



# Bounds on efficient outcomes for large-scale cardinality-constrained Markowitz problems

Janusz Miroforidis<sup>1</sup> 

Received: 3 March 2020 / Accepted: 30 March 2021 / Published online: 23 April 2021  
© The Author(s) 2021

## Abstract

When solving large-scale cardinality-constrained Markowitz mean–variance portfolio investment problems, exact solvers may be unable to derive some efficient portfolios, even within a reasonable time limit. In such cases, information on the distance from the best feasible solution, found before the optimization process has stopped, to the true efficient solution is unavailable. In this article, I demonstrate how to provide such information to the decision maker. I aim to use the concept of lower bounds and upper bounds on objective function values of an efficient portfolio, developed in my earlier works. I illustrate the proposed approach on a large-scale data set based upon real data. I address cases where a top-class commercial mixed-integer quadratic programming solver fails to provide efficient portfolios attempted to be derived by Chebyshev scalarization of the bi-objective optimization problem within a given time limit. In this case, I propose to transform purely technical information provided by the solver into information which can be used in navigation over the efficient frontier of the cardinality-constrained Markowitz mean–variance portfolio investment problem.

**Keywords** Bi-objective optimization · Multiple criteria decision making · Cardinality-constrained Markowitz portfolio investment problem · Two-sided efficient frontier approximations · Chebyshev scalarization

## 1 Introduction and motivation

Due to the globalization of financial markets, investors have access to thousands of financial instruments (assets). The portfolio selection problem can be modelled with the use of the famous Markowitz’s mean–variance (bi-objective) model [30]—the mean–variance problem (MVP)—or its extensions. The portfolio selection problem has long attracted the attention of researchers. A recent survey of modern portfolio theory can be found in [29]. The MVP is still a base for other practical models of the portfolio selection problem with additional real-life constraints, e.g., transaction costs, not present in the MVP. As nowadays a potentially large set of assets is available to investors, efficient (or: Pareto optimal) portfolios of the

---

✉ Janusz Miroforidis  
janusz.miroforidis@ibspan.waw.pl

<sup>1</sup> Systems Research Institute, Polish Academy of Sciences, ul. Newelska 6, 01-447 Warsaw, Poland

MVP might consist of a relatively large number of assets (often fractions of an efficient portfolio invested in these assets are very small). Portfolio management with a large number of assets might be costly, especially when an investor makes investment decisions relatively often. In order to limit the number of assets in efficient portfolios, a cardinality constraint can be added to the MVP [34]. This leads to the Cardinality-Constrained Markowitz mean–variance portfolio investment Problem (CCMP). Instead of adding the additional constraint on portfolio cardinality, one can add the third criterion (portfolio cardinality) to the MVP (see, e.g., [1]). My focus is the (bi-objective) CCMP.

The essence of solving practical instances of the CCMP is to select an efficient portfolio (or: solution) by the decision maker (DM) (or: investor) which best fits his/her preferences (see, e.g., [12]). I make a rational assumption that the DM's focus is only on the set of efficient portfolios (or: efficient set). In order to grasp the mean–variance trade-off, he/she navigates over the efficient frontier (image of the efficient set in the mean–variance space). This, however, requires the derivation of an a priori unknown number of efficient portfolios during the decision making process. It may be problematic for large-scale instances of the CCMP, especially when their efficient portfolios have to be derived during the interactive decision making process on demand, and an effective search among potential decisions to such problems can be time consuming or even beyond allotted limits.

For a large-scale bi-objective optimization problem, the cost of the derivation of a subset of its efficient set cannot be ignored during the decision making process. I want to emphasize that in the interactive multiple criteria decision making (MCDM) methods (see, e.g., [27,32,38]), efficient solutions are determined by solving a number of single-objective optimization problems using the so-called scalarization techniques. Such (exact) computations may be time consuming. The DM should take this into account in the case of large-scale problems, especially when he/she wishes to evaluate many efficient solutions during the interactive decision making process.

An alternative way to determine (exact) efficient solutions in interactive MCDM methods is to use approximate (metaheuristic) methods, e.g., evolutionary single-objective optimization (see, e.g., [31]). However, they produce, in general, only approximate solutions. This is the main drawback of these methods—one is never sure how far from the efficient solution its approximation lies. Even if they work well for a set of test problems (benchmarks), there is no guarantee that they give a good approximation of the efficient solution to a given problem. In other words, the DM may reveal his/her preferences during the decision making process, evaluating only the possibly rough approximations of efficient solutions found by means of the metaheuristic method. This may, however, result in making a bad (i.e., suboptimal) decision (the decision making process is biased).

The approximation of the whole efficient set of the CCMP can be found using metaheuristics (see, e.g., [3,41]). However, when making use of metaheuristic-based multiobjective optimization methods, efficiency of derived solutions is also not guaranteed. Evolutionary multiobjective optimization [5,6] is just an example. The review of other methods of approximating the efficient set can be found in [36]. Some of them are dedicated to a specific class of the bi-objective optimization problem (e.g., linear problems).

The CCMP can be modelled as the bi-objective mixed-integer quadratic programming (MIQP) problem. With the use of either the linear scalarization technique or  $\epsilon$ -constraint method (see, e.g., [32]), commercial or open-source MIQP solvers can be applied to derive efficient solutions to the CCMP. However, for large-scale instances of the CCMP, even commercial top-class MIQP solvers (e.g., CPLEX, Gurobi) may be unable to find some efficient solutions within a reasonable time limit on optimization. The existence of this time limit is

the rational assumption, as the DM may wish to evaluate a possibly numerous set of efficient portfolios during the decision making process.

To account for this, when the MIQP solver is not able to derive an efficient solution within the assumed time limit on optimization (computational budget), I propose to navigate over the efficient frontier of the CCMP with the use of *lower bounds* and *upper bounds* on values of objective functions of implicit efficient solutions. In the decision making process, such bounds serve as replacements for the exact values of the objective functions [26]. Formulas for calculating lower and upper bounds are relatively simple—they contain only arithmetic operators and operators of deriving minimal and maximal elements in a set of numbers [18,24]. The sources of such bounds are two finite sets. I shall call them *lower shells* and *upper shells*. Images of these two sets in the mean–variance space form (in a sense) finite two-sided efficient frontier approximations.

There are also other ideas for two-sided efficient frontier approximations in the literature, e.g., bound sets for bi-objective combinatorial optimization problems [11]. Some methods are dedicated to a specific type of the bi-objective optimization problem, e.g., inner and outer approximations of the efficient frontier for linear bi-objective optimization problems [37]. Inner and outer approximations can also be derived for convex efficient sets by so-called sandwich algorithms (see, e.g., [35]). However, in the current article I make use of lower and upper shells, whose images in the objective space are *local* (dependent on the DM's preferences) two-sided efficient frontier approximations. I use them to calculate lower and upper bounds on components of implicitly given efficient solution outcomes being the DM's focus. By doing so, unnecessary calculations focused on determining the two-sided approximation of the whole efficient frontier can be avoided. I show how to find lower and upper shells controlled by the DM's preferences with the use of purely technical information provided by a top-class MIQP solver solving computationally demanding instances of the parametrized single-objective optimization problem, which is Chebyshev scalarization (see, e.g., [39]) of the CCMP, under time limit on optimization. I investigate experimentally the accuracy of lower and upper bounds with an instance of the CCMP from my own large-scale data set based upon actual data.

It is worth noting here that inventive methods (which fully exploit properties of MIQP problems) to find efficient solutions to large-scale instances of the CCMP have been proposed in recent years (see, e.g., [2,4,7,9,13,14]). However, none of these methods use Chebyshev scalarization to derive efficient solutions. For example, in [4], an efficient portfolio is derived by minimizing the portfolio variance with a given expected return, and in [2], the weighted sum of the portfolio variance and portfolio expected return is used.

Navigating over the efficient frontier with the help of Chebyshev scalarization is a special case of the well-known Wierzbicki's reference point method for interactive-iterative MCDM frameworks [40]. In the navigation method based on Chebyshev scalarization, the DM must make concessions to the so-called ideal (or: reference) point representing the ideal portfolio, which usually does not exist. This trade-off guided method of unveiling the DM's preferences is an alternative to other methods and it is worth considering when solving the CCMP.

The outline of the article is as follows. In Sect. 2, I recall the bi-objective problem formulation in its most general form and recall a method for the derivation of efficient solutions which is instrumental for the subsequent development. In Sect. 3, the CCMP is formulated and some of its properties that are my motivation to apply lower and upper bounds are shown. In Sect. 4, I show how to calculate lower and upper bounds on objective function values of implicitly given efficient solutions, as well as a method of populating lower and upper shells to the CCMP. In Sects. 5 and 6, I illustrate how the bounds work on a large-scale instance of the CCMP when a time limit on optimization is assumed. Section 7 contains my conclusions.

## 2 Bi-objective optimization

### 2.1 Preliminaries

Let  $x$  denote a solution,  $X$  a space of solutions,  $X \subseteq \mathbb{R}^n$ ,  $X_0$  a set of feasible solutions,  $X_0 \subseteq X$ . Then the general bi-objective optimization problem is defined as:

$$\begin{aligned} & \text{vmax} f(x) \\ & \text{s.t.} \\ & x \in X_0, \end{aligned} \tag{1}$$

where  $f : X \rightarrow \mathbb{R}^2$ ,  $f = (f_1, f_2)$ ,  $f_l : X \rightarrow \mathbb{R}$ ,  $l = 1, 2$ , are objective functions, and *vmax* denotes the operator of deriving all efficient solutions in  $X_0$ .  $\mathbb{R}^2$  is called the objective space (or: outcome space).

Solution  $\bar{x}$  is efficient if  $f_l(x) \geq f_l(\bar{x})$ ,  $l = 1, 2$ , implies  $f(x) = f(\bar{x})$ . If  $f_l(x) \geq f_l(\bar{x})$ ,  $l = 1, 2$ , and  $f(x) \neq f(\bar{x})$ , then it is said that  $x$  dominates  $\bar{x}$ , which is written  $\bar{x} \prec x$ . Solution  $\bar{x}$  is weakly efficient (or: weakly Pareto optimal) if there is no such  $x$  that  $f_l(x) > f_l(\bar{x})$ ,  $l = 1, 2$ . Element  $f(x)$ ,  $x \in X_0$ , is the outcome of  $x$ . Below, I shall denote the set of efficient solutions to (1) by  $N$ . Set  $f(N)$  is called the efficient frontier (EF) or the set of efficient outcomes.

### 2.2 Chebyshev scalarization

It is a well-established result (cf. [10,18,32]) that solution  $x$  is efficient if and only if it solves the augmented Chebyshev weighted optimization problem

$$\begin{aligned} & \min\{\max_l \lambda_l((y_l^* - f_l(x)) + \rho e^\top (y^* - f(x)))\} \\ & \text{s.t.} \\ & x \in X_0, \end{aligned} \tag{2}$$

where weights  $\lambda_l > 0$ ,  $l = 1, 2$ ,  $e$  is the all-ones vector,  $y_l^* = \hat{y}_l + \varepsilon$ ,  $\hat{y}_l = \max_{x \in X_0} f_l(x)$ ,  $l = 1, 2$ ,  $\varepsilon > 0$ , and  $\rho$  is a positive “sufficiently small” number. Actually, with the use of (2) no the so-called improperly efficient solution [15] can be derived (see, e.g., [18,32]). This, however, is not a barrier when solving practical decision making problems [18]. By solving problem (2) with  $\rho = 0$ , weakly efficient solutions are derived (cf., e.g., [32]).

By the “only if” part of this result, no efficient solution is a priori excluded from being derived by solving an instance of optimization problem (2). In contrast to that, maximization of a weighted sum of objective functions over  $X_0$  does not, in general (and especially in the case of problems with discrete variables), possess this property (cf., e.g., [10,18,27,32]).

At first glance the objective function in (2) seems to be difficult to handle. However, problem (2) is equivalent to

$$\begin{aligned} & \min s \\ & \text{s.t.} \\ & s \geq \lambda_l((y_l^* - f_l(x)) + \rho e^\top (y^* - f(x))), \quad l = 1, 2, \\ & x \in X_0. \end{aligned} \tag{3}$$

In the sequel, I will assume that efficient solutions are derived by solving problem (2) with varying  $\lambda$ .

### 2.3 Chebyshev scalarization for interactive MCDM methods

In interactive MCDM methods, the DM navigates over the set of efficient outcomes unveiling his/her preferences gradually, step by step (see, e.g., [18,32]). The decision making process stops when the DM points out the most preferred efficient outcome. Of the many existing MCDM methods, interactive ones seem to be the most reasonable. In these methods, there is no formal stopping rule of the decision making process—the DM is a sovereign actor in it. The process stops when the DM’s preferences are firmly set or the DM no longer wishes to continue it. An example of an interactive MCDM method is the well-known Wierzbicki’s reference point method [40].

The Chebyshev scalarization technique can be used in interactive MCDM methods (see, e.g., [38]). In [18], the preference-driven navigation method (valid for the number of objective functions greater than or equal to two) over the set of efficient outcomes is proposed. This method relies on the notion of the *vector of concessions* serving the DM as the preference carrier. In practical decision making problems, element  $y^*$  is usually unattainable. Hence, to get an efficient outcome, the DM has to compromise on  $y^*$ . In the case of two objective functions, he/she can do this by selecting a vector of concessions  $\tau$ ,  $\tau_1 > 0$ ,  $\tau_2 > 0$ . Vector  $\tau$  defines proportions in which the DM agrees to sacrifice unattainable values of components of  $y^*$  in a quest for an element of the efficient frontier. The DM can explicitly define vector  $\tau$  by indicating its components. The DM can also do this implicitly by indicating a *base point*  $y^b \in \{y \mid y_1 \in y_1^* - \mathbb{R}_+^2, y_2 \in y_2^* - \mathbb{R}_+^2\}$  ( $\mathbb{R}_+^2$  is the positive orthant) which defines  $\tau$  as  $\tau_1 = y_1^* - y_1^b$ ,  $\tau_2 = y_2^* - y_2^b$ . It is shown in [18] that  $\lambda_1 = \frac{1}{\tau_1}$ ,  $\lambda_2 = \frac{1}{\tau_2}$ . Hence, translation from  $\tau$  or  $y^b$  to  $\lambda$  is straightforward. Vector  $\lambda$  (a parameter of the objective function of problem (2)) determines the so-called search direction in the objective space (Fig. 1). The DM can express his/her preferences using  $\lambda$  vectors explicitly.

The trade-off guided method of unveiling the DM’s preferences presented above can be used to solve practical decision making problems (see, e.g., [17,26,33]). The benefit of using Chebyshev scalarization is that it can be applied in the user-friendly process of unveiling the DM’s preferences [19], which can be expressed by vectors of concessions, weight vectors (the atomistic manner), or base points (the holistic manner).

## 3 The CCMP

In the case of “the more, the better” shown in the previous section, the CCMP is formulated as follows.

$$\begin{aligned}
 \text{vmax} \quad & \begin{cases} f_1(x) := -x^\top Qx \\ f_2(x) := r^\top x \end{cases} \\
 \text{s.t.} \quad & \\
 & x \in X_0,
 \end{aligned} \tag{4}$$

where  $f_1(x)$  (*-VARIANCE*) is the negative value of the portfolio  $x$  variance (maximizing the negative value of the portfolio variance is equivalent to minimizing the portfolio variance),  $f_2(x)$  (*EXP. RETURN*) is the expected return (mean) of portfolio  $x$ ,  $X_0 := \{x \mid e^\top x = 1, \text{card}(x) \leq C, x \geq 0\}$ ,  $x := (x_i) \in \mathbb{R}^n$ , and  $x_i$  is the fraction of the portfolio invested in the  $i$ -th asset.  $Q \in \mathbb{R}^{n \times n}$  is the covariance matrix,  $r \in \mathbb{R}^n$  is the vector of expected returns,  $\text{card}(x) := \sum_{i=1}^n \text{sgn}(x_i)$ , where  $\text{sgn}(\cdot)$  is the signum function,  $C, C > 0$ , is the upper bound on a number of assets in the portfolio, and  $\text{card}(x) \leq C$  is the cardinality constraint.

By introducing  $n$  binary variables  $z_i$  ( $z_i = 1$  if the  $i$ -th asset is selected;  $z_i = 0$ , otherwise), as well as additional constraints coupling continuous variables  $x_i$  with binary ones, problem (4) can be formulated as the bi-objective MIQP problem in the following way.

$$\begin{aligned}
 & \text{vmax} \begin{cases} f_1(x) \\ f_2(x) \end{cases} \\
 & \text{s.t.}
 \end{aligned} \tag{5}$$

$$x \in X_0 := \{x \mid e^\top x = 1, e^\top z \leq C, \forall_{i=1, \dots, n} (-x_i + Mz_i \geq 0), x \geq 0\},$$

where  $M$  is a “sufficiently big number”, and  $z = (z_i)$ .

In the general case, the efficient frontier of this problem is not a continuous curve (see, e.g., [3]). It may also contain elements which cannot be derived by linear scalarization (the so-called unsupported elements). With the help of scalarization (2) and taking into account (3), each efficient solution to (5) can be derived by solving an instance of the MIQP problem (in fact, the instance with the linear objective function subject to linear and quadratic constraints).

As it is shown in Sect. 2.3, the DM can navigate over the efficient frontier using  $\lambda$  parameters of the scalarization function in problem (2) as preference carriers. In this article, it is assumed that the DM acts in this way. Further on, I assume that  $\lambda_1 + \lambda_2 = 1$ .

It is well known that the derivation of low-risk efficient portfolios is much more difficult than high-risk ones (see, e.g., [2,13]).

### 4 Lower and upper bounds on objective function values of efficient solutions

Given  $\lambda$ , let  $x^{\text{ef}}(\lambda)$  denote the efficient solution to problem (2), which would be derived if this problem were solved to optimality (the implicit efficient solution designated by  $\lambda$ ).  $f(x^{\text{ef}}(\lambda))$  denotes the implicit efficient outcome designated by  $\lambda$ . In this section, I briefly present the theory of general <sup>1</sup> lower and upper bounds on components of implicit efficient outcomes I developed earlier [20–22].

#### 4.1 Lower bounds

Lower shell to problem (1) is a finite nonempty set  $S_L \subseteq X_0$  whose elements satisfy

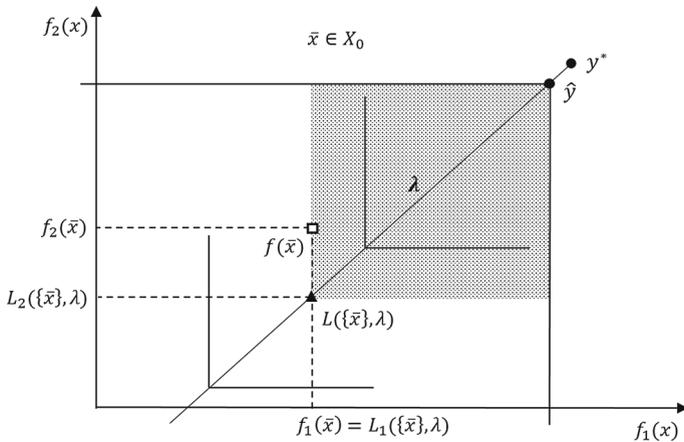
$$\forall x \in S_L \nexists x' \in S_L \ x < x'. \tag{6}$$

$S_L$  can be a singleton.

For the given  $y^*$  and  $\rho$ , when deriving efficient solutions to (1) with the use of problem (2), the formulas for lower bounds are

$$\begin{aligned}
 & f_1(x^{\text{ef}}(\lambda)) \geq L_1(S_L, \lambda) \\
 & := \max \left\{ y_1^* - (\lambda_1(1 + \rho))^{-1} \min_{x \in S_L} [\max_{j=1,2} \lambda_j (y_j^* - f_j(x)) \right. \\
 & \quad \left. + \rho e^\top (y^* - f(x))] + \frac{\rho}{1 + \rho} (y_2^* - U_2(\lambda)), \bar{L}_1 \right\}, \\
 & f_2(x^{\text{ef}}(\lambda)) \geq L_2(S_L, \lambda)
 \end{aligned} \tag{7}$$

<sup>1</sup> The term “general” means that I made no additional assumptions on properties of problem (1), e.g., convexity of set  $f(X_0)$ .



**Fig. 1** The idea of the derivation of formulas for lower bounds with  $\rho = 0$ :  $\square$ —image of lower shell  $\{\bar{x}\}$  in the objective space,  $\blacktriangle$ —vector of lower bounds on components of  $f(x^{ef}(\lambda))$

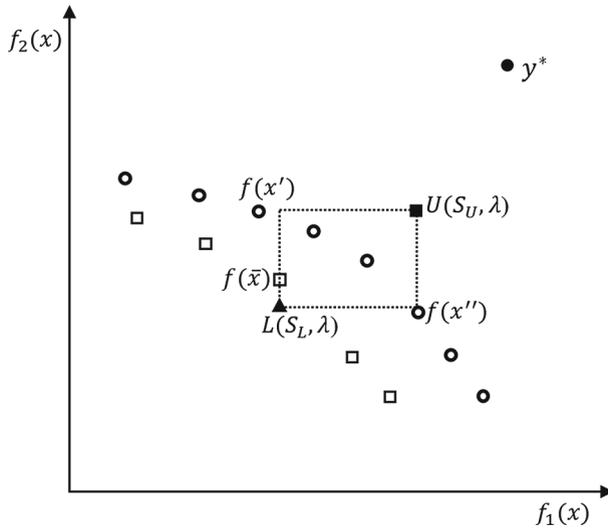
$$\begin{aligned}
 & := \max \left\{ y_2^* - (\lambda_2(1 + \rho))^{-1} \min_{x \in S_L} [\max_{j=1,2} \lambda_j (y_j^* - f_j(x)) \right. \\
 & \quad \left. + \rho e^\top (y^* - f(x))] + \frac{\rho}{1 + \rho} (y_1^* - U_1(\lambda)), \bar{L}_2 \right\}, \tag{8}
 \end{aligned}$$

where  $\bar{L}_l, l = 1, 2$ , are static lower bounds implied by some a priori information (e.g., when all objective functions are non-negative) on  $f_l(x^{ef}(\lambda))$ . If static lower bounds are not known, then it is possible to simply set  $L_l := -\infty$ .  $U_l(\lambda), l = 1, 2$ , are known upper bounds on  $f_l(x^{ef}(\lambda))$  (e.g., when one can infer these bounds based on properties of problem (1)). For details, see [18,24,25]. If  $U_l(\lambda)$  are not known, then it is possible to simply set  $U_l(\lambda) := y_l^*$ . If this is the case, in formulas (7) and (8), elements with factor  $\frac{\rho}{1+\rho}$  vanish.

Given  $\lambda$ , the idea of the derivation of formulas (7) and (8) for a simpler case with  $\rho = 0$  and a lower shell  $\{\bar{x}\}$  is shown in Fig. 1. Apexes of displaced cones (the cones correspond to contours of the augmented Chebyshev metric used in (2)) lie on the half line (determined by  $\lambda$ ) emanating from element  $y^*$ . The apex, being the interception point of the half line with straight line  $f_1(x) = f_1(\bar{x})$ , determines the position of the cone most distant from  $y^*$  (border cone) on which  $f(x^{ef}(\lambda))$  would lie. Thus, this point determines vector of lower bounds  $L(\{\bar{x}\}, \lambda)$ . If  $f(x^{ef}(\lambda))$  lay on the border of a cone which is farther from  $y^*$  than the border cone,  $f(x^{ef}(\lambda))$  would not be efficient. Therefore, element  $f(x^{ef}(\lambda))$  is located somewhere in the dashed rectangle in Fig. 1. When taking into account lower shell  $S_L, |S_L| > 1$ , one has to consider the best lower bound on  $f_l(x^{ef}(\lambda)), l = 1, 2$ . Hence, the “ $\min_{x \in S_L} [ \dots ]$ ” appears in formulas (7) and (8). If  $\rho > 0$ , then displaced cones are open, and terms with  $\rho$  appear in the formulas. For details see, e.g., [18,24].

A lower shell is simply any finite approximation (composed of feasible elements only) of the efficient set containing non-dominated elements only (see (6)). Given two lower shells  $S_L$  and  $S'_L$ , set  $S''_L := S_L \cup S'_L$  is a lower shell after removing dominated elements in this set.

In case of the CCMP, an element of a lower shell can be derived by solving, within a given time limit  $T$ , problem (2) (being scalarization of problem (5)) with a given  $\lambda$ . In that case, either the exact solution or incumbent (approximate solution) can be obtained.



**Fig. 2** The illustration to the idea of general lower and upper bounds:  $\square$ —image of lower shell  $S_L$  in the objective space,  $\circ$ —image of upper shell  $S_U$  in the objective space,  $\blacktriangle$ —vector of lower bounds,  $\blacksquare$ —vector of upper bounds

### 4.2 Upper bounds

Upper shell to problem (1) is a finite nonempty set  $S_U \subseteq \mathbb{R}^n$ , elements of which satisfy

$$\forall x \in S_U \nexists x' \in S_U \quad x' \prec x, \tag{9}$$

$$\forall x \in S_U \nexists x' \in N \quad x \prec x'. \tag{10}$$

$S_U$  can be a singleton.

Elements of an upper shell may be infeasible. If feasible, they are efficient. By (10) they are not dominated by any efficient solution to problem (1). Outcomes of elements of an upper shell are located either on or (in a sense) above the efficient frontier of problem (1).

Given  $\lambda$  and  $S_L$ , element  $x$  of the upper shell can be a source of the valid upper bound on  $f_l(x^{ef}(\lambda))$  for some  $l$ , when  $f(x)$  is appropriately located with respect to the vector of lower bounds  $L(S_L, \lambda) := (L_1(S_L, \lambda), L_2(S_L, \lambda))$ . This is specified by the following lemma (the proof of this lemma for more than two objective functions is found in [22]).

**Lemma 1** *Given lower shell  $S_L$  and upper shell  $S_U$ . Suppose  $x \in S_U$  and  $L_{\bar{l}}(S_L, \lambda) \leq f_{\bar{l}}(x)$  for some  $\bar{l}$  and  $L_l(S_L, \lambda) \geq f_l(x)$  for all  $l = 1, 2, l \neq \bar{l}$ . Then  $x$  provides an upper bound for  $f_{\bar{l}}(x^{ef}(\lambda))$ , namely  $f_{\bar{l}}(x^{ef}(\lambda)) \leq f_{\bar{l}}(x)$ .*

Given  $\lambda$ , if for upper shell  $S_U$  its elements provide upper bounds  $U_l(S_U, \lambda)$  only on  $f_l(x^{ef}(\lambda))$ ,  $l \in I, I \subset \{1, 2\}, I \neq \emptyset$ , then one can set  $U_l(S_U, \lambda) := \hat{y}_l$  (or  $U_l(S_U, \lambda) := y_l^*$ , when  $\hat{y}_l$  is unknown, and  $y_l^*, \hat{y}_l \leq y_l^*$ , is set a priori),  $l \in \{1, 2\} \setminus I$ . Since many elements of  $S_U$  may provide upper bounds on  $f_l(x^{ef}(\lambda))$ , I take the smallest upper bound, namely  $U_l(S_U, \lambda) = \min_{x \in S_U} U_l(\{x\}, \lambda)$ ,  $l = 1, 2$ . I call  $U(S_U, \lambda) := (U_1(S_U, \lambda), U_2(S_U, \lambda))$  the vector of upper bounds.

The idea of general lower and upper bounds is shown in Fig. 2. I assume here  $\rho = 0$ . Given the lower shell  $S_L$ , upper shell  $S_U$  and vector  $\lambda$  (determining the search direction in the objective space), the vector of lower bounds  $L(S_L, \lambda)$  and vector of upper

bounds  $U(S_U, \lambda)$  determine the region in the objective space, where efficient outcome  $f(x^{ef}(\lambda))$  is located. Amongst all elements of  $S_L$ , element  $\bar{x}$  provides the best lower bounds on values of  $f_l(x^{ef}(\lambda))$ ,  $l = 1, 2$ . Element  $x' \in S_U$  is the source of upper bound  $U_2(S_U, \lambda) = U_2(\{x'\}, \lambda) = f_2(x')$ , and element  $x'' \in S_U$ —the source of upper bound  $U_1(S_U, \lambda) = U_1(\{x''\}, \lambda) = f_1(x'')$ . These two elements fulfil conditions of Lemma 1. Elements of set  $S_U$  with their images in the objective space to the left of  $f(x')$  also are sources of valid upper bounds on  $f_2(x^{ef}(\lambda))$ ; however, these bounds are greater than the upper bound determined by  $f_2(x')$ . And similarly, elements of set  $S_U$  with their images in the objective space to the right of element  $f(x'')$  are also sources of valid upper bounds on  $f_1(x^{ef}(\lambda))$ ; however, these bounds are greater than the upper bound determined by  $f_1(x'')$ . Elements of  $S_U$  with their images inside the rectangle are not sources of valid upper bounds on  $f_l(x^{ef}(\lambda))$ ,  $l = 1, 2$ , as they do not fulfil conditions of Lemma 1 (are not properly located with regards to vector  $L(S_L, \lambda)$ ).

### 4.3 Derivation of upper shells in case of the CCMP

At first glance, it seems difficult to derive upper shells when the efficient set ( $N$ ) is unknown. However, in case of the CCMP in the form of problem (5), it can be done as follows.

When an MIQP solver provides no solution to problem (2) for some  $\lambda$  within a given time limit  $T$ , we are left with incumbent  $INC(\lambda)$  and the (mixed-integer programming) best bound  $B(\lambda)$  on values of the objective function of problem (2). By solving the following set of equations

$$\begin{aligned} B(\lambda) &= \lambda_1[(y_1^* - f_1^{B(\lambda)}) + \rho e^\top (y^* - f^{B(\lambda)})] \\ B(\lambda) &= \lambda_2[(y_2^* - f_2^{B(\lambda)}) + \rho e^\top (y^* - f^{B(\lambda)})] \end{aligned} \tag{11}$$

we can derive element  $f^{B(\lambda)} := (f_1^{B(\lambda)}, f_2^{B(\lambda)})$  belonging to the objective space.

**Lemma 2** *If for  $x^{B(\lambda)} \in \mathbb{R}^n$   $(f_1(x^{B(\lambda)}), f_2(x^{B(\lambda)})) = (f_1^{B(\lambda)}, f_2^{B(\lambda)})$  holds, then set  $\{x^{B(\lambda)}\}$  is the upper shell.*

**Proof** If efficient solution  $\tilde{x}$  to (5) dominated  $x^{B(\lambda)}$ , then  $\tilde{x}$  would provide a smaller value of the objective function in problem (2) ( $s$  in (3)) with  $\lambda$  than bound  $B(\lambda)$ , a contradiction to  $x^{B(\lambda)}$  defining the best (in this case, lower) bound on values of this function.  $\square$

If element  $x^{B(\lambda)}$  is unknown, then I say that it is a *virtual element* of upper shell  $\{x^{B(\lambda)}\}$ . By Lemma 1, one actually needs  $f_1(x^{B(\lambda)})$  and  $f_2(x^{B(\lambda)})$  to calculate upper bounds on components of an implicit efficient outcome.

Let us assume that  $S_L := \{INC(\lambda)\}$  and  $S_U := \{x^{B(\lambda)}\}$  are the lower shell and upper shell, respectively. The following remark arises from the derivation of formulas (7) and (8) [24].

**Remark 1** Element  $x^{B(\lambda)}$  is not a valid source for upper bounds  $U_1(S_U, \lambda)$ ,  $U_2(S_U, \lambda)$  because  $f_1(x^{B(\lambda)}) \geq L_1(S_L, \lambda)$  and  $f_2(x^{B(\lambda)}) \geq L_2(S_L, \lambda)$  hold (see Lemma 1).

According to Remark 1, element  $f^{B(\lambda)}$  is not appropriately located with respect to vector of lower bounds  $L(S_L, \lambda)$ . However, in order to find an upper shell whose element can be a source of a valid upper bound for some  $l$ ,  $l = 1, 2$ , problem (2) can be solved with other  $\tilde{\lambda}$ ,  $\tilde{\lambda}$  is close to  $\lambda$ . The following two cases are possible.

**Case 1.** When problem (2) with  $\tilde{\lambda}, \tilde{\lambda} \neq \lambda$ , is solved to optimality within time limit  $T^U$ , its solution  $x$  belongs to upper shell  $S_U := \{x\}$ .  $x$  might be a valid source of either  $U_1(S_U, \lambda)$  or  $U_2(S_U, \lambda)$ . (As  $x$  is feasible, it can even be an element of a lower shell, e.g.,  $\{x\}$ .)

**Case 2.** When problem (2) with  $\tilde{\lambda}, \tilde{\lambda} \neq \lambda$ , is not solved to optimality within time limit  $T^U$ , by solving set of equations (11) with the best bound  $B(\tilde{\lambda})$ , one can provide the outcome  $f^{B(\tilde{\lambda})}$  of element  $x^{B(\tilde{\lambda})}$ ,  $f^{B(\tilde{\lambda})} = f(x^{B(\tilde{\lambda})})$ . By Lemma 2, element  $x^{B(\tilde{\lambda})}$  belongs to upper shell  $S_U := \{x^{B(\tilde{\lambda})}\}$ .

However, in both cases, in order to find upper shells composed of elements suitable for providing valid upper bounds (see Lemma 1) on values of both objective functions, it may be necessary to select more than one  $\tilde{\lambda}$  vectors being close to  $\lambda$ .

The time of solving set of equations (11) is negligible, so I will assume that  $T^U$  is the time limit on optimization when deriving an upper shell which is a singleton.

The method of deriving upper shells presented here is based on fact that the best bound  $B(\lambda)$  is the result of solving a certain (implicit) relaxation of problem (2) by an MIQP solver. A method of deriving an upper shell by solving an explicit relaxation of the multiobjective optimization problem is proposed by Kaliszewski and Miroforidis in [21]. In this subsection, I presented the method with the implicit relaxation first shown in [23] and applied to bi-objective multidimensional 0–1 knapsack problems.

## 5 Numerical experiments

### 5.1 Experimental settings

In order to illustrate the presented idea, I applied the development of Sect. 4 to an instance of the CCMP with a real-world data set with 600 assets. There is a description of this data set in Appendix A.

I set:  $C = 5$ ,  $\rho = 0.001$ . I also assumed that  $y^* := (0, 0.00911)$  not deriving element  $\hat{y}$ . 0.00911 is the highest expected return among all assets—the portfolio composed only of the asset with the highest expected return belongs to the efficient set of the CCMP. 0 is my arbitrarily selected upper bound on negative values of the portfolio variance. As  $\hat{y}_l \leq y_l^*$ ,  $l = 1, 2$ , the use of  $y^*$  was legitimate.

I selected Gurobi version 8.1.1 for Windows 10 (x64) as the MIQP solver. The optimizer was installed on the Intel Core i7-7700HQ based laptop with 16 GB of RAM.

In all experiments below (after conducting some preliminary experiments), I used arbitrarily selected  $\lambda$  vectors. These vectors designate relatively risky portfolios.

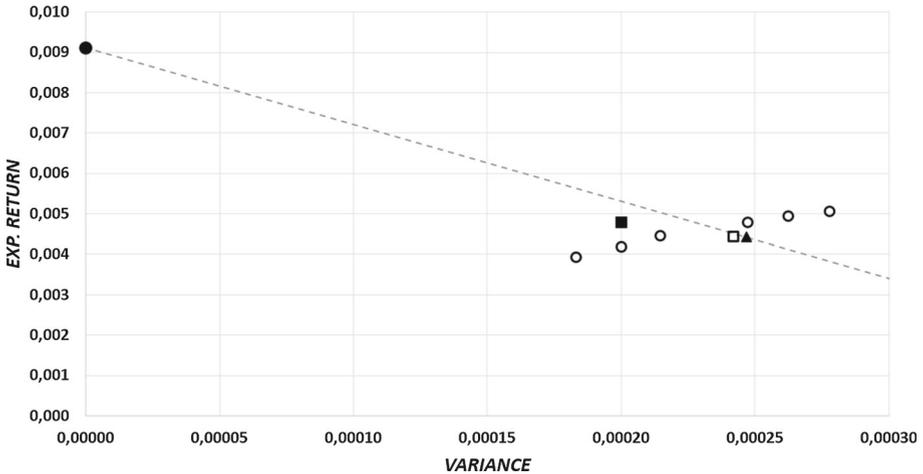
### 5.2 Experiment 1

In the first experiment, I assumed that  $\lambda := (0.95, 0.05)$ . I set time limits as follows.  $T := 300$  s,  $T^U := 300$  s. There were 6 selected  $\lambda^i$  vectors I used to derive upper shells  $S_U^i$  (see Table 1). The optimization time to derive lower shell  $S_L^A := \{INC(\lambda)\}$  as well as 6 upper shells  $S_U^1, \dots, S_U^6$  was  $300 + 6 \times 300 = 2100$  s. As all elements in set  $S_U^A := \bigcup_{i=1}^6 S_U^i$  are non-dominated in this set (see Fig. 3),  $S_U^A$  is an upper shell as well. By calculating lower and upper bounds on values of objective functions of  $x^{ef}(\lambda)$ , I obtained:

$$\begin{aligned} -0.0002467 &\leq f_1(x^{ef}(\lambda)) \leq -0.0002001, \\ 0.0044222 &\leq f_2(x^{ef}(\lambda)) \leq 0.0047897. \end{aligned}$$

**Table 1** The set of  $\lambda$  vectors I used to derive upper shells  $S_U^i$  in Experiment 1

$i$	$\lambda_1^i$	$\lambda_2^i$
1	0.935	0.065
2	0.940	0.060
3	0.945	0.550
4	0.955	0.045
5	0.960	0.040
6	0.965	0.035



**Fig. 3** Experiment 1:  $\blacktriangle$ —vector of lower bounds,  $\blacksquare$ —vector of upper bounds,  $\square, \circ$ —images, respectively, of lower shell  $S_L^A$  and upper shell  $S_U^A$  in the “expected return–variance” space,  $\bullet$ — $y^*$ , dashed line—search direction determined by  $\lambda$  in the “expected return–variance” space

### 5.3 Experiment 2

In the second experiment, I assumed that  $\lambda := (0.98, 0.02)$ . This vector designates a less risky portfolio than one designated by vector  $\lambda$  used in the previous experiment. I set time limits as follows.  $T := 1200$  s,  $T^U := 600$  s. There were 5 selected  $\lambda^i$  vectors I used to derive upper shells (see Table 2). The optimization time to derive lower shell  $S_L^B := \{INC(\lambda)\}$  as well as 5 upper shells  $S_U^1, \dots, S_U^5$  was  $1200 + 5 \times 600 = 4200$  s. As all elements in set  $S_U^B := \bigcup_{i=1}^5 S_U^i$  are non-dominated in this set (see Fig. 4),  $S_U^B$  is an upper shell as well. By calculating lower and upper bounds on values of objective functions of  $x^{ef}(\lambda)$ , I obtained:

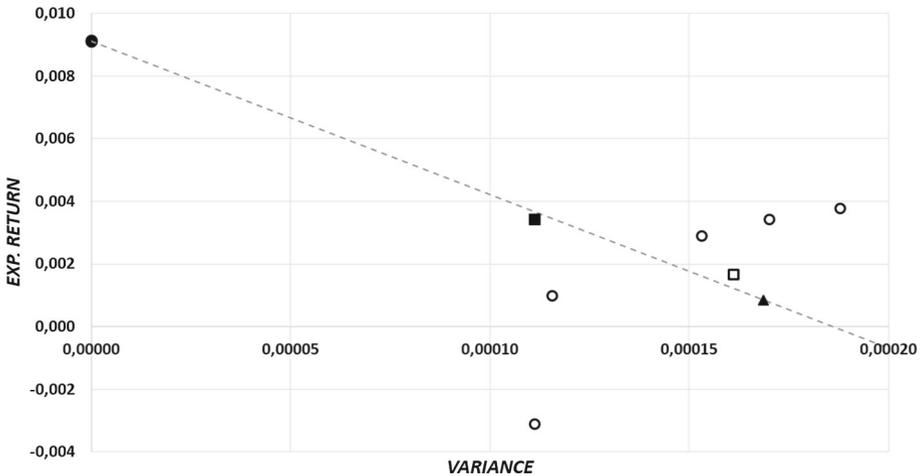
$$\begin{aligned}
 -0.0001686 &\leq f_1(x^{ef}(\lambda)) \leq -0.0001112, \\
 0.0008468 &\leq f_2(x^{ef}(\lambda)) \leq 0.0034285.
 \end{aligned}$$

### 5.4 Experiment 3

In the third experiment, I assumed that  $\lambda := (0.97, 0.03)$ . I set time limits as follows.  $T := 1200$  s,  $T^U := 600$  s. There were 7 selected  $\lambda^i$  vectors I used to derive upper shells (see Table 3). The optimization time to derive lower shell  $S_L^C := \{INC(\lambda)\}$  as well as 7 upper

**Table 2** The set of  $\lambda$  vectors I used to derive upper shells  $S_U^i$  in Experiment 2

$i$	$\lambda_1^i$	$\lambda_2^i$
1	0.965	0.035
2	0.970	0.030
3	0.975	0.025
4	0.985	0.015
5	0.990	0.010



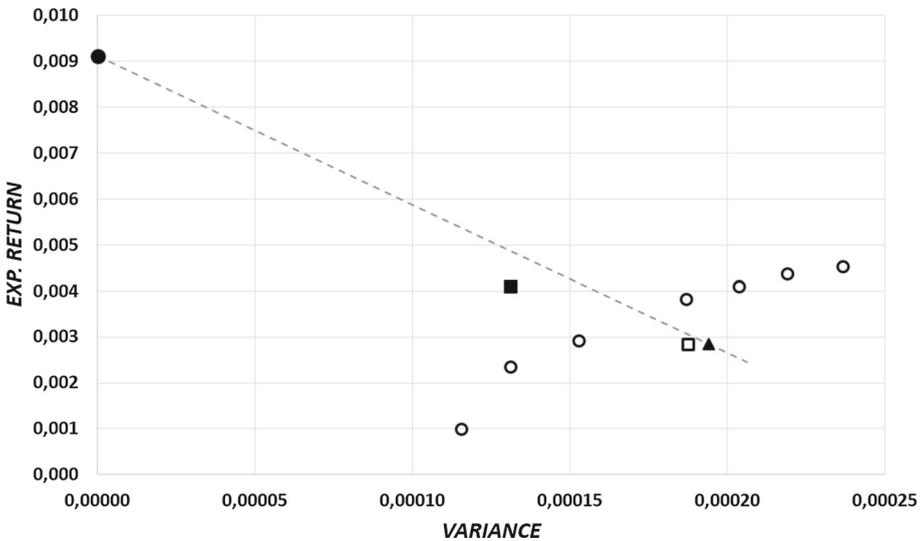
**Fig. 4** Experiment 2:  $\blacktriangle$ —vector of lower bounds,  $\blacksquare$ —vector of upper bounds,  $\square, \circ$ —images, respectively, of lower shell  $S_L^B$  and upper shell  $S_U^B$  in the “expected return–variance” space,  $\bullet$ — $y^*$ , dashed line—search direction determined by  $\lambda$  in the “expected return–variance” space

**Table 3** The set of  $\lambda$  vectors I used to derive upper shells  $S_U^i$  in Experiment 3

$i$	$\lambda_1^i$	$\lambda_2^i$
1	0.950	0.050
2	0.955	0.045
3	0.960	0.040
4	0.965	0.035
5	0.975	0.025
6	0.980	0.020
7	0.985	0.015

shells  $S_U^1, \dots, S_U^7$  was  $1200 + 7 \times 600 = 5400$  s. As all elements in set  $S_U^C := \bigcup_{i=1}^7 S_U^i$  are non-dominated in this set (see Fig. 5),  $S_U^C$  is an upper shell as well. By calculating lower and upper bounds on values of objective functions of  $x^{ef}(\lambda)$ , I obtained:

$$\begin{aligned}
 -0.0001939 &\leq f_1(x^{ef}(\lambda)) \leq -0.0001313, \\
 0.0028390 &\leq f_2(x^{ef}(\lambda)) \leq 0.0040999.
 \end{aligned}$$



**Fig. 5** Experiment 3:  $\blacktriangle$ —vector of lower bounds,  $\blacksquare$ —vector of upper bounds,  $\square, \circ$ —images, respectively, of lower shell  $S_L^C$  and upper shell  $S_U^C$  in the “expected return–variance” space,  $\bullet$ — $y^*$ , dashed line—search direction determined by  $\lambda$  in the “expected return–variance” space

### 5.5 Discussion

In Figs. 3, 4 and 5, the triangle represents the vector of lower bounds on components of implicit efficient outcome  $f(x^{ef}(\lambda))$  in the “expected return–variance” space. The filled rectangle represents the vector of upper bounds.

For vector  $\lambda$  used in Experiment 1, the (a priori unknown) optimization time for finding the exact efficient solution  $x^{ef}(\lambda)$  was  $\sim 733$  s (I checked this a posteriori). However, the optimization time to derive lower and upper shell was 2100 s. So, in this experiment, I could get the exact solution, instead of only the bounds, in less than 2100 s.

However, for vector  $\lambda$  used in Experiment 2, I tried to find the efficient solution  $x^{ef}(\lambda)$ . Gurobi reached the relative mixed-integer programming gap ( $MIP_{gap}$ ) equal to 5.78% within 7476 s. The optimization time to derive the lower shell and upper shells (thus, lower and upper bounds) was 4200 s. Within this time, I could provide the DM with the bounds on components of  $f(x^{ef}(\lambda))$ . Please note that even though I know the  $MIP_{gap} > 0$  (related to the single-objective optimization problem (2)), I cannot infer the gaps related to components of  $f(x^{ef}(\lambda))$ .

For vector  $\lambda$  used in Experiment 3, I tried to find efficient solution  $x^{ef}(\lambda)$  setting the time limit on optimization to 10,000 s. Gurobi reached the  $MIP_{gap}$  equal to 1.1828% within this time limit.

For each  $\lambda$  vector used in Experiments 1–3, portfolio  $INC(\lambda)$ , as well as its outcome can be shown the DM. Furthermore, lower and upper bounds on components of  $f(x^{ef}(\lambda))$  can be shown as well. The DM evaluates a portfolio  $x$  on the basis of values of components of  $f(x)$ . As  $MIP_{gap}$  value is purely technical information related to solving the single-objective optimization problem (2), presenting it to the DM is unreasonable.

The proposed approach requires a systematic scheme of selecting neighbouring  $\lambda^i$  vectors to derive elements of upper shells which can provide upper bounds. In Experiments 1–3, those

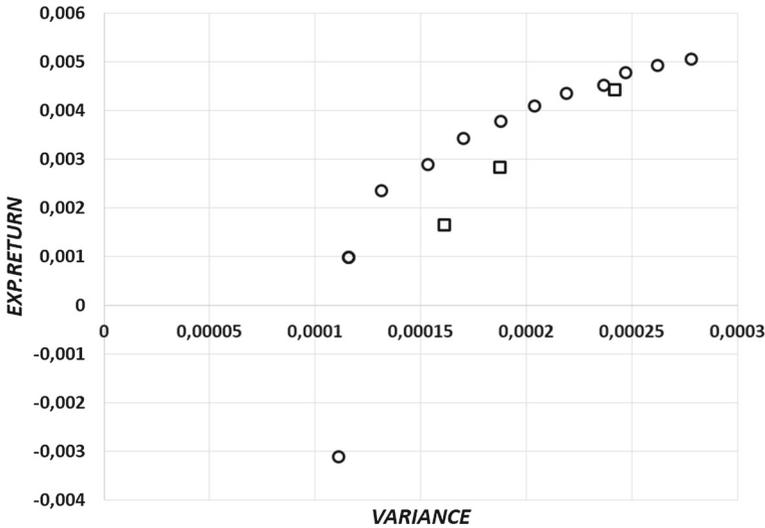


Fig. 6 □, ○—images of  $S_L$  and  $S_U$ , respectively, used in Sect. 6 in the “expected return–variance” space

vectors were arbitrarily selected. Given  $\lambda$ , tighter bounds on components of  $f(x^{ef}(\lambda))$  might be provided to the DM, e.g., by deriving extra upper shells or by deriving a new incumbent within another time limit—the extra computing effort is required then. There is a compromise between the tightness of lower and upper bounds and the time budget for optimization.

The lower and upper shells derived in Experiments 1–3 can be reused to navigate over the efficient frontier. This is demonstrated in the following section.

### 6 Navigating over the efficient frontier using pre-calculated lower and upper shells

By summing up lower shells  $S_L^A, S_L^B$  and  $S_L^C$  obtained from Experiments 1–3 and by removing dominated elements in the resulting set, a new lower shell  $S_L$  can be obtained. In a similar way, by summing up upper shells  $S_U^A, S_U^B$  and  $S_U^C$ , as well as by removing dominating elements in the resulting set (since dominating elements are a source of either worse or equal upper bounds—see Lemma 1), a new upper shell  $S_U$  can be obtained. Let us recall that the total optimization time needed to determine  $S_L$  and  $S_U$  was 3 h and 15 min. I can treat this as a total computational budget on optimization in the decision making process. Images of these new lower and upper shells in the “expected return–variance” space are shown in Fig. 6.

Taking this into account, one can use  $S_L$  and  $S_U$  to calculate lower and upper bounds on components of  $f(x^{ef}(\lambda))$  for other  $\lambda$  vectors. This means that the DM can navigate over the efficient frontier using pre-calculated lower shell and upper shell. When the DM is satisfied with the tightness of the bounds, there is no need to solve any new optimization problem.

In order to present the above idea, I selected three  $\lambda$  vectors:  $\lambda^a := (0.96, 0.04)$ ,  $\lambda^b := (0.975, 0.025)$ ,  $\lambda^c := (0.94, 0.06)$  which are different than ones used in Experiments 1–3.

Lower and upper bounds on values of components of  $x^{ef}(\lambda^\alpha)$ ,  $\alpha \in \{a, b, c\}$ , are as follows.

$$\begin{aligned}
 -0.0002467 &\leq f_1(x^{ef}(\lambda^a)) \leq -0.0001532, \\
 0.0031886 &\leq f_2(x^{ef}(\lambda^a)) \leq 0.0047897,
 \end{aligned}$$

$$\begin{aligned}
-0.0001911 &\leq f_1(x^{\text{ef}}(\lambda^b)) \leq -0.0001157, \\
0.0016580 &\leq f_2(x^{\text{ef}}(\lambda^b)) \leq 0.0040999, \\
-0.0002989 &\leq f_1(x^{\text{ef}}(\lambda^c)) \leq -0.0002190, \\
0.0044280 &\leq f_2(x^{\text{ef}}(\lambda^c)) \leq y_2^*.
\end{aligned}$$

There is no element of  $S_U$  being a source of a valid upper bound on values of  $f_2(x^{\text{ef}}(\lambda^c))$ . Hence, the only available upper bound on values of  $f_2(x^{\text{ef}}(\lambda^c))$  is equal to  $y_2^*$ .

Thus, when developing an interactive decision-support system for navigating over the efficient frontier of large-scale instances of the CCMP with the help of the presented method, it seems reasonable to maintain (global) sets  $S_L$  and  $S_U$ . E.g., by storing them in a database to reuse.

Also, please note that images of  $S_L$  and  $S_U$  in the “expected return–variance” space form a finite two-sided approximation of the efficient frontier. This approximation depends on  $\lambda$  vectors chosen by the DM during the decision making process. As for any given  $\lambda$  one can calculate lower and upper bounds on components of  $f(x^{\text{ef}}(\lambda))$  with the use of  $S_L$  and  $S_U$ , I can say that (parametrized) lower and upper bounds form a *continuous two-sided approximation* of the efficient frontier induced on sets  $S_L$  and  $S_U$ .

## 7 Final remarks

The derivation of some efficient solutions (portfolios) to the CCMP with a large number of assets might be difficult for commercial MIQP solvers, when there is a time limit on optimization. It is clear that this time limit would depend on number of assets in the CCMP as well on the right-hand side of the cardinality constraint. The time limit value can be set by the DM before the decision making process begins, but it also could be changed by him/her on demand during the decision making process.

When the time limit is reached by a solver, lower and upper bounds on components of an implicit efficient outcome can be calculated by making use of the data provided by it. When an interactive MCDM method based on Chebyshev scalarization is used to derive efficient solutions to the CCMP, the DM can use the bounds to evaluate implicit efficient portfolios when components of their images in the objective space are unknown due to the fixed time limit on optimization. It can be said that preference-driven two-sided efficient frontier approximations are sources of the bounds. The lower and upper bounds are calculated with the help of these approximations.

Any subset of efficient solutions to the CCMP can be used to calculate lower and upper bounds. To populate this subset (which is both a lower and an upper shell), sophisticated methods of finding efficient solutions, e.g., methods described in [2,4,7,9,13,14], can be applied to large-scale instances of the CCMP. However, even for such sophisticated methods the derivation of a numerous subset of the efficient set (being its fine representation) may be time-consuming for large-scale instances. In this case (in particular, when there is a time limit on exact optimization), a smaller representation of the efficient set can be derived. If  $y^*$  is given, then each element of the image of this representation in the objective space corresponds to a search direction (direction of concessions) in the objective space [18]. If, without further optimization, the DM wishes to evaluate an efficient outcome corresponding to another direction of concessions (expressed by one of these ways: a vector of concessions, a base point, or a weight vector), bounds on components of this implicit efficient outcome can be calculated based on the representation and provided to him/her. In this mixed scheme,

the DM evaluates either the explicit efficient outcomes using their components or the implicit ones using lower and upper bounds on the values of their components.

From a practical perspective, the advantage of the presented method is that the DM can use off-the-shelf MIQP solvers. The strength of the presented approach is that it uses purely technical (from the point of view of the DM) information provided by a commercial MIQP solver (or any solver which is able to provide the mixed-integer best bound when solving MIQP problems) to build upper shells and provide upper bounds. Its weakness is that it is general (it uses theory of lower and upper bounds which is valid for any kind of multiobjective optimization problem) and does not fully exploit the properties of the CCMP to populate these sets or provide tighter upper bounds.

The proposed approach based on MIQP solvers is valid for any extension of the CCMP (when other practical constraints, e.g., upper bounds on assets' shares, the pre-assignment constraint [3,8], the round lot constraint [16,28], are introduced) if the extended model belongs to the MIQP class. Since the general theory of lower and upper bounds is valid for more than two objective functions [22], after some modifications the approach can also be used for the CCMP with, e.g., three objective functions.

In my future research, I plan to develop a rule-controlled method of populating lower and upper shells. In order to populate lower shells, I also plan to make use of metaheuristics.

## Declarations

**Conflict of interest** The author declares that he has no conflict of interest.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## Appendix A

### The data set description

The data set called JKMP 600 was created from openly available data. Weekly stock quotes from the New York Stock Exchange were collected for 600 companies, for the period from July 2001 until June 2018, with no missing quotes. The data set in the format analogous to the Beasley problems from OR Library is available at: [http://www.ibspan.waw.pl/~kaliszew/JKMP\\_portfolio\\_problems/JKMP\\_600/JKMP\\_600.txt](http://www.ibspan.waw.pl/~kaliszew/JKMP_portfolio_problems/JKMP_600/JKMP_600.txt).

### Lower and upper shells used in Sect. 6

Images of elements of lower shell  $S_L$  and upper shell  $S_U$  in the “expected return–variance” space are shown in Tables 4 and 5, respectively. In Fig. 6, images of elements number 5 and 12 are represented by overlapping circles.

**Table 4** Images of elements of lower shell  $S_L$  in the “expected return–variance” space

Nos.	Variance	EXP.RETURN
1	0.0002420	0.0044282
2	0.0001877	0.0028418
3	0.0001612	0.0016581

**Table 5** Images of elements of upper shell  $S_U$  in the “expected return–variance” space

Nos.	Variance	EXP.RETURN
1	0.0002367	0.0045265
2	0.0002190	0.0043620
3	0.0002038	0.0040999
4	0.0001313	0.0023457
5	0.0001157000	0.0009790821
6	0.0002779	0.0050571
7	0.0002624	0.0049344
8	0.0002471	0.0047897
9	0.0001879	0.0037821
10	0.0001700	0.0034285
11	0.0001532	0.0028918
12	0.0001156996	0.0009791133
13	0.0001112	−0.0031058

## References

- Anagnostopoulos, K.P., Mamanis, G.: A portfolio optimization model with three objectives and discrete variables. *Comput. Oper. Res.* **37**, 1285–1297 (2010)
- Bertsimas, D., Cory-Wright, R.: A scalable algorithm for sparse portfolio selection (2019). [arXiv:1811.00138v3](https://arxiv.org/abs/1811.00138v3) [math.OC]
- Chang, T.-J., Meade, N., Beasley, J.E., Sharaiha, Y.M.: Heuristics for cardinality constrained portfolio optimisation. *Comput. Oper. Res.* **27**, 1271–1302 (2000)
- Cesarone, F., Scozzari, A., Tardella, F.: A new method for mean–variance portfolio optimization with cardinality constraints. *Ann. Oper. Res.* **205**, 213–234 (2013)
- Coello Coello, C.A., Van Veldhuizen, D.A., Lamont, G.B.: *Evolutionary Algorithms for Solving Multi-objective Problems*. Kluwer, New York (2002)
- Coello Coello, C.A., Lamont, G.B. (eds.): *Applications of Multi-objective Evolutionary Algorithms*. World Scientific Printers, Singapore (2004)
- Cui, X., Zheng, X., Zhu, S., Sun, X.: Convex relaxations and MIQCQP reformulations for a class of cardinality constrained portfolio selection problems. *J. Glob. Optim.* **56**(4), 1409–1423 (2013)
- Di Gaspero, L., Di Tollo, G., Roli, A., Schaerf, A.: Hybrid metaheuristics for constrained portfolio selection problems. *Quant. Finance* **11**(10), 1473–1487 (2011)
- Dong, Z., Xu, F., Dai, Y.: Fast algorithms for sparse portfolio selection considering industries and investment styles. *J. Glob. Optim.* (2020). <https://doi.org/10.1007/s10898-020-00911-1>
- Ehrgott, M.: *Multicriteria Optimization*. Springer, Berlin (2005)
- Ehrgott, M., Gandibleux, X.: Bound sets for biobjective combinatorial optimization problems. *Comput. Oper. Res.* **34**, 2674–2694 (2007)
- Ehrgott, M., Klamroth, K., Schwehm, Ch.: An MCDM approach to portfolio optimization. *Eur. J. Oper. Res.* **155**(3), 752–770 (2004)
- Frangioni, A., Gentile, C.: Perspective cuts for a class of convex 0–1 mixed integer programs. *Math. Program.* **106**, 225–236 (2006)
- Gao, J., Li, D.: Optimal cardinality constrained portfolio selection. *Oper. Res.* **61**, 745–761 (2013)

15. Geoffrion, A.M.: Proper efficiency and the theory of vector maximization. *J. Math. Anal. Appl.* **22**(3), 618–630 (1968)
16. Jobst, N., Horniman, M., Lucas, C., Mitra, G.: Computational aspects of alternative portfolio selection models in the presence of discrete asset choice constraints. *Quant. Finance* **1**(5), 489–501 (2001)
17. Juszczak, P., Kaliszewski, I., Miroforidis, J.: Trade-off guided search for approximate pareto optimal portfolios. *Multiple Criteria Decis. Mak.* **12**, 49–59 (2017)
18. Kaliszewski, I.: *Soft Computing for Complex Multiple Criteria Decision Making*. Springer, New York (2006)
19. Kaliszewski, I.: Out of the mist—towards decision-maker-friendly multiple criteria decision making support. *Eur. J. Oper. Res.* **158**(2), 293–307 (2004)
20. Kaliszewski, I., Miroforidis, J.: Two-sided pareto front approximations. *J. Optim. Theory Appl.* **162**, 845–855 (2014)
21. Kaliszewski, I., Miroforidis, J.: On upper approximations of pareto fronts. *J. Glob. Optim.* **72**(3), 475–490 (2018)
22. Kaliszewski, I., Miroforidis, J.: Lower and upper bounds for the general multiobjective optimization problem. *AIP Conf. Proc.* **2070**, 020038 (2019)
23. Kaliszewski, I., Miroforidis, J.: Cooperative multiobjective optimization with bounds on objective functions. *Systems Research Institute Report RB/15/2019* (2019)
24. Kaliszewski, I., Miroforidis, J., Podkopayev, D.: Interactive multiple criteria decision making based on preference driven evolutionary multiobjective optimization with controllable accuracy—the case of  $\rho$ -efficiency. *Systems Research Institute Report RB/1/2011* (2011)
25. Kaliszewski, I., Miroforidis, J., Podkopayev, D.: Interactive multiple criteria decision making based on preference driven evolutionary multiobjective optimization with controllable accuracy. *Eur. J. Oper. Res.* **216**, 188–199 (2012)
26. Kaliszewski, I., Kiczkowiak, T., Miroforidis, J.: Mechanical design, multiple criteria decision making and pareto optimality gap. *Eng. Comput.* **33**(3), 876–895 (2016)
27. Kaliszewski, I., Miroforidis, J., Podkopayev, D.: *Multiple Criteria Decision Making by Multiobjective Optimization—A Toolbox*. Springer, Berlin (2016)
28. Kellerer, H., Mansini, R., Speranza, M.: Selecting portfolios with fixed costs and minimum transaction lots. *Ann. Oper. Res.* **99**, 287–304 (2000)
29. Kolm, P.N., Tütüncü, R., Fabozzi, F.J.: 60 Years of portfolio optimization: practical challenges and current trends. *Eur. J. Oper. Res.* **234**, 356–371 (2014)
30. Markowitz, H.M.: Portfolio selection. *J. Finance* **7**, 77–91 (1952)
31. Michalewicz, Z.: *Genetic Algorithms + Data Structures = Evolution Programs*. Springer, Berlin (1996)
32. Miettinen, K.M.: *Nonlinear Multiobjective Optimization*. Kluwer Academic Publishers, New York (1999)
33. Miroforidis, J.: Multiple criteria analysis of the airport terminal effectiveness by multi-objective optimization and simulation. *Int. J. Inf. Manag. Sci.* **27**(2), 103–115 (2016)
34. Mitra, G., Kyriakis, T., Lucas, C., Pirbhai, M.: A review of portfolio planning: Models and systems. In: Satchell, S., Scowcroft, A. (eds.) *Advances in Portfolio Construction and Implementation*, pp. 1–39. Butterworth and Heinemann, Oxford (2003)
35. Rennet, G., van Dam, E.R., den Hertog, D.: Enhancement of sandwich algorithms for approximating higher-dimensional convex pareto sets. *INFORMS J. Comput.* **23**(4), 493–517 (2011)
36. Ruzika, S., Wiecek, M.M.: Approximation methods in multiobjective programming. *J. Optim. Theory Appl.* **126**, 473–501 (2005)
37. Solanki, R., Appino, P., Cohon, J.: Approximating the noninferior set in multiobjective linear programming problems. *Eur. J. Oper. Res.* **68**, 356–73 (1993)
38. Steuer, R.E.: Multiobjective programming. In: Gass, S.I., Harris, C.M. (eds.) *Encyclopedia of Operations Research and Management Science*. Springer, Berlin (2001). [https://doi.org/10.1007/1-4020-0611-X\\_652](https://doi.org/10.1007/1-4020-0611-X_652)
39. Steuer, R.E., Choo, E.-U.: An interactive weighted Tchebycheff procedure for multiple objective programming. *Math. Program.* **26**, 326–344 (1983)
40. Wierzbicki, A.P.: Reference point approaches. In: Gal, T., Stewart, T., Hanne, T. (eds.) *Multicriteria Decision Making—Advances in MCDM: Models, Algorithms, Theory and Applications*. Kluwer Academic Publishers, New York (1999)
41. Woodside-Oriakhi, M., Lucas, C., Beasley, J.E.: Heuristic algorithms for the cardinality constrained efficient frontier. *Eur. J. Oper. Res.* **213**, 538–550 (2011)