INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
Online ISSN 1841-9844, ISSN-L 1841-9836, Volume: 21, Issue: 1, Month: February, Year: 2026
Article Number: 7267, https://doi.org/10.15837/ijccc.2026.1.7267

CCC Publications

Solution Algorithm Involving Heuristics and Criterion Set Dissection to Solve Large Scale Multicriteria Knapsack Problems

N. Mladenović, B. Stanojević

Nada Mladenović*
Faculty of Organizational Sciences, University of Belgrade
Jove Ilića 154, 11000 Belgrade, Serbia
*Corresponding author: nada.mladenovic@fon.bg.ac.rs

Bogdana Stanojević
Faculty of Organizational Sciences, University of Belgrade
Jove Ilića 154, 11000 Belgrade, Serbia
bogdana.stanojevic@fon.bg.ac.rs

Abstract

Multiobjective combinatorial optimization is an essential approach for handling trade-offs in complex control and communications systems, when multiple conflicting objectives (performance, cost, reliability) must be counterbalanced. Many well known combinatorial problems can be easier solved by customizing a general approach, namely by employing problem specific algorithms in key points of the general framework. In this paper we adapt the general framework PESA (2020) from continual to the discrete case taking advantage of the particular shapes of both decision and criterion space of combinatorial problems, and replace the exact optimization algorithms with new developed heuristics. The results is an anytime heuristics approach that progressively dissecting the Pareto front and deriving approximated non-dominated points. To test the performances of the approach we customized the algorithms and carried out experiments on multiple criteria multidimensional 0/1 Knapsack problems recalled from the literature. The comparative numerical results reported in the paper support the conclusion that the novel approach has the ability to generate good first-level approximations to the Pareto fronts of the addressed problems in reasonable time.

Keywords: Multiple objective combinatorial optimization, knapsack problem, heuristics

1 Introduction

Combinatorial optimization (CO) problems, also known as discrete optimization problems, involving multiple objectives are widely studied in the recent literature. Providing efficiently solutions to such problems brings a fundamental support to artificial intelligence, since many of its key tasks require searching, planning, decision-making, and reasoning over discrete structures.

Most of CO problems, including for instance the classical ones like traveling salesman problem, knapsack problem, graph coloring, set covering, scheduling problems, assignment and matching problems, network flow problems can be formulated as integer linear programs (ILP) or mixed-integer linear programs (MILP). However, generally, the mathematical models of combinatorial problems are often hard to solve especially for large-size instances. In this paper we introduce a general anytime heuristics to solve large-scale multiple objective (MO) mixed-integer programming (MIP) problems.

Anytime algorithms are the algorithms that monotonic improves the quality of the provided solution with respect to the allocated runtime. Heuristics is any solution approach that uses practical experience or intuition to derive approximate solutions to a clearly formulated problem (combinatorial and/or based on mathematical models). Even though heuristics cannot assure that it reaches an optimal solution, it derives good solutions and in reasonable time to complex and large-scale problems. Due to this quality, heuristics are widely used in many important areas where soft computing can be applied. Particularly, in this paper, we report results obtained for multiple criteria (MC), multidimensional (MD) 0/1 knapsack problems (KP).

MO-MIP problems cannot be easily solved mainly because of the shape of their Pareto fronts that generally contain a great number of unsupported non-dominated points; and their non-continual decision spaces. Recently, PESA [23] was introduce to generate a pattern efficient set of non-dominated vectors to many-objective optimization problems. Embedding the specific characteristics of CO problems within the PESA framework, we succeeded to solve small size multiple objectives mixed integer programming problems. However, for large-scale problems approximate algorithms are desired, since any integer constraint on the decision variables complicates a lot the procedure of inspecting the optimality of each improved solution.

The novelty brought by our approach is in the special ways of exploiting the particularity of discrete Pareto front of integer problems, and in developing heuristics that can solve large scale combinatorial optimization problems. Based on the mathematical model, we firstly propose a matheuristics that, combined with the progressive dissection framework, is able to generate the entire Pareto front of MC-MD-KP problems for small-sized instances, providing a 0th-order approximation containing well distributed non-dominated vectors on the Pareto front. Secondly, we propose a greedy randomized adaptive search procedure (GRASP) based on a constructive heuristics, customized to solve multiple objective knapsack problems, and providing 1-level approximation to the true Pareto front of the addressed problems. For large-scale instances, both the matheuristics and constructive heuristics provide a good 1st-level approximation to the Pareto front.

Our solution approach is essentially based on the Targeted Directional Model (TDM) introduced in [23] to solve multiple objective continual problems. The original TDM model was formulated with the help of a hypothetical bound defined as a convex combination of certain already generated non-dominated vectors. In this study, we advance two different ways to chose the hypothetical bounds: (i) as non-dominated vectors obtained by PESA applied to the integer relaxation of the original problem; and (ii) as the nadir point of a set of certain already generated (approximated) non-dominated points.

Our paper is structured as follows: a short literature review is presented in Section 2; Section 3 provides basic notation and terminology, and formalizes the addressed problem; Section 4 describes our solution approach, while Section 5 reports some numerical results. The final conclusion and some directions for further researches are included in Section 6.

2 Brief literature review

Multiple objective optimization is an essential tool for handling the trade-offs in complex control and communications systems, especially when multiple conflicting objectives (performance, cost, reliability) must be counterbalanced. Many real-life problems were addressed and solved using multiple objective models. Kaur et al. [15] proposed solutions to hydrothermal, pumped hydro, and solar generators' non-convex, highly constrained, and non-linear power generation scheduling problem. They handled the conflictiong objectives by using the non-interactive approach, and exploiting a specific penalty method. One approach to handle bi-objective optimization is based on fractional programming models that aggregate the two conflicting objectives in a single function, and optimize their ratio. The wide survey provided in [22] well emphasized the importance of fractional programming under uncertainty, and its valuable effects on making decisions.

Combinatorial problems on graphs have a wide range of applicability nowadays. For example, recently, Montoya et al. [17] introduced a mathematical optimization model to solve the routing problem of different services in an IoT network. The authors considered different levels of priority of services, aiming to reduce the energy consumption.

Multiple objective combinatorial optimization (MOCO) problems are widely studied in the recent literature from both theoretical and practical points of view. The basic reference to MOCO problems was provided by Ehrgott et al. [12] who surveyed the exact methods. [24] surveyed the field based on social keywords and journal maps, [10] that proposed an anytime algorithm to solve MOCO problems, and [2] that developed a decomposition-based coevolutionary multiobjective local search for MOCO problems.

Recently, Nagy et al. [18] addressed a suppliers selection problem providing emergency responses by applying robust optimization models. They employed multiple criteria decision making tools to deal with uncertainty and risk, and cope with computational issues. Wand and Xu [25] built a multiple objective location decision making model for emergency shelters by giving higher priority to subjective evaluation of residents. Dusadeerungsikul and Nof [11] proposed a collaborative control protocol with artificial intelligence for medical student work scheduling aiming to optimize clinical training schedule.

Kaliszewski and Miroforidis [14] solved large-scale instances of multiple objective optimization problems using mixed integer programming solvers, and discussed the quality of the obtained Pareto fronts. Charkhgarda et al. [5] focused on a smaller class of multiple objective optimization problems, and proposed a novel approach to solve hard bi-objective knapsack problems using deep reinforcement learning.

Wang et al. [26] defined the reliability level between two points in a network from two points of view: minimal edge cut and path, respectively, and proved their equivalence. As a consequence, they were able to propose a multiple objective optimization model whose first goal was to minimize the total service distance, while the second one was to maximize the network reliability level.

Pardalos et al. [20] contributed to the field with their monograph on non-convex multiple objective optimization. Cai et al. [3] introduced a new diversity indicator based on reference vectors that showed to be easy to use to estimate the performance of an optimizer applied to many-objective optimization problems. To estimate the quality of our numerical results we used the coverage gap and inverted generational distance metrics introduced in [6] and [4], respectively. Strengths and weaknesses of various performance indicators were been systematized in [1].

Deb and Miettinen [9] provided a review of nadir point estimation procedures based on evolutionary approaches, and discussed the dimension-reduction issue.

Ozpeynirci and Koksalan [19] introduced an exact algorithm for finding extreme supported non-dominated points to multiple objective mixed integer programming problems. We carried out experiments on benchmarks introduced in [19] and [13]. A scatter search method based on surrogate relaxation to bi-criterion knapsack problems was proposed by Silva et al. in [7]. Our illustrative example is recalled from [7]. A reduction dynamic programming algorithm for the bi-objective integer knapsack problem was presented by Rong and Figueira in [21].

3 Preliminaries

The general model of a MO optimization problem is given by

$$\max_{x \in X} \left(f_k\left(x\right)\right)_{k=\overline{1,p}}, \tag{1}$$

where $p \geq 2$ is the number of objectives, $X \subseteq R^n$ is the feasible set of the decision variables $x \in R^n$, and $f_k : X \to R$, $k = \overline{1,p}$ are the objective functions that all have to be maximized. Within combinatorial problems the some of the decision variables are integer.

The main concept used in defining solutions to MO optimization problems is the concept of dominance. Let $F : X \to R^p$ be defined by $F\left(x\right) = \left(f_1\left(x\right), f_2\left(x\right), \ldots, f_p\left(x\right)\right)$. The image of the feasible set in R^p defined by $F\left(X\right)$ represents the criterion space. A feasible solution $x' \in X$ is said to dominate $x'' \in X$ if and only if $f_k\left(x'\right) \geq f_k\left(x''\right)$ for all $k = \overline{1,p}$. In addition, $x' \in X$ strictly dominates $x'' \in X$ if and only if x' dominates x'' and $f_k\left(x'\right) > f_k\left(x''\right)$ for at least one index $k = \overline{1,p}$.

A feasible solution is called Pareto optimal or Pareto efficient if there is no feasible solution that strictly dominates it. Whenever x' dominates x'' with respect to function F, we can use notation $F(x') \geq F(x'')$.

A vector y from the criterion space is called non-dominated if it is the image of a Pareto optimal solution, i.e. there is an Pareto efficient $x \in X$ such that $F(x) = y$. The set of all non-dominated points of an MO optimization problem represents the Pareto front of the given problem.

The pattern efficient framework PESA was introduced in [23] to provide a good approximation to the Pareto front of continuous MO optimization problems. Algorithm 1 contains its adapted description to the current notation and terminology.

Algorithm 1 PESA framework, adapted from [23]

Require: An instance of (1): p, X, $f_k : X \to R$, $k = \overline{1,p}$, and the number of iterations $nrit$.

1: Obtain marginal solutions and their corresponding images in the criterion space $y^1, y^2, \ldots, y^p$.

2: Define $\alpha^k = \left(\alpha_s^k\right)_{s=\overline{1,p}}$, $\alpha_s^k = \begin{cases} 1, & s = k, \\ 0 & s \neq k. \end{cases}$

3: Normalize the objective functions and obtain $\overline{f}_k$, $k = \overline{1,p}$.

4: Initialize the set of non-dominated vectors $E \leftarrow \{y^1, y^2, \ldots, y^p\}$.

5: Compute the hyper-volume $H(y^1, y^2, \ldots, y^p)$.

6: Define problem $P = [H(y^1, y^2, \ldots, y^p), y^1, y^2, \ldots, y^p, \alpha^1, \alpha^2, \ldots, \alpha^p]$ related to marginal solution, using the hyper-volume H, non-dominated vectors y^k, and coefficient vectors α^k, $k = \overline{1,p}$.

7: Initialize the list of problems $L \leftarrow \{P\}$.

8: **for** $q = \overline{1, nrit}$ **do**

9: Identify P^* in L that has the greatest hyper-volume.

10: Compute y^* and α^* based on information provided in P^*.

11: Solve (2) with $y^0 = y^*$, $\alpha^0 = \alpha^*$, derive $\left(x^{eff}, \lambda\right)$ as its optimal solution, and construct $y^{new} = F\left(x^{eff}\right)$. Update $E \leftarrow E \cup \{y^{new}\}$.

12: Insert p new problems in L with respect to P^0 and y^{new}. Update $L \leftarrow L \setminus \{P^0\}$.

13: **end for**

Ensure: The set E of generated non-dominated vectors.

The general TDM model, recalled in Step 11 of Algorithm 1, was presented in detail in [23]. In what follows we use its simplified form:

$$\max \left\{ \sum_{k=1}^{p} \alpha_k^0 f_k(x) + p_0 \lambda | x \in X, f_k(x) \geq \lambda y_k^0, k = \overline{1,p}, \lambda \geq 0 \right\}, \quad (2)$$

where $y^0 = \left(y_1^0, y_2^0, \ldots, y_p^0\right)$ is the hypothetical bound that is aimed through optimization using the penalty term $p_0 \lambda$ in the objective function; and $\alpha^0 = \left(\alpha_1^0, \alpha_2^0, \ldots, \alpha_p^0\right)$ is the coefficient vector used to aggregate the original objective functions.

Model (2) relays on parameter $p_0 \geq 0$ that can be tuned in accordance to the specificity of each approached instance. We empirically concluded that TDM model applied to MO-MIP problems performs properly even though only parameter p_0 is kept active.

3.1 Problem formulation

Particular models can be obtained from (1) by imposing various special conditions on variables, feasible sets, and/or objective functions. Within MO-MIP problems each variable is either integer or continual. The MO-MIP problem that we address in this paper is then modeled by

$$\begin{array}{lll} \max & f_k(x) & k = \overline{1,p}, \\ \text{s.t.} & x \in X, & \\ & x_s \in Z, & s \in S, \end{array} \quad (3)$$

where S is the set of indexes that correspond to integer variables, $S \subset \{1, 2, \ldots, n\}$.

In the sequel, we will refer to the Pareto front of Problem (3) by true Pareto front, and to the Pareto front of the integer relaxation of Problem (3) by continual Pareto front.

In Section 4 we propose a general approach to solve Problem (3), but also formulate a customized variant that solve multiple criteria knapsack problems, whose mathematical model is a particular form of (3), having linear objective functions, linear “less or equal”-type constraints, both with strictly positive coefficients and binary variables, i.e.

$$f_k(x) = \sum_{j=1}^{n} c_{kj} x_j, \quad c_{kj} > 0, \quad k = \overline{1,p}, \quad j = \overline{1,n},$$
$$X = \left\{ (x_1, x_2, ..., x_n) \in \{0,1\}^n \mid \sum_{j=1}^{n} a_{ij} x_j \le b_i, a_{ij} > 0, i = \overline{1,m}, j = \overline{1,n} \right\}. \tag{4}$$

Within our experiments we use the following indicators to evaluate the quality of the solutions:

1. The coverage gap (CG) indicator was introduced in [4]. Given the discrete Pareto front P^* and the subset A of P^*, the value
$$\alpha_A(z) = \min_{y \in A} \left(\max_{i=\overline{1,p}} (z_i - y_i) \right) \tag{5}$$
represents a measure of how well the vector z is covered by A. Next, the vector z^* having the worst coverage is determined
$$z^* = \arg\max_{z \in P} (\alpha_A(z)), \tag{6}$$
and then, the coverage gap of A is the value $CG(A, P^*) = \alpha_A(z^*)$.

2. The inverted generational distance (IGD) metric was introduced in [6], and it calculates the average distance from the given points on the reference Pareto front (P^*) to discrete Pareto front approximation, A, i.e.
$$IGD(A, P^*) = \frac{\sum_{v \in P^*} d(v, A)}{|P^*|}, \tag{7}$$
where $d(v, A)$ is the minimal distance (in Euclidean sense) between v and points in A. More details can be found in [8].

3. The hypervolume (HV) indicator measures the hypervolume (area, in the bi-objective case; volume in the three-objective case) of the objective space that is dominated by the generated approximated front A with respect to a reference point.

Note that the first two indicators demand a reference set P^* that is a subset of the Pareto front, while the third one can be evaluated with respect to an arbitrary fixed reference point.

4 Solution algorithm

Our approach conveniently combines a progressive dissection framework able to provide well dispersed hypothetical bounds for specific mathematical models, and novel heuristics able to efficiently solve the given optimization models within a reasonable time.

4.1 The criterion set dissection framework

Aiming to solve general MO-MIP problems (3), we exploit the specific shape of the Pareto front retrieved by the integer components of its non-dominated vectors, and conveniently change the manner of choosing the hypothetical bounds to be used in the integer form of the classic TDM model [23].

PESA [23] was proposed to handle continual MO optimization problems. Our first idea is to employ PESA to solve the integer relaxation of Problem (3), and use each obtained non-dominated point of the continual Pareto front as hypothetical bound y^0 in the integer TDM model related to the original MO-MIP problem. In other words, we now use PESA to generate well-spread enhanced hypothetical bounds, that are close to the true Pareto front of the MO-MIP problem. Let us denote by C-TDM the optimization model that involves such constructed hypothetical bounds.

Alternatively to setting continual hypothetical bounds, given a set of already derived (approximated) non-dominated points to the original MO-MIP problem, we may identify the "lowest" vector that can be defined using their components (in the manner of nadir point), and choose it as hypothetical bound y^0 in our second N-TDM model. Namely, given the non-dominated vectors $y^1, y^2, \ldots, y^h$, the components of y^0 are defined as

$$y_k^0 = \min \left\{ y_k^q | q = \overline{1,h} \right\}, k = \overline{1,p}.$$

Figure 1 illustrates the two ways of choosing the hypothetical bounds for the particular case $p = 2$. On the left, the hypothetical bound y^0 is a point on the continual Pareto front (black line) that is targeted from the origin; and by solving C-TDM model, it derives the closest point of the true Pareto front y^{new}. On the right, given the non-dominated points A and B on the true Pareto front, the nadir-like hypothetical bound y^0 has the first component of A, and the second component of B; and together with the origin defines a direction, and relative to it, derives the closest non-dominated point y^{new} on the true Pareto front.

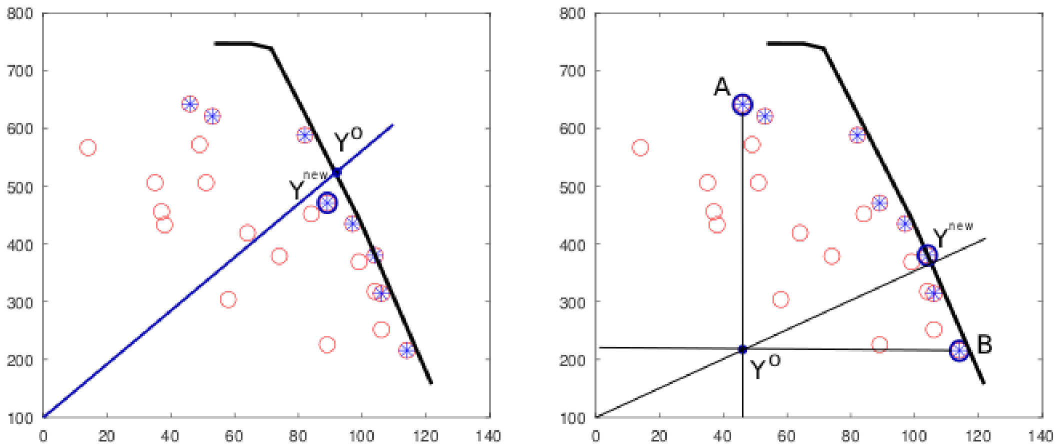

Figure 1: Targeted hypothetical bounds y^0, and corresponding non-dominated points y^{new} on true Pareto front of Problem (3) (continual bound, on the left; nadir-like bound, on the right)

In the next section, we propose a matheuristics, and a GRASP based on constructive heuristics to solve single objective C- and N-TDM models within Algorithm 1. As it will be concluded from the results of the experiments, models C-TDM and N-TDM used within the progressive dissection framework beneficially complement each other: continual hypothetical bounds are close to the true Pareto front, thus reducing the solver's runtime; while nadir-like hypothetical bounds assure that non of the given non-dominated points can be reached by the optimal solution to N-TDM model.

4.2 Heuristics approaches to C- and N-TDM optimization models

Algorithm 1 combined with any of C- or N-TDM optimization models, and interlinked with any MIP solver can derive the zero-level approximations to the true Pareto front of MO-MIP problem (3). However, MIP solvers are generally based on heuristics that easily can find a feasible – close to optimal – solution, and then spends time to prove that the provided solution is optimal. Practically, such solvers provide exact optimal solution given that their integer-tolerance parameter is set to 0.

Small size instances of MIP problems can be solved to optimality in real-time, but this is not the case for large-size ones. Increasing the integer-tolerance parameter of MIP solvers from 0 to a predefined value makes the exact algorithm to be a matheuristics, since it is an approximate algorithm based on relaxed mathematical models. Within our experiments, we denote a such obtained matheuristics either by Math-C or Math-N, depending on which optimization model it uses.

Our next goal is to formulate a greedy constructive heuristics to solve a specific class of MIP problems. Knapsack problems are the most approachable MIP problems. Even though they are linear integer problems, that does not necessarily make them easy to be solved. Since our original problem is a multiple objective one, we focus on solving the TDM model associated to a multiple criteria, multidimensional 0/1 knapsack problem. Using the notation given in (4), the particular form of Model (2) which corresponds to the above mentioned multiple criteria problem is

$$\begin{array}{ll} \max & \sum_{j=1}^{n} w_j x_j + p_0 \lambda \\ \text{s.t.} & x \in X \\ & f_k(x) \geq \lambda y_k^0, \ k = \overline{1,p}, \\ & \lambda \geq 0, \end{array} \tag{8}$$

obtained by rewriting $\sum_{k=1}^{p}\left(\alpha_k \sum_{j=1}^{n} c_{kj} x_j\right)$ as $\sum_{j=1}^{n}\left(x_j \sum_{k=1}^{p} c_{kj}\alpha_k\right)$, and denoting $\sum_{k=1}^{p} c_{kj}\alpha_k$ by w_j.

We propose Algorithm 2 to construct, in a greedy manner, a feasible solution to Problem (8).

Algorithm 2 Greedy constructive heuristics

Require: An instance of Problem (8).

1: Compute the vector $v = \left(\min_{i=\overline{1,m}}\left(\frac{w_j}{a_{ij}}\right)\right)_{j=\overline{1,n}}$.

2: Order the components of vector v in the decreasing order, denoting by σ the corresponding permutation of the indexes.

3: Identify the index s such that $\sum_{j=1}^{s} a_{i\sigma(j)} \leq b_i$ for each $i = \overline{1,m}$, and $\sum_{j=1}^{s+1} a_{i\sigma(j)} > b_i$ for at least one $i = \overline{1,m}$.

4: Allocate value 1 to all variables $x_{\sigma(j)}$, $j = \overline{1,s}$, and value 0 to $x_{\sigma(j)}$, $j = \overline{s+1,n}$, thus obtaining a solution $\overline{x} = \left(x_{\sigma^{-1}(l)}\right)_{l=\overline{1,n}}$ that fulfills the first set of constraints.

Ensure: The permutation σ, feasible solution $\overline{x}$ and its corresponding vector $\left(f_k(\overline{x})\right)_{k=\overline{1,p}}$ from the criterion set of Problem (1).

Aiming to obtain a feasible solution and increase the value of the objective function of Problem (8), we adopt GRASP principles, and extend Algorithm 2 with following randomization step: given $\overline{x}$ and its corresponding permutation σ, for each $t \in \overline{1,n}$ such that

$$\sigma(t) \in \left[\max\{1, s-10\}, \min\{n, s+10\}\right]$$

we replace the value of $\overline{x}_t$ by a random value from $\{0,1\}$. Then, we include the new obtained $\overline{x}'$ in the set of restricted candidate list if and only if it fulfills the first group of constraints, i.e. $\sum_{j=1}^{s} a_{ij}\overline{x}'_j \leq b_i$ for each $i = \overline{1,m}$. We apply this randomization $\#it$ times. For each $\overline{x}'$ from the restricted candidate list, we compute $\overline{\lambda}'$ as described in Step 5. From the such obtained set of pairs $\left(\overline{x}', \overline{\lambda}'\right)$, we choose the final solution $\left(\overline{x}^*, \overline{\lambda}^*\right)$ that yields the greatest value of the single objective function $\sum_{j=1}^{n} w_j \overline{x}_j^* + p_0\overline{\lambda}^*$. Algorithm 3 describes the enhanced GRASP heuristics.

We interlink Algorithms 1 and 3, and obtain Grasp-C and Grasp-N heuristics variants of our general approach, depending on the manner of choosing the hypothetical bounds in the formulation of Model (8).

Algorithm 3 GRASP variant of the greedy constructive heuristics

Require: An instance of Problem (8).

Employ Algorithm 2, and derive the feasible solution $\overline{x}$ and its corresponding permutation σ.

Initialize $A = \left\{ \left(\overline{x}, \min \left\{ \frac{f_k(\overline{x})}{y_k^0} | k = \overline{1,p} \right\} \right) \right\}$.

for $h = \overline{1, \#it}$ **do**

  Set $\overline{x}' = \overline{x}$.

  For each $t \in \overline{1,n}$ such that $\sigma(t) \in [\max\{1, s-10\}, \min\{n, s+10\}]$: $\overline{x}_t = rand\{0,1\}$.

  if $\sum_{j=1}^{s} a_{ij}\overline{x}_j \le b_i$ for each $i = \overline{1,m}$ **then**

    Update $A = A \bigcup \left\{ \left(\overline{x}, \min \left\{ \frac{f_k(\overline{x})}{y_k^0} | k = \overline{1,p} \right\} \right) \right\}$.

  end if

end for

for each $\left(\overline{x}, \overline{\lambda}\right) \in A$ **do**

  Compute $\left(\overline{x}^*, \overline{\lambda}^*\right) = \underset{(\overline{x},\overline{\lambda}) \in A}{arg\,max} \left\{ \sum_{k=1}^{p} \alpha_k f_k(\overline{x}) + p_0 \overline{\lambda} \right\}$, and the corresponding ND $y^* = (f_1(\overline{x}^*), f_2(\overline{x}^*), ..., f_p(\overline{x}^*))$ to Problem (8).

end for

Ensure: The non-dominated vector y^* corresponding to the hypothetical bound y^0.

4.3 Illustrative example

We consider Problem (9) of a bi-objective MD-KP with 4 constraints and 10 items (recalled from [7]) to illustrate the main differences between the proposed variants of our solution algorithm.

$$
\begin{array}{ll}
\max & x_1 + 87x_2 + 28x_3 + 32x_4 + 38x_5 + 9x_6 + 8x_7 + 6x_8 + 92x_9 + 78x_{10}, \\
\max & 4x_1 + 21x_2 + 68x_3 + 17x_4 + 43x_5 + 48x_6 + 85x_7 + 30x_8 + 37x_9 + 33x_{10}, \\
\text{s.t.} & \\
& 70x_1 + 85x_2 + 72x_3 + 31x_4 + 17x_5 + 33x_6 + 47x_7 + 25x_8 + 83x_9 + 28x_{10} \le 246, \\
& 49x_1 + 15x_2 + 88x_3 + 29x_4 + 78x_5 + 98x_6 + 50x_7 + 89x_8 + 83x_9 + 3x_{10} \le 291, \\
& 15x_1 + 15x_2 + 51x_3 + 3x_4 + 60x_5 + x_6 + 78x_7 + 66x_8 + 78x_9 + 71x_{10} \le 219, \\
& 56x_1 + 21x_2 + 69x_3 + 60x_4 + 96x_5 + 65x_6 + 100x_7 + 25x_8 + 68x_9 + 30x_{10} \le 295, \\
& x_i \in \{0,1\}, i = \overline{1,10}.
\end{array} \tag{9}
$$

Both the true Pareto front and its approximations obtained by applying the above procedures are depicted in Figure 2.

We provide below detailed explanations:

- the true Pareto front of MO-MD-KP problem, is the set of non-dominated points represented by red circles in Figure 2), namely $\{M, V, S, Q, W, P, N\}$;
- the continual Pareto front (the blue line in Figure 2) is the Pareto front of the integer relaxation of Problem (9). Several points on the continual Pareto front were used as hypothetical bounds in the Math-C variant of the algorithm;
- the constructive heuristics Grasp-C provided the approximate Pareto front consisting in the set of points $\{M, V, S, W, P, N, U, T\}$. It failed to reach the non-dominated vector Q and generated U and T that are dominated by $\{M, V\}$ and $\{W, P\}$, respectively;
- the matheuristics Math-C, with the tolerance set to 2%, provided the approximate Pareto front containing $\{M, V, S, W, P, N, Q\}$. It succeeded to cover entire exact Pareto front, and generated also the point R that is dominated by W.

Several steps of Math-N running on Problem (9) can be traced in Figure 2 as follows.

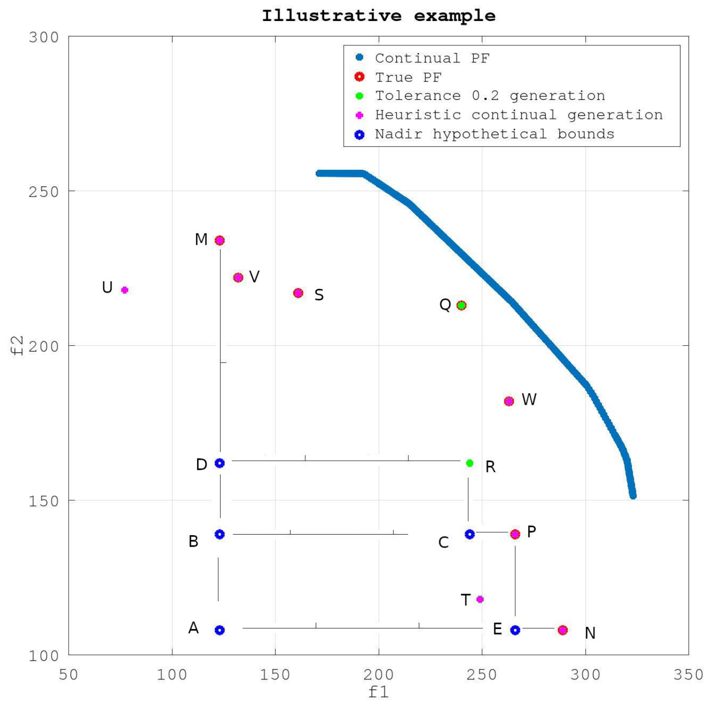

Figure 2: Approximations of the true Pareto front derived to Problem (9)

- M and N are the extreme non-dominated points, and they define the first gap $[MN]$. Taking their minimum component-wise values the hypothetical bound A is obtained. Solving the corresponding TDM model the non-dominated point P is generated, and the new two gaps $[MP]$ and $[PN]$ will be further aimed;
- M and P form a greater gap than N and P, thus $[MP]$ will be adressed first: M and P define the hypothetical bound B that derives the approximate non-dominated point R. Gap $[MP]$ is removed and two new gaps $[MR]$ and $[RP]$ are created;
- on the gap $[MR]$ the hypothetical bound is D, and the generated non-dominated point is Q. Note that the non-dominated vector R helped defining gaps and generating new non-dominated vectors;
- $[PN]$ has smaller length, than $[MR]$ and greater than $[RP]$, so next P and N define the hypothetical bound E, construct the TDM model with empty feasible set, derive no solution and declare the gap $[PN]$ covered, thus do not define any additional sub-gap;
- $[RP]$ is the smallest gap, it creates the hypothetical bound C, and generates a new non-dominated point W.

5 Experiments and discussions

We carried out our experiments solving 2- and 3-objective KP problems. The 3-objective problems involve multiple constraints that makes them multidimensional. We report in Section 5.1 the results obtained for small-size instances with known Pareto front. Such instances are convenient for analyzing performances using indicators that demand a reference Pareto front. Within our numerical results, we report the values of coverage gap (CG), inverted generational distance (IGD).

In Section 5.2 we summarize our results obtained large-scale instances with unknown Pareto fronts. For each of these instances, we report statistically relevant results, i.e. results obtained by multiple runs with variable seeds; and evaluate the heuristics performances using the HV indicator that makes use of an arbitrary reference point instead of an entire reference Pareto front.

Table 1: Summary of the run times needed to derive the entire Pareto front to 1.A 2KP instances [13]. Tolerance was set to $1e-7$.

Instances (tightness ratio)	2KP50-11 (0.11)	2KP50-50 (0.50)	2KP50-92 (0.92)	2KP100-50 (0.50)
Math-N's runtime (s)	1.5598	2.3458	0.0553	12.8020
Math-C's runtime (s)	1.5620	3.1002	2.7507	59.4820

Table 2: IGD vs CG values for various runs with different number of iterations, for Instance 2KP50-50, and tolerance set to $1e-7$.

Math-C					Math-N				
#it	#NDs	IGD	CG	Time (s)	#it	#NDs	IGD	CG	Time (s)
8	8	18.4153	0.0138	0.1541	6	8	20.4150	0.0207	0.0998
32	27	3.1162	0.0028	0.4099	27	27	3.1798	0.0067	0.6067
100	39	0.9906	0.0028	0.9181	45	39	1.1479	0.0021	1.0473
300	50	0.0608	0.0005	2.4536	80	50	0.0608	0.0014	1.9778
395	51	0	0	3.1002	90	51	0	0	2.2213

One group of experiments were carried out on instances form Sets 1.A and 1.B downloaded from [27], and having two objectives ($p = 2$), one constraint only ($k = 1$), and several tightness ratio (r). See also [13] where the instances were firstly introduced.

The second group of experiments, having three-objective functions were carried out on instances form the benchmark described in [19].

5.1 Results related to small-size instances

5.1.1 Two objectives KP instances

Table 1 summarizes the results obtained for Set 1.A instances on which we run our variants Math-C and Math-N. Incipient results obtained by Math-C were presented in [16]. For each instance, both matheuristics succeeded to derive the entire true Pareto front in short time.

The values of the diversity metrics, IGD and CG, for various runs with different number of iterations for Instance 2KP50-50 are reported in Table 2. Values $IGD = 0$ and $CG = 0$ obtained after 395 iterations prove that all points on the true Pareto front were obtained.

Analyzing the values reported in Table 2, several conclusions can be we drawn:

- less iterations are needed in Math-N, and consequently less time, to reach the same number of generated non-dominated vectors;
- based on the IGD metric values, the quality of the generated approximations to the true Pareto front is better when using Math-C than Math-N;
- based on the CG metric values, the qualities of the generated approximations to the true Pareto front by using Math-C and Math-N, respectively are not comparable, since the dispersion of the first 39 non-dominated vectors obtained by running Math-N vs Math-C while the first 50 obtained by Math-C are better dispersed than those generated by Math-N.

5.1.2 Three objectives KP instances

This second group of experiments were carried out on several MOKP instances with three objectives ($p = 3$), and three constraints ($k = 3$) with known Pareto front addressed in [19]. Five instances with $n = 25$ were solved using the variant Math-C2N3, and their summarized results are reported in Table 3. The acronym Math-C2N3 stands for matheuristics variant that uses continual hypothetical bounds when solving the pair of bi-objective problems (C2) and nadir-like hypothetical bounds when involving three-objective functions in optimization (N3).

Detailed results related to Instance 5 and tolerance $1e-7$ are shown in Figure 3. On the left, the dependency of the performance metric IGD is graphed with respect to the *multp* parameter. A greater

Table 3: Summary of the results obtained using Math-C2N3 (with various integer tolerances) to solve $p = 3$, $n = 25$ [19] instances.

Instance	Tolerance $1e-7$			Tolerance $1e-3$			Tolerance $1e-1$		
	#NDs	IGD	Time (s)	#NDs	IGD	Time (s)	#NDs	IGD	Time (s)
1	175	1.620	12.46	175	1.629	12.35	175	1.629	12.49
2	167	0.298	14.85	166	0.478	12.08	166	0.478	11.94
3	64	3.850	4.13	59	6.097	2.08	59	6.097	2.09
4	143	2.880	27.43	134	3.974	6.28	134	3.973	6.26
5	465	0.097	134.85	386	2.041	50.94	386	2.041	51.18

Table 4: Comparative results: Grasp-C2C3 and "gamultiobj" for $p = 3$, $n = 25$ [19].

		"gamultiobj"				Graps-C2C3			
		pop=100, maxit=10000, tol.=1e-4				maxit2=20, maxit3=10, randit=50, length=5			
Inst.	$\|PF\|$	#NDs	Time(s)	IGD	HV	#NDs	Time(s)	IGD	HV
1	182	70	19.2	140.7	9.811e+6	65.1	0.48	32.0	3.648e+07
2	168	70	18.3	120.7	5.039e+6	56.5	0.48	41.3	1.499e+07
3	76	70	18.8	108.3	5.366e+6	44.8	0.50	38.6	1.161e+07
4	163	70	17.3	213.2	3.856e+5	45.1	0.48	39.2	1.156e+07
5	470	1	15.6	310.4	0	46.0	0.48	39.3	1.146e+07

number of non-dominated vectors means a better, smaller IGD value when the generated vectors are well spread on the PF. The graph in the middle illustrates how the number of non-dominated vectors generated in a fixed number of iterations varies with respect to the *multp* parameter. Note that the number of generated non-dominated points has an abrupt increase from 0 to 20, and then stabilizes around 465. On the right, one can see the dependency of the values of the IGD metric with respect to the runtime.

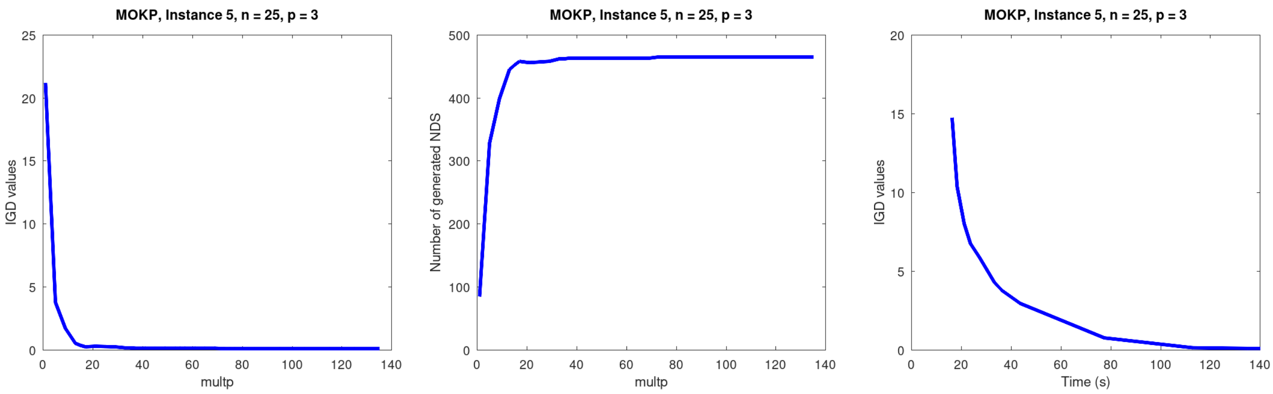

Figure 3: The number of generated non-dominated vectors (on the left) and the IGD values (in the middle) with respect to the *multp* values; and the IGD diversity metric values versus time (on the right) for Instance 5 ($p = 3$, $n = 25$, [19])

On the same set of instances we run Grasp-C2C3, that used continual hypothetical bounds in all optimization models, and compared its performances with the general function "gamultiobj" from matlab. Table 4 reports the comparative results.

The value 0 for the hypervolume obtained by "gamultiobj" (reported in Table 4) shows that the only point generated is dominated by the reference point defined based on the reference Pareto front recalled from the literature.

The characteristic of the next group of instances is $n = 50$. Table 5 summarizes the results obtained by Math-C2N3, Grasp-C2C3, and "gamultiobj". The number of iterations imposed to our heuristics was set to provide relevant comparative results, namely the performance metrics that correspond to approximations of about 65-70 non-dominated points. For all five instances "gamultiobj" run the longest, and derived low-quality approximations to the true PFs. Grasp-C2C3 has the best running times for all instances providing more than three times better IGD values than "gamultiobj".

Table 5: Comparative results: Grasp-C2C3, Math-C2N3, and "gamultiobj" for $p = 3$, $n = 50$ [19].

		"gamultiobj" pop=100, maxit=10000 tolerance $1e-4$,			Grasp-C2C3 maxit2=20, maxit3=10, randit=50, length= 5			Math-C2N3 maxit2=35, maxit3=30, tolerance $1e-7$		
Inst	$\|PF\|$	#NDs	Time(s)	IGD	#NDs	Time(s)	IGD	#NDs	Time(s)	IGD
1	784	70	18.93	380.5	66.2	0.56	87.68	72	4.62	51.37
2	912	70	17.98	395.1	58.8	0.56	79.69	85	3.99	46.61
3	519	70	18.06	505.2	62	0.57	82.22	72	3.62	45.09
4	280	70	18.24	242.2	57.7	0.57	78.97	60	3.91	29.21
5	356	70	23.11	463.0	58.9	0.57	162.26	82	4.78	71.11

Table 6: Average values obtained by running Grasp-C2C3 10 times for Instance 1 ($p = 3$, $n = 50$, [19]), with fixed parameters $maxit3 = 100$, $randit = 500$, $length = 5$, and various values for $maxit2$.

Metrics		$maxit2$ 20	50	100	150
#NDs	mean	5.64E+01	1.02E+02	1.48E+02	1.83E+02
	var. coef.	2.39E-02	2.66E-02	2.38E-02	3.58E-02
Time (s)	mean	3.98E+01	4.06E+01	4.14E+01	4.29E+01
	var. coef.	3.05E-02	5.73E-02	8.04E-02	1.13E-01
IGD	mean	8.72E+01	8.29E+01	7.70E+01	7.57E+01
	var. coef.	2.58E-02	2.96E-02	4.18E-02	5.52E-02
HV	mean	6.93E+07	7.23E+07	7.50E+07	7.57E+07
	var. coef.	1.76E-02	1.02E-02	1.01E-02	1.49E-02

Finally, Math-C2N3 provided the best IGD values for all instances, but it needed about 8 times more time than Math-C2N3. A better tuning of parameter p might improve the runtime of Math-C2N3. In the same way, a small increase of the runtime of Grasp-C2C3 may yield significant better values for IGD metric values.

Tabels 6, 7 and 8 provide detailed results obtained for Instance 1 ($p = 3$, $n = 50$, benchmark [19]). We set the parameter *length* to value 5 for this group of experiments, and sequentially varied one of the other three parameters $maxit2$ (number of iterations used in bi-objective optimizations), $maxit3$ (number of iterations used in three-objective optimizations), and *randit* (number of random iterations for each hypothetical bound, separately) while keeping the other two fixed.

Varying the number of random iterations when keeping all other parameters fixed, no variance within the number of generated points was expected. However, analyzing the values reported in Table 8 it is interesting to notice that the average number of derived points is negatively correlated with the values of *randit*. The explanation is that, by increasing the value of *randit*, different $maxit2$ and/or $maxit3$ iterations derive the same points, belonging to the Pareto front, and consequently providing a better quality of the generated approximation.

Figure 4 shows the approximations obtained for two different runs of Math-C2N3 and Grasp-C2C3 on Instance 1 ($p = 3$, $n = 50$, benchmark [19]), together with the true Pareto front recalled from the

Table 7: Average values obtained by running Grasp-C2C3 10 times for Instance 1 ($p = 3$, $n = 50$, [19]), with fixed parameters $maxit2 = 20$, $randit = 500$, $length = 5$, and various values for $maxit3$.

Metrics		$maxit3$ 10	20	50	100
#NDs	mean	57.3	56.9	58.5	56.4
	var. coef.	2.86E-02	3.04E-02	3.25E-02	2.39E-02
Time (s)	mean	4.49	8.11	19.6	39.8
	var. coef.	1.88E-02	1.56E-03	4.76E-03	3.05E-02
IGD	mean	85.1	86	83.9	87.2
	var. coef.	3.12E-02	4.44E-02	3.79E-02	2.58E-02
HV	mean	7.01E+07	6.99E+07	7.05E+07	6.93E+07
	var. coef.	1.54E-02	2.19E-02	1.71E-02	1.76E-02

Table 8: Average values obtained by running Grasp-C2C3 10 times for Instance 1 ($p = 3$, $n = 50$, [19]), with fixed parameters $maxit2 = 20$, $maxit3 = 10$, $length = 5$, and various values for $randit$.

Metrics		*randit*			
		10	50	100	500
#NDs	mean	68.7	67.5	63.2	57.3
	var. coef.	2.06E-02	2.44E-02	1.95E-02	2.8558e-02
Time (s)	mean	0.43	0.74	1.14	4.49
	var. coef.	8.09E-03	3.96E-03	3.32E-03	1.8787e-02
IGD	mean	92.6	86.5	86.4	85.1
	var. coef.	5.70E-02	3.95E-02	2.68E-02	3.1208e-02
HV	mean	6.22E+07	6.77E+07	6.80E+07	7.0089e+07
	var. coef.	4.14E-02	2.01E-02	1.70E-02	1.5361e-02

literature. In this way, the quality improvements of the approximation with respect to the number of iterations can be visualized.

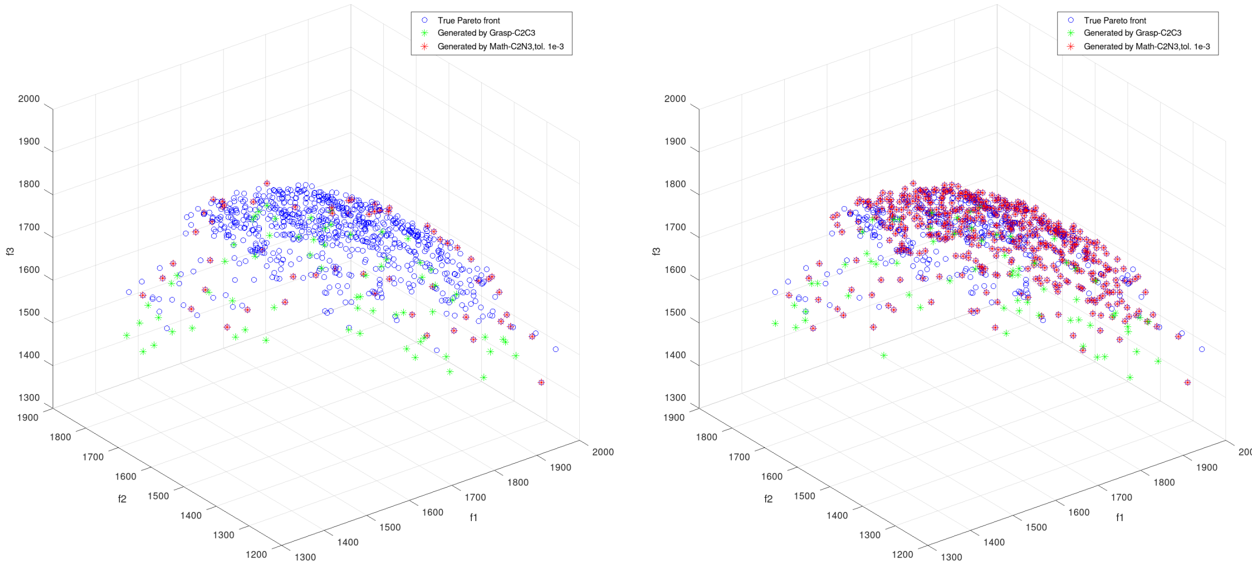

Figure 4: Instance 1, $p = 3$, $n = 50$, [19]. The red NDs (58 on the left, 514 on the right) were generated by Math-C2N3. The green approximated NDs (66 on both sides) were generated by Grasp-C2C3.

5.2 Results for large-scale instances

The numerical results obtained by running the heuristics on instances recalled from the literature are summarized in Table 9. Matheuristics Math-N and Math-C were run one time on each instance, since they do not relay on any randomization. Heuristics Grasp-N and Grasp-C were run ten times to assure statistical relevance to the results, and Table 9 reports the mean values and the coefficients of variance obtained for each performance metric. All runs contained 100 iterations within the progressive dissection framework. The GRASP variants used additional 100 random iterations to improve each constructed solution.

Several conclusions can be drawn from the results reported in Table 9:

- there are no big differences between the quality of approximations generated by the four variants of our solution approach, and different performance metrics establish different ranks for the four variants;
- comparing the runtimes, Math-N2 obtained the best quality results in three out of four cases, while Math-C2 derived the best quality in the other case;
- comparing the HV metric values, Math-C2 provided the best quality results for all solved instances;

Table 9: Results of heuristics on large-scale instances. The best values obtained are shown in bold.

Heuristics	Metrics	2KP500-411	2KP500-1B	2KP450-1B	2KP400-1B
Math-N2	#ND	98	97	96	100
	Time (s)	9.7913	8.1772	6.7736	6.0942
	HV	1.1546e+7	2.0669e+7	1.7689e+7	1.2982e+7
Math-C2	#ND	101	100	101	72
	Time (s)	8.1193	36.567	6.961	23.383
	HV	1.1562e+7	2.0717e+7	1.7717e+7	1.2983e+7
Grasp-N2	#ND	89 0.01790	98 0.00108	101 0.00475	101 0.01247
	Time (s)	55.938 0.01485	53.689 0.00810	48.705 0.01253	43.555 0.01016
	HV	1.1521e+7 0.00102	2.0495e+7 0.00127	1.7471e+7 0.00133	1.2655e+7 0.00314
Grasp-C2	#ND	78 0.02815	94 0	94 0.00773	76 0.03338
	Time (s)	55.873 0.00855	80.029 0.00425	107.24 1.1535	57.34 0.00431
	HV	1.1482e+7 0.00130	2.0482e+7 0.0011765	1.7414e+7 0.00099	1.2817e+7 0.00098

- generally, Grasp-N2 outperforms Grasp-C2 considering both runtime and HV metric values;
- even though Grasp-N2 succeeded to derive more approximated non-dominated points than all other variants for one out of four instances, the quality of the approximation does not overcome the quality obtained by matheuristics.

In what follows we provide some detailed results generated by our approach for Instance *2KP500-411*. Figure 5 clearly illustrates the good dispersion of the generated non-dominated points obtained after 10, 20 and 100 iterations, respectively. In addition, the non-dominated vectors of the continual Pareto front are also reported to show the closeness of continual hypothetical bounds to the true Pareto front.

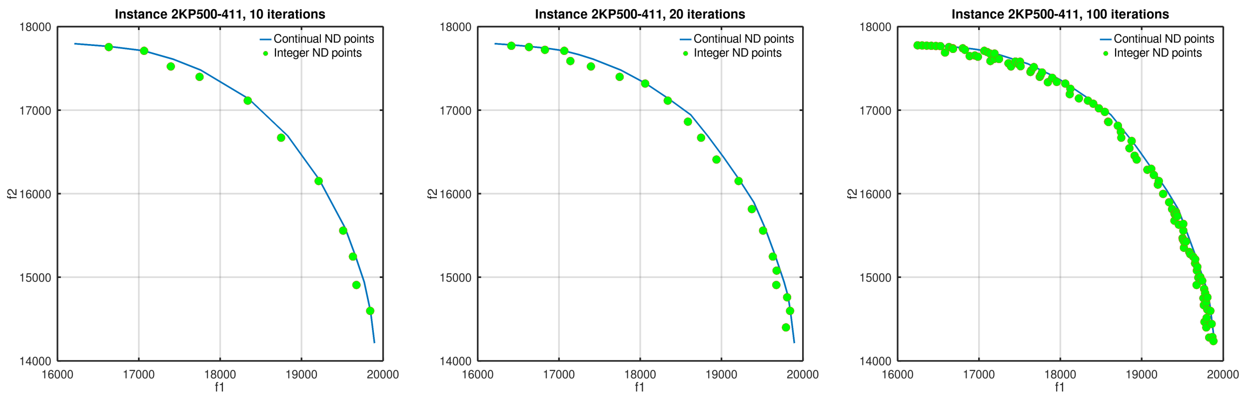

Figure 5: True Pareto front versus continual Pareto front, and approximations obtained for different number of iterations for Instance 2KP500-411

6 Conclusion and further researches

In this study we theoretically addressed the general multiple objective combinatorial optimization problem, and carried out experiments on multicriteria knapsack problems with two and three objective functions. Being able to handle trade-offs among multiple conflicting objectives, as performance, cost, and/or reliability, the multiobjective combinatorial optimization spread its applicability in the field of complex control and communications systems.

We adapted the general framework PESA proposed in 2020 to solve continual problems to the combinatorial case taking advantage of the particular shapes of both decision and criterion space of discrete problems, and replacing the exact optimization algorithms with new developed heuristics. The main result is an anytime heuristics approach that is progressively dissecting the criterion set to derive approximated non-dominated points.

Based on an optimization model formulated with the help of a hypothetical bound, our approach advanced two different ways to chose the hypothetical bounds: (i) as non-dominated vectors obtained by PESA applied to the integer relaxation of the original problem; and (ii) as the nadir point of a set of certain already generated (approximated) non-dominated points. We also developed heuristics that can solve large scale combinatorial optimization problems. Following the mathematical model, we firstly proposed a matheuristics that, combined with a progressive dissection framework, was able to generate the entire Pareto front for small-size instances. Secondly, we proposed a greedy randomized adaptive search procedure based on a constructive heuristics, customized to solve multiple objective knapsack problems.

The comparative numerical results reported in the paper support the conclusion that the new introduced approach generated good first-level approximations to the Pareto fronts of the addressed problems in reasonable time. Our approach was tested so far only on knapsack problems. This limitation can be overcome by developing customized variants of heuristics applicable to other classes of combinatorial problems, but keeping the same dissection framework.

In our future researches we will carry out more experiments, especially on instances with more than three objective functions, aiming to incorporate more specialized indicators able to evaluate the quality of generated Pareto front approximations of many-objective combinatorial optimization problems. We will also investigate the behavior of our matheuristics when solving mixed integer non-linear programming problems.

Another direction for further researches is to adapt our findings to solve multicriteria optimization problems under uncertainty. Using fractional programming to aggregate the objective functions of a bi-objective optimization problem is also of interest in our next researches.

Acknowledgements

This research was supported in part by the Ministry of Science, Technological Development and Innovation of the Republic of Serbia (grant number 200151).

Conflict of interest

The authors declare no conflict of interest.

References

[1] C. Audet, J. Bigeon, D. Cartier, S. Le Digabel, and L. Salomon (2021). Performance indicators in multiobjective optimization. *European J. of Operational Research*, 292(2):397–422, 2021.

[2] X. Cai, M. Hu, D. Gong, Y. Guo, Y. Zhang, Z. Fan, and Y. Huang (2019). A decomposition-based coevolutionary multiobjective local search for combinatorial multiobjective optimization. *Swarm and Evolutionary Computation*, 49:178–193, 2019.

[3] X. Cai, H. Sun, and Z. Fan (2018). A diversity indicator based on reference vectors for many-objective optimization. *Information Sciences*, 430-431:467 – 486, 2018.

[4] G. Ceyhan, M. Köksalan, and B. Lokman (2019). Finding a representative nondominated set for multi-objective mixed integer programs. *European J. of Operational Research*, 272(1):61–77, 2019.

[5] H. Charkhgarda, H.R. Moghaddama, A. Eshraghb, and S. Mahmoudinazlou (2025). Solving hard bi-objective knapsack problems using deep reinforcement learning. *Computers & Operations Research*, 55, 2025.

[6] C. Coello, A. Carlos, and M. Reyes Sierra (2004). A study of the parallelization of a coevolutionary multi-objective evolutionary algorithm. In Raúl Monroy, Gustavo Arroyo-Figueroa, Luis Enrique Sucar, and Humberto Sossa, editors, *MICAI 2004: Advances in Artificial Intelligence*, pages 688–697, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[7] C. Gomes da Silva, J. Climaco, and J. Figueira (2024). A scatter search method for the bi-criteria multi-dimensional 0,1-knapsack problem using surrogate relaxation. *Journal of Mathematical Modelling and Algorithms*, 3:183–208, 2004.

[8] K. Deb and H. Jain (2014). An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: Solving problems with box constraints. *IEEE Transactions on Evolutionary Computation*, 18(4):577–601, 2014.

[9] K. Deb and K. Miettinen (2009). *A Review of Nadir Point Estimation Procedures Using Evolutionary Approaches: A Tale of Dimensionality Reduction*, pages pp. 1–14. Proceedings of the Multiple Criterion Decision Making (MCDM-2008) Conference., 2009.

[10] M.A. Dominguez-Rios, F. Chicano, and E. Alba (2021). Effective anytime algorithm for multi-objective combinatorial optimization problems. *Information Sciences*, 565:210–228, 2021.

[11] P.O. Dusadeerungsikul and S.Y. Nof (2024). A collaborative control protocol with artificial intel-ligence for medical student work scheduling. *International Journal of Computers, Communications, Control*, 19(4):6686, 2024.

[12] Matthias Ehrgott, X. Gandibleux, and A. Przybylski (2016). *Exact Methods for Multi-Objective Combinatorial Optimisation*, pages 817–850. Springer New York, New York, NY, 2016.

[13] X. Gandibleux and A. Freville (2000). Tabu search based procedure for solving the 0-1 multiobjective knapsack problem : the two objectives case. *Journal of Heuristics*, 6(3):361–383, 2000.

[14] I. Kaliszewski and J. Miroforidis (2022). Probing the pareto front of a large-scale multiobjective problem with a mip solver. *Operational Research*, 22:5617–5673, 2022.

[15] A. Kaur, J.S. Dhillon, and M. Singh (2024). Emended snake optimizer to solve multiobjective hybrid energy generation scheduling. *Yugoslav Journal of Operations Research*, 34(4):627–668, 2024.

[16] N. Mladenović and B. Stanojević (2025). Anytime algorithm to generate non-dominated vectors to bi-objective integer programming problems. In B. Andrić Guvsavac M. Kuzmanović, D. Makajić-Nikolić, editor, *Proceedings of SymOpIs 2025*, pages 324–329. FON, 2025.

[17] G.A. Montoya, C. Lozano-Garzon, C. Paternina-Arboleda, and Y. Donoso (2025). Mathematical optimization approach for prioritized services in IoT networks for energy-constrained smart cities. *International Journal of Computers, Communications, Control*, 20(1):6912, 2025.

[18] M. Nagy, J.L. de Miranda, and N. Popescu-Bodorin. Decision making and robust optimization for information systems oriented to emergency events. *International Journal of Computers, Communications, Control*, 19(6):6861, 2024.

[19] O. Ozpeynirci and M. Koksalan (2010). An exact algorithm for finding extreme supported non-dominated points of multiobjective mixed integer programs. *Management Science*, 56(12):2302–2315, 2010.

[20] P. M. Pardalos, A. Žilinskas, and J. Žilinskas (2017). *Non-Convex Multi-Objective Optimization.* Springer, 2017.

[21] A. Rong and J.R. Figueira (2013). A reduction dynamic programming algorithm for the bi-objective integer knapsack problem. *European J. of Operational Research*, 231(2):299–313, 2013.

[22] B. Stanojević, S. Dzitac, and I. Dzitac (2020). Fuzzy numbers and fractional programming in making decisions. *International Journal of Information Technology & Decision Making*, 19(04):1123–1147, 2020.

[23] B. Stanojević and F. Glover (2020). A new approach to generate pattern-efficient sets of non-dominated vectors for multi-objective optimization. *Information Sciences*, 530:22–42, 2020.

[24] M. Toloo, S. Talatahari, A.H. Gandomi, and I. Rahimi (2022). 1 - multiobjective combinatorial optimization problems: social, keywords, and journal maps. In M. Toloo, S. Talatahari, and I. Rahimi, editors, *Multi-Objective Combinatorial Optimization Problems and Solution Methods*, pages 1–9. Academic Press, 2022.

[25] Y. Wang and Z. Xu (2022). A multi-objective location decision making model for emer-gency shelters giving priority to subjective evaluation of residents. *International Journal of Computers, Communications, Control*, 17(4):4749, 2022.

[26] Y. Wang, Z. Xu, and F.G. Filip (2022). Multi-objective model to improve network reliability level under limitedbudget by considering selection of facilities and total service distance in rescue operations. *International Journal of Computers, Communications, Control*, 17(1):4573, 2022.

[27] [Online] Available at: http://web.archive.org/web/20061205225020/ and http://www.univ-valenciennes.fr:80/road/mcdm/.

Cite this paper as:

Mladenović, N.; Stanojević, B. (2026). Solution Algorithm Involving Heuristics and Criterion Set Dissection to Solve Large Scale Multicriteria Knapsack Problems, *International Journal of Computers Communications & Control*, 21(1), 7267, 2026.
https://doi.org/10.15837/ijccc.2026.1.7267