unit costs of an offset plate. Column 8 (E#Wp) reports the extreme solution that minimizes the total number of offset plates used subject to the constraint that the minimum number of sheets of paper is printed (no wasted sheets of paper). For each solution, we indicate the number of offset plates used (#Op), an upper bound on the number of offset plates (OpUB), the number of sheets of paper printed (#Sp), the percentage of reduction in the amount of wasted sheets of paper (WpRD) and the total number of wasted sheets of paper produced in the case of the two extreme solutions.

Table 9
Ratio C_f/C_t sensitivity on instances with 50 book covers.

P50-1	E#Op	C_f*4	C_f*2	C_f	$C_f/2$	$C_f/4$	E#Wp
#Op	13	13	16	16	20	22	26
OpUB		16	18	21	22	24	
#Sp	442 000	442 000	428 400	428 400	423 100	422 200	421 300
WpRD	20 700	0%	66%	66%	91%	96%	0
P50-2	E#Op	C_f*4	C_f*2	C_f	$C_f/2$	$C_f/4$	E#Wp
#Op	13	15	15	18	19	23	25
OpUB		16	18	20	22	25	
#Sp	576 800	562 700	562 700	555 400	554 200	552 500	551 800
WpRD	25 000	56%	56%	86%	90%	97%	0
P50-3	E#Op	C_f*4	C_f*2	C_f	$C_f/2$	$C_f/4$	E#Wp
#Op	13	13	19	21	22	23	28
OpUB		18	21	22	24	26	
#Sp	653 050	653 050	672 700	623 400	622 500	621 900	620 700
WpRD	32 350	0%	78%	92%	94%	96%	0
P50-4	E#Op	C_f*4	C_f*2	C_f	$C_f/2$	$C_f/4$	E#Wp
#Op	13	14	20	21	21	26	31
OpUB		19	21	23	25	27	
#Sp	698 800	692 400	667 000	665 500	665 500	663 100	662 625
WpRD	36 175	18%	88%	92%	92%	99%	0
P50-5	E#Op	C_f*4	C_f*2	C_f	$C_f/2$	$C_f/4$	E#Wp
#Op	13	13	21	21	24	24	29
OpUB		20	22	23	26	28	
#Sp	733 700	733 700	697 500	697 500	695 400	695 400	694 000
WpRD	39 700	0%	91%	91%	96%	96%	0

The CPU times are not mentioned because they are in the same order as the CPU times reported in Table 5.

From Table 7–9, it can be seen that:

- If we consider C_f as the cost of an offset plate (as before), the results presented in Table 6 are confirmed. The performance is similar and similar conclusions can be drawn. It is possible to obtain good solutions in acceptable time.

- When the cost of an offset plate is varying between the two extreme values $C_f * 4$ and $C_f/4$, we clearly observe the trade-off between the number of offset plates used and the number of sheets of paper printed (and consequently the amount of wasted sheets of paper produced). This information is very valuable to a decision maker.
- When the cost of an offset plate is fixed to $C_f * 4$, we obtain most of the time 0% reduction in the amount of wasted sheets of paper produced. In this case, it is clearly very difficult to find an assignment of the book covers to the compartments of each offset plate reducing enough the number of wasted sheets of paper produced in comparison with the cost of an offset plate.
- When the cost of an offset plate decreases, additional offset plates are used in the solutions. We observe that the first offset plates added are the most interesting ones at the point of view of reduction of wasted sheets of paper. The problem is clearly non-linear.
  - From C_f*4 to C_f*2, the average percentage reduction of wasted sheets of paper for an additional offset plate is 27.4% for $I=30$, 17.7% for $I=40$ and 13.3% for $I=50$.
  - From C_f*2 to C_f, the average percentage reduction of wasted sheets of paper for an additional offset plate is 21.8% for $I=30$, 8.6% for $I=40$ and 8.0% for $I=50$.
  - From C_f to $C_f/2$, the average percentage reduction of wasted sheets of paper for an additional offset plate is 9.7% for $I=30$, 3.7% for $I=40$ and 4.0% for $I=50$.
  - From $C_f/2$ to $C_f/4$, the average percentage reduction of wasted sheets of paper for an additional offset plate is 3.8% for $I=30$, 1.9% for $I=40$ and 1.8% for $I=50$.
  - From $C_f/4$ to E#Wp, the average percentage reduction of wasted sheets of paper for an additional offset plate is 1.8% for $I=30$, 1.0% for $I=40$ and 0.8% for $I=50$.
- When the cost of an offset plate decreases, we observe also that strong reduction (more than 90%) in the amount of wasted sheets of paper can be obtained. The complete reduction of wasted sheets of paper (extreme solution E#Wp) is very expensive in terms of additional offset plates required.
- The upper bound on the number of offset plates is an indicator of the difficulty of finding an assignment of the book covers to the compartments of each offset plate reducing enough the number of wasted sheets of paper in comparison with the cost of an offset plate. In most cases, the difference between the upper bound and the number of offset plates used is small.

5. Conclusion

In the paper we proposed a greedy random adaptative search procedure to solve the cover printing problem. The problem is hard and hence heuristic methods are needed. The experimental results reveal that the procedure outperforms simulated annealing and genetic algorithms used in previously published results. Taking into account both the quality of solution and the computational requirement, we see that the GRASP procedure is particularly well suited to this problem. Certainly, the proposed method can still be improved, for example by proposing other local search algorithms.

Acknowledgment

We thank anonymous referee for his comments, which helped to improve this paper.

Appendix A. Instance data

Table A1

	$P_1(I=3)$	$P_2(I=4)$	$P_3(I=5)$	$P_4(I=8)$	$P_5(I=12)$	$P_6(I=15)$
N_1	16 000	20 000	135 500	15 000	12 000	15 000
N_2	9000	18 000	114 500	12 000	9000	14 000
N_3	4500	15 000	103 500	10 000	7000	13 000
N_4		8500	94 500	8000	6000	12 000
N_5			84 500	5000	5200	11 000
N_6				3000	4500	10 000
N_7				3000	3000	9000
N_8				3000	2500	8000
N_9					2000	7000
N_{10}					2000	6000
N_{11}					1500	5000
N_{12}					1000	4000
N_{13}						3000
N_{14}						2000
N_{15}						1000

Table A2

$P_7(I=30)$			
N_1	30 000	N_{16}	15 000
N_2	28 000	N_{17}	14 000
N_3	27 000	N_{18}	13 500
N_4	26 000	N_{19}	13 000
N_5	26 000	N_{20}	11 000
N_6	23 000	N_{21}	10 500
N_7	22 000	N_{22}	10 000
N_8	22 000	N_{23}	9000
N_9	20 000	N_{24}	9000
N_{10}	20 000	N_{25}	7500
N_{11}	19 000	N_{26}	6000
N_{12}	18 000	N_{27}	5000
N_{13}	17 000	N_{28}	2500
N_{14}	16 000	N_{29}	1500
N_{15}	15 000	N_{30}	1000

Table A3

P_8 $(I=40)$			
N_1	50 000	N_{26}	10 000
N_2	47 000	N_{27}	10 000
N_3	45 500	N_{28}	9100
N_4	41 500	N_{29}	8000
N_5	37 000	N_{30}	6300
N_6	32 500	N_{31}	5000
N_7	30 500	N_{32}	5000
N_8	29 000	N_{33}	4300
N_9	27 000	N_{34}	4200
N_{10}	27 000	N_{35}	4000
N_{11}	26 700	N_{36}	3000
N_{12}	25 000	N_{37}	2650
N_{13}	22 000	N_{38}	1800
N_{14}	19 000	N_{39}	1100
N_{15}	16 100	N_{40}	700
N_{16}	15 000		
N_{17}	14 000		
N_{18}	13 500		
N_{19}	13 000		
N_{20}	13 000		
N_{21}	12 900		
N_{22}	12 000		
N_{23}	12 000		
N_{24}	11 300		
N_{25}	10 700		

Table A4

P_9 ($I=50$)			
N_1	95 000	N_{26}	30 000
N_2	90 000	N_{27}	30 000
N_3	85 500	N_{28}	28 300
N_4	80 000	N_{29}	28 000
N_5	77 000	N_{30}	26 350
N_6	72 300	N_{31}	25 500
N_7	70 500	N_{32}	22 400
N_8	69 000	N_{33}	21 000
N_9	67 000	N_{34}	21 000
N_{10}	67 000	N_{35}	19 000
N_{11}	61 700	N_{36}	16 050
N_{12}	60 000	N_{37}	14 000
N_{13}	57 650	N_{38}	11 900
N_{14}	55 500	N_{39}	11 000
N_{15}	52 100	N_{40}	10 000
N_{16}	51 000	N_{41}	10 000
N_{17}	50 000	N_{42}	7800
N_{18}	43 500	N_{43}	6000
N_{19}	43 000	N_{44}	4500
N_{20}	39 000	N_{45}	4000
N_{21}	38 900	N_{46}	3000
N_{22}	37 000	N_{47}	2900
N_{23}	36 000	N_{48}	1450
N_{24}	34 300	N_{49}	1000
N_{25}	32 700	N_{50}	750

Appendix B. Optimal solutions

Problem P1 ($I=3$): total cost $= 2 * C_f + 7375 * C_t = 136\,472$.

$$\begin{array}{c}\begin{array}{|c|c|}\hline 1 & 2 \\ \hline 2 & 3 \\ \hline\end{array}\\ (3,2) \\ z_1 = 4500 \\ w_1 = 0\end{array} \quad \begin{array}{c}\begin{array}{|c|c|}\hline 1 & 1 \\ \hline 1 & 1 \\ \hline\end{array}\\ (1,1) \\ z_2 = 2875 \\ w_2 = 0\end{array}$$

Problem P2 ($I=4$): total cost $= 2 * C_f + 15\,666.6667 * C_t = 247\,912$.

$$\begin{array}{c}\begin{array}{|c|c|}\hline 2 & 2 \\ \hline 3 & 4 \\ \hline\end{array}\\ (3,2) \\ z_1 = 9000 \\ w_1 = 125\end{array} \quad \begin{array}{c}\begin{array}{|c|c|}\hline 1 & 1 \\ \hline 1 & 3 \\ \hline\end{array}\\ (2,2) \\ z_2 = 6666.6667 \\ w_2 = 166.6667\end{array}$$

Remark: With $z_2 = 6667$ and $w_2 = 167$, total cost $= 247\,916$ as in [5,22].

Problem P3 ($I=5$): total cost $= 3 * C_f + 133\,625 * C_t = 1\,851\,948$.

$$\begin{array}{c}\begin{array}{|c|c|}\hline 1 & 1 \\ \hline 2 & 3 \\ \hline\end{array}\\ (3,1) \\ z_1 = 67750 \\ w_1 = 0\end{array} \quad \begin{array}{c}\begin{array}{|c|c|}\hline 2 & 4 \\ \hline 4 & 5 \\ \hline\end{array}\\ (3,2) \\ z_2 = 47250 \\ w_2 = 125\end{array} \quad \begin{array}{c}\begin{array}{|c|c|}\hline 3 & 3 \\ \hline 5 & 5 \\ \hline\end{array}\\ (2,2) \\ z_3 = 18625 \\ w_3 = 375\end{array}$$

Problem P4 ($I=8$): total cost $= 3 * C_f + 15\,500 * C_t = 264\,348$.

$$\begin{array}{c}\begin{array}{|c|c|}\hline 1 & 2 \\ \hline 3 & 4 \\ \hline\end{array}\\ (4,2) \\ z_1 = 10000 \\ w_1 = 500\end{array} \quad \begin{array}{c}\begin{array}{|c|c|}\hline 1 & 6 \\ \hline 7 & 8 \\ \hline\end{array}\\ (4,3) \\ z_2 = 3000 \\ w_2 = 0\end{array} \quad \begin{array}{c}\begin{array}{|c|c|}\hline 1 & 2 \\ \hline 5 & 5 \\ \hline\end{array}\\ (3,3) \\ z_3 = 2500 \\ w_3 = 250\end{array}$$

Remark: The optimal solution is not unique.

Appendix C. Best known solutions

Problem P5 ($I=12$): total cost $= 4 * C_f + 14\,500 * C_t = 269\,584$.

$$\begin{array}{c}\begin{array}{|c|c|}\hline 1 & 1 \\ \hline 4 & 5 \\ \hline\end{array}\\ (3,3) \\ z_1 = 6000 \\ w_1 = 200\end{array} \quad \begin{array}{c}\begin{array}{|c|c|}\hline 2 & 2 \\ \hline 3 & 6 \\ \hline\end{array}\\ (3,2) \\ z_2 = 4500 \\ w_2 = 0\end{array} \quad \begin{array}{c}\begin{array}{|c|c|}\hline 3 & 8 \\ \hline 10 & 9 \\ \hline\end{array}\\ (4,4) \\ z_3 = 2500 \\ w_3 = 250\end{array} \quad \begin{array}{c}\begin{array}{|c|c|}\hline 7 & 7 \\ \hline 11 & 12 \\ \hline\end{array}\\ (3,3) \\ z_4 = 1500 \\ w_4 = 125\end{array}$$

Problem P6 ($I=15$): total cost $= 6 * C_f + 30\,000 * C_t = 515\,256$.

$$\begin{array}{c}\begin{array}{|c|c|}\hline 1 & 2 \\ \hline 3 & 4 \\ \hline\end{array}\\ (4,1) \\ z_1 = 12000 \\ w_1 = 0\end{array} \quad \begin{array}{c}\begin{array}{|c|c|}\hline 5 & 6 \\ \hline 7 & 8 \\ \hline\end{array}\\ (4,1) \\ z_2 = 8000 \\ w_2 = 0\end{array} \quad \begin{array}{c}\begin{array}{|c|c|}\hline 9 & 10 \\ \hline 11 & 12 \\ \hline\end{array}\\ (4,1) \\ z_3 = 4000 \\ w_3 = 0\end{array} \quad \begin{array}{c}\begin{array}{|c|c|}\hline 1 & 5 \\ \hline 9 & 13 \\ \hline\end{array}\\ (4,4) \\ z_4 = 3000 \\ w_4 = 0\end{array} \quad \begin{array}{c}\begin{array}{|c|c|}\hline 2 & 6 \\ \hline 10 & 14 \\ \hline\end{array}\\ (4,4) \\ z_5 = 2000 \\ w_5 = 0\end{array}$$

$$\begin{array}{c}\begin{array}{|c|c|}\hline 3 & 7 \\ \hline 11 & 15 \\ \hline\end{array}\\ (4,4) \\ z_6 = 1000 \\ w_6 = 0\end{array}$$

Problem P7 ($I=30$): total cost $= 10 * C_f + 117\,000 * C_t = 1\,759\,240$.

$$\begin{array}{c}\begin{array}{|c|c|}\hline 1 & 3 \\ \hline 4 & 5 \\ \hline\end{array}\\ (4,3) \\ z_1 = 27000 \\ w_1 = 500\end{array} \quad \begin{array}{c}\begin{array}{|c|c|}\hline 2 & 6 \\ \hline 7 & 8 \\ \hline\end{array}\\ (4,3) \\ z_2 = 23000 \\ w_2 = 500\end{array} \quad \begin{array}{c}\begin{array}{|c|c|}\hline 9 & 10 \\ \hline 11 & 12 \\ \hline\end{array}\\ (4,4) \\ z_3 = 20000 \\ w_3 = 750\end{array} \quad \begin{array}{c}\begin{array}{|c|c|}\hline 13 & 14 \\ \hline 15 & 16 \\ \hline\end{array}\\ (4,2) \\ z_4 = 15000 \\ w_4 = 0\end{array} \quad \begin{array}{c}\begin{array}{|c|c|}\hline 17 & 18 \\ \hline 20 & 21 \\ \hline\end{array}\\ (4,2) \\ z_5 = 11000 \\ w_5 = 125\end{array}$$

$$\begin{array}{c}\begin{array}{|c|c|}\hline 19 & 23 \\ \hline 24 & 25 \\ \hline\end{array}\\ (4,3) \\ z_6 = 9000 \\ w_6 = 375\end{array} \quad \begin{array}{c}\begin{array}{|c|c|}\hline 2 & 22 \\ \hline 22 & 27 \\ \hline\end{array}\\ (3,3) \\ z_7 = 5000 \\ w_7 = 0\end{array} \quad \begin{array}{c}\begin{array}{|c|c|}\hline 1 & 26 \\ \hline 26 & 17 \\ \hline\end{array}\\ (3,3) \\ z_8 = 3000 \\ w_8 = 0\end{array} \quad \begin{array}{c}\begin{array}{|c|c|}\hline 13 & 18 \\ \hline 19 & 28 \\ \hline\end{array}\\ (4,3) \\ z_9 = 2500 \\ w_9 = 125\end{array} \quad \begin{array}{c}\begin{array}{|c|c|}\hline 14 & 19 \\ \hline 29 & 30 \\ \hline\end{array}\\ (4,4) \\ z_{10} = 1500 \\ w_{10} = 250\end{array}$$

Problem P8 ($I=40$): total cost $= 14 * C_f + 172\,950 * C_t = 2\,585\,912$.

$$\begin{array}{c}\begin{array}{|c|c|}\hline 1 & 2 \\ \hline 3 & 4 \\ \hline\end{array}\\ (4,2) \\ z_1 = 45500 \\ w_1 = 1000\end{array} \quad \begin{array}{c}\begin{array}{|c|c|}\hline 5 & 6 \\ \hline 7 & 8 \\ \hline\end{array}\\ (4,1) \\ z_2 = 29000 \\ w_2 = 0\end{array} \quad \begin{array}{c}\begin{array}{|c|c|}\hline 9 & 10 \\ \hline 11 & 12 \\ \hline\end{array}\\ (4,4) \\ z_3 = 27000 \\ w_3 = 575\end{array} \quad \begin{array}{c}\begin{array}{|c|c|}\hline 13 & 13 \\ \hline 14 & 15 \\ \hline\end{array}\\ (3,1) \\ z_4 = 11000 \\ w_4 = 0\end{array} \quad \begin{array}{c}\begin{array}{|c|c|}\hline 17 & 18 \\ \hline 19 & 20 \\ \hline\end{array}\\ (4,4) \\ z_5 = 14000 \\ w_5 = 625\end{array}$$

$$\begin{array}{c}\begin{array}{|c|c|}\hline 16 & 21 \\ \hline 22 & 23 \\ \hline\end{array}\\ (4,2) \\ z_6 = 12000 \\ w_6 = 0\end{array} \quad \begin{array}{c}\begin{array}{|c|c|}\hline 24 & 25 \\ \hline 26 & 27 \\ \hline\end{array}\\ (4,2) \\ z_7 = 10000 \\ w_7 = 0\end{array} \quad \begin{array}{c}\begin{array}{|c|c|}\hline 5 & 14 \\ \hline 28 & 29 \\ \hline\end{array}\\ (4,3) \\ z_8 = 8000 \\ w_8 = 0\end{array} \quad \begin{array}{c}\begin{array}{|c|c|}\hline 15 & 30 \\ \hline 31 & 32 \\ \hline\end{array}\\ (4,2) \\ z_9 = 5000 \\ w_9 = 0\end{array} \quad \begin{array}{c}\begin{array}{|c|c|}\hline 1 & 33 \\ \hline 34 & 35 \\ \hline\end{array}\\ (4,4) \\ z_{10} = 4500 \\ w_{10} = 250\end{array}$$

$$\begin{array}{c}\begin{array}{|c|c|}\hline 16 & 36 \\ \hline 37 & 38 \\ \hline\end{array}\\ (4,4) \\ z_{11} = 3000 \\ w_{11} = 387\end{array} \quad \begin{array}{c}\begin{array}{|c|c|}\hline 2 & 6 \\ \hline 6 & 7 \\ \hline\end{array}\\ (3,3) \\ z_{12} = 1750 \\ w_{12} = 125\end{array} \quad \begin{array}{c}\begin{array}{|c|c|}\hline 24 & 28 \\ \hline 30 & 39 \\ \hline\end{array}\\ (4,4) \\ z_{13} = 1300 \\ w_{13} = 100\end{array} \quad \begin{array}{c}\begin{array}{|c|c|}\hline 15 & 21 \\ \hline 25 & 40 \\ \hline\end{array}\\ (4,4) \\ z_{14} = 900 \\ w_{14} = 300\end{array}$$

Problem P9 ($I=50$): total cost $=21*C_f+457\,325*C_t=6\,538\,644$.

1	2
3	4

$(4,1)$
$z_1=80000$
$w_1=0$

5	6
7	8

$(4,1)$
$z_2=69000$
$w_2=0$

9	10
11	12

$(4,1)$
$z_3=60000$
$w_3=0$

14	15
16	17

$(4,2)$
$z_4=51000$
$w_4=250$

18	19
20	21

$(4,2)$
$z_5=39000$
$w_5=25$

22	22
23	23

$(2,2)$
$z_6=18500$
$w_6=250$

24	25
26	27

$(4,2)$
$z_7=30000$
$w_7=0$

13	13
28	29

$(3,2)$
$z_8=28300$
$w_8=75$

31	32
33	34

$(4,2)$
$z_9=21000$
$w_9=0$

1	35
36	37

$(4,2)$
$z_{10}=15000$
$w_{10}=250$

2	30
30	40

$(3,2)$
$z_{11}=10000$
$w_{11}=0$

5	38
41	42

$(4,2)$
$z_{12}=8000$
$w_{12}=50$

9	10
30	39

$(4,3)$
$z_{13}=7000$
$w_{13}=163$

3	14
18	31

$(4,3)$
$z_{14}=4500$
$w_{14}=0$

24	35
39	44

$(4,4)$
$z_{15}=4500$
$w_{15}=300$

6	19
38	45

$(4,4)$
$z_{16}=4000$
$w_{16}=200$

25	43
46	47

$(4,3)$
$z_{17}=3000$
$w_{17}=100$

7	11
43	48

$(4,3)$
$z_{18}=1700$
$w_{18}=113$

15	32
41	43

$(4,3)$
$z_{19}=1400$
$w_{19}=100$

3	13
36	49

$(4,4)$
$z_{20}=1050$
$w_{20}=25$

41	41
50	50

$(2,2)$
$z_{21}=375$
$w_{21}=37$

References

[1] Alvarez-Valdes R, Parreño F, Tamarit JM. Reactive GRASP for the strip-packing problem. Computers & Operations Research 2008;35:1065–83.

[2] Bautista J, Pereira J. A GRASP algorithm to solve the unicost set covering problem. Computers & Operations Research 2007;34:3162–73.

[3] Carrein C. Problème du mariage des couvertures par la méthode tabou. Dissertation. Belgium: Faculté Polytechnique de Mons; 1990.

[4] Cook WJ, Cunningham WH, Pulleyblank WR, Schrijver A. Combinatorial optimization. NY: Wiley; 1998.

[5] Elaoud S, Teghem J, Bouaziz B. Genetic algorithms to solve the cover printing problem. Computers & Operations Research 2007;34:3346–61.

[6] Fasbender G. A branch & price algorithm for the book cover printing problem. Dissertation, Université Libre de Bruxelles; 2000.

[7] Feo TA, Resende MGC. Greedy randomized adaptive search procedures. Journal of Global Optimization 1995;6:109–33.

[8] Feo TA, Resende MGC. A probabilistic heuristic for a computationally difficult set covering problem. Operations Research Letters 1989;8:67–71.

[9] Festa P, Resende MGC. An annotated bibliography. In: Ribeiro CC, Hansen P, editors. Essays and surveys in metaheuristics. Dordrecht: Kluwer Academic Publishers; 2001. p. 325–67.

[10] Fortemps Ph, Ost C, Pirlot M, Teghem J, Tuyttens D. A production scheduling case study in a chemical firm using metaheuristics. International Journal of Production Economics 1996;46–47:13–26.

[11] Glover R, Kockenberger G. Handbook of metaheuristics. Boston: Kluwer Academic Publishers; 2003.

[12] Glover F, Greenberg HJ. New approaches for heuristic search: a bilateral linkage with artificial intelligence. European Journal of Operational Research 1989;39:119–30.

[13] ILOG CPLEX 9.0 documentation. Ilog Cplex Optimization Inc., 2005.

[14] Naya M. Problème du mariage des couvertures posé par la S.A. Casterman. Dissertation. Belgium: Faculté Polytechnique de Mons; 1990.

[15] Osman IH. An introduction to metaheuristics. In: Lawrence M, Wilsdon C, editors. Operational research tutorial papers. Hampshire, UK: Stockton Press, Publication of the Operation Society; 1995. p. 92122.

[16] Osman IH, Laporte G. Meta-heuristics: a bibliography. Annals of Operations Research 1996;63:513–628.

[17] Papadimitriou CH, Steiglitz K. Combinatorial optimization. NJ: Prentice-Hall; 1982.

[18] Pirlot M. General local search heuristics in combinatorial optimization: a tutorial. JORBEL 1992;32(1–2):7–69.

[19] Resende MGC, Ribeiro CC. Greedy randomized adaptative search procedures. In: Glover F, Kockenberger G, editors. Handbook of metaheuristics. Dordrecht: Kluwer Academic Publishers; 2003. p. 219–49.

[20] Resende MGC, Ribeiro CC. GRASP with path-relinking: recent advances and applications. In: Ibaraki T, Nonobe K, Yagiura M, editors. Metaheuristics: progress as real problem solvers. NY: Springer; 2005. p. 29–63.

[21] Ribeiro CC. Une métaheuristique gloutonne et probabiliste. In: Teghem J, Pirlot M, editors. Optimisation approchée en recherche opérationnelle, Hermès, 2002. p. 153–76.

[22] Teghem J, Pirlot M, Antoniadis C. Embedding of linear programming in a simulated annealing algorithm for solving a mixed integer production planning problem. Journal of Computational and Applied Mathematics 1995;64:91–102.

[23] Tuyttens D, Pirlot M, Teghem J. Implémentation des métaheuristiques au travers d'études de cas. In: Teghem J, Pirlot M, editors. Résolution de problèmes de RO par les métaheuristiques, Hermès, 2004. p. 91–126.