Optimization Letters manuscript No.
(will be inserted by the editor)

Finding representations for an unconstrained bi-objective combinatorial optimization problem[⋆]

Alexandre D. Jesus · Luís Paquete · José Rui Figueira

Abstract Typically, multi-objective optimization problems give rise to a large number of optimal solutions. However, this information can be overwhelming to a decision maker. This article introduces a technique to find a representative subset of optimal solutions, of a given bounded cardinality for an unconstrained bi-objective combinatorial optimization problem in terms of ϵ-indicator. This technique extends the Nemhauser-Ullman algorithm for the knapsack problem and allows to find a representative subset in a single run. We present a discussion on the representation quality achieved by this technique, both from a theoretical and numerical perspective, with respect to an optimal representation.

Keywords Multiobjective Combinatorial Optimization, Representation Problem, Dynamic Programming

1 Introduction

In this article, we consider a particular *unconstrained bi-objective combinatorial optimization problem* (UBCOP), which is closely related to the single-objective knapsack problem. The goal is to select a subset of items whose total profit is maximal and total weight is minimal [6]. Under the notion of Pareto optimality, it might not exist only one solution that is optimal, but several *efficient* solutions, each of which cannot improve the outcome of one objective without deteriorating the outcome of the other one. However, two levels of difficulty arise in this problem [6]: Finding an efficient solution is

[⋆] This is a post-peer-review, pre-copyedit version of an article published in *Optimization Letters*. The final authenticated version is available online at: `https://doi.org/10.1007/s11590-017-1129-6`

A. D. Jesus · L. Paquete
CISUC, Department of Informatics Engineering, University of Coimbra, Pólo II, 3030-290 Coimbra, Portugal
E-mail: ajesus@student.dei.uc.pt,paquete@dei.uc.pt

J. R. Figueira
CEG-IST, Instituto Superior Técnico, Universidade de Lisboa, Av. Rovisco Pais 1, 1049-001, Lisboa, Portugal
E-mail: figueira@tecnico.ulisboa.pt

NP-hard and the total number of efficient solutions is exponential with respect to the instance size. As the number of efficient solutions may be too large for a decision maker to choose the most preferable one, procedures that produce succint representations of this set are of particular interest.

In this article, we extend the algorithm by Nemhauser and Ullman [12], a classical approach to solve the knapsack problem, to find a "good" representation of the set of efficient solutions, where the quality of the representation is measured in terms of ϵ-indicator, given an upper bound on the representation cardinality. This notion of representation was initially proposed in Vaz et al. [18] and is closely related to known approximation results in multiobjective optimization [13].

The technique proposed in this article assumes an *a priori* definition of targets in the objective space that, once reached, should provide a reasonable good representation to the set of efficient solutions. Then, during the run of the algorithm, a pruning step is applied to discard partial solutions that *provably* will not contribute to reach the targets.

The performance of this technique is evaluated in terms of representation quality, both experimentally and theoretically, and compared with an optimal representation. As opposed to known approaches in the literature, the technique described in this article is able to obtain a representation in a single run.

2 Definitions and notation

2.1 An unconstrained bi-objective combinatorial optimization problem

The UBCOP can be stated as follows:

$$\begin{aligned} \max\ P(x_1, x_2, \ldots, x_n) &:= \sum_{i=1}^{n} p_i x_i \\ \min\ W(x_1, x_2, \ldots, x_n) &:= \sum_{i=1}^{n} w_i x_i \end{aligned} \tag{UBCOP}$$

where $x_i \in \{0, 1\}$ are the decision variables, for $i = 1, \ldots, n$, p and w are the coefficients of such variables in the linear objective functions P and W, respectively; $p_i \in \mathbb{R}_{\geq 0}$ and $w_i \in \mathbb{R}_{\geq 0}$ are known as the *profit* and *weight* of *object* x_i, respectively.

Let X denote the set of feasible solutions in the decision space. We introduce the following dominance relation for this problem.

Definition 1 (Dominance) Let $x, x' \in X$. Then x dominates x', which is denoted by $(P(x), W(x)) \mathbin{\Delta} \big(P(x'), W(x')\big)$, iff $P(x) \geqslant P(x')$ and $W(x) \leqslant W(x')$ with at least one strict inequality.

A solution $x \in X$ is *efficient* iff there is no other feasible solution $x' \in X$ such that $\big(P(x'), W(x')\big) \mathbin{\Delta} (P(x), W(x))$ and its corresponding image in the objective space is a *nondominated outcome vector*. The set of all efficient solutions is called the *efficient set* and its image in the objective space, called the *nondominated set*, is denoted by Y.

Let $Z \subset \mathbb{R}^2$ denote the image of the feasible set X in the *objective space*. Let $Z^{\leqq} = \{z \in \mathbb{R}^2 : z \leqslant 0\}$ denote the *negative cone* formed by all the possible coordinates of the negative orthant; $Z^{\leqq}_{\bar{z}} = \bar{z} \oplus Z^{\leqq}$ denote the displaced cone $Z^{\leqq}$ at point $\bar{z}$; $\hat{Z} = \{z \in R^2 : z \in Z^{\leqq}_{\bar{z}} \text{ for all } \bar{z} \in Y\}$ denote the union of $Z^{\leqq}_{\bar{z}}$ for all $\bar{z} \in Y$; $\mathcal{Z} = \operatorname{conv}(\hat{Z})$

denote the convex hull of $\hat{Z}$; $\mathrm{int}(\mathcal{Z})$ denote the interior of $\mathcal{Z}$; and $\mathrm{bd}(\mathcal{Z})$ the boundary of $\mathcal{Z}$. Set Y is composed by *supported non-dominated outcome vectors*, those that belong to Y and are placed in $\mathrm{bd}(\mathcal{Z})$, which can easily be computed, for example, by the approach proposed in [5], and *unsupported non-dominated vectors*, those that belong to Y and are located in $\mathrm{int}(\mathcal{Z})$.

The Nemhauser-Ullman algorithm is a dynamic programming approach that was originally proposed to solve the knapsack problem [12] but it can easily be modified to solve UBCOP as well. The sequential process consists of n stages. At any stage i, the algorithm generates a set of states S_i that corresponds to a subset of the feasible solutions made up of items belonging exclusively to the first i items, $i = 1, \ldots, n$. A state $s = \left(s^1, s^2\right) \in S_i$ represents a solution of profit s^1 and weight s^2. At each stage $i = 1, \ldots, n$, the states are generated according to the following recursion:

$$S_i := ND\left(T_i := S_{i-1} \cup \left\{\left(s^1 + p_i, s^2 + w_i\right) : s \in S_{i-1}\right\}\right) \tag{1}$$

with the basis case $S_0 = \{(0,0)\}$. Operator $ND(\cdot)$ removes dominated states. Note that $S_n = Y$.

The running time of this approach depends on the size of set S_i, $i = 1, \ldots, n$. Although this set can grow exponentially in the worst case, Beier and Vöcking have shown that it grows only polynomially in the smooth case [2].

2.2 Representation quality

A large number of non-dominated outcome vectors can overwhelm a decision maker. Therefore, it is important to find good representations of set Y. The ϵ-indicator corresponds to the smallest factor that when multiplied to each element of a set $R \subseteq Y$, every point in the set Y becomes dominated [20]. Although well-known in the context of performance assessment of heuristics (see Zitzler et al. [21]), it has been only recently suggested as a representation measure [18]. It can be defined as

$$E(R, Y) := \max_{y \in Y} \min_{r \in R} \epsilon(r, y) \tag{2}$$

where

$$\epsilon(r, y) := \max\left(\frac{y_p + \alpha}{r_p + \alpha}, \frac{\widehat{W} - y_w + \alpha}{\widehat{W} - r_w + \alpha}\right) \tag{3}$$

for a very small α (since $(0,0)$ is always a non-dominated outcome vector for UBCOP), where $r = (r_p, r_w)$, $y = (y_p, y_w)$ and $\widehat{W} = \sum_{i=1}^{n} w_i$. The problem of finding a representation in terms of the ϵ-indicator can be formalized as finding a subset $R^* \subseteq Y$, $|R^*| \leqslant k$, that minimises $E(R, Y)$, that is,

$$R^* \in \underset{\substack{R \subseteq Y \\ |R| \leqslant k}}{\arg\min}\ E(R, Y) \tag{4}$$

Algorithms to compute an optimal representation, given that Y is known, have been introduced in Vaz et al. [18].

2.3 Solution techniques for representation

For many multiobjective combinatorial optimization problems, it is not possible to obtain set Y in a reasonable amount of time. In that case, it is preferable to find a representation without computing set Y. However, if finding an element in set Y is NP-hard, then, finding an element of an optimal representation is NP-hard as well. This is very often the case in multiobjective combinatorial optimization [17]. For that reason, some focus has been given on developing algorithms that obtain an approximation to the optimal representation with a certain representation guarantee. The following approaches solve a sequence of scalarizing problems in order to ensure a given representation quality.

Sayın [15] proposes a method to find discrete representations for multiobjective linear problems with coverage guarantees or a target cardinality. Coverage is understood as the minimum distance of the elements in Y to their closest elements in the representation. The method consists of an iterative process that at every step finds the point that provides the largest coverage value. It stops when either a given cardinality or a given coverage value is reached. Sayın and Kouvelis [16] use parametric search over weights with min-max sub-problems, where the representation guarantee is controlled by refining an interval between two previously generated non-dominated solutions until it falls below a specified threshold.

Hamacher et al. [9] propose a method to find a representative subset for bi-objective discrete optimisation problems with *box algorithms*. The idea is to start with a box given by two points that contains all non-dominated outcome vectors. Whenever a new non-dominated outcome vector is found, the box is split into smaller boxes. The algorithm stops when a given criterion based on the area of the box is reached. In the method proposed, there is a guarantee on the maximum cardinality with respect to the criterion.

Finally, Eusébio et al. [7] propose a method based on branch-and-bound to find a representation for the bi-objective integer network flow problem. It consists of solving a sequence of constrained formulations of the original problem, which allows to find a new non-dominated outcome vector, until the representation has the guaranteed quality.

It should be noted that these approaches provide some form of representation quality guarantee under certain conditions, but no guarantee on the approximation to the optimal representation is given. Moreover, these approaches need to solve several problems until reaching a certain representation quality.

Approximation results in terms of ϵ-indicator are known[1,4] when the goal is to find a representation of minimal cardinality, not necessarily containing only non-dominated outcome vectors.

3 A modified Nemhauser-Ullman algorithm

In the following, we describe a modification of Nemhauser-Ullman algorithm that returns an approximation to the optimal representation for UBCOP, containing only elements of set Y. It assumes that k weight values, $W_j, j = 1, \ldots, k$, are defined *a priori*; we name these weight values as *targets*. The representation returned by our approach consists of a set of at most k states, each of which has the lexicographically largest profit and smallest weight that satisfies a capacity constraint value defined by target $W_j, j = 1, \dots, k$. More formally, the goal is to find set R_{lex}

$$R_{\text{lex}} := \{ \operatorname*{lex}_{x \in X} (P(x), W(x)) \mid W(x) \leqslant W_j, j = 1, \dots, k\} \tag{5}$$

where lex denotes the lexicographic operator.

Definition 2 For $x \in X$, we define that x is lexicographically optimal if there is no $x' \in X$ such that $P(x') > P(x)$ or $(P(x') = P(x)$ and $W(x') < W(x))$.

Note that an optimal solution to the corresponding knapsack problem for a given capacity constraint value W_j, $j \in \{1, \dots, k\}$, may not be lexicographically optimal.

Our approach is based on the Nemhauser-Ullman dynamic programming algorithm, which allows to find a representation in a single run, as opposed to the methods mentioned in Section 2.3, and is able to find a lexicographically optimal solution for each target $W_j, j = 1, \dots, k$, if it exists. In addition to the removal of all dominated states at each iteration i, $i = 1, \dots, n$, as performed by the operator ND given in the recursive formula in Eq. (1), it also removes states that provably do not lead to elements in R_{lex}; we call *pruning step* to this additional filtering of set S_i. In the following sections, we introduce the technique in more detail.

3.1 Target definition

In our approach, we consider that targets W_j, $j = 1, \dots, k$, are equally spaced in the range $(0, \sum_{i=1}^{n} w_i)$, that is:

$$W_j := \frac{j-1}{k-1} \sum_{i=1}^{n} w_i \tag{6}$$

Note that due to the lexicographic operator, set R_{lex} contains only non-dominated outcome vectors. However, it may only provide an approximation to the optimal representation value. Furthermore, Proposition 1 shows that finding a representation for a given k may not even be possible.

Proposition 1 *There exists an instance for which $|R_{lex}| < k \leqslant |Y|$ holds.*

Proof Consider an instance with two variables and the following profits and weights: $p_1 = w_1 = \theta$ and $p_2 = w_2 = 3 - \theta$, with $0 < \theta < 1$. Then, set $Y = \{(0, 0), (\theta, \theta), (3 - \theta, 3 - \theta), (3, 3)\}$. For $k = 4$, there exists no efficient solution with total weight in the range $(1, 2)$; therefore $|R_{\text{lex}}| < k \leqslant |Y|$.

3.2 Pruning step

The pruning step is based on the computation of k lower and upper bounds of each state in $S_i, i = 1, \dots, n$ for a sequence of related k knapsack problems with capacity constraint values $W_j, j = 1, \dots, k$. For each state $s \in S_i, i = 1, \dots, n$, an extension of s is calculated, whose profit value is a lower bound on the maximum total profit that can be reached from that state for each constraint W_j. Note that if the lower bound is sufficiently tight, then the corresponding extension may be a candidate state for the representation. In addition, the upper bound on the profit value of state s for each constraint W_j is also computed; if the upper bound is inferior or equal to the profit of the candidate element for the representation for every constraint W_j, then state s can be discarded.

Let W_j be the value of a capacity constraint, $j = 1, \ldots, k$. At stage $i = 1, \ldots, n$, let $e_{W_j^i}(s) = (e^1_{W_j^i}(s), e^2_{W_j^i}(s))$ denote the extension of a state $s \in S_i$ as follows

$$e_{W_j^i}(s) := \left(s^1 + \sum_{\ell \in L_i} p_\ell, s^2 + \sum_{\ell \in L_i} w_\ell \right) \tag{7}$$

where $L_i \subseteq \{i+1, \ldots, n\}$ such that $e^2_{W_j^i}(s) \leqslant W_j$. This extension can be obtained by Dantzig's greedy algorithm for the knapsack problem[3]. Without loss of generality, assume that $i < \ell$ implies that $p_i/w_i > p_\ell/w_\ell$. Equation 7 can be reformulated as follows

$$e_{W_j^i}(s) := \left(s^1 + \sum_{\ell = i+1}^{c-1} p_\ell, s^2 + \sum_{\ell = i+1}^{c-1} w_\ell \right) \tag{8}$$

where $c \in \{i+1, \ldots, n\}$ denotes the index of the critical variable whose weight cannot be added to $e^2_{W_j^i}(s)$ without breaking the capacity constraint W_j.

For a given constraint W_j, an upper bound $u_{W_j^i}(s)$ on the profit value of a given state $s \in S_i, i = 1, \ldots, n$ is also given by [3]. Let $\overline{W}(s) = W_j - s^2$ be the residual capacity associated to s. Let

$$\bar{c} := \overline{W}(s) - \sum_{\ell = i+1}^{c-1} w_\ell \tag{9}$$

Thus, given the same ordering of the items, the upper bound is computed as

$$u_{W_j^i}(s) := s^1 + \sum_{\ell = i+1}^{c-1} p_\ell + \left\lfloor \bar{c} \frac{p_c}{w_c} \right\rfloor \tag{10}$$

A state $s' \in S_i$ can be pruned if, for each $j = 1, \ldots, k$, there exists a state $s \in S_i$ such that $u_{W_j^i}(s') \leqslant e^1_{W_j^i}(s)$. Note that the calculation of the extension and upper bounds for a given state takes $O(n)$-time for all capacity constraints $W_j, j = 1, \ldots, k$.

4 Quality representation guarantee

In the following, we give a negative result on the quality of the approximation to the optimal representation value.

Proposition 2 *There exists an instance for an even k, $\theta = \sum_{i=1}^n w_i = \sum_{i=1}^n p_i$ and $W_j = \frac{j-1}{k-1} \sum_{i=1}^n w_i$, for which, $E(R_{lex}, Y) = \theta$, and $E(R^*_\epsilon, Y) = \theta/(\theta - 1) \approx 1$ holds, where R^*_ϵ denotes an optimal representation with cardinality k in terms of ϵ-indicator.*

Proof Consider an instance with two variables with the following coefficients: $p_1 = \theta - 1$, $w_1 = 1$, $p_2 = 1$, $w_2 = \theta - 1$. Then, we have that $Y = \{(0,0), (\theta - 1, 1), (\theta, \theta)\}$. For $k = 2$ define $W_1 = 0$ and $W_2 = \theta$ and obtain the following sets

$$R_{\text{lex}} = \{(0,0), (\theta, \theta)\} \tag{11}$$

$$R^*_\epsilon = \{(0,0), (\theta - 1, 1)\} \tag{12}$$

Then it holds that $E(R^*_\epsilon, Y) = \theta/(\theta - 1) \approx 1$ and $E(R_{\text{lex}}, Y) = \theta$.

5 Experimental results

In the following, we describe an experimental analysis to assess the performance of our approach described in the previous section on a set of benchmark instances. Two measures are used to characterize its performance, the CPU-time and the representation value obtained. For comparison purpose, an optimal representation is also computed by solving the corresponding UBCOP to optimality with the original Nemhauser-Ullman algorithm and, then, applying the dynamic programming algorithm described in Vaz [18] to extract an optimal representation from set Y.

The implementations were written in `C` and compiled with `clang` 7.3.0 using the optimization flag `O2`. The tests were run in a computer with operating system OSX 10.11.4, with Intel i5, 2.7 GHz and 16 Gb 1867MHz DDR3 RAM. The instances were generated with three parameters: problem size (n), representation cardinality (k) and correlation betwen profits and weight vectors (ρ). Note that the positive (negative) correlation increases (decreases) the degree of conflict between the two objectives. For this reason, a positive (negative) correlation between weights and profits should give rise to a large (small) nondominated set. Profit and weight values were randomly generated according to a uniform distribution in $[1, 2^{31}/n]$. Ten instances were generated to each combination of the three parameters: $\rho = \{-0.8, 0.0, 0.8\}$, $n = \{200, 400, 600, 800, 1000\}$ and $k = \{0.1n, 0.4n, 0.7n, n\}$. The generation of correlated data follows the procedure described in Verel et al. [19].

Table 1 presents the experimental results obtained for the exact approach and our approach, averaged over the instances considered. We report the mean cardinality of the non-dominated set (column $|Y|$). For the exact approach, we report the mean of the CPU-time in seconds taken to find set Y with the original Nemhauser-Ullman algorithm (column t_{NU}) and the overall time to find an optimal representation (column t_{NU^*}), that is, the time reported in column t_{NU} plus the time taken by the approach described in Vaz et al. [18]; note that the time to find Y takes less than one minute for the instances considered, whereas extracting an optimal representation can take almost one hour for the largest instances. We also report the harmonic mean of the optimal representation value (column ϵ^*) as well as the mean of the CPU-time taken by our modified version of Nemhauser-Ullman algorithm (column t_{MNU}), the harmonic mean of the ratio between the cardinality of the representation obtained and k (column $|R_{\text{lex}}|/k$), the harmonic mean of the representation value (column ϵ) and harmonic mean of the representation ratio (column ϵ^*/ϵ).

The results in Table 1 clearly indicate that our modified version of Nemhauser-Ullman algorithm is faster than the exact approach by several orders of magnitude, mainly when the correlation is positive. Moreover, the performance of our approach seems to be independent of the induced correlation and the size of the instances. The representation quality obtained is also very good, with a representation ratio larger than 99% in most of the cases. Finally, the cardinality of the representation obtained is more than 99% of the value of k, which suggest that very few solutions are missed.

Note that the advantage of our approach is only meaningful for $k \leq 0.4n$, since it is possible to obtain the nondominated set with the original Nemhauser-Ullman algorithm and extract set R_{lex} in comparable time to our approach for larger values of k; compare the times reported in column t_{NU} with those in t_{MNU} for different values of k. Figure 1 plots the CPU-time taken by the original Nemhauser-Ullman algorithm (NU) and our approach for $k = n$ and $k = 0.1n$ for instance sizes from $n = 400$ up to $n = 5000$ and correlation equals to 0.8, where the largest times were obtained; the results of our

ρ	n	$\lvert Y\rvert$	t_{NU}	k	t_{NU^*}	ϵ^*	t_{MNU}	$\lvert R_{\text{lex}}\rvert/k$	ϵ	ϵ^*/ϵ
-0.8	200	4 151	0.05	20	0.10	1.0217	0.01	1.000	1.0433	0.979
				80	0.17	1.0055	0.02	1.000	1.0100	0.995
				140	0.24	1.0031	0.04	0.995	1.0070	0.996
				200	0.28	1.0022	0.05	0.979	1.0058	0.996
	400	15 622	0.31	40	1.21	1.0111	0.04	1.000	1.0205	0.991
				160	2.08	1.0028	0.16	0.999	1.0054	0.997
				280	2.80	1.0016	0.27	0.994	1.0034	0.998
				400	3.36	1.0011	0.41	0.981	1.0035	0.998
	600	32 894	1.14	60	5.09	1.0074	0.12	1.000	1.0134	0.994
				240	8.41	1.0018	0.49	0.999	1.0036	0.998
				420	11.21	1.0011	1.02	0.990	1.0029	0.998
				600	13.44	1.0007	2.02	0.981	1.0024	0.998
	800	55 177	2.60	80	14.76	1.0055	0.27	1.000	1.0099	0.996
				320	23.67	1.0014	1.28	0.998	1.0027	0.999
				560	31.17	1.0008	3.51	0.996	1.0020	0.999
				800	37.35	1.0006	5.48	0.985	1.0018	0.999
	1000	84 153	4.79	100	34.73	1.0044	0.52	1.000	1.0079	0.997
				400	54.19	1.0011	2.80	0.999	1.0020	0.999
				700	70.54	1.0006	7.51	0.996	1.0016	0.999
				1000	84.10	1.0004	11.89	0.983	1.0015	0.999
0.0	200	7 414	0.08	20	0.26	1.0285	0.01	1.000	1.0453	0.984
				80	0.43	1.0072	0.03	0.999	1.0113	0.996
				140	0.58	1.0041	0.05	0.999	1.0068	0.997
				200	0.70	1.0029	0.07	0.986	1.0082	0.995
	400	25 435	0.52	40	3.15	1.0144	0.06	1.000	1.0214	0.993
				160	5.02	1.0036	0.22	0.999	1.0053	0.998
				280	6.52	1.0021	0.38	0.997	1.0037	0.998
				400	7.86	1.0014	0.78	0.991	1.0037	0.998
	600	54 786	2.05	60	14.29	1.0096	0.17	1.000	1.0140	0.996
				240	21.80	1.0024	0.74	0.999	1.0037	0.999
				420	27.59	1.0014	1.70	0.997	1.0030	0.998
				600	32.89	1.0010	3.33	0.991	1.0027	0.998
	800	87 572	4.11	80	38.04	1.0071	0.39	1.000	1.0104	0.997
				320	56.79	1.0018	2.07	1.000	1.0026	0.999
				560	71.38	1.0010	4.98	0.996	1.0024	0.999
				800	84.29	1.0007	8.96	0.992	1.0019	0.999
	1000	139 850	8.43	100	98.75	1.0057	0.80	1.000	1.0083	0.997
				400	143.50	1.0014	4.63	0.999	1.0021	0.999
				700	177.19	1.0008	10.16	0.998	1.0018	0.999
				1000	207.34	1.0006	18.72	0.991	1.0016	0.999
0.8	200	19 230	0.20	20	1.51	1.0400	0.02	1.000	1.0497	0.991
				80	2.42	1.0100	0.08	1.000	1.0116	0.998
				140	3.04	1.0057	0.12	0.999	1.0078	0.998
				200	3.59	1.0040	0.20	0.991	1.0080	0.996
	400	93 008	2.12	40	38.02	1.0200	0.18	1.000	1.0235	0.997
				160	54.97	1.0050	0.68	1.000	1.0057	0.999
				280	65.46	1.0029	1.48	0.997	1.0048	0.998
				400	74.93	1.0020	2.59	0.995	1.0045	0.998
	600	194 479	7.46	60	177.90	1.0134	0.74	1.000	1.0154	0.998
				240	250.49	1.0034	2.94	1.000	1.0038	1.000
				420	294.17	1.0019	6.34	0.998	1.0033	0.999
				600	332.36	1.0013	10.70	0.996	1.0029	0.998
	800	394 141	18.66	80	855.14	1.0100	1.73	1.000	1.0115	0.999
				320	1168.64	1.0025	8.28	1.000	1.0031	0.999
				560	1346.39	1.0014	17.22	0.999	1.0024	0.999
				800	1504.93	1.0010	31.03	0.994	1.0022	0.999
	1000	546 147	34.87	100	1685.44	1.0080	3.87	1.000	1.0091	0.999
				400	2269.41	1.0020	17.73	1.000	1.0023	1.000
				700	2608.02	1.0011	40.68	0.998	1.0022	0.999
				1000	2885.25	1.0008	78.17	0.996	1.0018	0.999

Table 1 Experimental results (see text for more details)

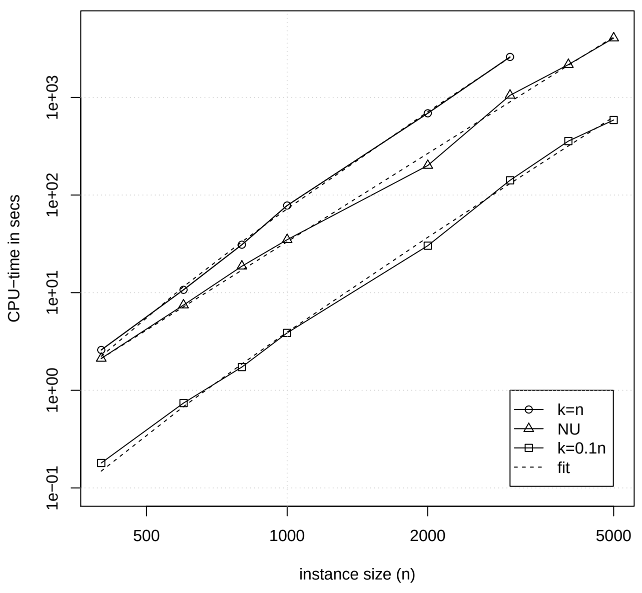

Fig. 1 CPU-time taken by the several approaches and corresponding regression lines, with respect to instance size.

approach for $k = n$ are not shown for $n > 3000$ since they were above the cut-off time of 5000 seconds. The CPU-time to extract set R_{lex} from the nondominated set is not taken into account since it is very neglible. The plot also shows regression (dashed) lines for each of the three approaches; we have used a cubic regression model as sugested by the Box-Cox procedure, with $R^2 = 0.9971$ for the original Nemahauser-Ullman algorithm, and $R^2 = 0.9997$ and $R^2 = 0.9972$ for our approach with $k = n$ and $k = 0.1n$, respectively. These models suggest that the algorithms tested on those type of instances have a cubic running-time with respect to instance size. These experimental results confirm that our modified version of Nemhauser-Ullman is preferable for smaller values of k. Note that a small value of k is often seen as a requirement for a representation [14]; in our approach, the representation quality in terms of ϵ-indicator does not seem to degrade too much for decreasing values of k, as shown in column ϵ^*/ϵ in Table 1.

6 Conclusions and discussion

In this article, we described a modification of the Nemhauser-Ullman algorithm to find a "good" representation of the non-dominated set in terms of a representation measure for an unconstrained biobjective combinatorial optimization problem. The techniques are based on the definition of several constraint values on the range of the weights. Our theoretical results indicate that the gap between the optimal representation value and the representation value achieved by this technique can be arbitrarly large. However, the experimental results obtained show that this approach is able to perform very well in terms of CPU-time and representation quality, in a wide range of instances and mainly for small values of k.

Other targets can be considered, for instance, by taking into account some knowledge of set Y, in particular, the set of supported non-dominated outcome vectors, which can be computed with the dichotomic search algorithm described in [5]. In this variant, targets W_j, $j = 1, \ldots, k$, are chosen such that they are equally spaced in the polyline that connects the supported non-dominated outcome vectors. However, preliminary experimental results indicate that there is almost no gain in terms of representation quality and running time [10].

The major drawback of this approach is that small gaps may arise in the representation obtained since the desirable cardinality k may not be possible to obtain. However, in practice, one may re-run the algorithm for slightly different values of k until an "ungapped" representation is obtained.

Finally, it would be interesting to consider other well-known representation measures, such as hypervolume indicator, which measures the area of the dominated region, bounded by a reference point [20]. If the non-dominated outcome vectors are explictly given, it is possible to find an optimal representation in the bi-objective case in polynomial time [11] as well as to find an approximation in the three-objective case with a greedy strategy [8]. Other representation measures of interest are uniformity and coverage [14]. However, preliminary experiments indicated that the variants proposed in this article only obtain good results in some combinations of instance parameters [10].

Acknowledgments This work was partially supported by the European Regional Development Fund (FEDER), through the COMPETE 2020 — Operational Program for Competitiveness and Internationalization (POCI).

References

1. C. Bazgan, F. Jamain, and D. Vanderpooten. Approximate Pareto sets of minimal size for multi-objective optimization problems. *Operation research letters*, 43(1):1–6, 2015.
2. R. Beier and B. Vöcking. Random knapsack in expected polynomial time. In L.L. Larmore and M.X. Goemans, editors, *Proc. of the 35rd Annual ACM Symposium on Theory of Computing (STOC)*, pages 232–241. ACM Press, 2003.
3. G.B. Dantzig. Discrete-variable extremum problems. *Operations Research*, 5(2):266–288, 1957.
4. I. Diakonikolas and M. Yannakakis. Small approximate Pareto sets for bi-objective shortest paths and other problems. *SIAM Journal on Computing*, 39(4):1340–1371, 2009.
5. M. Ebem-Chaime. Parametric solution for linear bricriteria knapsack models. *Management Science*, 42(11):1565–1575, 1996.
6. Matthias Ehrgott. *Multicriteria optimization.* Springer, 2nd edition, 2005.
7. A. Eusébio, J.R. Figueira, and M. Ehrgott. On finding representative non-dominated points for bi-objective integer network flow problems. *Computers & Operations Research*, 48:1–10, 2014.
8. A.P. Guerreiro, C.M. Fonseca, and L. Paquete. Greedy hypervolume subset selection in low dimensions Evolutionary Computation, 24(3):521–544, 2016.
9. H.W. Hamacher, C.R. Pedersen, and S. Ruzika. Finding representative systems for discrete bicriterion optimization problems. *Operations Research Letters*, 35(3):336–344, 2007.
10. A. Jesus. Implicit enumeration for representation systems in multiobjective optimization. Master's thesis, University of Coimbra, Portugal, 2015.
11. T. Kuhn, C.M. Fonseca, L. Paquete, S. Ruzika, M.M. Duarte, and J.R. Figueira. Hypervolume subset selection in two dimensions: Formulations and algorithms. *Evolutionary Computation*, 24(3):411–425, 2016.

12. G. Nemhauser and Z. Ullman. Discrete dynamic programming and capital allocation. *Management science*, 15(9):494–505, 1969.
13. C. H. Papadimitriou and M. Yannakakis. On the approximability of trade-offs and optimal access of web sources. In *Proceedings of the 41st Annual Symposium on Foundations of Computer Science*, FOCS '00, pages 86–92, Washington, DC, USA, 2000. IEEE Computer Society.
14. S. Sayın. Measuring the quality of discrete representations of efficient sets in multiple objective mathematical programming. *Mathematical Programming*, 87(3):543–560, 2000.
15. S. Sayın. A procedure to find discrete representations of the efficient set with specified coverage errors. *Operations Research*, 51(3):427–436, 2003.
16. S. Sayın and P. Kouvelis. The multiobjective discrete optimization problem: A weighted min-max two-stage optimization approach and a bicriteria algorithm. *Management Science*, 51(10):1572–1581, 2005.
17. P. Serafini. Some considerations about computational complexity for multiobjective combinatorial optimization. In *Recent Advances and Historical Development of Vector Optimization*, Lecture Notes in Economics and Mathematics, pages 221–231, Berlin, Germany, 1986. Springer.
18. D. Vaz, L. Paquete, C.M. Fonseca, K.Klamroth, and M.Stiglmayr. Representation of the non-dominated set in biobjective discrete optimization. *Computers & Operations Research*, 63:172 – 186, 2015.
19. S. Verel, A. Liefooghe, L. Jourdan, and C. Dhaenens. Analyzing the effect of objective correlation on the efficient set of MNK-landscapes. In *Proceedings of the 5th Conference on Learning and Intelligent OptimizatioN (LION 5)*, Lecture Notes in Computer Science, pages 116–130. Springer, 2011.
20. E. Zitzler and L. Thiele. Multiobjective optimization using evolutionary algorithms—a comparative case study. In *Proceedings of the International Conference on Parallel problem solving from nature—PPSN V*, pages 292–301. Springer, 1998.
21. E. Zitzler, L. Thiele, M. Laumanns, C.M. Fonseca, and V.G. da Fonseca. Performance assessment of multiobjective optimizers: an analysis and review. *IEEE Transactions on Evolutionary Computation*, 7(2):117–132, April 2003.