FOUNDATIONS OF COMPUTING AND DECISION SCIENCES
Vol. 44 (2019) No. 2
ISSN 0867-6356
DOI: 10.2478/fcds-2019-0010 e-ISSN 2300-3405

Toward an Efficient Resolution for a Single-machine Bi-objective Scheduling Problem with Rejection

Atefeh Moghaddam *, Jacques Teghem, Daniel Tuyttens †,
Farouk Yalaoui, Lionel Amodeo ‡

Abstract. We consider a single-machine bi-objective scheduling problem with rejection. In this problem, it is possible to reject some jobs. Four algorithms are provided to solve this scheduling problem. The two objectives are the total weighted completion time and the total rejection cost. The aim is to determine the set of efficient solutions. Four heuristics are described; they are implicit enumeration algorithms forming a branching tree, each one having two versions according to the root of the tree corresponding either to acceptance or rejection of all the jobs. The algorithms are first illustrated by a didactic example. Then they are compared on a large set of instances of various dimension and their respective performances are analysed.

Keywords: Production scheduling, bi-objective optimization, single-machine, rejection cost.

1. Introduction

The majority of the researches in the scheduling problem literature [2] considered that all the jobs must be scheduled in the workshop. However, there are particular practical situations where all the jobs may not be accepted and thus some of them rejected for specific reasons as congestion of the workshop, impossibility to meet customers requirements or as very long processing time or less importance of a job.

*Siemens Mobility SAS, 150 Avenue de la République, 92323 Chatillon Cedex France, e-mail: atefeh.moghaddam@gmail.com

†Mathematics and Operational Research Unit, University of Mons, Polytechnic Faculty, Belgium, e-mail: {jacques.teghem,daniel.tuyttens}@umons.ac.be

‡Charles Delaunay Institute (ICD-LOSI), University of Technology of Troyes, France, e-mail: {farouk.yalaoui, lionel.amodeo}@utt.fr

In such a case, the rejection of a job induces a penalty which can be considered like an outsourcing cost or an opportunity loss.

To our knowledge, the paper of Bartel et al. in 2000 [1] is the first to introduce the possible rejection of a job. After this initial study, several authors were interested to propose specific situations of scheduling problems with rejection of jobs. Their works are listed in the paper of Moghaddam et al. [8]. The majority of them analyse single-objective problems where the rejection cost is aggregated with a classical scheduling criterion or the rejection cost is limited by a constraint. Researches on the rejection cost as a separate objective in a multi-objective problem is still rare in the literature [9]. That is the reason why we address the specific problem in this paper.

Moreover, except the papers of Moghaddam et al. [6], [7] and [8], even in a bi-objective context, none of these works proposes algorithms to determine the set of efficient solutions. It is also the case in two most recent studies: the one of Wang et al. [11] where the authors introduce a special mechanism to control the processing times and the one of Zhang et al. [12] in which the off-line and the on-line two machines flow-shop scheduling problems with rejection are analysed.

An exception is the paper of Jia et al. [5]: some ant colony optimization (ACO) metaheuristics are proposed to find the Pareto efficient solutions set for the bi-criteria optimization of the makespan and the total rejection cost in a batch scheduling model on parallel machines.

Using the formalism of T'Kindt and Billaut [9], the model analysed in the present paper can be denoted by $1//(C_w, R)$, where C_w is the total weighted completion time of the accepted jobs and R is the total cost of the rejected jobs, and it consists to determine the set of efficient solutions for these two criteria. This model has been first introduced by Moghaddam et al. In [7], various simulated annealing metaheuristics are compared to obtain the efficient solutions based on the Pareto dominance. In [6], these algorithms are also analysed but based on the Lorenz dominance. Both dominance concepts are also considered in [8] where the results of two main metaheuristics, MOSA [10] and NSGAII [3] are compared.

The same model with Pareto dominance is analysed in the present paper, but dedicated implicit enumeration heuristics are developed instead of general metaheuristics. The rest of the paper is organized as follows. In Section 2, we define the problem in more details and we introduce in Section 3 some concepts used in the algorithms. The first four sub-sections of Section 4 describe respectively four implicit enumeration heuristics $H1, H2, H3$ and $H4$, each one having two versions according to the root of the branching tree corresponding either to the acceptance or the rejection of all the jobs. These eight heuristics are illustrated with a didactic example and in the sub-section 4.5, further comparison between them is illustrated. In Section 5, a large set of instances, with the number of jobs varying from 5 till 200, are solved with all the heuristics and a detailed analysis of their respective results is presented. The practical implementation of the algorithms is commented in sub-section 5.4. Section 6 provides a short conclusion of the study.

2. The problem

We consider a single-machine scheduling problem. There are n jobs and each job $j \in \{1, \ldots, n\}$ is characterized by its processing time p_j and its own weight w_j.

A first objective is to minimize the total weighted completion time C_w, for which it is well known [2] that an optimal solution consists to schedule the jobs in the increasing order of the ratio p_j/w_j. For this reason, we assume that the n jobs are initially ranked in this order:

$$p_1/w_1 \ \leq \ p_2/w_2 \ \leq \ \ldots \ \leq \ p_n/w_n. \tag{1}$$

Nevertheless, it is possible to decide not to schedule all the jobs and thus to reject some of them. If the job $j \in \{1, \ldots, n\}$ is rejected, a rejection cost or penalty r_j is incurred. A second objective R is to minimize the total rejection cost i.e. the sum of the penalties r_j of the rejected jobs. A solution s is characterized by $A \subseteq \{1, \ldots, n\}$ the set of rejected jobs and by $\overline{A} = \{1, \ldots, n\} \setminus A$ the set of accepted jobs scheduled in the order (1).

The two considered objectives to minimize are thus

$$C_w(s) \ = \sum_{j \in \overline{A}} w_j \ C_j \tag{2}$$

$$R(s) \ = \sum_{j \in A} r_j \tag{3}$$

where C_j is the completion time of job $j \in \overline{A}$ in the optimal schedule of the jobs in subset $\overline{A}$ obtained by the order of relation (1).

Let us note that as

$$C_j \ = \sum_{\substack{k \in \overline{A} \\ k \leq j}} p_k \tag{4}$$

relation (2) can be written equivalently

$$C_w(s) \ = \sum_{j \in \overline{A}} p_j \ (\sum_{\substack{k \in \overline{A} \\ k \geq j}} w_k) \tag{5}$$

For this bi-objective scheduling problem, our aim is to determine (or to approximate) the set of efficient solutions and the corresponding Pareto front.

We recall [9] that a feasible solution s is efficient if it does not exist any other feasible solution s' such that

$$C_w(s') \ \leq \ C_w(s)$$
$$R(s') \ \leq \ R(s)$$

with at least one strict inequality.

We denote by E the set of efficient solutions. The points $\{(C_w(s), R(s)) \ |s \in E\}$ correspond to the non-dominated points in the bi-objective space and form the so called Pareto front.

Obviously the two extreme efficient solutions, denoted by s_a and s_b, correspond respectively to $A = \emptyset$ and $A = \{1, \ldots, n\}$, giving the two extreme points of the Pareto front $(\overline{C}_w = C_w(s_a) = \sum_{j=1}^{n} w_j \; C_j, \underline{R} = R(s_a) = 0)$ and $(\underline{C}_w = C_w(s_b) = 0, \overline{R} = R(s_b) = \sum_{j=1}^{n} r_j)$. For all efficient solutions $s \in E$, we have thus

$$\underline{C}_w = C_w(s_b) = 0 \;\leq\; C_w(s) \;\leq\; C_w(s_a) = \overline{C}_w$$

$$\underline{R} = R(s_a) = 0 \;\leq R(s) \;\leq\; R(s_b) = \overline{R}$$

3. Terminology

3.1. Potential efficient solution

In the algorithms proposed in this paper, in a certain iteration, a solution is called a "potential efficient solution" if it is not dominated by any other solutions.

The set of potential efficient solutions is denoted by PE. Each time new solutions are generated, the list PE is updated.

3.2. Algorithms of type a and type b

A solution s is of level k, if the number of rejected jobs is equal to k, i.e. if $|A| = k$. The algorithms in Section 4 are implicit enumeration forming a tree. An algorithm is said of type a, if the root of the tree is the solution s_a (i.e. the unique solution of level 0) and the branching scheme increases progressively the considered level, creating children solutions at level $k + 1$ by rejecting an additional job from a parent solution at level k.

An algorithm is said of type b, if the root of the tree is the solution s_b (i.e. the unique solution of level n) and the branching scheme decreases progressively the considered level, creating children solutions at level $k - 1$ by the acceptance of an additional job from a parent solution at level k.

For both types, the considered children solutions of a parent solution s will be denoted by $E(s)$.

3.3. Children nodes generated by non-dominated variations

3.3.1. Algorithm of type a

At each level, the branching scheme creating $E(s)$ is based on a dominance relation defined by decreasing the first objective value C_w and increasing the second objective value R.

$E(s_a)$ at level 1

The parent of this solution is the initial solution s_a at level 0 with $C_w(s_a) = \overline{C}_w$ and $R(s_a) = 0$.

If a child solution s_1 at level 1 is characterized by $A = \{j\}$, using relation (5), it is easy to see that

$$\begin{aligned} C_w(s_1) &= \overline{C}_w - \Delta_j^{(1)} \\ R(s_1) &= 0 + r_j \end{aligned} \tag{6}$$

where

$$\Delta_j^{(1)} = p_j \; (\sum_{k=j}^{n} w_k) + w_j \; (\sum_{l=1}^{j-1} p_l) \tag{7}$$

Definition: Dominance relation of variation $(\Delta_j^{(1)}, r_j)$

In an algorithm of type a, the variation $(\Delta_j^{(1)}, r_j)$ is dominated if there exists another job $k \in \{1, \ldots, n\} \setminus \{j\}$ such that

$$\begin{aligned} \Delta_k^{(1)} &\geq \Delta_j^{(1)} \\ r_k &\leq r_j \end{aligned} \tag{8}$$

with at least one strict inequality.

In such a case, by rejecting job k, it is possible to have a larger decrease of the objective C_w at a smaller increase of the objective R. It is clear that the non-dominated variations are the only onces able to possibly generate an efficient solution at level 1.

So the branching scheme will only generate the set of children solutions $E(s_a)$ with each solution s_1 corresponding to a non-dominated variation $(\Delta_j^{(1)}, r_j)$.

At this first iteration, all the solutions $E(s_a)$ are potential efficient solutions and there are all assigned in the list PE.

$E(s)$ at level $k = 2, \ldots, n$

From a parent solution s_{k-1}, with $A = \{j_1, \ldots, j_{k-1}\}$ at level $k-1$, an additional job j can be rejected to build a child solution s_k with $A = \{j_1, \ldots, j_{k-1}, j\}$.

Using relation (5), the variation of the two objectives is iteratively determined:

$$\begin{aligned} C_w(s_k) &= C_w(s_{k-1}) - \Delta_j^{(k)} \\ R(s_k) &= R(s_{k-1}) + r_j \end{aligned} \tag{9}$$

where

$$\begin{aligned} \Delta_j^{(k)} &= \Delta_j^{(k-1)} - p_j \; w_{j_{k-1}} \quad \text{if} \quad j < j_{k-1} \\ \Delta_j^{(k)} &= \Delta_j^{(k-1)} - p_{j_{k-1}} \; w_j \quad \text{if} \quad j > j_{k-1} \end{aligned} \tag{10}$$

The non-dominated variations $(\Delta_j^{(k)}, r_j)$ are the only ones able to possibly generate potential efficient solutions s_k. So the branching scheme will only generate the set of children $E(s_{k-1})$ with each solution s_k corresponding to a non-dominated variation $(\Delta_j^{(k)}, r_j)$.

Remark 1: At each level k, the set of non-dominated variations $(\Delta_j^{(k)}, r_j)$ is always non-empty, containing at least the job j with the smallest r_j and the job j with the largest $\Delta_j^{(k)}$. If it is the same job, then there exists only one non-dominated variation.

3.3.2. Algorithm of type b

When an algorithm of type b is considered, with the root s_b at level n, with $C_w(s_b) = 0$ and $R(s_b) = \overline{R}$, at each level the branching scheme creating $E(s)$ consists to schedule an additional job and is also based on a dominance relation defined by the increase of the first objective C_w and the decrease of the second objective R.

The concept is similar and the equations (6),(7),(8),(9) and (10), are replaced by the following equations (11),(12),(15),(13) and (14).

$$\begin{array}{rcl} C_w(s_{n-1}) & = & \Delta_j^{(n-1)} \\ R(s_{n-1}) & = & \overline{R} - r_j \end{array} \tag{11}$$

where

$$\Delta_j^{(n-1)} = p_j \; w_j \tag{12}$$

$$\begin{array}{rcl} C_w(s_k) & = & C_w(s_{k+1}) + \Delta_j^{(k)} \\ R(s_k) & = & R(s_{k+1}) - r_j \end{array} \tag{13}$$

where

$$\begin{array}{rcllll} \Delta_j^{(k)} & = & \Delta_j^{(k+1)} + p_j \; w_{j_{k-1}} & \text{if} & j < j_{k-1} \\ \Delta_j^{(k)} & = & \Delta_j^{(k+1)} + p_{j_{k-1}} \; w_j & \text{if} & j > j_{k-1} \end{array} \tag{14}$$

with s_{k+1} characterized by $\overline{A} = \{j_1, \ldots, j_{k-1}\}$ and s_k by $\overline{A} = \{j_1, \ldots, j_{k-1}, j\}$.

The dominance relation of variation $(\Delta_j^{(1)}, r_j)$ could be defined as following:

$$\begin{array}{rcl} \Delta_k^{(n-1)} & \leq & \Delta_j^{(n-1)} \\ r_k & \geq & r_j \end{array} \tag{15}$$

with at least one strict inequality.

As for the algorithm of type a, the branching scheme will only generate the set of children $E(s_{k+1})$ with each solution s_k corresponding to a non-dominated variation $(\Delta_j^{(k)}, r_j)$.

4. Algorithms $H1$,$H2$,$H3$ and $H4$

Each of these four algorithms can be of type a or type b. In these algorithms, we denote by P the pool of solutions which must be still considered by the branching scheme. These algorithms will be illustrated on the following didactic example with 5 jobs (already in the order of increasing ratio p_j/w_j).

Table 1: Data of the didactic example

j	1	2	3	4	5
p_j	7	5	10	8	20
w_j	8	4	6	4	2
r_j	5	1	5	3	2

4.1. Algorithms $H1$

4.1.1. $H1a$

In this algorithm, to describe a solution we will assign a status to each job with one of the three possibilities:

status	0	scheduled job free to be rejected,
status	-1	definitively rejected job,
status	1	definitively scheduled job.

The initial solution is $s_a = (0, \ldots, 0)$ and assigned to the pool P. The solutions of P are ranked in the increasing order of the objective R; in case of equality of this objective, the solutions are sorted in the order of their generation (see remark 2 below). When a new solution is examined by the branching scheme, the first solution of P is chosen and this solution is removed. The algorithm comes to the end when $P = \emptyset$.

When a set of children solutions is created, they are compared with those of PE. Only the new potential solutions are assigned to the pool P, the others are fathomed in the tree, i.e. they are no more considered for the branching scheme.

The aim of the status "1" of a job is to avoid non-empty intersections between the sets of solutions corresponding to the different non-fathomed branches of the tree. For instance at the first iteration of $H1a$ on the didactic example where the children solutions consist to reject one of the three jobs $2, 4$ or 3, these three children solutions will be $(0, -1, 0, 0, 0), (0, 1, 0, -1, 0)$ and $(0, 1, -1, 1, 0)$.

The rejection of job 2, considered in the first branch, will no more be examined in the second and third branches, and the rejection of job 4, considered in the second branch, will no more be examined in the third branch.

Table 2: Iterations of algorithm $H1a$ on the didactic example

It	s	$E(s)$	(R, C_w)	Level	P	PE
0		$s_0 =$ (0,0,0,0,0)	(0,456)	0	s_0	s_0
1	s_0	$s_1 =$ (0,-1,0,0,0) $s_3 =$ (0,1,0,-1,0) $s_5 =$ (0,1,-1,1,0)	(1,348) (3,320) (5,264)	1 1 1	s_1, s_3 s_5	s_0, s_1, s_3, s_5
2	s_1	$s_3^2 =$ (0,-1,0,0,-1) $s_4 =$ (0,-1,0,-1,1) $s_6 =$ (0,-1,-1,1,1)	(3,258) (4,232) (6,186)	2 2 2	s_3, s_3^2 s_4, s_5 s_6	s_0, s_1, s_3^2, s_4 s_6
3	s_3	$s_5^2 =$ (0,1,0,-1,-1) $s_8 =$ (0,1,-1,-1,0)	(5,236) (8,168)	$2 \prec s_4$ 2	s_3^2, s_4 s_5, s_6 s_8	s_0, s_1, s_3^2, s_4 s_6, s_8
4	s_3^2	$s_6^2 =$ (0,-1,0,-1,-1) $s_8^2 =$ (0,-1,-1,1,-1)	(6,158) (8,116)	3 3	s_4, s_5 s_6, s_6^2 s_8, s_8^2	s_0, s_1, s_3^2, s_4 s_6^2, s_8^2
5	s_4	$s_9 =$ (0,-1,-1,-1,1)	(9,110)	3	s_5, s_6 s_6^2, s_8 s_8^2, s_9	s_0, s_1, s_3^2, s_4 s_6^2, s_8^2, s_9
6	s_5	$s_7 =$ (0,1,-1,1,-1) $s_{10} =$ (-1,1,-1,1,0)	(7,184) (10,138)	$2 \prec s_6^2$ $2 \prec s_9$	s_6, s_6^2 s_8, s_8^2 s_9	s_0, s_1, s_3^2, s_4 s_6^2, s_8^2, s_9
7	s_6	$s_{11} =$ (-1,-1,-1,1,1)	(11,88)	3	s_6^2, s_8 s_8^2, s_9	s_0, s_1, s_3^2, s_4 $s_6^2, s_8^2, s_9, s_{11}$
8	s_6^2	$s_{11}^2 =$ (0,-1,-1,-1,-1)	(11,56)	4	s_8, s_8^2 s_9, s_{11}^2	s_0, s_1, s_3^2, s_4 $s_6^2, s_8^2, s_9, s_{11}^2$
9	s_8	$s_{10}^2 =$ (0,1,-1,-1,-1) $s_{13} =$ (-1,1,-1,-1,0)	(10,104) (13,70)	3 $3 \prec s_{11}^2$	s_8^2, s_9 s_{10}^2, s_{11}^2	s_0, s_1, s_3^2, s_4 $s_6^2, s_8^2, s_9, s_{10}^2$ s_{11}^2
10	s_8^2	$s_{13}^2 =$ (-1,-1,-1,1,-1)	(13,32)	4	s_9, s_{10}^2 s_{11}^2	s_0, s_1, s_3^2, s_4 $s_6^2, s_8^2, s_9, s_{10}^2$ s_{11}^2, s_{13}^2
11	s_9	$s_{14} =$ (-1,-1,-1,-1,1)	(14,40)	$4 \prec s_{13}^2$	s_{10}^2, s_{11}^2	s_0, s_1, s_3^2, s_4 $s_6^2, s_8^2, s_9, s_{10}^2$ s_{11}^2, s_{13}^2
12	s_{10}^2	$s_{15} =$ (-1,1,-1,-1,-1)	(15,20)	4	s_{11}^2	s_0, s_1, s_3^2, s_4 $s_6^2, s_8^2, s_9, s_{10}^2$ $s_{11}^2, s_{13}^2, s_{15}$
13	s_{11}^2	$s_{16} =$ (-1,-1,-1,-1,-1)	(16,0)	5		s_0, s_1, s_3^2, s_4 $s_6^2, s_8^2, s_9, s_{10}^2$ $s_{11}^2, s_{13}^2, s_{15}, s_{16}$

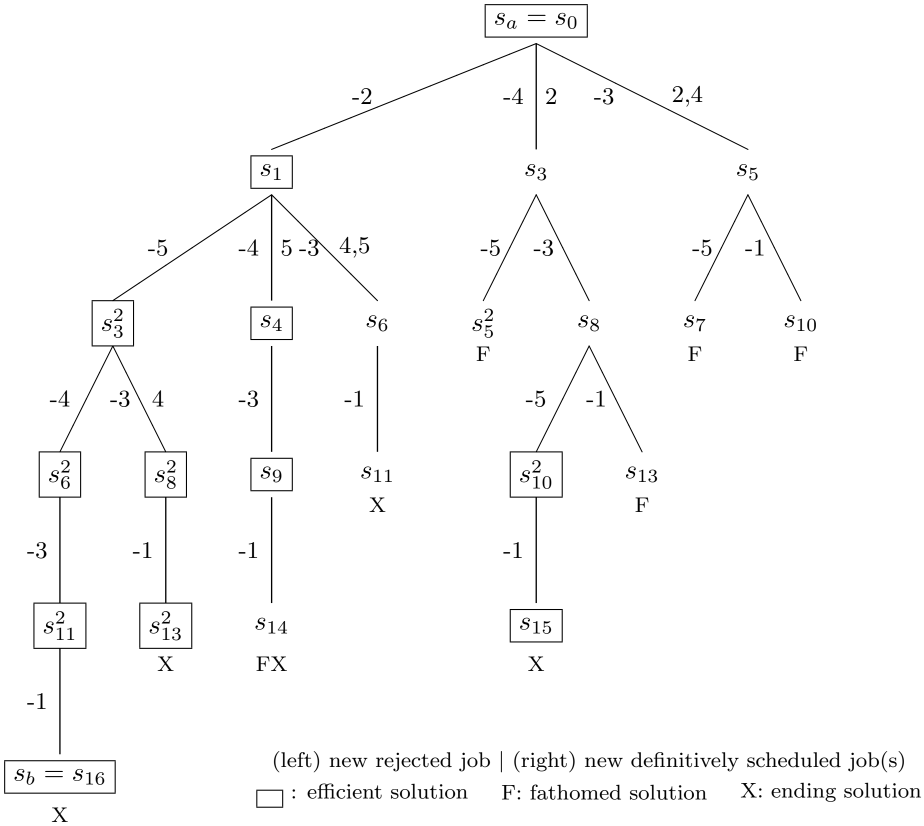


Figure 1: Branching scheme of $H1a$

For a solution with no job with the status "0", this solution is an ending solution of the branch. Table 2 describes the successive iterations of the algorithm $H1a$ on the didactic example. Figure 1 represents the corresponding branching scheme. The twelve efficient solutions of this instance are generated. For this didactic example, the subindex of a solution is the value for the objective R; in case of identical value, the upper index indicates the order of their generation.

Remark 2: An impact of the order of the solutions in the pool P

Let us suppose that inside the pool P, the solutions with identical value of objective R are ordered in the increasing order of objective C_w in contrary to the order of their generation. Table 3 indicates what happens for the didactic example with this change. The solution s_3^2, with $C_w(s_3^2) = 258$, will be treated before the solution s_3, with $C_w(s_3) = 320$. So the solution s_8^2, a child solution of s_3^2, will be generated before the solution s_8, a child solution of s_3, and as s_8 is dominated by s_8^2, the solution s_8 will now be fathomed and no more included in the pool P. The consequence is (see iteration 9 of Table 2 and also Figure 1) that it will be impossible to still generate the efficient solution s_{10}^2 and so the efficient solution s_{15}, a child of s_{10}^2 (see iteration

12 of Table 2).

So it appears that with such a change in the order to treat the solutions of the pool, only 10 efficient solutions will be generated instead of 12. We have seen that this case can happen in several instances and it is the reason why we have chosen to order the solutions of the pool as indicated before.

Table 3: Impact of the lexicographic order

It	s	$E(s)$	(R, C_w)	Level	P	PE
2	s_1	$s_3^2 =$ (0,-1,0,0,-1)	(3,258)	2	$\boxed{s_3^2}, \boxed{s_3}, s_4, s_5$	s_0, s_1, s_3^2, s_4
		$s_4 =$ (0,-1,0,-1,1)	(4,232)	2	s_6	s_6
		$s_6 =$ (0,-1,-1,1,1)	(6,186)	2		
3	$\boxed{s_3^2}$	$s_6^2 =$ (0,-1,0,-1,-1)	(6,158)	3	s_3, s_4, s_5, s_6^2	s_0, s_1, s_3^2, s_4
		$s_8^2 =$ (0,-1,-1,1,-1)	(8,116)	3	s_6, s_8^2	s_6^2, s_8^2
4	$\boxed{s_3}$	$s_5^2 =$ (0,1,0,-1,-1)	(5,236)	$2 \prec s_4$	s_4, s_5, s_6^2, s_6	s_0, s_1, s_3^2, s_4
		$s_8 =$ (0,1,-1,-1,0)	(8,168)	$2\ \boxed{\prec s_8^2}$	s_8^2	s_6^2, s_8^2

4.1.2. $H1b$

Algorithm $H1b$ is similar to algorithm $H1a$. The unique difference is that this algorithm is of type b so that the initial solution is thus the extreme efficient solution s_b, initializing the pool P.

There are two logical modifications in comparison to $H1a$. First, the solutions of P are ranked in the decreasing order of the objective R; in case of equality of this objective, the solutions are sorted according to their generation. Nevertheless, the management of the pool P is identical to the one of $H1a$ and thus the dominated children solutions are not placed in the pool P.

Secondly, in the description of a solution, the three following status of a job are used:

status	0	rejected job free to be scheduled,
status	-1	definitively rejected job,
status	1	definitively scheduled job,

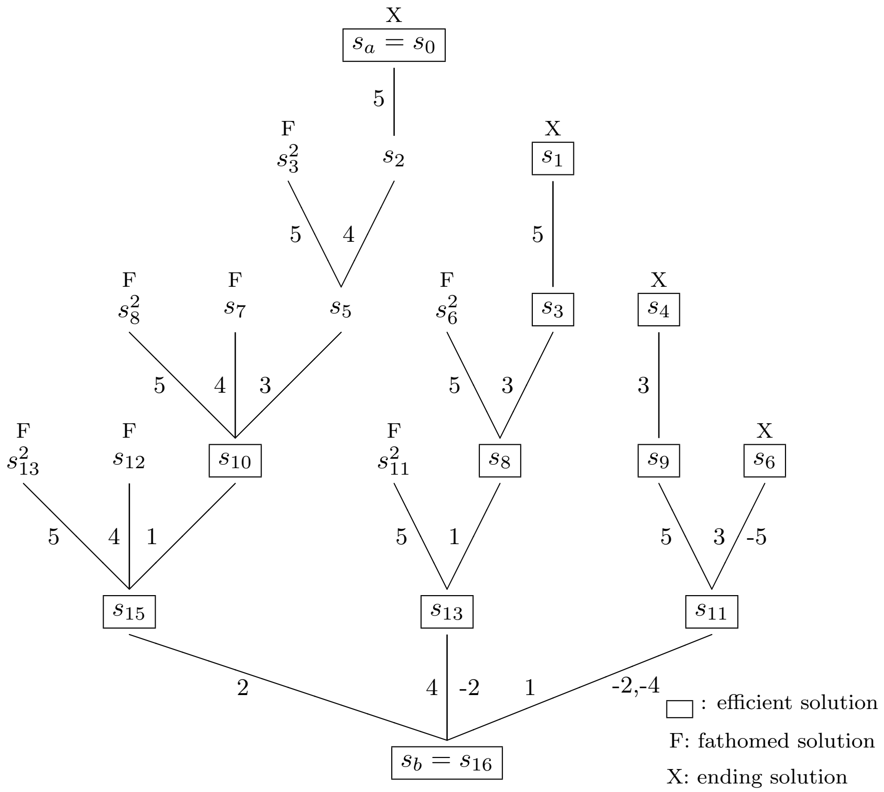
so that $s_b = (0, \ldots, 0)$.

But the aim of the status ”-1” is also to avoid non-empty intersections between the sets of solutions corresponding to the non-fathomed different branches of the tree.

Table 4 describes the successive iterations of the algorithm $H1b$ on the didactic example. Figure 2 represents the corresponding branching scheme. The twelve efficient solutions of this instance are again generated even if the way to obtain those solutions in Figure 1 is quite different to the one in Figure 2.

Table 4: Iterations of algorithm $H1b$ on the didactic example

It	s	$E(s)$	(R, C_w)	Level	P	PE
0		s_{16} = (0,0,0,0,0)	(16,0)	5	s_{16}	s_{16}
1	s_{16}	s_{15} = (0,1,0,0,0) s_{13} = (0,-1,0,1,0) s_{11} = (1,-1,0,-1,0)	(15,20) (13,32) (11,56)	4 4 4	s_{15}, s_{13}, s_{11}	$s_{11}, s_{13}, s_{15}, s_{16}$
2	s_{15}	s_{13}^2 = (0,1,0,0,1) s_{12} = (0,1,0,1,0) s_{10} = (1,1,0,0,0)	(13,70) (12,72) (10,104)	$3 \prec s_{13}$ $3 \prec s_{11}$ 3	s_{13}, s_{11}, s_{10}	$s_{10}, s_{11}, s_{13}, s_{15}$ s_{16}
3	s_{13}	s_{11}^2 = (0,-1,0,1,1) s_8 = (1,-1,0,1,0)	(11,88) (8,116)	$3 \prec s_{11}$ 3	s_{11}, s_{10}, s_8	$s_8, s_{10}, s_{11}, s_{13}$ s_{15}, s_{16}
4	s_{11}	s_9 = (1,-1,0,-1,1) s_6 = (1,-1,1,-1,-1)	(9,110) (6,158)	3 3	s_{10}, s_9, s_8	s_6, s_8, s_9, s_{10} $s_{11}, s_{13}, s_{15}, s_{16}$
5	s_{10}	s_8^2 = (1,1,0,0,1) s_7 = (1,1,0,1,0) s_5 = (1,1,1,0,0)	(8,168) (7,184) (5,236)	$2 \prec s_8$ $2 \prec s_6$ 2	s_9, s_8, s_5	s_5, s_6, s_8, s_9 $s_{10}, s_{11}, s_{13}, s_{15}$ s_{16}
6	s_9	s_4 = (1,-1,1,-1,1)	(4,232)	2	s_8, s_5	s_4, s_6, s_8, s_9 $s_{10}, s_{11}, s_{13}, s_{15}$ s_{16}
7	s_8	s_6^2 = (1,-1,0,1,1) s_3 = (1,-1,1,1,0)	(6,186) (3,258)	$2 \prec s_6$ 2	s_5, s_3	s_3, s_4, s_6, s_8 $s_9, s_{10}, s_{11}, s_{13}$ s_{15}, s_{16}
8	s_5	s_3^2 = (1,1,1,0,1) s_2 = (1,1,1,1,0)	(3,320) (2,356)	$1 \prec s_3$ 1	s_3, s_2	s_2, s_3, s_4, s_6 s_8, s_9, s_{10}, s_{11} s_{13}, s_{15}, s_{16}
9	s_3	s_1 = (1,-1,1,1,1)	(1,348)	1	s_2	s_1, s_3, s_4, s_6 s_8, s_9, s_{10}, s_{11} s_{13}, s_{15}, s_{16}
10	s_2	s_0 = (1,1,1,1,1)	(0,456)	0		s_0, s_1, s_3, s_4 s_6, s_8, s_9, s_{10} $s_{11}, s_{13}, s_{15}, s_{16}$



(left) new scheduled job | (right) new definitively rejected job(s)

Figure 2: Branching scheme of $H1b$

4.1.3. $H1ab$

As we will see in the numerical experiments of Section 5, the two sets PE obtained by $H1a$ and $H1b$, can be different, contrary to what happens in the didactic example. If these two sets are filtered by pairwise comparisons of the solutions, this filtering will be denoted by $H1ab$. Thus only the non-dominated solutions present in the union of these two sets are saved.

4.2. Algorithms $H2$

4.2.1. $H2a$

The algorithm $H2a$ presents a minor modification comparing to the algorithm $H1a$. As we will see in the numerical experiments (and see also above-mentioned remark 1), an efficient solution can be missed in the case where all the children solutions of the same parent solution are dominated and thus fathomed and not placed in the pool P

of solutions still to be examined.

It is the reason why in such a situation, in $H2a$ one of the dominated children solutions is kept in the pool P. To choose which solution to retain in the pool, different criteria are possible. For $H2a$, we decide to keep the dominated child solution with the minimum value of R in the pool. Another possibility investigated is to keep the solution corresponding to the minimum ratio r_j/Δ_j. In this case, the other dominated children solutions are fathomed.

In Table 5, an example of modification due to $H2a$ comparing to Table 2 for $H1a$ is given. As the two children solutions s_7 and s_{10} are dominated, the solution s_7 is kept in the pool.

Table 5: Algorithm $H2a$ when all children solutions are dominated solutions

It	s	$E(s)$	(R, C_w)	Level	P	PE
6	s_5	$s_7 = (0, 1, -1, 1, -1)$	(7,184)	$2 \prec s_6^2$	$s_6, s_6^2, \boxed{s_7}, s_8$	s_0, s_1, s_3^2, s_4
		$s_{10} = (-1, 1, -1, 1, 0)$	(10,138)	$2 \prec s_9$	s_8^2, s_9	s_6^2, s_8^2, s_9

4.2.2. $H2b$

In $H1b$ a similar modification that the one made in $H1a$ is introduced. When all the children solutions of the same parent solution are dominated, one of them is kept in the pool P: the dominated child solution with the maximal value of R is assigned to the pool. The other dominated children solutions are fathomed.

4.2.3. $H2ab$

In this algorithm, the two sets of PE obtained with $H2a$ and $H2b$ are filtered, keeping only the non-dominated solutions in the union of these two sets.

4.3. Algorithms $H3$

4.3.1. $H3a$

The principle of this algorithm is identical to the one $H1a$. The unique difference is that in $H3a$ the status "1" of a job is not considered. Thus there are only two possibilities for the status of a job:

status	0	scheduled job free to be rejected,
status	-1	definitively rejected job.

In $H1a$, a job j with the status "1" is no more considered to be rejected in the branching scheme so that its variation (Δ_j, r_j) is not taken into consideration. As a consequence, the set of non-dominated variations can be influenced by this fact. In $H3a$, all the variations for the jobs with "0" status are analyzed. This mechanism helps generating new branches, which can lead to efficient solutions ignored by $H1a$.

Of course, another consequence is that the intersection of the sets of solutions in non-fathomed branches are no more empty. The algorithm simply ignores redundant solutions, which have been generated in another branch.

The numerical experiments of Section 5 will prove that such a situation appears in the case of large-size instances.

Table 6 illustrates the first three iterations of $H3a$. We see that the non-dominated variations of iteration 3 are now obtained for jobs 2 and 3, providing the solutions s_4 (already obtained) and s_8 instead in $H1a$ (see Table 2) for jobs 5 and 3 providing the solutions s_5^2 and s_8.

Table 6: Three first iterations of algorithm $H3a$ on the didactic example

It	s	$E(s)$	(R, C_w)	Level	P	PE
0		$s_0 = (0,0,0,0,0)$	(0,456)	0	s_0	s_0
1	s_0	$s_1 = (0,-1,0,0,0)$	(1,348)	1	s_1, s_3, s_5	s_0, s_1, s_3, s_5
		$s_3 = (0,\boxed{0},0,-1,0)$	(3,320)	1		
		$s_5 = (0,\boxed{0},-1,\boxed{0},0)$	(5,264)	1		
2	s_1	$s_3^2 = (0,-1,0,0,-1)$	(3,258)	2	s_3, s_3^2, s_4, s_5	s_0, s_1, s_3^2, s_4
		$s_4 = (0,-1,0,-1,\boxed{0})$	(4,232)	2	s_6	s_6
		$s_6 = (0,-1,-1,\boxed{0},\boxed{0})$	(6,186)	2		
3	s_3	$s_4 = (0,-1,0,-1,0)$	(4,232)	2 $\boxed{= s_4}$	s_3^2, s_4, s_5, s_6	s_0, s_1, s_3^2, s_4
		$s_8 = (0,0,-1,-1,0)$	(8,168)	2	s_8	s_6, s_8

4.3.2. $H3b$

A similar modification is made in $H1b$, i.e. the status "-1" of a job is no more considered, keeping only the two status:

status	0	rejected job free to be scheduled,
status	1	definitively scheduled job.

The consequences of this modification are identical to those described for $H3a$.

4.3.3. $H3ab$

The result of this algorithm is the filtering of the two sets $H3a$ and $H3b$.

4.4. Algorithms $H4$

4.4.1. $H4a$

In this algorithm, as in $H3a$, we do not consider the status "1" of a job, keeping only the status "0" and "-1". But here the status "-1" means "rejected job free to be scheduled" but no more "definitively rejected job". This change differs the $H4a$ algorithm to $H3, H1$ and $H2$ by one main element: if the solution parent is of level k, the possible children solutions will be of level $k-1$ or of level $k+1$. Effectively, the considered variations consist to change successively the status of each job; thus:

- the status "0" becomes "-1" generating solutions of level $k+1$,
- the status "-1" becomes "0" generating solutions of level $k-1$.

Among these n variations, as in the three preceding algorithms, only the non-dominated variations are kept to define $E(s)$. Redundant solutions are ignored as they were in $H3$ algorithm. Nevertheless, Table 7 illustrates this new branching scheme on the iterations 4 and 5 of Table 2 for the didactic example.

Table 7: First five iterations of algorithm $H4a$ on the didactic example

It	s	$E(s)$	(R, C_w)	Level	P	PE
0		$s_0 = (0,0,0,0,0)$	(0,456)	0	s_0	s_0
1	s_0	$s_1 = (0, \boxed{\text{-1}}, 0, 0, 0)$	(1,348)	1	s_1, s_3, s_5	s_0, s_1, s_3, s_5
		$s_3 = (0,0,0, \boxed{\text{-1}}, 0)$	(3,320)	1		
		$s_5 = (0,0, \boxed{\text{-1}}, 0, 0)$	(5,264)	1		
2	s_1	$s_3^2 = (0,-1,0,0, \boxed{\text{-1}})$	(3,258)	2	s_3, s_3^2, s_4, s_5	s_0, s_1, s_3^2, s_4
		$s_4 = (0,-1,0, \boxed{\text{-1}}, 0)$	(4,232)	2	s_6	s_6
		$s_6 = (0,-1, \boxed{\text{-1}}, 0, 0)$	(6,186)	2		
3	s_3	$s_4 = (0,-1,0,-1,0)$	(4,232)	2 $\boxed{= s_4}$	s_3^2, s_4, s_5, s_6	s_0, s_1, s_3^2, s_4
		$s_8 = (0,0,-1,-1,0)$	(8,168)	2	s_8	s_6, s_8
4	s_3^2	$s_2 = (0, \boxed{0}, 0, 0, -1)$	(2,356)	1 $\prec s_1$	s_4, s_5, s_6, s_6^2	s_0, s_1, s_3^2, s_4
		$s_6^2 = (0,-1,0, \boxed{\text{-1}}, -1)$	(6,158)	3	s_8, s_8^2	s_6^2, s_8^2
		$s_8^2 = (0,-1, \boxed{\text{-1}}, 0, -1)$	(8,116)	3		
5	s_4	$s_3 = (0, \boxed{0}, 0, 0, -1)$	(3,320)	1 $\boxed{= s_3}$	s_5, s_6, s_6^2, s_8	s_0, s_1, s_3^2, s_4
		$s_6^2 = (0,-1,0,-1, \boxed{\text{-1}})$	(6,158)	3 $\boxed{= s_6^2}$	s_8^2, s_9	s_6^2, s_8^2, s_9
		$s_9 = (0,-1, \boxed{\text{-1}}, -1, 0)$	(9,110)	3		

In the next section, the positive impact of $H4$ on two larger instances is presented.

4.4.2. $H4b$

By changing the initial efficient solution, algorithm $H4b$ can also be developed. Nevertheless, for all the instances treated in Section 5, the results of $H4b$ will be systematically the same as those of $H4a$. For this reason, in the following, we will only mention $H4$.

4.5. Illustrations

In this section, we will analyse the impact of the algorithms using the first two instances of Section 5 presenting some differences between the results of the algorithms. The data of these two instances 12 (13-3) and 13 (15-1), are given in Table 8 and Table 9 in the Annex.

Instance 12 (13-3)

This instance with 13 jobs has 41 efficient solutions. They are all generated by the algorithms $H1b, H2a, H2b, H3b$ and $H4$. $H1a$ provides only 39 efficient solutions (see Table 10 in the Annex); the two missing solutions are:

- s_1 at level 8, with rejected jobs 1,4,5,6,7,8,11 and 12,
- s_2 at level 11, with rejected jobs 1,2,3,4,5,7,8,10,11,12 and 13.

$H3a$ provides 40 efficient solutions (see Table 14 in the Annex); s_2 is the missing solution.

Remarks on the observations made.

$H1a$ **versus** $H2a$

In $H1a$, at parent solution at level 6 corresponding to the successive rejects of jobs 7,12,5,4,8,6 and the job 2 with status "1", all the children solutions are dominated so that it is impossible to obtain s_1 corresponding to the reject of the additional jobs 1 and 11.

But with $H2a$, the dominated child solution with the minimal rejection cost is kept in the pool; and during the two next iterations these two jobs are successively rejected providing s_1.

Similarly in $H1a$, at parent solution at level 9 corresponding to the successive rejects of jobs 7,12,5,4,8,2,11,1,10 and the job 6 with status "1", all the children solutions are dominated and it is impossible to obtain s_2 corresponding to the reject of the additional jobs 3 and 13.

Again, with $H2a$, the dominated child solution with the minimal rejection cost is kept in the pool. This allows to reject jobs 3 and 13 during the two next iterations providing s_2.

$H3a$ **versus** $H4$

In $H3a$, from the same parent solution at level 9 that was in $H1a$, it is impossible to obtain s_2.

It is useful to note that s_2 at level 11 is generated with $H4$ from the parent solution at level 12 corresponding to the successive rejects of jobs 7,12,5,4,8,2,11,6,1,13,10,3, changing the status "-1" of job 6 to status "0".

Instance 13 (15-1)

This instance with 15 jobs has 66 efficient solutions. They are all provided by algorithms $H1b, H2b, H3b$ and $H4$. But $H1a, H2a$ and $H3a$ generate only 65 efficient solutions (see Table 10, Table 12 and Table 14).

$H3a$ **versus** $H4$

It is important to note that the missing efficient solution at level 11, corresponding to the reject of jobs 1,3,4,5,7,8,9,10,11,13 and 14 is obtained by $H4$ as child solution of a solution at level 12, changing the status "-1" of job 12 to status "0", despite that this job was initially rejected already at level 3.

5. Numerical experiments

5.1. The data

To evaluate the performance of the different algorithms described in Section 4, we use the instances proposed in [8]. These instances are randomly generated within predefined intervals as follows:

- the processing times p_j: random integer numbers within [10,80],
- the weight values w_j: random integer numbers within [1,30],
- the rejection costs r_j: $exp(5+\sqrt{a}\ b)$ where a is a random integer number within [1,80] and b is a random number within [0,1].

The number of jobs is fixed to 21 values:

- small instances with $n \in \{5, 7, 9, 13, 15, 20\}$,

- medium instances with $n \in \{25, 30, 35, 40, 45, 50\}$,
- larger instances with $n \in \{60, 70, 80, 90, 100, 125, 150, 175, 200\}$.

Three instances are generated for each value of n so that 63 instances are solved.

5.2. The results

The complete results are presented in Table 10, Table 12, Table 14 and Table 16 (see the Annex). In these tables we use the following notations.

$|PE(H)|$ the number of solutions obtained by algorithm H
and for the comparison between two algorithms H and $\bar{H}$:

- $C(H, \bar{H}) = |\{x \in PE(H) \bigcap PE(\bar{H})\}|$ is the number of common solutions in $PE(H)$ and $PE(\bar{H})$,
- $D(H, \bar{H}) = |\{x \in PE(H) \mid \exists y \in PE(\bar{H}), y$ dominating $x\}|$ is the number of solutions in $PE(H)$ dominated by a solution of $PE(\bar{H})$,
- $ND(H, \bar{H}) = |PE(H)| - C(H, \bar{H}) - D(H, \bar{H})$ is the number of solutions in $PE(H)$ not present in $PE(\bar{H})$ and not dominated by any solution of $PE(\bar{H})$.

One can find that, $H4$ is by so far the most performant algorithm: no of the solutions obtained by $H4$ is dominated by any solution generated with another algorithm H; moreover all the solutions generated by any H are either found by $H4$ or dominated by a solution of $H4$. We have thus

$$ND(H, H4) = D(H4, H) = 0 \ \ \forall H.$$

For this reason the results of all the algorithms will be compared with those of $H4$.

In Table 10 till Table 16, the first two columns characterize the instance:

(1) index of the 63 instances

(2) name of the instance ($n - x$, with $x \in \{1, 2, 3\}$)

Table 10 is dedicated to the algorithm $H1$ and its comparison with $H4$. Columns 3 till 9, present the comparison of the results of $H1a$ and $H1b$:

(3) $|PE(H1a)|$

(4) $D(H1a, H1b)$

(5) $ND(H1a, H1b)$

(6) $C(H1a, H1b)$

(7) $ND(H1b, H1a)$

(8) $D(H1b, H1a)$

(9) $|PE(H1b)|$

We have $(3) = (4) + (5) + (6)$ and $(9) = (6) + (7) + (8)$. Columns 10 to 12 present the comparison of the results of $H1a$ and $H4$:

(10) $D(H1a, H4)$

(11) $C(H1a, H4)$

(12) $ND(H4, H1a)$

We have $|PE(H1a)| = (10) + (11)$, $|PE(H4)| = (11) + (12)$ (as $ND(H1a, H4) = 0$ and $D(H4, H1a) = 0$).

In the same way, columns 13 to 15 present the comparison of the results of $H1b$ and $H4$ and columns 16 to 18 present the comparison of the results of $H1ab$ and $H4$.

With the same structure of Table 10, Table 12 and Table 14 consider respectively the comparison of $H2$ and $H3$ with the algorithm $H4$.

Table 16 analyses the performance of the different algorithms, indicating the ratio

$$\%(H, H4) \ = \ \frac{C(H, H4)}{|PE(H4)|}$$

i.e. the percentage of the solutions of $H4$ generated by algorithm H.

This percentage is given in columns 5, 7 and 9 for $H1a$, $H1b$ and $H1ab$, in columns 10, 12 and 14 for $H2a$, $H2b$ and $H2ab$, and in columns 15, 17 and 19 for $H3a$, $H3b$ and $H3ab$. Table 16 gives also the CPU time for the different algorithms, in columns 6 and 8 for $H1a$ and $H1b$, columns 11 and 13 for $H2a$ and $H2b$, columns 16 and 18 for $H3a$ and $H3b$. Column 3 gives the number of solutions obtained with algorithm $H4$ and column 4 gives the CPU time for algorithm $H4$. Only the CPU times greater or equal to 1 second are reported.

Let us note that the filtering process-time to obtain the result of an algorithm Hab is almost immediate.

Finally, Table 18 presents a more synthetic view of all these results, gathering the instances by three successive values of n. These values are indicated in column 1 and correspond to 9 instances. The nine other columns give the average value of $\%(H, H4)$ in these instances, respectively for the 9 algorithms from $H1a$ till $H3ab$. The last line of Table 18 resumes all the results by the average value of $\%(H, H4)$ for all the instances.

Remark 3:

In [8], all these instances are solved to also generate PE, with two metaheuristics: MOSA [10] and NSGAII [3]. Unfortunately, it appears impossible to compare our results with those of [8] for the following reasons:

- with NSGAII, the number of solutions in PE is always limited to 100,

- for MOSA, the parameters are not well fixed because they do not dynamically vary in function of the dimension of the instances.

So in the results of [8] the average numbers of solutions in PE for the instances with $n = 200$, are respectively equal to 87 with MOSA and 100 with NSGAII, to compare with the thousands solutions in PE in the results of Table 10, Table 12 and Table 14.

5.3. Analysis of the results

5.3.1. $H3$ versus $H4$

First of all, Table 14 and Table 16 indicate that the two algorithms $H3a$ and especially $H3b$ give excellent results according to $H4$, with very large ratios $\%(H3a, H4)$ and $\%(H3b, H4)$. But it is remarkable that $D(H3ab, H4) = ND(H4, H3ab) = 0$ so that $\%(H3ab, H4) = 100$ for all the instances. $H3ab$ and $H4$ produce thus the same sets $PE(H3ab)$ and $PE(H4)$. Is it the exact set of efficient solutions ? It is an open question to further research. In any case, we take thus this set, denoted by $\hat{E}$, as a reference set to compare the different algorithms.

5.3.2. The small-size instances ($n \leq 20$)

For this set of instances, it has been possible to verify that $\hat{E}$ is the exact set of efficient solutions by a complete enumeration of all the feasible solutions.

For the 18 small instances, even if the results of the 3 algorithms $H1a$, $H2a$ and $H3a$ are quite good, nevertheless sometimes they miss some solutions of $\hat{E}$; it is the case for 5 instances (12-13-16-17-18) for $H1a$ and $H3a$ and for 3 instances (13-16-18) with $H2a$ (see Table 16). But it is not the case for the versions $H1b$, $H2b$ and $H3b$ which generate exactly $\hat{E}$ (see Table 16). In these instances, it appears that the missing solutions of the versions a, are located at a high level of the branching scheme and for this reason, they are more easily obtained with the versions b.

5.3.3. The medium-size instances ($25 \leq n \leq 50$)

Often, the versions b of the algorithms produce a larger percentage of $\hat{E}$ than the corresponding versions a, but it is not always the case. Effectively, in particular for $H1$, we have opposite results: for example, with instance 27, we have $\%(H1b, H4) = 98.9$ however only $\%(H1a, H4) = 54.1$ otherwise with the instance 36 the results are different, $\%(H1b, H4) = 42.0$ instead $\%(H1a, H4) = 95.4$. The respective performance of the versions a and b depends on the data of each instance. Nevertheless, for these medium-size instances, the results of $H1ab$ are excellent with an average percentage $\%(H1ab, H4)$ larger than 99.83% (see Table 18).

It is even the case when either the result of $H1a$ is bad (see the instance 27 in

Table 16 where $\%(H1ab, H4) = 99.3$) or the result of $H1b$ is bad (see the instance 36 in Table 16 $\%(H1ab, H4) = 99.0$).

We can note that $H2$ becomes a little bit more competitive than $H1$ for these medium-size instances, and $H3$ is more competitive than $H2$.

The version b of $H2$ and $H3$ are almost always more performant than their version a, with some rare exceptions like instances 27, 35 and 36 for $H2$ and instance 27 for $H3$ (see Table 16). But, as with $H3$, we have always $\%(H2ab, H4) = 100$ so that for the medium instances $PE(H2ab) = \hat{E}$.

5.3.4. The large-size instances ($60 \leq n \leq 200$)

For the large instances, clearly algorithms $H1a$ and $H1b$ become inadequate to obtain a good approximation of $\hat{E}$, with small ratios $\%(H1a, H4)$ and $\%(H1b, H4)$. Table 10 shows also that the numbers $D(H1a, H4)$ and $D(H1b, H4)$ are very large, especially when $n \geq 150$ (with rare exception like $D(H1a, H4)$ for instance 59); it appears that these algorithms produce thousand solutions which in fact are dominated.

It should be underlined that in many of these large instances, we have only $C(H1a, H1b) = 2$ (see Table 10). These two unique common solutions in $PE(H1a)$ and $PE(H1b)$ are in fact the two extreme efficient solutions corresponding to rejecting none of the jobs or rejecting all of them. Except these two obvious solutions, $H1a$ and $H1b$ completely diverge to generate $PE(H1a)$ and $PE(H1b)$.

Even if the results of $H2$ improve those of $H1$, we note (see Table 12) that $D(H2a, H4)$ and $D(H2b, H4)$ increase with the dimension of the instances to reach several thousands when $n \geq 150$. So these algorithms do not provide a good approximation of $\hat{E}$ for instances with very large dimension.

For such instances, there is a clear superiority of $H3$ versus $H2$. The performance of $H3$ remains excellent and very stable with, in particular of $H3b$, a ratio $\%(H3b, H4)$ larger than 99.1 (see Table 16). We recall that for all the instances $\%(H3ab, H4) = 100$.

5.3.5. The CPU time

Concerning the CPU time of all the different algorithms, we can see from Table 16 (columns 4,6,8,11,13,16 and 18) that:

- for $H1$ and $H3$, it is always inferior to 2 seconds, even when $n = 200$,
- logically, $H2$ needs a larger CPU time till 35 seconds for $H2a$ and 16 seconds for $H2b$ when $n = 200$,
- $H4$ takes 19 seconds to treat the instances with $n = 200$, thus a larger time than $H3a$ and $H3b$. We must note that the filtering process of their respective set PE to obtain $PE(H3ab)$ is almost immediate.

In conclusion of this analysis, $H4$ appears the more efficient and stable algorithm to approximate E even if $H3ab$ appears as an excellent alternative, a little faster.

5.4. Comment on the implementation of the algorithms

The algorithms presented in the paper are all implicit enumeration. At each iteration, a solution s (related to a node of the tree) is selected from the pool and is examined by the branching scheme. The set $E(s)$ of children solutions of the parent solution s is created with only solutions corresponding to a non-dominated variation. For the solution s, the variations corresponding to the two objectives C_w and R can be efficiently obtained using relation (9) and relation (10). Nevertheless relation (10) is computationally very efficient only when the vector of variations $\Delta^{()}$ can also be stored in the pool. For information, to treat the instances with $n = 200$, the maximal size of pool is equal to 1600 for $H1$, $H3$ and $H4$, and is equal to 8000 for $H2$. In case of limited memory (very large scale problems), we can proceed differently and these variations can still be calculated efficiently using relation (16) which is a generalization of relation (7).

$$\Delta_j^{(k)} \;=\; p_j \;(\sum_{\substack{k\in\overline{A} \\ k\geq j}} w_k) \;+\; w_j \;(\sum_{\substack{l\in\overline{A} \\ l<j}} p_l) \tag{16}$$

The verification of the non-dominance of the variations can be done efficiently when considering these variations in the order of increasing rejection cost. For this reason, an additional vector is created $(i_1, i_2, \ldots, i_n)$ corresponding to the increasing order $r_{i_1} \leq r_{i_2} \leq \ldots \leq r_{i_n}$. For the didactic example given in Table 1, we have that $r_2 \leq r_5 \leq r_4 \leq r_3 \leq r_1$ so that we create the additional vector $(2, 5, 4, 3, 1)$. We compare the successive variations $(r_{j_k}, \Delta^{()}_{j_k})$ and $(r_{j_k+1}, \Delta^{()}_{j_k+1})$. These variations are non-dominated if $\Delta^{()}_{j_k} < \Delta^{()}_{j_k+1}$.

6. Conclusion

For the bi-objective model $1//(C_w, R)$ several heuristics which are implicit enumeration following a branching scheme are proposed. They all use the concept of non-dominated variations to define the set $E(s)$ of children solutions of a parent solution s. These heuristics differ either by the definition of fathomed node or by the branching scheme. The large set of experimental results proofs that the heuristic $H1$ and $H2$ are not well adapted to large instances; secondly the superiority of the heuristic $H4$, even if $H3ab$ obtains the same results.

For the future, one can first analyse if some metaheuristics are able to improve the results; of course to propose an exact method is also a challenge. It will also be interesting, in the spirit of [4], [6] and [8], to analyse if the concept of non-dominated variations can be extended to the Lorenz domination variations instead of the Pareto domination. Another direction of further research is to adapt these algorithms to other classical scheduling models, like two parallel machines or two sequential machines, with rejection.

References

[1] Y. Bartal, S. Leonardi, A. Marchetti-Spaccamela, J. Sgall, L. Stougie, "Multiprocessor scheduling with rejection", *Journal of Discrete Mathematics* 13(1), 64–78, (2000).

[2] J. Blazewicz, K.H. Ecker, E. Pesch, G. Schmidt, J. Weglarz, "Handbook on scheduling : From theory to applications", Springer Berlin Heidelberg New York (2007).

[3] K. Deb, S. S. Agrawal, A. Pratap, T. Meyarivan, "A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II", *Lecture Notes in Computer Sciences*, 1917, 849–858, (2000).

[4] F. Dugardin, F. Yalaoui, L. Amodeo "New multi-objective method to solve reentrant hybrid flow shop scheduling problem", *European Journal of Operational Research* 203(1), 22–31, (2010).

[5] Z.-H. Jia, M.-L. Pei, J.Y.-T. Leung "Multi-objective ACO algorithms to minimize the makespan and the total rejection cost on BPMs with arbitrary job weights", *International Journal of Systems Science* 48 (16), 3542–3557, (2017).

[6] A. Moghaddam, F. Yalaoui, L. Amodeo, "Lorenz versus Pareto dominance in a single machine scheduling problem with rejection", *Lecture Notes in Computer Science* 6576, 520–534, (2011).

[7] A. Moghaddam, L. Amodeo, F. Yalaoui, B. Karimi "Single machine scheduling with rejection minimizing total weighted completion time and rejection cost", *International Journal of Applied Evolutionary Computation* 3 (2), 42–61, (2012).

[8] A. Moghaddam, F. Yalaoui, L. Amodeo, "Efficient meta-heuristics based on various dominance criteria for a single-machine bi-criteria scheduling problem with rejection", *Journal of Manufacturing Systems* 34(C), 12–22, (2015).

[9] V. T'Kindt, J.-Ch. Billaut, "Multicriteria scheduling. Theory, Models and Algorithms", Springer-Verlag Berlin Heidelberg New York (2002).

[10] E.L. Ulungu, J. Teghem, Ph. Fortemps, D. Tuyttens, "MOSA method : A tool for solving multiobjective combinatorial optimization problems", *Journal of Multi-Criteria Decision Analysis* 8(4), 221–236, (1999).

[11] D.-J. Wang, Y. Yin, M. Liu, "Bicriteria scheduling problems involving job rejection, controllable processing times and rate-modifying activity", *International Journal of Production Research* 54(12), 3691–3705, (2016).

[12] L. Zhang, L. Lu, S. Li, "New results on two-machine flow-shop scheduling with rejection", *Journal of Combinatorial Optimization* 31 (4), 1493–1504, (2016).

Received 4.10.2018, Accepted 13.11.2018

Annex: Instance data and complete results

Table 8: Data of instance 12

j	p_j	w_j	r_j
1	18	22	1602
2	14	16	592
3	26	18	15738
4	40	21	432
5	51	26	408
6	23	8	601
7	76	26	149
8	80	27	809
9	16	4	4017
10	69	15	5879
11	74	16	1053
12	41	4	282
13	63	4	1652

Table 9: Data of instance 13

j	p_j	w_j	r_j
1	11	25	9097
2	26	29	39771
3	53	28	1960
4	73	27	2372
5	44	13	11277
6	22	6	20548
7	42	11	2476
8	75	19	1838
9	77	13	427
10	62	9	17825
11	53	7	2273
12	41	5	383
13	66	6	2227
14	64	2	175
15	68	1	3782

Table 10: Results $H1$ (Part I)

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)
	n	$\|PE(H1a)\|$	D	ND	$C(H1a, H1b)$	ND	D	$\|PE(H1b)\|$
1	5-1	11	0	0	11	0	0	11
2	5-2	8	0	0	8	0	0	8
3	5-3	7	0	0	7	0	0	7
4	7-1	20	0	0	20	0	0	20
5	7-2	17	0	0	17	0	0	17
6	7-3	16	0	0	16	0	0	16
7	9-1	22	0	0	22	0	0	22
8	9-2	26	0	0	26	0	0	26
9	9-3	26	0	0	26	0	0	26
10	13-1	30	0	0	30	0	0	30
11	13-2	41	0	0	41	0	0	41
12	13-3	39	0	0	39	2	0	41
13	15-1	65	0	0	65	1	0	66
14	15-2	39	0	0	39	0	0	39
15	15-3	81	0	0	81	0	0	81
16	20-1	124	1	0	123	3	0	126
17	20-2	93	0	0	93	1	0	94
18	20-3	96	21	0	75	53	0	128
19	25-1	144	0	0	144	5	0	149
20	25-2	108	0	0	108	6	0	114
21	25-3	156	0	3	153	3	2	158
22	30-1	218	0	6	212	5	1	218
23	30-2	133	1	0	132	7	0	139
24	30-3	221	0	0	221	7	0	228
25	35-1	288	0	24	264	17	14	295
26	35-2	242	1	0	241	33	0	274
27	35-3	192	39	1	152	128	0	280
28	40-1	298	0	0	298	7	0	305
29	40-2	292	0	74	218	12	49	279
30	40-3	296	0	0	296	18	0	314
31	45-1	290	0	2	288	6	0	294
32	45-2	352	0	79	273	7	47	327
33	45-3	348	0	54	294	12	36	342
34	50-1	374	2	123	249	4	107	360
35	50-2	401	2	177	222	12	156	390
36	50-3	559	0	334	225	21	290	536
37	60-1	526	1	148	377	6	56	439
38	60-2	481	317	0	164	335	0	499
39	60-3	485	14	0	471	76	0	547
40	70-1	622	5	10	607	72	0	679
41	70-2	708	621	85	2	677	97	776
42	70-3	776	69	602	105	76	495	676
43	80-1	983	903	0	80	926	0	1006
44	80-2	676	1	531	144	6	506	656
45	80-3	839	22	622	195	98	535	828
46	90-1	970	542	426	2	558	406	966
47	90-2	854	741	93	20	869	94	983
48	90-3	1083	265	647	171	445	638	1254
49	100-1	1400	18	241	1141	47	127	1315
50	100-2	1037	412	623	2	440	613	1055
51	100-3	1190	21	641	528	60	512	1100
52	125-1	1597	17	981	599	76	790	1465
53	125-2	1687	1401	284	2	1601	347	1950
54	125-3	1691	3	1128	560	82	1072	1714
55	150-1	3245	1071	2172	2	784	1670	2456
56	150-2	2262	306	1954	2	589	2074	2665
57	150-3	2322	1615	705	2	1988	714	2704
58	175-1	3054	1477	1575	2	1484	1571	3057
59	175-2	3317	2	3198	117	16	3011	3144
60	175-3	3377	1032	2343	2	1103	2273	3378
61	200-1	3608	1762	1844	2	1833	1961	3796
62	200-2	3860	2488	1370	2	2775	1361	4138
63	200-3	3888	1924	1962	2	2210	1949	4161

Table 11: Results $H1$ (Part II)

(10) D	(11) $C(H1a, H4)$	(12) ND	(13) D	(14) $C(H1b, H4)$	(15) ND	(16) D	(17) $C(H1ab, H4)$	(18) ND
0	11	0	0	11	0	0	11	0
0	8	0	0	8	0	0	8	0
0	7	0	0	7	0	0	7	0
0	20	0	0	20	0	0	20	0
0	17	0	0	17	0	0	17	0
0	16	0	0	16	0	0	16	0
0	22	0	0	22	0	0	22	0
0	26	0	0	26	0	0	26	0
0	26	0	0	26	0	0	26	0
0	30	0	0	30	0	0	30	0
0	41	0	0	41	0	0	41	0
0	39	2	0	41	0	0	41	0
0	65	1	0	66	0	0	66	0
0	39	0	0	39	0	0	39	0
0	81	0	0	81	0	0	81	0
1	123	3	0	126	0	0	126	0
0	93	1	0	94	0	0	94	0
21	75	53	0	128	0	0	128	0
0	144	5	0	149	0	0	149	0
0	108	6	0	114	0	0	114	0
0	156	3	2	156	3	0	159	0
0	218	5	1	217	6	0	223	0
1	132	7	0	139	0	0	139	0
0	221	7	0	228	0	0	228	0
0	288	17	14	281	24	0	305	0
1	241	33	0	274	0	0	274	0
39	153	130	0	280	3	0	281	2
0	298	7	0	305	0	0	305	0
0	292	12	49	230	74	0	304	0
0	296	18	0	314	0	0	314	0
0	290	6	0	294	2	0	296	0
0	352	7	47	280	79	0	359	0
0	348	12	36	306	54	0	360	0
2	372	5	107	253	124	0	376	1
3	398	13	156	234	177	1	410	1
0	559	27	290	246	340	0	580	6
1	525	9	56	383	151	0	531	3
317	164	341	2	497	8	2	497	8
14	471	76	0	547	0	0	547	0
5	617	72	0	679	10	0	689	0
638	70	760	162	614	216	82	682	148
72	704	86	495	181	609	3	780	10
903	80	930	2	1004	6	2	1004	6
1	675	13	507	149	539	1	680	8
25	814	125	535	293	646	3	912	27
727	243	783	484	482	544	263	723	303
764	90	899	210	773	216	121	861	128
374	709	669	740	514	864	190	1073	305
18	1382	53	127	1188	247	0	1429	6
875	162	1010	765	290	882	615	450	722
23	1167	78	512	588	657	2	1227	18
25	1572	125	806	659	1038	20	1636	61
1529	158	1922	993	957	1123	774	1113	967
38	1653	128	1078	636	1145	35	1735	46
2021	1224	1587	1943	513	2298	1223	1735	1076
1756	506	2582	2335	330	2758	1711	834	2254
1762	560	2832	1407	1297	2095	840	1855	1537
2556	498	2860	2329	728	2630	1837	1224	2134
133	3184	314	3011	133	3365	131	3200	298
3238	139	3474	2962	416	3197	2895	553	3060
3116	492	3340	2738	1058	2774	2131	1548	2284
3500	360	3914	2577	1561	2713	2228	1919	2355
3341	547	3877	3324	837	3587	2792	1382	3042

Table 12: Results $H2$ (Part I)

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)
	n	$\|PE(H2a)\|$	D	ND	$C(H2a, H2b)$	ND	D	$\|PE(H2b)\|$
1	5-1	11	0	0	11	0	0	11
2	5-2	8	0	0	8	0	0	8
3	5-3	7	0	0	7	0	0	7
4	7-1	20	0	0	20	0	0	20
5	7-2	17	0	0	17	0	0	17
6	7-3	16	0	0	16	0	0	16
7	9-1	22	0	0	22	0	0	22
8	9-2	26	0	0	26	0	0	26
9	9-3	26	0	0	26	0	0	26
10	13-1	30	0	0	30	0	0	30
11	13-2	41	0	0	41	0	0	41
12	13-3	41	0	0	41	0	0	41
13	15-1	65	0	0	65	1	0	66
14	15-2	39	0	0	39	0	0	39
15	15-3	81	0	0	81	0	0	81
16	20-1	124	1	0	123	3	0	126
17	20-2	94	0	0	94	0	0	94
18	20-3	123	0	0	123	5	0	128
19	25-1	144	0	0	144	5	0	149
20	25-2	114	0	0	114	0	0	114
21	25-3	158	1	0	157	2	0	159
22	30-1	218	0	3	215	5	1	221
23	30-2	136	2	0	134	5	0	139
24	30-3	226	0	0	226	2	0	228
25	35-1	300	0	2	298	5	1	304
26	35-2	258	0	0	258	16	0	274
27	35-3	220	7	0	213	70	0	283
28	40-1	303	0	0	303	2	0	305
29	40-2	293	0	0	293	11	0	304
30	40-3	298	0	0	298	16	0	314
31	45-1	293	0	2	291	3	0	294
32	45-2	355	1	5	349	5	0	354
33	45-3	353	1	0	352	8	0	360
34	50-1	374	2	9	363	5	2	370
35	50-2	405	3	37	365	9	24	398
36	50-3	565	2	82	481	23	46	550
37	60-1	527	1	3	523	8	0	531
38	60-2	485	68	0	417	88	0	505
39	60-3	516	12	0	504	43	0	547
40	70-1	622	5	7	610	72	0	682
41	70-2	786	14	69	703	53	37	793
42	70-3	775	43	24	708	57	9	774
43	80-1	768	333	2	433	574	0	1 007
44	80-2	678	1	3	674	11	3	688
45	80-3	845	23	296	526	108	217	851
46	90-1	964	475	84	405	494	53	952
47	90-2	900	49	7	844	144	4	992
48	90-3	1089	323	2	764	613	0	1 377
49	100-1	1410	13	438	959	32	378	1369
50	100-2	1056	63	34	959	176	10	1145
51	100-3	1203	4	20	1179	42	5	1226
52	125-1	1632	14	389	1229	51	354	1634
53	125-2	1415	937	0	478	1572	0	2050
54	125-3	1730	10	665	1055	58	621	1734
55	150-1	2631	715	889	1027	722	725	2474
56	150-2	2209	630	1176	403	1235	1197	2835
57	150-3	2222	1379	698	145	1858	707	2710
58	175-1	2833	2044	123	666	2344	93	3103
59	175-2	3317	29	1934	1354	92	1861	3307
60	175-3	3173	2700	244	229	2950	213	3392
61	200-1	3566	1986	1466	114	2116	1421	3651
62	200-2	3778	2675	690	413	3092	638	4143
63	200-3	3443	2319	8	1116	3253	3	4372

Table 13: Results $H2$ (Part II)

(10)	(11)	(12)	(13)	(14)	(15)	(16)	(17)	(18)
D	$C(H2a, H4)$	ND	D	$C(H2b, H4)$	ND	D	$C(H2ab, H4)$	ND
0	11	0	0	11	0	0	11	0
0	8	0	0	8	0	0	8	0
0	7	0	0	7	0	0	7	0
0	20	0	0	20	0	0	20	0
0	17	0	0	17	0	0	17	0
0	16	0	0	16	0	0	16	0
0	22	0	0	22	0	0	22	0
0	26	0	0	26	0	0	26	0
0	26	0	0	26	0	0	26	0
0	30	0	0	30	0	0	30	0
0	41	0	0	41	0	0	41	0
0	41	0	0	41	0	0	41	0
0	65	1	0	66	0	0	66	0
0	39	0	0	39	0	0	39	0
0	81	0	0	81	0	0	81	0
1	123	3	0	126	0	0	126	0
0	94	0	0	94	0	0	94	0
0	123	5	0	128	0	0	128	0
0	144	5	0	149	0	0	149	0
0	114	0	0	114	0	0	114	0
1	157	2	0	159	0	0	159	0
0	218	5	1	220	3	0	223	0
2	134	5	0	139	0	0	139	0
0	226	2	0	228	0	0	228	0
0	300	5	1	303	2	0	305	0
0	258	16	0	274	0	0	274	0
7	213	70	0	283	0	0	283	0
0	303	2	0	305	0	0	305	0
0	293	11	0	304	0	0	304	0
0	298	16	0	314	0	0	314	0
0	293	3	0	294	2	0	296	0
1	354	5	0	354	5	0	359	0
1	352	8	0	360	0	0	360	0
2	372	5	2	368	9	0	377	0
3	402	9	24	374	37	0	411	0
2	563	23	46	504	82	0	586	0
1	526	8	0	531	3	0	534	0
68	417	88	0	505	0	0	505	0
12	504	43	0	547	0	0	547	0
5	617	72	0	682	7	0	689	0
16	770	60	41	752	78	4	821	9
43	732	58	9	765	25	0	789	1
333	435	575	0	1 007	3	0	1 009	1
1	677	11	3	685	3	0	688	0
25	820	119	217	634	305	2	928	11
546	418	608	136	816	210	154	829	197
105	795	194	90	902	87	86	909	80
325	764	614	5	1 372	6	5	1 374	4
17	1393	42	380	989	446	4	1425	10
63	993	179	11	1134	38	1	1168	4
5	1198	47	5	1221	24	1	1240	5
18	1614	83	354	1280	417	4	1665	32
938	477	1603	35	2015	65	35	2015	65
38	1692	89	625	1109	672	28	1750	31
784	1847	964	769	1705	1106	83	2555	256
1367	842	2246	1963	872	2216	1499	1315	1773
1528	694	2698	1403	1307	2085	843	1858	1534
2198	635	2723	861	2242	1116	888	2245	1113
133	3184	314	1868	1439	2059	106	3274	224
2887	286	3327	887	2505	1108	861	2562	1051
3062	504	3328	2409	1242	2590	2064	1632	2200
3362	416	3858	1920	2223	2051	1969	2226	2048
2322	1121	3303	61	4311	113	58	4319	105

Table 14: Results $H3$ (Part I)

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)
	n	$\|PE(H3a)\|$	D	ND	$C(H3a, H3b)$	ND	D	$\|PE(H3b)\|$
1	5-1	11	0	0	11	0	0	11
2	5-2	8	0	0	8	0	0	8
3	5-3	7	0	0	7	0	0	7
4	7-1	20	0	0	20	0	0	20
5	7-2	17	0	0	17	0	0	17
6	7-3	16	0	0	16	0	0	16
7	9-1	22	0	0	22	0	0	22
8	9-2	26	0	0	26	0	0	26
9	9-3	26	0	0	26	0	0	26
10	13-1	30	0	0	30	0	0	30
11	13-2	41	0	0	41	0	0	41
12	13-3	40	0	0	40	1	0	41
13	15-1	65	0	0	65	1	0	66
14	15-2	39	0	0	39	0	0	39
15	15-3	81	0	0	81	0	0	81
16	20-1	124	1	0	123	3	0	126
17	20-2	93	0	0	93	1	0	94
18	20-3	122	1	0	121	7	0	128
19	25-1	144	0	0	144	5	0	149
20	25-2	108	0	0	108	6	0	114
21	25-3	157	0	0	157	2	0	159
22	30-1	219	0	0	219	4	0	223
23	30-2	138	0	0	138	1	0	139
24	30-3	221	0	0	221	7	0	228
25	35-1	289	0	0	289	16	0	305
26	35-2	274	0	0	274	0	0	274
27	35-3	282	0	3	279	1	0	280
28	40-1	301	0	0	301	4	0	305
29	40-2	293	0	0	293	11	0	304
30	40-3	310	0	0	310	4	0	314
31	45-1	291	0	0	291	5	0	296
32	45-2	355	0	1	354	4	0	358
33	45-3	355	1	0	354	6	0	360
34	50-1	374	2	0	372	5	0	377
35	50-2	405	3	0	402	9	0	411
36	50-3	577	0	0	577	9	0	586
37	60-1	530	1	0	529	5	0	534
38	60-2	496	1	0	495	10	0	505
39	60-3	547	0	0	547	0	0	547
40	70-1	686	1	0	685	4	0	689
41	70-2	777	3	4	770	56	1	827
42	70-3	778	0	1	777	12	0	789
43	80-1	1007	1	1	1 005	4	0	1 009
44	80-2	676	1	0	675	13	0	688
45	80-3	919	1	2	916	21	1	938
46	90-1	985	0	4	981	41	3	1 025
47	90-2	969	6	1	962	26	0	988
48	90-3	1352	3	0	1 349	29	0	1 378
49	100-1	1410	13	6	1391	38	1	1430
50	100-2	1110	11	1	1098	73	0	1171
51	100-3	1200	6	0	1194	51	0	1245
52	125-1	1624	4	5	1615	77	2	1694
53	125-2	2041	3	3	2035	42	0	2077
54	125-3	1753	8	8	1737	36	6	1779
55	150-1	2726	12	13	2701	97	2	2800
56	150-2	2933	13	7	2913	168	5	3086
57	150-3	3341	14	30	3297	65	11	3373
58	175-1	3261	4	22	3235	101	4	3340
59	175-2	3433	11	4	3418	76	0	3494
60	175-3	3519	19	9	3491	113	2	3606
61	200-1	3786	8	9	3769	54	0	3823
62	200-2	3984	41	17	3926	331	1	4258
63	200-3	4362	2	10	4350	64	5	4419

Table 15: Results $H3$ (Part II)

(10)	(11)	(12)	(13)	(14)	(15)	(16)	(17)	(18)
D	$C(H3a, H4)$	ND	D	$C(H3b, H4)$	ND	D	$C(H3ab, H4)$	ND
0	11	0	0	11	0	0	11	0
0	8	0	0	8	0	0	8	0
0	7	0	0	7	0	0	7	0
0	20	0	0	20	0	0	20	0
0	17	0	0	17	0	0	17	0
0	16	0	0	16	0	0	16	0
0	22	0	0	22	0	0	22	0
0	26	0	0	26	0	0	26	0
0	26	0	0	26	0	0	26	0
0	30	0	0	30	0	0	30	0
0	41	0	0	41	0	0	41	0
0	40	1	0	41	0	0	41	0
0	65	1	0	66	0	0	66	0
0	39	0	0	39	0	0	39	0
0	81	0	0	81	0	0	81	0
1	123	3	0	126	0	0	126	0
0	93	1	0	94	0	0	94	0
1	121	7	0	128	0	0	128	0
0	144	5	0	149	0	0	149	0
0	108	6	0	114	0	0	114	0
0	157	2	0	159	0	0	159	0
0	219	4	0	223	0	0	223	0
0	138	1	0	139	0	0	139	0
0	221	7	0	228	0	0	228	0
0	289	16	0	305	0	0	305	0
0	274	0	0	274	0	0	274	0
0	282	1	0	280	3	0	283	0
0	301	4	0	305	0	0	305	0
0	293	11	0	304	0	0	304	0
0	310	4	0	314	0	0	314	0
0	291	5	0	296	0	0	296	0
0	355	4	0	358	1	0	359	0
1	354	6	0	360	0	0	360	0
2	372	5	0	377	0	0	377	0
3	402	9	0	411	0	0	411	0
0	577	9	0	586	0	0	586	0
1	529	5	0	534	0	0	534	0
1	495	10	0	505	0	0	505	0
0	547	0	0	547	0	0	547	0
1	685	4	0	689	0	0	689	0
3	774	56	1	826	4	0	830	0
0	778	12	0	789	1	0	790	0
1	1 006	4	0	1 009	1	0	1 010	0
1	675	13	0	688	0	0	688	0
1	918	21	1	937	2	0	939	0
0	985	41	3	1 022	4	0	1 026	0
6	963	26	0	988	1	0	989	0
3	1 349	29	0	1 378	0	0	1 378	0
13	1397	38	1	1429	6	0	1435	0
11	1099	73	0	1171	1	0	1172	0
6	1194	51	0	1245	0	0	1245	0
4	1620	77	2	1692	5	0	1697	0
3	2038	42	0	2077	3	0	2080	0
8	1745	36	6	1773	8	0	1781	0
12	2714	97	2	2798	13	0	2811	0
13	2920	168	5	3081	7	0	3088	0
14	3327	65	11	3362	30	0	3392	0
4	3257	101	4	3336	22	0	3358	0
11	3422	76	0	3494	4	0	3498	0
19	3500	113	2	3604	9	0	3613	0
8	3778	54	0	3823	9	0	3832	0
41	3943	331	1	4257	17	0	4274	0
2	4360	64	5	4414	10	0	4424	0

Table 16: Percentage of common solutions with $H4$ and CPU time (Part I)

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)
	n	$\|PE(H4)\|$	cpu	$\%H1a$	cpu	$\%H1b$	cpu	$\%H1ab$
1	5-1	11		100		100		100
2	5-2	8		100		100		100
3	5-3	7		100		100		100
4	7-1	20		100		100		100
5	7-2	17		100		100		100
6	7-3	16		100		100		100
7	9-1	22		100		100		100
8	9-2	26		100		100		100
9	9-3	26		100		100		100
10	13-1	30		100		100		100
11	13-2	41		100		100		100
12	13-3	41		95.1		100		100
13	15-1	66		98.5		100		100
14	15-2	39		100		100		100
15	15-3	81		100		100		100
16	20-1	126		97.6		100		100
17	20-2	94		98.9		100		100
18	20-3	128		58.6		100		100
19	25-1	149		96.6		100		100
20	25-2	114		94.7		100		100
21	25-3	159		98.1		98.1		100
22	30-1	223		97.8		97.3		100
23	30-2	139		95.0		100		100
24	30-3	228		96.9		100		100
25	35-1	305		94.4		92.1		100
26	35-2	274		88.0		100		100
27	35-3	283		54.1		98.9		99.3
28	40-1	305		97.7		100		100
29	40-2	304		96.1		75.7		100
30	40-3	314		94.3		100		100
31	45-1	296		98.0		99.3		100
32	45-2	359		98.1		78.0		100
33	45-3	360		96.7		85.0		100
34	50-1	377		98.7		67.1		99.7
35	50-2	411		96.8		26.9		99.8
36	50-3	586		95.4		42.0		99.0
37	60-1	534		98.3		71.7		99.4
38	60-2	505		32.5		98.4		98.4
39	60-3	547		96.1		100		100
40	70-1	689	1	89.6		98.5		100
41	70-2	830	1	8.4		74.0		82.2
42	70-3	790	1	89.1		22.9		98.7
43	80-1	1010	1	7.9		99.4		99.4
44	80-2	688	1	98.1		21.7		98.8
45	80-3	939	1	86.7		31.2		97.1
46	90-1	1026	5	23.7		47.0		70.5
47	90-2	989	5	9.1		78.2		87.1
48	90-3	1378	8	51.5		37.3		77.9
49	100-1	1435	8	96.3	1	82.8		99.6
50	100-2	1172	7	18.8	1	24.7		38.4
51	100-3	1245	7	93.7	1	47.2		98.6
52	125-1	1697	12	92.6	1	38.8	1	96.4
53	125-2	2080	12	7.6	1	46.0	1	53.5
54	125-3	1781	9	92.8	1	35.7	1	97.4
55	150-1	2811	9	43.5	1	18.2	1	61.7
56	150-2	3088	11	16.4	1	10.7	1	27.0
57	150-3	3392	11	16.5	1	38.2	1	54.7
58	175-1	3358	16	14.8	2	21.7	1	36.5
59	175-2	3498	16	91.0	2	3.8	1	91.5
60	175-3	3613	17	3.8	2	11.5	1	15.3
61	200-1	3832	17	12.8	2	27.6	1	40.4
62	200-2	4274	19	8.4	2	36.5	1	44.9
63	200-3	4424	19	12.4	2	18.9	1	31.2

Table 17: Percentage of common solutions with $H4$ and CPU time (Part II)

(10)	(11)	(12)	(13)	(14)	(15)	(16)	(17)	(18)	(19)
$\%H2a$	cpu	$\%H2b$	cpu	$\%H2ab$	$\%H3a$	cpu	$\%H3b$	cpu	$\%H3ab$
100		100		100	100		100		100
100		100		100	100		100		100
100		100		100	100		100		100
100		100		100	100		100		100
100		100		100	100		100		100
100		100		100	100		100		100
100		100		100	100		100		100
100		100		100	100		100		100
100		100		100	100		100		100
100		100		100	100		100		100
100		100		100	100		100		100
100		100		100	97.6		100		100
98.5		100		100	98.5		100		100
100		100		100	100		100		100
100		100		100	100		100		100
97.6		100		100	100		100		100
100		100		100	97.6		100		100
96.1		100		100	94.5		100		100
96.6		100		100	96.6		100		100
100		100		100	94.7		100		100
98.7		100		100	98.7		100		100
97.8		98.7		100	98.2		100		100
96.4		100		100	99.3		100		100
99.1		100		100	96.9		100		100
98.4		99.3		100	94.8		100		100
94.2		100		100	100		100		100
75.3		100		100	99.6		98.9		100
99.3		100		100	98.7		100		100
98.4		100		100	96.4		100		100
94.9		100		100	98.7		100		100
99.0		99.3		100	98.3		100		100
98.6		98.6		100	98.9		99.7		100
97.8		100		100	98.3		100		100
98.7	2	97.6		100	98.7		100		100
97.8	2	91.0		100	97.8		100		100
96.1	1	86.0		100	98.5		100		100
98.5	1	99.4		100	99.1		100		100
82.6	1	100		100	98.0		100		100
92.1	1	100		100	100		100		100
89.6	3	99.0	1	100	99.4		100		100
92.8	3	90.6	1	98.9	93.3		99.5		100
92.7	5	96.8	2	99.9	98.5		99.9		100
43.1	6	99.7	2	99.9	99.6		99.9		100
98.4	3	99.6	1	100	98.1		100		100
87.3	3	67.5	1	98.8	97.8		99.8		100
40.7	13	79.5	2	80.8	96.0		99.6		100
80.4	13	91.2	2	91.9	97.4		99.9		100
55.4	9	99.6	4	99.7	97.9		100		100
97.1	9	68.9	5	99.3	97.4	1	99.6		100
84.7	20	96.8	4	99.7	93.8	1	99.9		100
96.2	20	98.1	4	99.6	95.9	1	100		100
95.1	21	75.4	4	98.1	95.5	1	99.7	1	100
22.9	21	96.9	4	96.9	98.0	1	99.9	1	100
95.0	26	62.3	4	98.3	98.0	1	99.6	1	100
65.7	26	60.7	6	90.9	96.5	1	99.5	1	100
27.3	42	28.2	7	42.6	94.6	1	99.8	1	100
20.5	42	38.5	7	54.8	98.1	1	99.1	1	100
18.9	36	66.8	11	66.9	97.0	2	99.3	1	100
91.0	36	41.1	11	93.6	97.8	2	99.9	1	100
7.9	49	69.3	13	70.9	96.9	2	99.8	1	100
13.2	49	32.4	13	42.6	98.6	2	99.8	1	100
9.7	35	52.0	16	52.1	92.3	2	99.6	1	100
25.3	35	97.4	16	97.6	98.6	2	99.8	1	100

Table 18: Synthetic view of the results: average %

n	$H1a$	$H1b$	$H1ab$	$H2a$	$H2b$	$H2ab$	$H3a$	$H3b$	$H3ab$
5-7-9	100	100	100	100	100	100	100	100	100
13-15-20	94.30	100	100	99.13	100	100	98.56	100	100
25-30-35	90.60	98.48	99.92	95.30	99.77	100	97.64	99.87	100
40-45-50	96.86	78.22	99.83	97.62	96.94	100	98.25	99.96	100
60-87-800	66.30	68.64	97.11	83.64	94.73	99.72	98.20	99.90	100
90-100-125	53.45	48.63	79.93	74.16	85.41	96.03	96.65	99.80	100
150-175-200	24.40	20.78	44.80	31.05	54.04	68.00	96.71	99.64	100
All instances	75.13	73.53	88.79	83.37	90.12	94.82	98.00	99.88	100