A hybrid population-based algorithm for the bi-objective quadratic multiple knapsack problem

Meziane Aider, Oussama Gacem, Mhand Hifi

Version of Record: https://www.sciencedirect.com/science/article/pii/S0957417421015487
Manuscript_9274ccb6cb3816cd9f8f47ba775e485b

A Hybrid Population-Based Algorithm for the Bi-Objective Quadratic Multiple Knapsack Problem

Méziane Aïder
LaROMaD, DGRSDT, Université des Sciences et de la Technologie Houari Boumediene,
BP 32 Bab Ezzouar, 16111 Algiers, Algeria
email: m-aider@usthb.dz

Oussama Gacem
LaROMaD, DGRSDT, Université des Sciences et de la Technologie Houari Boumediene,
BP 32 Bab Ezzouar, 16111 Algiers, Algeria
email: oussamagacem@yahoo.com

Mhand Hifi*
EPROAD UR 4669, Université de Picardie Jules Verne,
7 rue du Moulin Neuf, 80000 Amiens, France
hifi@u-picardie.fr

*Corresponding author (All authors are listed in alphabetical order of their last name, according to the Operational Research domain in France).

Abstract. In this paper, the bi-objective quadratic multiple knapsack problem is tackled with a hybrid population-based method. The proposed method starts by computing two reference solutions, where a specialized powerful mono-objective algorithm is used. From both reference solutions, a starting population is built by using a series of perturbations around the solutions. Next, the so-called non-sorting genetic process is combined with a new drop/rebuild operator for generating a series of populations till converging toward an approximate Pareto front with high density. The performance of the hybrid population based algorithm (namely HBPA) is evaluated on a set of benchmark instances of the literature containing both medium and large-scale instances. Its provided results are compared to those achieved by the best methods available in the literature. Encouraging results have been obtained.

Keywords. Bi-objective, evolutionary, hybrid, knapsack, optimization.

1 Introduction

The knapsack problem arises in several real-world applications, like cutting and packing (Hifi, 2004), cryptography (Merkle & Hellman, 1978), logistics (Perboli et al., 2014), multimedia (Akbar et al., 2006), telecommunications and other themes (Plata-Gonzalez et al., 2019). Such a problem plays, on the one hand, a central role in modeling more higher NP-hard combinatorial optimization problems, where the used models serve as a guide to design effective exact and efficient approximate methods. On the other hand, it is often important to tackle large-scale instances, especially when several optimization objectives are needed for evaluating the performance of methods applied to solving it. Often, simple deterministic / stochastic approximate solution procedures may be used for solving some knapsack-type problems, it has often been remarked that the used methods may converge slowly and in some cases towards to solutions with poor quality.

The problem considered in this paper is a more complex variant of the classical knapsack problem: Bi-Objective Quadratic Multiple Knapsack Problem (BO-QMKP). As described in Chen & Hao (2016), such a problem occurs whenever a company leader tries to allocate staff to form a set of personnels to drive different available products. In addition to the total strength of overall personnel, a balance between the personnels in the interest of fairness and sustainability should be established. Another situation may occur in the portfolio investments, where an investor is often interested with maximizing the global long-term return on the mix of assets invested, and tries to ensure an expected return on the least profitable asset. Such a problem can be viewed as a combination of two well-known NP-hard combinatorial optimization problems with two different objective functions: Quadratic Knapsack Problem−QKP (Billionnet & Soutif,

2004) and, Multiple Knapsack Problem−MKP (Hung & Fisk, 1978). The proposed hybrid population-based method combines several operators incorporated into the non-sorting genetic algorithm for tackling large-scale instances. It is based upon the following features:

- To build *two reference solutions* according two both objective functions by using an adaptation of the algorithm designed in Aïder et al. (2020a), tailored for BO-QMKP.
- To use a *perturbation operator* for providing a starting population containing a set of diversified solutions.
- To use an *iterative non-sorting genetic procedure* combined with a *drop and rebuild operator* for achieving the final approximate Pareto set of solutions.

The proposed method is a tailored hybrid framework, where several operators are combined. As observed in the literature, hybridization of different strategies often leads to efficient methods. Moreover, a tailored hybridization using multiple operators can induce benefits from each of the operators used; as a result, the method built can be more powerful. Although there are different forms of hybridization, we investigate the use of a hybrid evolutionary algorithm that combines an iterative non-sorting genetic algorithm with a drop and complete operator to generate the final approximate Pareto front.

The outline of the paper is as follows. In Section 2 we provide the related work addressing the bi-objective quadratic knapsack problem and variants related to the mono-objective version of the problem. Section 2 discusses the main contribution of the paper: a hybrid method for providing a set of approximate Pareto front. The first two reference solutions, related to both objective functions, are detailed in Sections 3.5.2 and 3.5.3. The starting diverse population is presented in section 3.6. The new fusion operator is detailed in Sections 3.7 while Section 3.8 presents the mutation operator replaced by a new drop and rebuild operator. Section 4 evaluates the performance of the proposed method on reference instances taken from the literature. The results provided by the proposed method are compared to those achieved by recent methods of the literature. Finally, section 5 concludes the content of the paper.

2 Related work

The knapsack problems family contains a large variety of problems as underlined in Martello & Toth (1990) and in Kellerer et al. (2004). In addition to the classic binary knapsack problem, which received considerable attention (Feng et al., 2019, 2017, 2018a, 2016), the discounted binary knapsack problem (Feng et al., 2018c), the knapsack problem with setups (Boukhari et al., 2022), and the time-varying random knapsack problem (Feng et al., 2018b) can be considered as more complex variants.

The BO-QMKP can be viewed as a combination of two NP-hard knapsack problems: the quadratic knapsack problem (Billionnet & Soutif, 2004) and the multiple knapsack problem

(Hung & Fisk, 1978), where two different objective functions are incorporated to both problems. Due to its NP-hardness, there are few papers in the literature, which tackle BO-QMKP either exactly or approximately. Most published methods have been tailored for solving the combined problem separately, i.e., the two single-objective versions of the problem, where, in some cases, the quadratic objective function is coupled with multiple knapsack constraints.

Hiley & Julstrom (2006) are among the first authors who studied the single-objective version of the problem, where several nice heuristics have been designed. The first approach is based on selecting items with the highest density. A more complete solution procedure has been developed; that is based upon a genetic algorithm, such that both classical crossover and mutation operators were introduced. Finally, a random destruction and reconstruction operator was added for exploring the search space by maintaining the diversity of the solutions. In the experimental part, the behavior of all proposed algorithms was evaluated on a set of benchmark instances of various sizes.

Saraç & Sipahioglu (2007) designed another genetic algorithm, where the starting population is built by using a tournament selection operator combined with a specific mutation operator (in this case, two versions of the solution procedure were developed). The authors performed an extensive experimentation on a set of instances and showed that their method remains competitive.

Chen & Hao (2015) proposed a hybrid method, where several strategies were combined to design an iterated reactive threshold search procedure. First, the method applies the well-known greedy density procedure proposed by Hiley & Julstrom (2006) to start the method. Then, both the *threshold exploration phase* and *descent improvement phase* are iterated, the former combining a variety of neighborhoods in an attempt to diversify/intensify the search process and the latter mimicking a descent procedure by implementing complementary neighborhoods. In the experimental part, the performance of the method was evaluated on a set of large-scale instances from the literature, including instances ranging from 100 to 200 items.

Aïder et al. (2020a) designed a branch and solve strategy-based algorithm for the single-objective version of the problem. A starting solution was built by calling a special fix and solve solution procedure. A flexible memory structure based on a short memory coupled with a hashing function is added for enhancing the fix and solve procedure. The goal of the aforementioned strategy is to forbid already visited local optima. The provided solution procedure is embedded into a local branching-based method, where two branches are considered: a first branch that is used for intensifying the search process while a second branch tries to diversify the search process. Finally, the proposed method was computationally analyzed on a set of benchmark instances taken from Chen & Hao (2015).

For BO-QMKP, the studied problem, Chen et al. (2016) proposed a population-based method, where the path relinking strategy is used as the core of the scattered method. Their method combines (i) a construction procedure that ensures the creation of the initial reference set con-

taining the elite solutions, (ii) a path relinking strategy that generates a set of intermediate solutions provided from a set of initial solutions to the guiding solutions, (iii) a threshold search for enhancing the quality of the solution; that is based upon Chen & Hao (2015) algorithm used as a black-box solver and (iv) a pool updating strategy for maintaining the diversity of the reference set. In their experimental study, the authors underlined the competitiveness of their method, in this case by observing that their method was able to match the solutions provided by the methods tailored for the single-objective version of the problem and confirms the success of their method for achieving non-dominated solutions on instances tested in Chen & Hao (2015).

Table 1: Illustration of some published papers addressing tailored and general purpose algorithms

	Algorithms	
References	QMKP	BO-QMKP
Hiley & Julstrom (2006)	Three heuristic approaches	–
Saraç & Sipahioglu (2007)	A genetic algorithm	–
Chen & Hao (2015)	An iterated responsive threshold	–
Chen, Hao and Glover (2016)	An evolutionary path relinking	–
Chen & Hao (2016)	–	A hybrid two stage algorithm
Aider, Gacem and Hifi (2020a)	Branch and solve algorithm	–
Aider, Gacem and Hifi (2020b)	–	A two stage $\varepsilon-$constraint

Finally, Aïder et al. (2020b) designed a two-stage method for the BO-QMKP. The main principle of the method is based upon an iterative search, where (i) a starting solution is provided by applying a basic knapsack's greedy procedure, (ii) the current provided optimization problem is solved using a specialized reactive search and, (iii) a new restricting problem is built; that is a problem containing a new ε-constraint. The aforementioned steps were embedded into an iterative search till converging to a satisfactory set of approximate Pareto solutions. The behavior of the proposed method was evaluated on several benchmark instances of the literature and its provided results were compared to those achieved by the best available methods. The experimental part showed that their method was able to reach new dominating solutions when compared to those published in Chen et al. (2016). Table 1 illustrates a tentative resume of some algorithms tackling both QMKP and BO-QMKP.

In this work, we propose a hybrid method for solving BO-QMKP, where the following strategies are combined:

- To build two reference solutions, according to both objective functions, by using an adaptation of the powerful branch and solve approach (Aïder et al., 2020a). It is called once for solving $P_{BO-QMKP}$ favoring the first objective function (neglecting the second one), and recalled once again for solving the same problem by favoring the second objective function.

- To build a starting population of solutions generated according to the two reference solu-

tions. It is provided by using a perturbation operator which is based upon the drop and rebuild procedure: for each reference solution, (i) a drop strategy is applied for removing a subset of items; thus, a partial solution is induced, and (ii) a rebuild strategy for reaching a complete solution.

- To use an iterative non-dominating sorting strategy in order to generate a final approximate Pareto set. Each step of the iterative search uses (i) a tournament selection on the current population, (ii) a new fusion operator replacing the classical crossover operator, and (iii) a reactive local search which replaces the mutation operator, as used in memetic algorithms.

3 Tackling BO-QMKP with a hybrid method

This section starts by describing the formal model related to BO-QMKP (Section 3.1). Second, the principle of the generic approach in presented in Section 3.3 and the proposed method is discussed in Section 3.4. Third, the process used for generating the first diversified population of solutions is detailed in Section 3.5 and, the first approximate set of Pareto front is presented in Section 3.6, where a two-stage procedure is applied: generating a couple of non-dominated solutions and the first starting population. Fourth and last, the hybrid process is presented in Section 3.9, where the modified selection, crossover and mutation operators are introduced.

3.1 *The model*

An instance of BO-QMKP is characterized by a set $M = \{1, \ldots, m\}$ of m knapsacks of fixed capacity each, i.e., $c = (c_1, \ldots, c_m)$, and a set $N = \{1, \ldots, n\}$ of n items. Each item i, $\forall\ i \in N$, is characterized by a profit p_i and a weight w_i and each pair of distinct items (i, j) belonging to $N \times N$ $(i \neq j)$ has an augmented profit p_{ij} if both items belong to the same knapsack k, $k \in M$. The goal of the problem is to assign each item to at most one knapsack such that the total weight of the items in each knapsack k, $k \in M$, does not exceed its capacity c_k and both (i) the total profit of all the items included into the knapsacks and (ii) the makespan related to the knapsack with the lowest gain, are maximized.

Let $\boldsymbol{x_{ik}}$ be the decision variable set equal to 1 if the item i, $i \in I$, is assigned to the knapsack k, $k \in M$, 0 otherwise. The formal description of BO-QMKP (noted $\mathrm{P_{BO-QMKP}}$) is given as follows:

$$\mathrm{P_{BO-QMKP}}:$$

$$z_1(x) = \max \quad \sum_{i \in N} \sum_{k \in M} \boldsymbol{p_i x_{ik}} + \sum_{i \in N} \sum_{\substack{j \in N \\ i<j}} \sum_{k \in M} \boldsymbol{p_{ij} x_{ik} x_{jk}} \tag{1}$$

$$z_2(x) = \max \quad \min_{k \in M} \sum_{i \in N} \boldsymbol{p_i x_{ik}} + \sum_{i \in N} \sum_{\substack{j \in N \\ i<j}} \boldsymbol{p_{ij} x_{ik} x_{jk}} \tag{2}$$

$$\text{s.t.} \quad \sum_{i \in N} \boldsymbol{w_i x_{ik}} \leq \boldsymbol{c_k}, \forall k \in M, \tag{3}$$

$$\sum_{k \in M} \boldsymbol{x_{ik}} \leq 1, \forall i \in N, \tag{4}$$

$$\boldsymbol{x_{ik}} \in \{0,1\}, \forall i \in N, \ \forall k \in M, \tag{5}$$

where inequalities of type (3) represent the knapsack constraints of capacity $\boldsymbol{c_k}$, $k \in M$, and inequality of type (4) means that each item can be assigned to at most one knapsack (when such an item belongs to the solution). Finally, the objective function $z_1(x)$ (Eq. (1)) is composed of a linear term (on the left-hand) and a quadratic term (on the right hand) while the second objective function (Eq. (2)) is also composed of two terms such that its worst component is to be maximized.

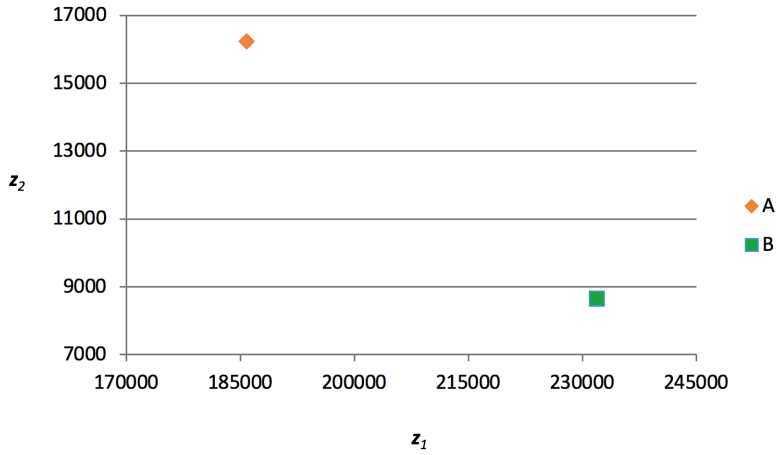

Figure 1: Illustration of the conflictual objective values of z_1 and z_2 for the instance `300-25-10-4`

3.2 The conflicting objectives z_1 and z_2

Often real-world problems are multi-objective by nature, where several contradictory objectives involve the optimization. As described by the original model of Section 3.1, there are two objectives including the total profit to maximize (Eq. (1)), and the optimization of the best knapsack realizing the minimum value (Eq. (2)) over all knapsacks (or maximizing the gain of the least profit knapsack). According to Chen et al. (2016), these objectives may be conflicting. Indeed, as illustrated in Fig. 1, the behavior of both objective functions, especially for the instance `300-25-10-4` (belonging to the benchmark instances of the literature), the solution A realizes a profit of 185787 ($z_1(A)$) with the best knapsack gain of 16225 ($z_2(A)$) whereas the solution B has a greatest total profit of 231923 ($z_1(B)$) but achieves a worse overall gain ($z_1(B) = 8646$).

One can observe that augmenting the total profit may provide the reduction on the gain

of the least profit (and vice versa). Hence, a hybrid multi-objective evolutionary algorithm becomes a good candidate for tackling $P_{BO-QMKP}$.

3.3 *A standard multi-objective approach*

Often each evolutionary method (Wang et al., 2020) starts with a randomly generated population of solutions $\mathcal{P}_0$, using a uniform generator for covering the search space of the related problem. At each step t, $t \geq 1$, of the search process, a set of offspring $\mathcal{O}_t$ is provided regarding the current parent population $\mathcal{P}_t$, where both selection and several operators are applied. In this case, for the current iteration t, the parental selection considers a randomly picking operator of two parents from $\mathcal{P}_t$. A variety of crossover operators may be applied to the aforementioned parents with a certain probability to create two offsprings while the variety of mutation operators may be applied to each offspring with a given probability in order to reach the final offspring. Hence, the new population of solutions at step $t+1$, i.e., $\mathcal{P}_{t+1}$, is then updated by considering a given survival selection of the mixed population of both parents and both two offsprings.
In order to adapt such a process to a problem with many objective functions, the multi-objective optimization uses the Pareto dominance rule related to the following definition.

Definition. *Dominance rule*

For a given maximization problem with m objective functions: $z_1, z_2, \ldots, z_m$, a solution $S^{(1)}$ dominates another solution $S^{(2)}$ ($S^{(1)} \succ S^{(2)}$), if and only if the following conditions hold:

1. $\forall k \in \{1, \ldots, m\} \ : \ z_k(S^{(1)}) \geq z_k(S^{(2)})$,
2. $\exists k \in \{1, \ldots, m\} \ : \ z_k(S^{(1)}) > z_k(S^{(2)})$.

According to the dominance rule, the set of Pareto optimal solutions can be represented by the set of *points* among the set of solutions $\mathcal{P}$, which are not dominated by any solution belonging to $\mathcal{P}$.

Among the optimization-based populations that have been designed for tackling multi-objective problems (Zhang et al., 2020; Gu & Wang, 2020); NSGA-II (Deb et al., 2002) remains one of the best methods to provide a balance between the Pareto front's density and the runtime required to converge. Such a method successfully used for solving various optimization problems (Bederina & Hifi, 2018; Haghighia et al., 2021; Che et al., 2021). In this paper, $P_{BO-QMKP}$ is solved by NSGA-II combined with a drop and rebuild procedure is investigated, where two starting reference points are provided by using a special branch and solve method. Of course, the goal of the combined operator is to prevent premature convergence of the method and stagnation in local optima.

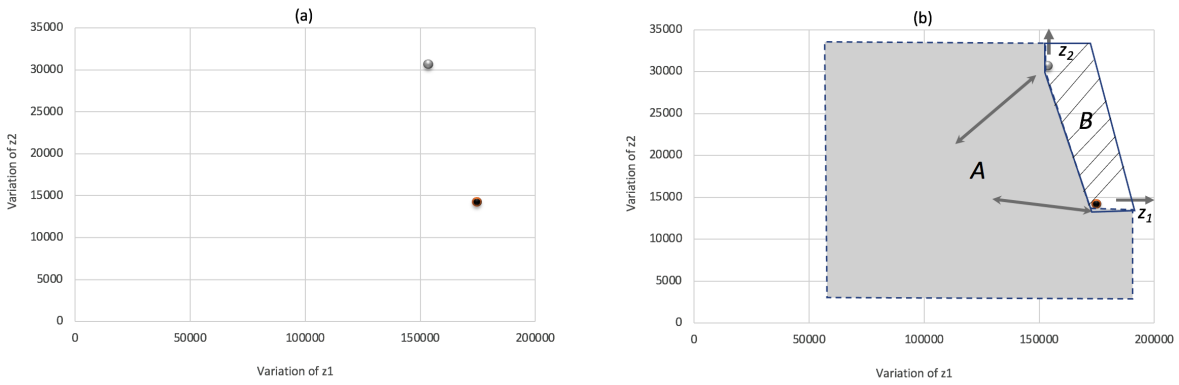

Figure 2: Representation (a) of reference points (on an instance of the literature `200-75-5-2`) and, (b) the target space-set of Pareto points

3.4 *Illustration of the principle of the hybrid method*

The main principle of the hybrid method can be summarized by the following steps:

1. To generate two reference solutions, which are related to both objective functions z_1 and z_2 (cf. Sections 3.5.2 and 3.5.3).

2. From both reference solutions, a starting population, that mimics an approximate Pareto front, is built (Section 3.6).

3. Selection, crossover and mutation-based modified operators are introduced for updating the current population with high degree of diversification (Sections 3.7.1, 3.7.2 and 3.8).

4. Steps (2) and (3) are iterated till converging to an approximate Pareto front of solutions.

First, Figure 2(a) mimics the first step of the proposed method (Step (2) above) for providing two reference solutions (reached by the algorithm ABSM, as discussed in Sections 3.5.2 and 3.5.3), especially for the instance `200-75-5-2` belonging to the second set of benchmark instances of the literature: (i) the first solution in the grey-color represents the first solution with objective value $z_1 = 174836$ (and $z_2 = 14220$) while (ii) the second one in the black-color denotes the second point with $z_2 = 30686$ (and $z_1 = 153695$).

Second, Figure 2(b) illustrates two potential (sub)sets A and B that an approach could target in order to iteratively create a(n) (approximate) Pareto front; i.e. a set of (non-)dominated points which can be obtained (or at least some parts of the shown (sub)sets) by an eventual optimization solution procedure. The set A can be provided by calling an optimization process when decreasing either z_1 or z_2, while the set B can be obtained by increasing either z_1 or z_2.

Third, from the two reference solutions, one can observe that a "guided search" can provide solutions belonging to either A, B, or A and B. Indeed, as illustrated in Figure 3(a), the

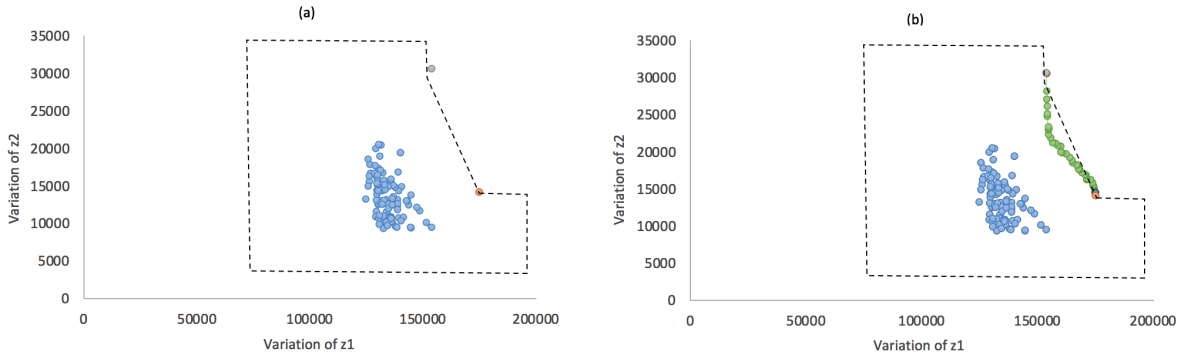

Figure 3: Illustration of the hybrid NSGA-II's behavior on an instance of the literature `200-75-5-2`: (a) a starting population and, (b) the final population (with its Pareto front)

initial population is provided using a "drop and rebuild" solution procedure (as described in sections 3.6 and 3.7).

Fourth and last, Figure 3(b) shows an eventual final population containing an approximate Pareto front of non-dominated points; that is the role of the search process proposed in Section 3.9.

The three steps shown in the figures above (Figures 2 and 3), which represent the main steps of the proposed hybrid genetic algorithm without sorting, are detailed in the rest of the paper. We note that in order to make the paper self-containing, some parts already described in Aïder et al. (2020a) are repeated in what follows.

3.5 *Generating a starting diversified population*

3.5.1 Solutions representation

For an instance of $\mathrm{P}_{\mathrm{BO-QMKP}}$, a standard binary representation scheme is an obvious choice since it represents the underlying assignment related to binary decision variables (Figure 4 shows a simple representation of $\mathrm{P}_{\mathrm{BO-QMKP}}$'s solution).

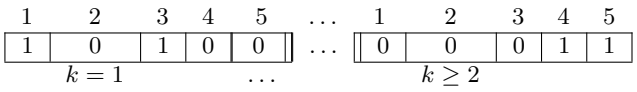

Figure 4: A binary representation of $\mathrm{P}_{\mathrm{BO-QMKP}}$'s solution S related to knapsacks k, $k \leq m$.

Herein, a feasible solution $S = (\vec{s}_k, \ldots, \vec{s}_m)$ is such that both inequalities (3) and (4) hold while fitness functions are directly computed by using equalities (1) and (2), respectively.

Because $\mathrm{P}_{\mathrm{BO-QMKP}}$ is more complex to solve, where its goal is to reach the best front containing a set of Pareto optimal solutions that optimizing both objective functions, we then try to adapt a quick variant of the branch and solve method (Aïder et al., 2020a). Herein,

the diversity of the starting population needs two extreme solutions $S^{(1)} = (\vec{s}\,_1^{(1)}, \ldots, \vec{s}\,_m^{(1)})$ and $S^{(2)} = (\vec{s}\,_1^{(2)}, \ldots, \vec{s}\,_m^{(2)})$ such that the first solution must be based on a better value related to z_1 while the second one must be constructed by favoring z_2. Indeed, the three key features of the branch and solve method follow:

1. Generating an initial solution $S^\star$ by applying a standard bin-packing procedure.
2. Enhancing the above solution by applying a tabu search.
3. Applying the local branching for providing a final solution $S^\star$.

Aïder et al.'s (2020a) local branching (namely LB) is used, as explained in Section 3.5.3, for generating two reference solutions. We recall that LB is a specialized mono-objective algorithm tailored for efficiently solving the mono-objective QMKP. The purpose of its use is to build two opposite solutions hoping for a better diversification of the solutions belonging to the future populations.

3.5.2 *The first stage*

This stage is used for providing a starting solution with a greedy optimization procedure by trying to solve the original problem step by step till examining all items of the problem. Therefore, the standard bin packing solution procedure is applied as follows:

- Let $S = (\vec{s}_1, \ldots, \vec{s}_m)$ be a feasible solution of $P_{\text{BO-QMKP}}$ (according either to the objective function z_1 or z_2) and $N_1(S)$ (resp. $N_0(S)$) be the subset of items with 1 (resp. 0) in each component $\vec{s}_j,\ j \in M$, of S.
- The *contribution* of item $i,\ i \in N$, of the objective function according to the solution S is given as follows:

$$c_i(S,k) = \boldsymbol{p_i} + \sum_{\substack{j \in N, \\ i \neq j}} \boldsymbol{p_{ij}} * \boldsymbol{x_{ik}} * \boldsymbol{x_{jk}}, \tag{6}$$

  and the *density* of item $i,\ i \in N$, of the objective function according to both the solution S and the knapsack $k,\ k \in M$, is given as follows:

$$d_i(S,k) = \frac{c_i(S,k)}{\boldsymbol{w_i}}. \tag{7}$$

Algorithm 1 describes the principle of the greedy Constructive Procedure (CP) that uses the criterion (7): it selects items who should be included (excluded) into (from) the current solution. Of course, we initially suppose that all items are set to free (no assignment is specified for all decision variables). According to the current residual capacity $\bar{c}$ related to c of the current knapsack (that is c minus all the weights related to the already assigned items in the current solution according to the current knapsack), all items are sorted in decreasing order of their

Algorithm 1 – A constructive procedure (CP) for $\mathrm{P}_{\mathrm{BO-QMKP}}$

Input. An instance of $P_{\mathrm{BO-QMKP}}$.
Output. A solution $S^{\star}$ with objective values z_1 and z_2.

Set $J' = N$ and $\overline{c} = c := (c_1, \ldots, c_m)$;
Sort the indices of N in decreasing order of their *densities*, i.e., $d_1 \geq d_2 \geq \ldots \geq d_n$;
Sort all knapsacks in decreasing order of their capacities, i.e., $\overline{c}_1 \geq \overline{c}_2, \ldots, \geq \overline{c}_m$;
Set $\ell = 0$; `// (ℓ + 1) is the first item to pack into the first bin/knapsack`
repeat
    Increment(ℓ);
    if $(\exists \rho \mid \overline{c}_\rho \geq w_\ell)$ **then**
        Set $s^{\star}_{\ell k} = 1$ and $\forall k \in M \setminus \{\rho\}, s^{\star}_{\ell k} = 0$;
        Set $\overline{c}_\rho = \overline{c}_\rho - w_\ell$;
    else
        Set $s^{\star}_{\ell k} = 0, \forall k \in M$;
    Set $J' = J' \setminus \{\ell\}$;
until $(J' = \emptyset)$
return $S^{\star}$

densities (as underlined above) and, all capacities in decreasing order (whenever these capacities are different). At each step of CP, an unassigned item ℓ with highest density d_ℓ is selected and added to the (current) partial solution. The ℓ-th item is packed into an *open knapsack* (*bin*), namely ρ, whenever its weight w_ℓ is smallest than or equal to its residual capacity $\overline{c}_\rho$ or, assigned to a new knapsack / bin or, fixed to zero otherwise (for all knapsacks / bins). Such a process is iterated till visiting all items and thus, CP stops with the best solution $S^{\star}$.

Algorithm 2 – CP with the first swapping operator for $\mathrm{P}_{\mathrm{BO-QMKP}}$: CP1.

Let $\hat{S}$ be the solution at hand (initially provided by the first call of CP).
Let (i, j), $i \neq j$ (belonging to N) denote a couple of items assigned to a couple of knapsacks (k_i, k_j) belonging to the set M (i.e., $\hat{s}_{ik_i} = \hat{s}_{jk_j} = 1$).
Let $\hat{S}'$ be a configuration built by swapping both indices i and j, i.e., setting $\hat{s}_{ik_j} = \hat{s}_{jk_i} = 1$ in the new configuration.
Set the inverse-move (j, i) tabu, by creating a tabu list (L_{tabu}) following a FIFO ranking.
Call `Algorithm 1` for making $\hat{S}'$ feasible (in case where $\hat{S}'$ is infeasible), and improving the quality of the solution at hand.

We note that CP can also be used as a *repairing procedure* of any infeasible solution. Indeed, on the one hand, CP proceeds by inverting the fixing process, i.e., CP removes items from the current (in)feasible solution step by step according to the smallest density related to the items to remove. On the other hand, CP is recalled for completing the provided partial solution. Because CP is a deterministic greedy procedure, it directly converges to a local optimum. In order to enhance the quality of the starting solution, a 2-opt operator with a tabu list is used for swapping two elements either (i) belonging to the same backpack (bin) or (ii) belonging to two

different backpacks (bins). Algorithm 2 (CP1) describes the process using the first swapping operator.

Algorithm 3 – A Repairing Procedure using the second swapping operator (RP).

1: Let $\hat{S}^{\star}$ be an infeasible solution at hand.
2: Apply the inverse CP to $\hat{S}^{\star}$ by removing step by step items with the smallest densities till reaching a partial feasible solution $\hat{S}'$.
3: Call `Algorithm 1` for completing $\hat{S}'$ and let $\hat{S}^{\star}$ the final improved solution.

The second swapping operator follows the same principle as described above except that two different items $\hat{s}_{ik}$ and $\hat{s}_{jk}$ (of a solution S) belonging to the same knapsack k, where one of the decision variables is fixed to 1 and 0 for the second one. Of course, that permutation may also induce the infeasibility of the solution and so, a simple repairing and improving process inspired from CP may be applied by using the main steps of Algorithm 3.

Algorithm 4 – Adaptation of the Branch and Solve Method (ABSM) for $\mathrm{P}_{\mathrm{BO-QMKP}}$.

Input. An instance of $P_{\mathrm{BO-QMKP}}$.
Output. A solution $S^{(1)}$.

1: **Initialization phase:**
2: Call `Algorithm 1` for generating a starting solution $S^{\star}$.
3: Iterative phase.
4: **repeat**
5: Add the local branch $\sum_{j \in S^{\star}} (1 - s_j^{\star}) \leq k$ to the subtree according to $S^{\star}$.
6: Call `Algorithm 2` (or `Algorithm 3`, if the solution is infeasible) for solving the provided problem, i.e., the (original / current) problem with the local constraint.
7: Let S' be the new achieved solution.
8: **if** $(z(S^{\star}) < z(S'))$ `//` $z = z_1$ `or` $z = z_2$ **then**
9: Update $S^{\star}$ with S' and create a new local tree.
10: **else**
11: Branch on the complementary subtree of the right-branch and remove the latest added constraints.
12: **until** (satisfying the stopping condition.)
13: Exit with $S^{\star}$.

3.5.3 *The second stage*

The second step consists in adapting the Local Branching (LB) for improving the quality of the solutions. In this case, a series of branches are introduced throughout a special branch and solve procedure, where LB supposes a starting reference solution that is used for the first local tree. Next, an iterative search is considered for enhancing the solution at hand. In this case, two cases can be distinguished: (i) the activated search that is able to improve the current solution

and so, the process is repeated and, (ii) the solution stagnates; in this case the search process is recalled with a new search space. The main principle of LB can be described as follows:

1. Suppose the existence of a starting solution.
2. Initialize the first local tree with the starting solution.
3. Iterative Step:
   (a) Solve completely the local tree.
   (b) When the local search terminates,
      i. if a better solution is provided, then create (from this local tree) a new local tree with the new solution (reference solution);
      ii. otherwise, improvement is reached; therefore, abort the local branching.
   (c) Solve the rest of the search tree.
   (d) Return the best solution found up to now.

Because two reference solutions are needed for generating the starting diversified population, we then call the above LB twice, where the first call generates the first reference solution optimizing the first objective function z_1 while the second call provides the second solution according to the second objective function z_2.

Algorithm 4 summarizes the main principle of the Adaptation of the Branch-and-Solve Method (noted ABSM), comprising two phases. The first phase (line 2) starts the first subtree by applying the procedure described in Section 3.5.2 (Algorithms from 1 to 3). The second phase is composed of three cases. For the first case, a local-branch (line 5) is injected and Algorithm 2 (line 7) is called as an *optimizer-tool* to the local subtree trying to provide a new solution. The second case (lines from 8 to 11) checks if the new provided bound enhances the incumbent solution, then it generates a new local tree for a new branching. The third case (line 11) branches on the right-branch of the tree and calls the optimizer-tool till satisfying the final stopping condition. This process is iterated till matching the stopping criterion.

3.6 *The first approximate Pareto set with two reference solutions*

3.6.1 *Generating two reference solutions*

As underlined above, ABSM (Algorithm 4) is applied for reaching the two reference solutions which could, on the one hand, be candidates to be contained in the starting population and, to generate members of that population, on the other hand. Indeed, a quick version of ABSM is called once on the instance of $\mathrm{P}_{\mathrm{BO-QMKP}}$ favoring the first objective function z_1 (neglecting the second one) and, to recall it on the same instance by favoring the second function z_2, ABSM is initialized with the best solution obtained for z_1. We recall that $s_k^* = (s_{ik}^*, \ldots, s_{nk}^*)$ denotes the decision variables of the knapsack k according to the solution S^*.

3.6.2 *The first population*

From the two referent points achieved by ABSM, the starting population is generated by the adaptation of an iterated greedy methodology to deal with bi-objective optimization problems. We do it by adapting the so-called *drop and rebuild* strategy to the bi-objective context, where-from a reference solution, the strategy is applied in order to generate the closest point while optimizing on the least objective. In this case, a random reference (or another created point) is selected by adding the following constraint to the original problem:

$$\max\{z_1, z_2\} - \varepsilon \geq 0, \tag{8}$$

where ε is randomly generated in the interval $\{1, \rho \times 500\}$n with $\rho = \frac{\max\{z_1,z_2\}}{\min\{z_1,z_2\}}$ and, the optimization procedure (described below) is called for providing a new solution. Note that the parameter ρ is used to balance the two objective functions because often the two values can be concurrent and the amplitude between these two values becomes important. Of course, one can observe that the quality of that solution may depend on the power of the used optimizer, and the constraint " $\geq 0''$ means that each time the optimization may provide a solution either with better z_1 or z_2.

Now, we are going to discuss the "guided solution procedure" used for generating the starting points when adding a series of constraints of type (8). As mentioned above, we introduce the so-called *drop and rebuild operator* for providing a series of solutions.

1. - Let $S^{(1)}$ and $S^{(2)}$ be the reference solutions and α be a random parameter generated from the interval $[20\%, 50\%]$.
   - Set the population $\mathcal{P} = \{S^{(1)}, S^{(2)}\}$.

2. Repeat

   (a) Let S be a solution randomly taken from $\mathcal{P}$.

      i. *Drop phase.*
      Remove α random items from S and let $S_{Partial}$ be the partial solution reached.

      ii. *Rebuild phase.*
      Let $N' = N \backslash \{$indices of items belonging to $S_{Partial}\}$.
      - Call `Algorithm 1` for solving the sub-instance with indices N' and, let S_{Com} be the resulting complementary solution.
      - Let $S' = S_{Partial} \cup S_{Com}$ be the new provided solution.

   (b) Set $\mathcal{P} = \mathcal{P} \cup S'$.

3. Until providing a satisfactory set of solutions.

Generally, when using population-based metaheuristics to solve combinatorial optimization problems, it is well-known that different parameter settings (like the population size for the proposed method) lead to a variability in the quality of the final solutions. Limited computational results showed that the population size of 500 (among values tested from 100 to 1000) provided a satisfactory set of Pareto points with a reasonable runtime.

3.7 *Selection and crossover operators*

3.7.1 *Selection*

From the current population, a simple *tournament selection* operator is used to select pairs of individuals. Each individual (parent) may be a candidate for the crossover. In order to select each parent, two individuals are randomly taken from the current population, and then the *best one* between both individuals is chosen. Since a bi-objective optimization problem is considered, we then select the so-called *best individual* by considering a random choice to maintain the diversity of the population. Indeed, one of the following cases is considered, where if the current case holds, then all others are neglected. Let S_A and S_B be two different solutions randomly taken from the current population, then:

1. The solution S_A (resp. S_B) is selected whenever $S_A \succ S_B$ (resp. $S_B \succ S_A$).
2. Let α be a randomly generated binary variable, where:
   (a) with $\alpha = 1$, S_A (resp. S_B) is selected if $z_1(S_A) \geq z_1(S_B)$ (resp. $z_1(S_B) \geq z_1(S_A)$);
   (b) with $\alpha = 0$, S_A (resp. S_B) is selected if $z_2(S_A) \geq z_2(S_B)$ (resp. $z_2(S_B) \geq z_2(S_A)$).

3.7.2 *Crossover:* fusion operator

The crossover operator is one of the main ingredients used in population-based methods (Yi et al., 2020). In our study, from the best selected solution, namely S_A, the crossover operator is replaced with the *fusion operator*, which tries to build a set of diversified solutions. In this case, it tries to provide a new trial solution by building a series of new solutions according to a series of pairs of solutions $(S_A, S_B), S_A \neq S_B$, where S_B belongs to the population at hand.

For knapsack type problems, it has been observed that the selection of some items belonging to diversified solutions leads to good partial solutions. It is therefore interesting to select these items and then to complete the current solution by a deep search, where specialized procedures for knapsack problems may be applied. In order to mimic such a process, we propose to use the so-called greedy random operator, called a Fusion Operator (FO). More precisely, let $S^{(1)}$ and $S^{(2)}$ be the solutions (belonging to the population $\mathcal{P}$) being combined and S_{new} be the new resulting solution that is built as follows:

1. Set $S_{Partial} = S^{(1)} \cap S^{(2)}$, where only items fixed to one in both solutions are stored.

2. For each item fixed to one either in $S^{(1)}$ and not in $S^{(2)}$ or in $S^{(2)}$ and not in $S^{(1)}$, let $\alpha = \text{random}\{0;1\}$ and for each α, add to $S_{partial}$ the index representing α whenever $\alpha = 1$.

3. If $S_{Partial}$ is unfeasible, then order all items according to their densities and, remove all items from the first item of $S_{Partial}$ until satisfying all constraints related to inequalities (3) and (4).

4. Call `Algorithm 2` (and `Algorithm 3`) to optimize the reduced problem P_{red} obtained as follows: let C be all indices out of $S_{Partial}$, i.e., $C = I \setminus S_{Partial}$ and

5. Let $S^{(new)}$ be the new complete solution $S_{Partial}$ and .

6. Return $S^{(new)}$.

3.8 *Drop and complete as a mutation operator*

In this part, we propose an adaptation of the reactive local search, as used in Hifi & Michrafy (2006) for tackling $\text{P}_{\text{BO-QMKP}}$. Such a principle is based upon the *drop* and *complete* solution that combines two strategies: *dropping strategy* and *greedy assignment* strategy used for *completing* the solution at hand. Indeed, let S be a feasible solution at hand and, S_1 and S_0 be two subsets related to S such that $|S_1 \cup S_0| = n$ and $S_1 \cap S_0 = \emptyset$, where S_1 denotes a subset containing the already fixed variables and S_0 is a subset of unfixed (free) variables.

Algorithm 5 − Adaptation of the drop and complete solution procedure for $\text{P}_{\text{BO-QMKP}}$.

Input. A feasible solution S of $\text{P}_{\text{BO-QMKP}}$ with a parameter $\alpha_{\max}$.
Output. A better solution $S^{\star}$.

```
Set S* = S and let α_min ∈ ]0%, 50%[;
repeat
    Set α = α_min;
    repeat
        Let S_1 be the set with (1−α)|S^(1)| variables fixed to 1 in S and, Sol_1 its corresponding partial solution;
        Let S_free = N \ S_1 and Sol_free be a complementary solution reached by Algorithm 1 when applied to the set of free items S_free.
        Set Ŝ = Sol_1 ∪ Sol_free as the new resulting solution.
        Improve Ŝ with Algorithm 2 and, Algorithm 3;
        if (z_1(S*) ≤ z_1(Ŝ)) or (z_2(S*) ≤ z_2(Ŝ)) then
            Set S* = Ŝ and α = α_min;
        else
            Increment(α);
    until (α ≥ α_max)
until (satisfying the stopping condition)
return S*.
```

On the one hand, both S_1 and S_0 of (un)assigned variables can be generated as follows:

1. Suppose that S be a feasible solution at hand;
2. Let α be a percentage generated in the interval $]0\%,\ 50\%[$;
3. Let S_1 be a copy of S, where $\alpha \times |S_1|$ variables are randomly dropped from S; that represents $(1-\alpha)|S|$ variables fixed to 1 in S ($|S_1|$ means the number of component of the solution S fixed to one).
4. Set $S_{free} = N \setminus S_1$, apply `Algorithm 2` to S_{free}, and let S' be the resulting solution containing items either fixed to 1 or 0.
5. Set $\hat{S} = S_1 \cup S'$ as the new solution built.

On the other hand, one can observe that the above process can be added to a descent method, where the percentage α can be initially fixed to a minimum value $\alpha_{\min}$ that can be incremented till matching a maximum value $\alpha_{\max}$.

Of course, in case of the current solution is improved, the algorithm restarts the search process with $\alpha_{\min}$, choosing the greatest value for $\alpha_{\min}$ (increment where the new value belongs to the interval) otherwise. Because the used process mimics a heuristic, we then add a stopping criterion. We note that the drop and complete procedure with its used parameters are analyzed in Section 4.2 (Computational Results).

Algorithm 6 – A Hybrid Population-Based Algorithm for $\mathrm{P}_{\mathrm{BO-QMKP}}$.

Input. An instance of $\mathrm{P}_{\mathrm{BO-QMKP}}$.
Output. A population $\mathcal{P}$ of Pareto solutions for $\mathrm{P}_{\mathrm{BO-QMKP}}$.

Set Iter= 0 and generate both reference solutions $S^{(1)}$ (with z_1) and $S^{(2)}$ with (z_2) (cf. Section 3.6).
Build a starting population $\mathcal{P}_0$ according to both $S^{(1)}$ and $S^{(2)}$ (cf. Section 3.6.2).
repeat
    Apply the tournament selection to $\mathcal{P}_{\mathrm{Iter}}$ for selecting the best individual S_{best} (cf. Section 3.7.1)
    Apply the fusion operator for generating a temporary expanded population $\mathcal{Q}_{\mathrm{Iter}}$ (without solutions of $\mathcal{P}_{\mathrm{Iter}}$), when combining S_{best} with all other solutions of $\mathcal{P}_{\mathrm{Iter}}$ (cf. Section 3.7.2).
    For each solution of $\mathcal{Q}_{\mathrm{Iter}}$, make the mutation operator; set $\mathcal{Q}_{\mathrm{Iter}} = \mathcal{Q}_{\mathrm{Iter}} \cup \mathcal{P}_{\mathrm{Iter}}$ (cf. Section 3.8).
    Make the Pareto ranking on the solutions of $\mathcal{Q}_{\mathrm{Iter}}$.
    Let $\mathcal{P}_{\mathrm{Iter}+1}$ be the new population containing N_{pop} best solutions of $\mathcal{Q}_{\mathrm{Iter}}$.
    Increment(Iter).
until (Iter = $\mathrm{Iter}_{\max}$)
return $\mathcal{P} := \mathcal{P}_{Iter_{\max}}$

3.9 *An overview of the proposed hybrid method*

Algorithm 6 summarizes the general principle of the Hybrid NSGA-II (HNSGA) adapted to the context of bi-objective quadratic multiple knapsack problem. The algorithm begins by building two reference solutions provided by using the adaptation of the branch and solve (Algorithm 4). Next, the starting population of solutions $\mathcal{P}_0$ is created by calling the "guided procedure" (as

described in Section 3.6.2). At each iteration, a tournament selection is applied in order to expand the search space (as detailed in Section 3.7.1). Of course, for each generation, namely Iter, a new population of solutions $\mathcal{Q}_{\text{Iter}}$ is generated by applying the fusion operator (as described in Section 3.7.2) and the so-called mutation operator (as described in Section 3.8) to the current population $\mathcal{P}_{\text{Iter}}$. Then, the intensification strategy, which is based-upon descent method (cf. Algorithm 5) is applied for each generation.

Thereafter, at iteration Iter, both the parent population $\mathcal{P}_{\text{Iter}}$ and the resulting population $\mathcal{Q}_{\text{Iter}}$ are merged to create a global population $\mathcal{R}_{\text{Iter}}$ of size $2.|\mathcal{P}|$, and NSGA-IIs' ranking operators are performed to classify the individuals: such a process follows the standard ranking used for clustering all solutions according to the scores related to their objective functions. The best $N_{\mathcal{P}}$ solutions of the whole population $\mathcal{R}_{\text{Iter}}$ are then selected to form the next parent population, namely $\mathcal{P}_{\text{Iter}+1}$. Such a process is iterated till matching the maximum number of iterations $\text{Iter}_{\max}$; thus, a final approximative Pareto front is reached. As considered for multi-objective optimization problems (Chergui et al., 2008; Ecker & Kouada, 1978; Kantour et al., 2019), the density of the provided Pareto front will be compared to those achieved by available methods of the literature.

4 Computational Results

The objective of this part is to assess the performance of the proposed Hybrid Population-Based Algorithm (namely HPBA) by comparing the results it provides with those obtained by the best available methods published in the literature. Note that all proposed solution procedures were coded in C++ and performed on a computer with an Intel Pentium Core i5 with 2.10 GHz.
Two subparts are considered: (i) a first subpart is devoted to the *qualitative study*, where only a single objective (related to the first function) is considered and, (ii) a second subpart which is related to the *quantitative study*, where both objective functions are considered and in which the density of the Pareto Front (PF) is analyzed.

Table 2: Characteristics of the benchmark instances

Set	$x \in \{3;\ 5;\ 10\}$ and $1 \leq y \leq 5$	
`Set I`	100-25_x_y,	100-75_x_y
	200-25_x_y,	200-75_x_y
`Set II`	300-25_x_y,	300-75_x_y
`Set III`	400-25_x_y,	400-75_x_y
	500-25_x_y,	500-75_x_y

4.1 *Benchmark instances and parameter settings*

In order to evaluate the behavior of HPBA, we used a set of benchmark instances taken from (Aïder et al., 2020a; Chen et al., 2016). These instances are composed of three sets (noted `Set I`, `Set II` and `Set III`), representing a total of 180 instances:

- `Set I`: It contains sixty instances such that each instance was provided by using Hiley & Julstrom (2006)'s generator, where m knapsacks are considered, and the overall capacity is divided by m. It is composed of two types of instances: there are thirty instances for each group, with the number of items n fixed to 100 for the first group and 200 for the second one. The density of each instance varies in the interval $\{0.25, 0.75\}$ (according to the coefficients of the objective function), and the number of knapsacks m varies in the discrete interval $\{3, 5, 10\}$ for each group. The capacity of each knapsack is fixed to 80% of the total weights of each knapsack over the number of available knapsacks.

- `Set II`: It contains thirty large instances (extracted from Chen et al. (2016)), where the number of items n is fixed to 300 and the rest of the information has the same characteristics as considered for `Set I`.

- `Set III`: It is composed of sixty more largest instances (extracted from Aïder et al. (2020b)), where thirty instances are characterized with their number of items $n = 400$ and the other thirty instances with $n = 500$ (the other parameters have the same characteristics as for the previous two sets).

As illustrated in Table 2, the labels of the instances are of the form `n-d_x_y`, where n denotes the number of items, d is the density of the instance, x denotes the number of the knapsack constraints while y represents the variation of the knapsack capacity.

4.2 *Effect of the drop and complete operator*

The Adaptation of the Drop and Rebuild Procedure (Algorithm 5 − ADRP) is used for generating the starting population, which will be used by the proposed hybrid algorithm.

On the one hand, ADRP needs two decision parameters: (i) the criterion used to stop the procedure, and (ii) the number of elements to drop. Its stopping condition is limited to five seconds (retuned after several tunings) while right value for the second parameter α is analyzed in what follows. The algorithm was run by varying the parameter α in the discrete interval $[3, 5, 10, 30, 50]\%$, and by considering other values in $[1,\ 3[\% \cup]30,\ 100[\%$.

On the other hand, ADRP is a stochastic procedure, where each run can obtain a different result. Thus, for each instance, ten trials (runs) are considered. Table 3 reports the global average bounds achieved by ADRP when varying the value of α over the ten trials. Column 1 displays the instance's information, columns from 2 to 7, under the value assigned to α, refer

Table 3: Behavior of ADRP on the instances of `Set I`: variation of α.

	The Enhancing Drop and Complete Solution (EFCS) over ten trials					
#Inst	3%	5%	10%	30%	50%	Others
100-25-3-1	28401.00	28383.05	28375.10	27792.50	27685.75	28194.80
100-25-3-2	27180.00	27312.20	26856.90	26897.00	26605.50	26970.00
100-25-3-3	25893.25	26618.10	25881.00	25764.70	24918.02	26709.00
100-25-3-4	27202.50	26851.30	28171.35	27304.00	27059.40	27776.88
100-25-3-5	26254.00	26611.10	26809.70	25529.18	25542.42	27126.50
100-25-5-1	21144.00	22029.50	21722.15	21478.60	20930.70	21710.02
100-25-5-2	21033.00	21199.01	20316.45	19996.18	19783.90	21692.80
100-25-5-3	21045.00	20908.50	20897.15	20712.35	20475.70	20817.20
100-25-5-4	21449.00	21789.05	20335.80	19985.60	19643.75	20451.09
100-25-5-5	20280.80	20654.10	19370.15	18929.75	18457.60	19251.80
100-25-10-1	14844.75	15232.05	13784.80	13871.78	13715.19	14253.30
100-25-10-2	14694.70	14741.30	13586.35	13466.90	13261.92	14705.70
100-25-10-3	13987.40	13844.87	12645.45	12777.80	12456.00	13480.47
100-25-10-4	14517.65	15425.28	12972.80	13529.00	13539.79	14741.60
100-25-10-5	14097.17	14121.05	12964.75	12391.64	12355.72	13051.00
100-75-3-1	69330.00	69816.00	69752.20	69074.00	69074.00	69708.62
100-75-3-2	68694.00	69192.85	68859.00	68147.00	68056.00	68601.30
100-75-3-3	68330.00	68073.00	68689.50	67628.00	67449.85	68650.05
100-75-3-4	69634.00	69634.00	69628.00	68999.00	68800.00	69535.17
100-75-3-5	68990.00	69224.60	69381.00	68554.00	68304.00	69158.98
100-75-5-1	48978.18	48821.25	48740.74	47990.02	47541.25	48463.60
100-75-5-2	48480.19	48637.90	48537.62	47843.50	47622.84	48441.02
100-75-5-3	47579.40	47635.00	47388.48	46882.25	46765.00	46994.60
100-75-5-4	49883.00	49484.17	49609.02	49139.60	48737.80	48961.18
100-75-5-5	47987.00	47722.95	47501.73	46852.20	46568.00	47638.80
100-75-10-1	28841.41	28851.80	28564.35	27414.17	27570.00	28484.50
100-75-10-2	29912.05	29818.10	29744.00	29113.17	29175.74	29813.00
100-75-10-3	27901.50	27958.25	28608.60	26962.05	27816.00	27685.80
100-75-10-4	30912.68	31012.90	30445.02	29684.27	29357.45	30675.50
100-75-10-5	28114.50	28464.03	28635.80	27967.84	28558.68	28575.18
200-25-3-1	99987.00	99306.17	99132.00	97843.00	98012.87	98606.05
200-25-3-2	106430.18	106385.64	106061.20	105321.00	105564.60	106031.07
200-25-3-3	103240.02	103495.00	103360.90	102875.42	102745.65	103266.16
200-25-3-4	99222.12	99316.00	99150.50	98396.20	98749.85	98820.25
200-25-3-5	101722.46	102026.61	101276.80	99835.40	99814.07	100512.64
200-25-5-1	74312.80	74565.97	74521.50	73765.04	73682.80	74488.90
200-25-5-2	79180.71	79124.36	78504.25	78098.80	78414.18	79113.95
200-25-5-3	76478.70	77088.65	76547.08	76001.85	76128.72	76845.40
200-25-5-4	73009.47	73260.08	73274.26	72581.84	72268.70	73395.43
200-25-5-5	75571.72	75802.90	75212.07	73478.46	73911.27	74767.67
200-25-10-1	51375.46	51001.19	51211.54	50312.60	50304.40	51094.91
200-25-10-2	53054.28	53387.40	52846.05	51945.15	52487.27	53021.94
200-25-10-3	52714.72	52141.26	52241.18	51625.03	51214.95	52514.67
200-25-10-4	50874.40	50462.86	50512.30	49746.10	49932.64	49945.39
200-25-10-5	52251.27	52511.83	52902.46	51894.80	51762.80	52843.50
200-75-3-1	269654.00	269427.00	268543.50	268212.08	268304.60	269401.15
200-75-3-2	256264.17	256444.00	255911.80	255147.00	255467.70	256437.65
200-75-3-3	268797.25	269103.51	268425.27	267815.52	267764.80	268588.19
200-75-3-4	245310.60	246054.10	246065.08	245377.74	244879.64	245697.29
200-75-3-5	278441.76	278697.19	278054.26	277328.37	276891.11	278041.84
200-75-5-1	184507.50	184287.60	184451.17	183863.90	183156.05	184343.40
200-75-5-2	173579.88	173278.90	173814.04	172807.85	172441.43	173337.30
200-75-5-3	185264.03	185414.40	185397.15	185012.43	184545.91	184845.44
200-75-5-4	166064.37	166076.27	166002.02	165742.46	165365.66	166111.10
200-75-5-5	192442.44	191995.82	192131.07	191510.12	191137.70	191867.19
200-75-10-1	112117.05	112628.42	112544.70	111437.60	111323.34	112587.50
200-75-10-2	104012.60	104521.19	104745.27	104031.80	104078.44	104743.25
200-75-10-3	112534.47	113002.08	112678.67	111731.18	111694.70	112398.50
200-75-10-4	97843.00	97912.80	97624.21	96931.50	96329.70	97527.60
200-75-10-5	116311.88	116617.08	116086.50	115371.61	115037.12	116657.80
AV.	82636.04	**82756.73**	82466.73	81812.00	81687.34	82502.88

to the bounds reached, and the last line (Av.) tallies the global average value over all tested instances (`Set I`).

From Table 3, we observe that the proposed algorithm seems more efficient with $\alpha = 5\%$, especially when comparing its provided results to the results reached with the other values of α. Indeed, for $\alpha = 5\%$, the global average solution value is equal to 82756.73 (last line, column 3 of Table 3), which represents a greatest global average bound among all reported values.

Table 4: Results with population size fixed either to 100, 500 or 1000 individuals on instances of `Set I`

#Inst.	PopSize: best bound 100		500		1000	
	z_1	z_2	z_1	z_2	z_1	z_2
I100_25_3_1	29286	6293	29286	6293	29286	6293
I100_25_3_2	28491	7446	28491	7446	28491	7446
I100_25_3_3	27179	8640	27179	8640	27179	8640
I100_25_3_4	28593	7551	28593	7551	28593	7551
I100_25_3_5	27892	8303	27892	8303	27892	8303
I100_25_5_1	22581	2896	22581	2896	22581	2896
I100_25_5_2	21622	3623	**21704**	3621	21622	3623
I100_25_5_3	21239	3256	21239	3256	21239	3256
I100_25_5_4	22181	73379	22181	3379	22181	3379
I100_25_5_5	21669	2904	21669	2904	21669	2904
I100_25_10_1	16095	953	**16221**	942	16026	978
I100_25_10_2	15645	797	15645	797	**15700**	714
I100_25_10_3	14654	971	**14927**	838	14846	964
I100_25_10_4	15988	1017	**16181**	1000	16085	1013
I100_25_10_5	15326	701	15326	701	15326	701
I100_75_3_1	69977	7823	69977	7823	69977	7823
I100_75_3_2	69504	8970	69504	8970	69504	8970
I100_75_3_3	68832	10614	68832	10614	68832	10614
I100_75_3_4	70028	8499	70028	8499	70028	8499
I100_75_3_5	69692	9373	69692	9373	69692	9373
I100_75_5_1	49110	3706	**49421**	3582	49231	3647
I100_75_5_2	49400	3452	49400	3452	49400	3452
I100_75_5_3	21622	3623	21622	3623	**21704**	3621
I100_75_5_4	50246	3692	50246	3692	50246	3692
I100_75_5_5	48753	4170	48753	4170	48753	4170
I100_75_10_1	29857	1112	**30296**	1052	30096	1070
I100_75_10_2	31207	1070	31207	1070	31207	1070
I100_75_10_3	29591	987	**29908**	926	29775	965
I100_75_10_4	31762	1057	31762	1057	31762	1057
I100_75_10_5	48230	4611	**48495**	3816	48272	4590
Average	35541.73	4382.96	**35608.60**	4342.86	35573.16	4375.80

4.3 *Parameter settings*

In order to evaluate the behavior of HPBA, we first focused the preliminary study on the quality of the provided bounds, especially by favoring the first objective function, consisting of maximizing the total profit. Of course, such a choice is not insignificant, because the algorithms designed in the literature are often interested in the aforementioned objective, excepting HTS (Chen & Hao, 2016), while we propose herein a double experimental study: (i) using a single objective function (z_1) and, (ii) considering both objective functions (z_1 and z_2). Further, HPBA uses several parameters, such as population size, probabilities of selection, mutation, and crossover operators. As discussed in Sections 3.7.1 and 3.8, the proposed method applies the following operators:

- Selection: A random binary variable is generated for selecting which objective value is selected for guiding the population generation.
- Crossover: It is replaced with a fusion operator, where a partial solution is built and completed by calling both Algorithm 2 and Algorithm 3.
- Mutation: It is replaced with drop and rebuild operator (discussed in Section 4.2).

With the above tunings, we studied the impact of the population size on the results achieved by HPBA. In this case, three values representing the population size (noted PopSize) have been considered for instances of `Set 1` (this choice is not trivial, because most of the algorithms in the literature have intensively tested the aforementioned instances, where most of the provided bounds are very close). Table 4 shows the provided results for the instances of `Set I`. The first version of the algorithm related to PopSize=100 is noted HPBA$_a$, the second one (with PopSize=500) is noted HPBA$_b$ and the third one (with PopSize=1000) is noted HPBA$_b$. According to the variation of PopSize, we also fixed the runtime limit of each version to 500 seconds, 700 seconds and 1000 seconds, respectively. From Table 4, we observe what follows:

1. HPBA$_a$ (with the population size fixed to 100) can match 73.33% of the best bounds (solution values reached by the three versions of HPBA) and it fails on 8 occasions to reach the best bounds.

2. The percentage of the best bounds achieved by HPBA$_b$ becomes more significant. Indeed, on the one hand, HPBA$_b$ can provide 8 better solutions when compared to both HPBA$_a$ and HPBA$_c$. On the other hand, it achieves all best bounds. In this case, its global average bound is equal to 35608.60, which is greater than that provided by HPBA$_a$, i.e., 35541.73.

3. The results reached by HPBA$_c$ seem also interesting even if the global average bound becomes less than that provided by HPBA$_c$ (and slightly higher than that obtained by HPBA$_a$). In this case, HPBA$_c$ reaches two better solutions and matches the rest of the instances (a total of 93.33%).

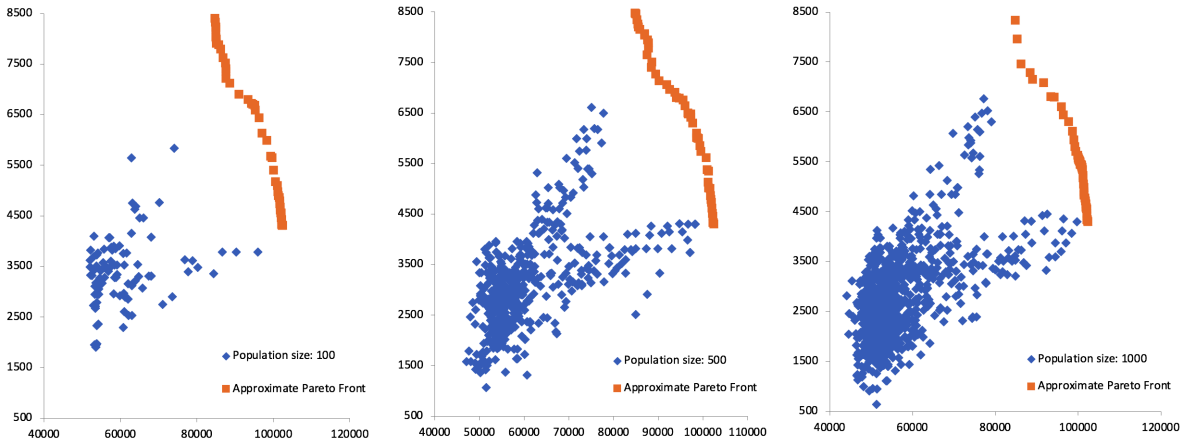

Figure 5: Illustration of the three starting populations and the final approximate Pareto front with varying the population size.

Figure 5 illustrates the variation of the starting population and its resulting Pareto front related to the three versions of the method. Therefore, we can conclude that a larger population size allows us to obtain better solutions thanks to a better exploration of the search space, but the

average runtime becomes more important. With the smallest population size, the percentage of better-achieved bounds remain smaller than that realized with the population size of 500 individuals. In this case, the average runtime remains acceptable for the intermediate value; hence, for the rest of the paper, the population size will be fixed to 500 individuals.

Table 5: Behavior of the five considered methods: HTS, EPR, BS and ε-CSBH and HPBA on instances of `Set I`.

	HTS		EPR		BS		ε-CSBH		HPBA	
#Inst	z_1	z_2	z_1	z_2	z_1	z_2	z_1	z_2	z_1	z_2
I100_25_3_1	29286	6293	29286	6293	29286	6293	29286	6293	29286	6293
I100_25_3_2	28491	7446	28491	7446	28491	7446	28491	7446	28491	7446
I100_25_3_3	27179	8640	27179	8640	27179	8640	27179	8640	27179	8640
I100_25_3_4	28593	9302	28593	7551	28593	7551	28593	7551	28593	7551
I100_25_3_5	27892	8303	27892	8303	27892	8303	27892	8303	27892	8303
I100_25_5_1	22581	2896	22581	2896	22581	2896	22581	2896	22581	2896
I100_25_5_2	21678	2744	21704	3621	21704	3621	21704	3621	21704	3621
I100_25_5_3	21239	3256	21239	3256	21239	3256	21239	3256	21239	3256
I100_25_5_4	22181	3379	22181	3379	22181	3379	22181	3379	22181	3379
I100_25_5_5	21669	2904	21669	2904	21669	2904	21669	2904	21669	2904
I100_25_10_1	16221	942	16221	942	16221	942	16221	942	16221	942
I100_25_10_2	15700	714	15700	714	15700	714	15700	714	15700	714
I100_25_10_3	14927	838	14927	838	14927	838	14927	838	14846	964
I100_25_10_4	16181	1000	16181	1000	16181	1000	16181	1000	16181	1000
I100_25_10_5	15326	701	15326	701	15326	701	15326	701	15276	925
I200_25_3_1	101465	18416	101471	19253	101471	19253	101471	19253	101471	19253
I200_25_3_2	107958	12870	107958	12870	107958	12870	107958	12870	107698	13422
I200_25_3_3	104567	15811	104589	16510	104589	16510	107698	13422	104321	17148
I200_25_3_4	100098	18553	100136	18149	100136	18149	99812	19132	99812	19132
I200_25_3_5	102311	18096	102311	18036	102311	18036	102139	18522	102139	18522
I200_25_5_1	75567	9131	75623	8435	75623	8435	75623	8435	75243	9485
I200_25_5_2	80033	6230	80033	6230	80033	6230	80033	6230	80033	6230
I200_25_5_3	78043	7293	78043	7293	78043	7293	78043	7293	77914	7702
I200_25_5_4	74073	8688	74140	8226	74140	8226	73906	8527	73906	8527
I200_25_5_5	76610	7811	76610	7811	76610	7811	76377	8198	76377	8198
I200_25_10_1	52259	2277	52293	2331	52293	2331	52010	2643	52010	2643
I200_25_10_2	54830	2169	54830	2169	54830	2169	54830	2169	54830	2169
I200_25_10_3	53586	2575	53678	2289	53678	2289	53678	2289	53470	2414
I200_25_10_4	51135	2458	51302	2985	51302	2985	51205	3049	51205	3049
I200_25_10_5	53598	2839	53621	2383	53621	2383	53611	2456	53611	2456
I100_75_3_1	69977	7823	69977	7823	69977	7823	69977	7823	69977	7823
I100_75_3_2	69504	8970	69504	8970	69504	8970	69504	8970	69504	8970
I100_75_3_3	68832	10614	68832	10614	68832	10614	68832	10614	68832	10614
I100_75_3_4	70028	8499	70028	8499	70028	8499	70028	8499	70028	8499
I100_75_3_5	69692	9373	69692	9373	69692	9373	69692	9373	69692	9373
I100_75_5_1	49421	3582	49421	3582	49421	3582	49421	3582	49421	3582
I100_75_5_2	49365	3858	49400	3452	49400	3452	49400	3452	49400	3452
I100_75_5_3	48495	3816	48495	3816	48495	3816	48495	3816	48495	3816
I100_75_5_4	50246	3692	50246	3692	50246	3692	50246	3692	50246	3692
I100_75_5_5	48753	4170	48753	4170	48753	4170	48753	4170	48753	4170
I100_75_10_1	30296	1052	30296	1052	30296	1052	30296	1052	30296	1052
I100_75_10_2	31129	1184	31207	1070	31207	1070	31207	1070	31207	1070
I100_75_10_3	29908	926	29908	926	29908	926	29908	926	29908	926
I100_75_10_4	31762	1057	31762	1057	31762	1057	31762	1057	31762	1057
I100_75_10_5	30465	958	30507	1049	30507	1049	30507	1049	30507	1049
I200_75_3_1	270718	29712	270718	29712	270718	29712	270718	29712	270199	31052
I200_75_3_2	257156	38420	257288	38726	257288	38726	257288	38726	257288	38726
I200_75_3_3	270069	31536	270069	31536	270069	31536	269657	33204	269657	33204
I200_75_3_4	246961	38794	246993	38734	246993	38734	246252	39824	246252	39824
I200_75_3_5	279598	31892	279598	31892	279598	31892	279458	32381	279458	32381
I200_75_5_1	185076	13671	185493	14372	185493	14372	185493	14372	185493	14372
I200_75_5_2	174836	14220	174836	14220	174836	14220	174836	14220	174835	14575
I200_75_5_3	186745	12999	186782	12692	186782	12692	186782	12692	186782	12692
I200_75_5_4	166815	14126	167142	14986	167142	14986	167142	14986	167142	14986
I200_75_5_5	193240	13558	193310	13852	193310	13852	193310	13852	193310	13852
I200_75_10_1	113140	4598	113324	4013	113324	4013	113324	4013	113175	4173
I200_75_10_2	105597	4849	105966	4028	105966	4028	105966	4028	105876	4101
I200_75_10_3	114551	4079	114860	3780	114860	3780	114860	3780	114496	4211
I200_75_10_4	99017	5490	99422	5224	99422	5224	99422	5224	99357	5318
I200_75_10_5	117026	3415	117309	3426	117309	3426	117309	3426	117309	3426
Nb./#inst.	40/60		60/60		60/60		59/60		60/60	
Av.	83728,25	8758,46	83782,43	8729,86	83782,43	8729,86	83790,15	8775,95	83695,61	8920,36

4.4 *Qualitative study*

This section discusses the behavior of HPBA when compared to the best available method of the literature, especially by focusing on the first objective function. There are three parts, where each of them is related to the set of instances considered: Section 4.4.1 for the `Set I`, Section 4.4.2 for the `Set II` while Section 4.4.3 analyzes the performance of HPBA on the third

set of instances `Set III`. Because the proposed method needs to fix the runtime limit, we just limited the number of iterations to 25 000 for all sets. In this case, the observed average runtime varies from 150 seconds to 1700 seconds, depending on the complexity of the instance tested.

4.4.1 *Behavior of HPBA versus four methods of the literature:* `Set I`

In this part, the performance of HPBA is evaluated and analyzed on the first set (`Set I`) containing sixty instances (divided into two groups). HPBAs' results are compared to those achieved by the four published methods:

- Hybrid Two-Stage (Chen & Hao (2016) − noted HTS),
- Evolutionary-Path Relinking (Chen et al. (2016) − noted EPR),
- Branch and Solve (Aïder et al. (2020a) − noted BS), and
- ε-Constraint Strategy Based Heuristic (Aïder et al. (2020b) − noted ε-CSBH).

We note that all results are extracted from (Aïder et al., 2020b).

Table 5 reports the provided results of all compared algorithms on the set of instances `Set I`; in this case, for each tested algorithm, both objective values (z_1, z_2) are reported. Column 1 presents the instance label (including the number of items n, the density d, and the number of knapsacks m). Columns 2 and 3 display HTS' bounds while columns 4 and 5 (resp. 6 and 7 and, 8 and 9) display those achieved by EPR (resp. BS and, ε-CSBH). Columns 10 and 11 report the bounds achieved by the proposed method HPBA while the last two lines report the number of times that the current method matches the best bounds and its global average bounds, respectively.

Because of the random aspect of HPBA, we then considered ten trials. We note that the value in "bold-face" means that the best solution value has been reached by the considered algorithm. From Table 5, we observe what follows:

- HPBA versus HTS method: HPBA dominates HTS on 10 occasions and it matches the rest of the bounds.
- HPBA versus EPR, BS and ε-CSBH: the four methods remain competitive since globally all methods can match the same bounds, except for the instance `200-75-5-2` for which HPBA provides a new bound (a new non-dominated point) for the second objective value (z_2).

4.4.2 *Behavior of HPBA versus four methods of the literature:* `Set II`

The second set of instances (`Set II`) was used (tested) by the following three algorithms of the literature: EPR, BS, and ε-CSBH. We then compare their provided results to those achieved by HPBA on all instances of that set.

Table 6 reports the results obtained by HPBA and the other three algorithms (as used in section 4.4.1). The columns of the table 6 have the same meaning as in the previous table. From this table, we observe what follows:

- ε-CSBH versus EPR and BS: For the objective function z_1, ε-CSBH dominates both EPR and BS on several occasions. Indeed, it dominates BS on 18 occasions and on 25 occasions when compared to the results achieved by EPR. For the objective function z_2, the ε-CSBH achieves a better average bound, i.e., 27355.43, compared to 27045.26 (resp. 27044.40) achieved by EPR (resp. BS).

- HPBA versus ε-CSBH: HPBA performs better than ε-CSBH. It is able to reach 4 new points (Pareto solutions) with both objective functions (z_1, z_2): instances `I300-25_10_3`, `I300-75_5_2`, `I300-75_10_2` and `I300-75_10_4` (as shown in Table 6)

Table 6: Performance of HPBA versus both BS and ε-CSTBH on the first group of `Set III`.

	BS	ε-CSTBH		HPBA	
#Inst	z_1	z_1	z_2	z_1	z_2
400-25-3-1	325978	326185	49254	326185	49254
400-25-3-2	315186	315393	54210	315277	54724
400-25-3-3	305532	306059	58098	306096	56369
400-25-3-4	324030	324775	57645	325032	57352
400-25-3-5	307315	**315805***	**53079***	315692	52788
400-25-5-1	242946	242084	20850	242373	22494
400-25-5-2	227374	228153	25447	228405	25440
400-25-5-3	226144	225017	22926	225161	22736
400-25-5-4	230912	231241	23648	231350	22400
400-25-5-5	229338	229645	21195	229795	20533
400-25-10-1	158601	156792	6386	157079	6753
400-25-10-2	147474	147410	6966	147555	6481
400-25-10-3	148183	145945	6335	146525	5614
400-25-10-4	153410	152527	6968	152922	6968
400-25-10-5	150742	149886	6976	**151088***	**7191***
400-75-3-1	895360	896240	121247	**896460***	**121465***
400-75-3-2	982436	981731	110181	981731	110014
400-75-3-3	913992	913934	117526	914000	116908
400-75-3-4	874363	874443	118885	**874539***	**120054***
400-75-3-5	940763	939655	124413	940364	120063
400-75-5-1	627864	627649	44850	**628931***	**45108***
400-75-5-2	684177	684192	44345	683897	44603
400-75-5-3	635732	636061	39632	635758	42219
400-75-5-4	616307	**617575***	**50897***	616909	49102
400-75-5-5	648835	**649992***	**51676***	649657	49350
400-75-10-1	372751	**378812***	**10149***	377008	9987
400-75-10-2	403730	408482	10274	**410695***	**11534***
400-75-10-3	369465	370956	8818	**372247***	**11995***
400-75-10-4	357526	361592	11393	**362680***	**11507***
400-75-10-5	376407	384445	10850	**384456***	**11429***
Av.	439762.43	440755.86	43170.63	440995.56	43081.16
ND./#inst.	18/30	22/30		26/30	

4.4.3 *Performance of HPBA on more complex instances:* `Set III`

In this section, we studied the behavior of HPBA on the more complex instances of `Set III`, which contains sixty instances. As discussed above, `Set III` is composed of two groups: (i) a first group containing thirty instances with $n = 400$ and the second group containing the rest of the instances with $n = 500$.

Table 7: Performance of HPBA versus both BS and ε-CSTBH on the second group of `Set III`.

	BS (Aïder et al., 2020a)	ε-CSTBH (Aïder et al., 2020b)		HPBA (This work)	
#Inst	z_1	z_1	z_2	z_1	z_2
500-25-3-1	487014	487114	57477	487200	56270
500-25-3-2	486271	486262	64583	486496	64040
500-25-3-3	482382	**482889***	**67383***	482814	65949
500-25-3-4	484119	484920	61904	**485001***	**63605***
500-25-3-5	478455	478727	63168	**478859***	**63589***
500-25-5-1	347263	347226	30905	347623	29284
500-25-5-2	344622	345523	25768	346202	25340
500-25-5-3	341223	342173	30432	**342186***	**30432***
500-25-5-4	342482	343385	34071	**343733***	**35312***
500-25-5-5	341624	341850	30667	**342878***	**31541***
500-25-10-1	219343	221121	8850	**221924***	**9159***
500-25-10-2	223344	223987	8755	**224724***	**10097***
500-25-10-3	219401	222575	9779	**223384***	**10192***
500-25-10-4	217238	219308	10328	220058	10079
500-25-10-5	220552	221920	9518	221920	9518
500-75-3-1	1391636	1390073	151730	1391467	147973
500-75-3-2	1441764	1441256	133560	1441310	134359
500-75-3-3	1406773	1406251	141065	1406701	146034
500-75-3-4	1356161	1356036	163115	1354988	169391
500-75-3-5	1437572	1437652	147308	1438384	142433
500-75-5-1	949075	950005	59856	950045	59725
500-75-5-2	994837	993441	62180	993649	64003
500-75-5-3	958707	**958728***	**66058***	958389	65912
500-75-5-4	918690	918194	71390	918278	74806
500-75-5-5	987326	985113	60122	986991	57709
500-75-10-1	568311	566975	21304	569325	18341
500-75-10-2	592648	594022	18974	595119	18549
500-75-10-3	565467	566152	18917	**566959***	**19978***
500-75-10-4	537670	538005	23721	539152	21782
500-75-10-5	582972	582685	17146	**583543***	**177331***
Av.	664164.73	651111.93	55667.8	621940.86	61091.10
ND./#inst.	18/30	20/30		28/30	

In what follows, we comment on Tables 6 and 7 which report the results achieved by HPBA and both BS and ε-CSTBH:

- HPBA versus BS: HPBA is very competitive when compared to BS. Indeed, it can reach 44 new bounds according to the first objective function (z_1) which represents, in this case, a percentage of 73% of improved bounds over the sixty instances tested.
- HPBA versus ε-CSBH: HPBA remains competitive when compared to ε-CSBH. Indeed, HPBA achieves 19 new bounds, matches 35 bounds, and fails on 6 occasions. Globally, HPBA realizes 31.67% of new upper bounds when compared to those achieved by ε-CSBH (10%), and its matches 58.33% (the rest) of the instances of `Set III`.

Table 8: Performance of HPBA versus both BS and ε-CSTBH on the more complex instances of `Set III`.

	BS (Aïder et al., 2020a)	ε-CSTBH (Aïder et al., 2020b)		HPBA (This work)	
`Set III`	z_1	z_1	z_2	z_1	z_2
Av.	**551963.58**	545933.90	49419.21	531468.21	**52086.13**
ND | #inst.	/	39 | 60		**54** | 60	

Finally, Table 8 summarizes the number of times that each tested method (HPBA, BS and

ε-CSTBH) is able to reach the better bound (line 1: Av.) and the global average bound over all the sixty instances of `Set III`. On the one hand, from the first line (according to the global average bound) there is no-dominance between both HPBA and ε-CSTBH. On the other hand, because we are seeking for a maximum number of the best bounds reached by the used algorithm, one can observe that HPBA can match 90% of the best bounds whereas ε-CSTBH matches 65% of the bounds. Moreover, HPBA can reach 19 new dominated solutions and it fails in 6 occasions to match the solutions reached by ε-CSTBH.

4.4.4 Statistical analysis: HPBA versus BS and ε-CSTBH

Furthermore, in order to evaluate the behavior of the proposed HPBA (when compared to other algorithms), we propose a statistical analysis where both the *sign test* and the *Wilcoxon signed-rank test* statistics are considered. We then use the following hypothesis:

$$H_0\text{: Algo}_1 \text{ - Algo}_2 = \mu,$$

to express that algorithm Algo_1 performs better than Algo_2 and, the hypothesis $\overline{H}_0$: algorithm Algo_2 is better than Algo_1 to express the rejection of the hypothesis H_0. Herein, the greatest the average bound and the greater the number of better bounds, the better the corresponding version of Algo.

Table 9: p-values for both *Sign test* and *Wilcoxon rank-test* on all instances (Set I, Set II and Set III) with the significance level $\theta = 0.05$.

	BS vs ε-CSTBH	BS vs HPBA	ε-CSTBH vs HPBA
p-value (Sign test)	0.999	0.829	0.745
N^+	32	39	23
N^-	39	44	47
$N^=$	49	37	50
p-value (Wilcoxon)	0.999	0.954	0.995

From Table 9 one can observe what follows:

- For BS vs ε-CSTBH: the p-value related to the sign test (resp. Wilcoxon test) is greatest to the significance level $\beta = 0.05$, indicating that ε-CSTBH performs better than BS (rejecting the hypothesis H_0). The number of times that BS provide best bounds for z_1 (the values related to N^+ for BS) is equal to 32 while that related to ε-CSTBH is equal to 39.
- BS vs HPBA: HPBA dominates BS since the p-value related to the sign test (resp. Wilcoxon test) is equal to 0.829 (resp. 0.954); that is greatest to the significance level

$\beta = 0.05$ (rejecting the hypothesis H_0). I this case, HPBA matches better number of best solution (44) than that of BS (39).

- ε-CSTBH vs HPBA: HPBA outperforms ε-CSTBH, since the sign test (resp. Wilcoxon test) is equal to 0.745 (resp. 0.995), which induces the rejection of the hypothesis H_0. Further, HPBA provides 47 best solutions when compared to the 23 realized by ε-CSTBH.

Therefore, although the HPBA method is not specialized for the single-objective version of the problem, it remains competitive with recent methods in the literature.

4.5 *Quantitative study*

In this section, a comparative study between ε-CSTBH and HPBA is proposed by using a performance indicator related to the bi-objective optimization problems. Although there are several performance indicators dedicated to analyzing the behavior of a given method, often the majority of studies focus on two types of indicators: (i) the *Hypervolume Indicator* (Cao et al., 2015) and, (Zitzler & Thiele, 1999), the *Epsilon-Indicator* (Zhao et al., 2003), the binary coverage indicator (Farmani et al., 2002) and the net-front contribution indicator (García-León et al., 2019).

4.5.1 *The density of the Pareto front*

Herein, in order to assess the performance of both HPBA and ε-CSTBH, we first use the *Hypervolume Indicator*, because it allows us to measure the density of the Pareto front (as a relative quality) in terms of both convergence and diversity of the final solutions. In this case, the normalized hypervolume performance indicator is computed according to their Pareto fronts obtained for both ε-CSTBH and HPBA.

Of course, for both computed measures, a unique "*reference point*" was used; that is the origin point $(0, 0)$ representing the two worst objective values for z_1 and z_2. Whenever the hypervolume related to each method is calculated, each of them is normalized by dividing each of them by the greatest value among both values. Table 10 shows the values related to the normalized hypervolume indicator for both versions of the algorithm. Column 1 of Table 10 reports the instance's label, column 2 tallies the normalized hypervolume indicator related to ε-CSTBH (denoted $I_{NH}^{(1)}$) whereas column 3 displays the hypervolume indicator related to HPBA (denoted $I_{NH}^{(2)}$).

From Table 10, we observe what follows.

- $I_{NH}^{(2)}$ value (related to the proposed method HPBA) is better than that $I_{NH}^{(1)}$ (related to ε-CSTBH) in 25 occasions over the thirty instances of `Set II` and it fails in 5 occasions to match the same normalized hypervolume measures.

Table 10: Hypervolume indicator values (normalized) of both ε-CSTBH and HPBA on instances of `Set II`.

#Inst.	Normalized Hypervolume $I_{NH}^{(1)}$	$I_{NH}^{(2)}$
I300_25_3_1	1	0.999898097
I300_25_3_2	0.998105203	1
I300_25_3_3	0.999865547	1
I300_25_3_4	0.998048473	1
I300_25_3_5	0.996629612	1
I300_25_5_1	1	0.983738565
I300_25_5_2	0.996914384	1
I300_25_5_3	0.995211207	1
I300_25_5_4	1	0.99963126
I300_25_5_5	0.99737228	1
I300_25_10_1	0.997548289	1
I300_25_10_2	0.998019678	1
I300_25_10_3	0.995506044	1
I300_25_10_4	0.993843728	1
I300_25_10_5	0.992423294	1
I300_75_3_1	0.99978488	1
I300_75_3_2	0.99993991	1
I300_75_3_3	1	0.9885468
I300_75_3_4	0.999744294	1
I300_75_3_5	0.999697785	1
I300_75_5_1	0.99063368	1
I300_75_5_2	0.997004272	1
I300_75_5_3	1	0.998818002
I300_75_5_4	0.98888497	1
I300_75_5_5	0.984031502	1
I300_75_10_1	0.977488808	1
I300_75_10_2	0.991828015	1
I300_75_10_3	0.99571554	1
I300_75_10_4	0.994973732	1
I300_75_10_5	0.997372127	1
Average	0.995886242	**0.999021091**

- Globally, ε-CSTBH's global average (normalized) hypervolume indicator is equal to 0.995886242 while HPBA realizes a greater global average hypervolume indicator of 0.999021091 (in the bold-space). As both indicators are normalized, one can easily compute the gap between both hypervolumes: in this case, the average hypervolume is equal to 0.997453666 and so, the average normalized gap between both methods is equal to 0.003134849.
- In terms of percentage, HPBA can provide 83.33% of best-normalized indicators, which can be considered as significant, especially for large-scale instances.

Finally, Figure 6 shows the variation of the (normalized) hypervolume indicator, related to the (normalized) density of Pareto front, achieved by both compared methods (ε-CSTBH $-$*black plot*$-$ and HPBA $-$*gray plot*$-$), on the instances of `Set II` while Figure 7 illustrates the

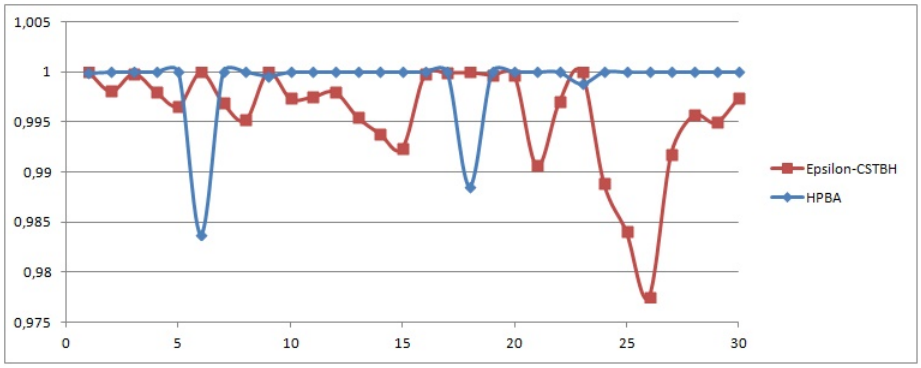

Figure 6: Variation of the normalized hypervolume for both ε-CSTBH and HPBA on instances of `Set II`.

starting population and the approximate Pareto front on two large-scale instances of `Set III`: instances "`inst-300-25-10-2`" (on the left-hand of the figure) and "`inst-300-25-10-3`" (on the right-hand of the figure).

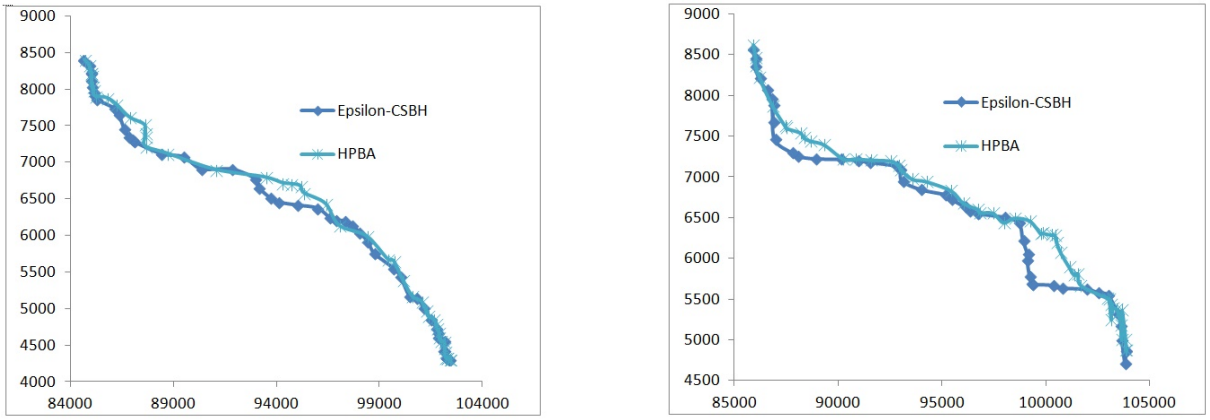

Figure 7: Illustration of the final approximate Pareto front achieved by both ε-CSTBH and HPBA on two instances "`inst-300-25-10-3`" and "`inst-300-25-10-2`" of `Set II`.

4.5.2 *Other indicators on some instances of* `Set II`

In addition to the indicator studied in Section 4.5, we propose to study the behavior of both ε-CSTBH and HPBA on some instances of `Set II`. In this case, such a comparison is related to both (i) the binary coverage indicator and, (ii) the net-front contribution indicator.

The binary coverage indicator For a given instance I, let $S_A(I)$ (resp. $S_B(I)$) be the set of solutions representing the Pareto front when applying algorithm A (resp. B). The binary coverage measure indicator between methods A and B may be represented by the number of solutions reached by B which dominate those provided by A.

Herein, we are looking (i) for the number of solutions achieved by ε-CSTBH dominating

those provided by HPBA and, (ii) those achieved by HPBA that dominate those of ε-CSTBH. Moreover, because both approximate Pareto fronts are available, we then also compare the cardinalities related to the non-dominated solutions achieved by both ε-CSTBH and HPBA.

Table 11: Variation of the binary coverage measure and the cardinality of the non-dominated solutions achieved by ε-CSTBH and HPBA.

	ε-CSBH		HPBA	
#Inst.	$I^{(1)}_{BCM}$	$\mathrm{C}^{(1)}$	$I^{(2)}_{BCM}$	$\mathrm{C}^{(2)}$
inst-300-25-10-3	116	41	**27**	**49**
inst-300-25-10-2	96	**47**	**56**	45
inst-300-75-10-2	523	44	**512**	**53**
inst-300-75-10-4	649	45	**557**	**52**

Table 11 reports the comparative study between both methods, where column 1 shows the instance's label, columns 2 and 3 tally the binary coverage indicator ($I^{(1)}_{BCM}$) and the number of non-dominated solutions of the approximate Pareto front ($\mathrm{C}^{(1)}$) related to ε-CSTBH whereas columns 4 and 5 display the binary coverage indicator ($I^{(2)}_{BCM}$) and the number of non-dominated solutions of the approximate Pareto front ($\mathrm{C}^{(2)}$) related to HPBA.

From Table 11, one can observe that HPBA has a better behavior than that ε-CSTBH. On the one hand, HPBA is able to provide a better indicator coverage since for all instances $\mathrm{C}^{(2)} > C^{(1)}$. On the other hand, according to the cardinality of the approximate Pareto front, HPBA dominates in four occasions ε-CSTBH, except for the instance `inst-300-25-10-2` for which both $I^{(1)}_{BCM}$ and $I^{(2)}_{BCM}$ are closest. We note that, in this case, HPBA provides a better final solution.

The net front contribution indicator Let I be an instance of the problem and, $S_A(I)$ (resp. $S_B(I)$) be the set of no-dominated solutions achieved by algorithm A (resp. B). The Net Front Contribution $NFC(S_A, S_A)$ denotes the subset of S_A belonging to $S_A \cup S_B$ and, $\overline{NFC}(exp_I, exp_J)$ denotes the average value of $NFC(S_{exp_I}, S_{exp_J})$ according to all considered instances. In this case, greater the value of S_{exp_i}, more the algorithm becomes interesting.

Table 12: Variation of the net contribution measure on the non-dominated solutions achieved by ε-CSTBH and HPBA.

	Net front contribution	
#Inst.	$I^{(1)}_{NFC}$	$I^{(2)}_{NFC}$
inst-300-25-10-3	0.45	**0.54**
inst-300-25-10-2	**0.51**	0.50
inst-300-75-10-2	0.45	**0.54**
inst-300-75-10-4	0.46	**0.57**
Av.	0.47	**0.54**

Table 12 shows the comparative study between both ε-CSBH and HPBA, where column 1

displays the instance's label, column 2 (resp. column 3) reports the net front contribution indicator $I^{(1)}_{NFC}$ (resp. $I^{(2)}_{NFC}$) related to ε-CSTBH (resp. HPBA).

We observe from Table 12 that HPBA performs better than ε-CSTBH. Indeed, HPBA can provide the best average net-front contribution indicator (0.54) when compared to that provided by ε-CSTBH (0.47). In this case, HPBA dominates ε-CSTBH on 3 occasions and, fails for the instance `inst-300-25-10-2` for which both related indicators are closest.

5 Conclusion

In this paper, we investigated the use of a hybrid multi-objective evolutionary algorithm for solving the bi-objective quadratic multiple knapsack problem. The proposed algorithm is based upon the so-called non-dominated sorting operator, where both the general profit and the min-max criterion on the second objective function are optimized. A starting population of solutions were provided by combining two starting solutions: a fist solution provided by a special ε-constraint heuristic and, (ii) a second solution achieved by an adaptive local-branching-based heuristic. However, in order to highlight the quality of the solutions reached, a drop/rebuild strategy was incorporated into the search process. Computational results showed that the proposed method remains competitive (in term of quality of the achieved bounds), especially when comparing its provided bounds to those reached by the more recent methods published in the literature. Its performance was also analyzed by using several indicators employed for evaluating multi-objective methods, where it demonstrated its effectiveness on the majority of considered instances. The knapsack problem and its variants are gaining renewed interest in recent years, especially (multi-)knapsacks with setups, and multi-objective versions. In this case, there are plenty of possibilities for further investigation involving powerful and efficient algorithms: (i) tailored heuristics and general purpose evolutionary algorithms. A first research direction, for the studied problem, can be focused on adding successive valid constraints to the original problem,where its goal is to drive the search process toward specific regions for a deep exploration. Second, there are several evolutionary algorithms which can be investigated for tackling the studied problem, like monarch butterfly optimization (MBO), earthworm optimization algorithm (EWA), elephant herding optimization (EHO) and others. Finally, almost of using simple adaptations, we are also interested in hybridization of evolutionary algorithms with tailored procedures, where its goal is to introduce these procedures as a learning strategy for augmenting the quality of the solutions and / or enhancing the density of the final approximate Pareto front.

Acknowledgement

We would like to thank anonymous referees for their relevant comments and suggestions that allowed us to improve the content of the paper.

Author contribution statements

- In this paper, all authors are listed in alphabetical order and all the work was supervised by Pr Mhand Hifi.
- Mhand Hifi proposed the main idea related to this work. He designed the first evolutionary algorithm combined with the ε-constraint procedure-based strategy.
- Oussama Gacem (Phd student under the supervision of Pr Mhand Hifi) coded the first version of the algorithm which is based upon the above idea. He tested the final version of the code on a set of benchmark instances of the literature.
- Méziane Aider is the second co-advisor of Oussama Gacem.
- Mhand Hifi investigated the sensitivity analysis, and extended the experimental part with a statistical analysis.
- All authors discussed the results and Mhand Hifi provided the final version of the manuscript.

References

Aïder, M., Gacem, O., & Hifi, M. (2020a). Branch and solve strategies-based algorithm for the quadratic multiple knapsack problem. *Journal of the Operational Research Society*, (pp. 1–18). doi:10.1080/01605682.2020.1843982.

Aïder, M., Gacem, O., & Hifi, M. (2020b). A two-stage ε-constraint strategy-based heuristic for bi-objective quadratic multiple knapsack problems. In *2020 7th International Conference on Soft Computing Machine Intelligence (ISCMI)* (pp. 51–55). doi:10.1109/ISCMI51676.2020.9311581.

Akbar, M.-M., Rahman, S., Kaykobad, M., Manning, E.-G., & Shoja, G.-C. (2006). Solving the multidimensional multiple-choice knapsack problem by constructing convex hulls. *Computers & Operations Research*, *33*, 1259–1273. doi:https://doi.org/10.1016/j.cor.2004.09.016.

Bederina, H., & Hifi, M. (2018). A hybrid multi-objective evolutionary optimization approach for the robust vehicle routing problem. *Applied Soft Computing*, *71*, 980–993. doi:10.1016/j.asoc.2018.07.014.

Billionnet, A., & Soutif, E. (2004). An exact method for the 0-1 quadratic knapsack problem based on lagrangian decomposition. *European Journal of Operational Research*, *157*, 565–575. doi:10.1016/S0377-2217(03)00244-3.

Boukhari, S., Dahmani, I., & Hifi, M. (2022). Computational power of a hybrid algorithm for solving the multiple knapsack problem with setup. In A. K. (Ed.), *Intelligent Computing, Lecture Notes in Networks and Systems*. volume 283. doi:doi.org/10.1007/978-3-030-80119-9_7.

Cao, Y., Smucker, B.-J., & Robinson, T.-J. (2015). On using the hypervolume indicator to compare pareto fronts: Applications to multi-criteria optimal experimental design. *Journal of Statistical Planning and Inference*, *160*, 60–74. doi:10.1016/j.jspi.

Che, Z.-H., Chiang, T.-A., & Lin, T.-T. (2021). A multi-objective genetic algorithm for assembly planning and supplier selection with capacity constraints. *Applied Soft Computing*, *101*, 107030. doi:10.1016/j.asoc.2020.107030.

Chen, Y., & Hao, J.-K. (2015). Iterated responsive threshold search for the quadratic multiple knapsack problem. *Annals of Operations Research*, *226*, 101–131. doi:10.1007/s10479-014-1720-5.

Chen, Y., & Hao, J.-K. (2016). The bi-objective quadratic multiple knapsack problem: Model and heuristics. *Knowledge-Based Systems*, *97*, 89–100. doi:10.1016/j.knosys.2016.01.014.

Chen, Y., Hao, J.-K., & Glover, F. (2016). An evolutionary path relinking approach for the quadratic multiple knapsack problem. *Knowledge-Based Systems*, *92*, 23–34. doi:10.1016/j.knosys.2015.10.004.

Chergui, M.-E.-A., Moulaï, M., & Ouaïl, F.-Z. (2008). Solving the multiple objective integer linear programming problem. In *Le Thi H.A., Bouvry P., Pham Dinh T. (eds) Modelling, Computation and Optimization in Information Systems and Management Sciences. MCO 2008*. Springer, Berlin, Heidelberg volume 14. doi:10.1007/978-3-540-87477-5_8.

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation*, *6*, 182–197. doi:10.1109/4235.996017.

Ecker, J.-G., & Kouada, I.-A. (1978). Finding all efficient extreme points for multiple objective linear programs. *Mathematical Programming*, *14*, 249–261. doi:https://doi.org/10.1007/BF01588968.

Farmani, R., Savic, D.-A., & Walters, G.-A. (2002). Evolutionary multi-objective optimization in water distribution network design. *Engineering Optimization*, *37*, 167–183. doi:10.1080/03052150512331303436.

Feng, Y., Wang, G.-G., Dong, J., & Gao, X.-Z. (2016). A novel hybrid cuckoo search algorithm with global harmony search for 0-1 knapsack problems. *International Journal of Computational Intelligence Systems*, (pp. 1174–1190). doi:10.1080/18756891.2016.1256577.

Feng, Y., Wang, G.-G., Dong, J., & Wang, L. (2018a). Opposition-based learning monarch butterfly optimization with gaussian perturbation for large-scale 0-1 knapsack problem. *Computers & Electrical Engineering*, *67*, 454–468. doi:10.1016/j.compeleceng.2017.12.014.

Feng, Y., Wang, G.-G., Dong, J., & Wang, L. (2018b). Solving randomized time-varying knapsack problems by a novel global firefly algorithm. *Engineering with Computers volume*, *34*, 621–635. doi:10.1007/s00366-017-0562-6.

Feng, Y., Yu, X., & Wang, G.-G. (2018c). Binary moth search algorithm for discounted 0-1 knapsack problem. *IEEE Access*, *6*, 10708 – 10719. doi:10.1109/ACCESS.2018.2809445.

Feng, Y., Yu, X., & Wang, G.-G. (2019). A novel monarch butterfly optimization with global position updating operator for large-scale 0-1 knapsack problems. *Mathematics 2019*, *7*, 1056. doi:10.3390/math7111056.

Feng, Y., Yu, X., Wang, G.-G., Deb, S., Lu, M., & Zhao, X.-J. (2017). Solving 0–1 knapsack problem by a novel binary monarch butterfly optimization. *Neural Computing and Applications*, *28*, 1619–1634. doi:10.1007/s00521-015-2135-1.

García-León, A.-A., Dauzère-Pérès, S., & Matì, Y. (2019). An efficient pareto approach for solving the multi-objective flexible job-shop scheduling problem with regular criteria. *Computers & Operations Research*, *108*, 187–200. doi:10.1016/j.cor.2019.04.012.

Gu, Z.-M., & Wang, G.-G. (2020). Improving nsga-iii algorithms with information feedback models for large-scale many-objective optimization. *Future Generation Computer Systems*, *107*, 49–69. doi:10.1016/j.future.2020.01.048.

Haghighia, B.-B., Taherinia, A.-H., Harati, A., & Rouhani, M. (2021). An optimized multipurpose blind watermarking in shearlet domain using mlp and nsga-ii. *Applied Soft Computing*, *101*. doi:10.1016/j.asoc.2020.107029.

Hifi, M. (2004). Exact algorithms for unconstrained three-dimensional cutting problems: a comparative study. *Computers & Operations Research*, *31*, 657–674. doi:10.1016/S0305-0548(03)00019-4.

Hifi, M., & Michrafy, M. (2006). A reactive local search-based algorithm for the disjunctively constrained knapsack problem. *Journal of the Operational Research Society*, *57*, 718–726. doi:10.1057/palgrave.jors.2602046.

Hiley, A., & Julstrom, B. (2006). The quadratic multiple knapsack problem and three heuristic approaches to it. In *Proceedings of the 8th annual conference on Genetic and evolutionary computation (GECCO'06)* (pp. 547–552). doi:10.1145/1143997.1144096.

Hung, M.-S., & Fisk, J.-C. (1978). An algorithm for 0-1 multiple-knapsack problems. *Naval Research Logistics Quarterly*, *25*, 571–579. doi:https://doi.org/10.1002/nav.3800250316.

Kantour, N., Bouroubi, S., & Chaabane, D. (2019). A parallel moea with criterion-based selection applied to the knapsack problem. *Applied Soft Computing*, *80*, 358–373. doi:10.1016/j.asoc.2019.04.005.

Kellerer, H., Perschy, U., & Pisinger, D. (2004). Knapsack problems. *Springer*, .

Martello, S., & Toth, P. (1990). *Knapsack Problems: Algorithms and Computer Implementations*. John Wiley & Sons Ltd.

Merkle, M., & Hellman, M. (1978). Hiding information and signatures in trapdoor knapsacks. *IEEE Transactions on Information Theory*, *24*, 525–530. doi:10.1109/TIT.1978.1055927.

Perboli, G., Gobbato, L., & Perfetti, F. (2014). Packing problems in transportation and supply chain: new problems and trends. *Procedia - Social and Behavioral Sciences*, *111*, 672–681. doi:10.1016/j.sbspro.2014.01.101.

Plata-Gonzalez, L.-F., Amaya, I., Ortiz-Bayliss, J.-C., Conant-Pablos, S.-E., Terashima-Marin, H., & Coello, C.-A. C. (2019). Evolutionary-based tailoring of synthetic instances for the knapsack problem. *Soft Computing*, *23*, 12711–12728. doi:10.1007/s00500-019-03822-w.

Saraç, T., & Sipahioglu, A. (2007). A genetic algorithm for the quadratic multiple knapsack problem. In *International Symposium on Brain, Vision, and Artificial Intelligence* 4729 (pp. 490–498). Springer, Berlin, Heidelberg. doi:https://doi.org/10.1007/978-3-540-75555-5_47.

Wang, G.-G., Cai, X., Cui, Z., Min, G., & J.Chen (2020). High performance computing for cyber physical social systems by using evolutionary multi-objective optimization algorithm. *IEEE Transactions on Emerging Topics in Computing*, *8*, 20–30. doi:10.1109/TETC.2017.2703784.

Yi, J.-H., Xing, L.-N., Wang, G.-G., Dong, J., Vasilakos, A.-V., Alavi, A.-H., & Wang, L. (2020). Behavior of crossover operators in nsga-iii for large-scale optimization problems. *Information Sciences*, *509*, 470–487. doi:10.1016/j.ins.2018.10.005.

Zhang, Y., Wang, G.-G., Li, K., Yeh, W.-C., M.Ji, & Dong, J. (2020). Enhancing moea/d with information feedback models for large-scale many-objective optimization. *Information Sciences*, *522*, 1–16. doi:10.1016/j.ins.2020.02.066.

Zhao, F., Asmus, T.-N., Assanis, D.-N., Dec, J.-E., Eng, J.-A., & Najt, P.-M. (2003). *Homogeneous charge compression ignition (HCCI) engines*. SAE International. URL: `https://doi.org/1000001356`.

Zitzler, E., & Thiele, L. (1999). Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. *IEEE transactions on Evolutionary Computation*, *3*, 257–271. doi:10.1109/4235.797969.