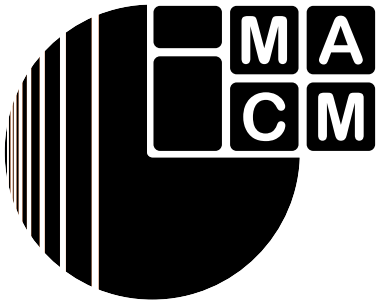



Bergische Universität Wuppertal

Fakultät für Mathematik und Naturwissenschaften

Institute of Mathematical Modelling, Analysis and Computational Mathematics (IMACM)

Preprint BUW-IMACM 19/35

Michael Stiglmayr, José Rui Figueira, Kathrin Klamroth, Luís Paquete und Britta Schulze

# Decision Space Robustness for Multi-Objective Integer Linear Programming

December 3, 2019

http://www.math.uni-wuppertal.de

# Decision Space Robustness for Multi-Objective Integer Linear Programming

Michael Stiglmayr[a], José Rui Figueira[b], Kathrin Klamroth[a], Luís Paquete[c], Britta Schulze[a]

[a] *School of Mathematics and Natural Sciences, University of Wuppertal, Germany*
[b] *CEG-IST, Instituto Superior Técnico, Universidade de Lisboa, Portugal*
[c] *CISUC, Department of Informatics Engineering, University of Coimbra, Portugal*

**Abstract**

In this article we introduce robustness measures in the context of multi-objective integer linear programming problems. The proposed measures are in line with the concept of decision space robustness, which considers the uncertainty with respect to the implementation of a specific solution. An efficient solution is considered to be decision space robust if many solutions in its neighborhood are efficient as well. This rather new area of research differs from robustness concepts dealing with imperfect knowledge (arbitrariness, uncertainty, imprecision, and ill-determination) of data parameters. Our approach implies a two-phase procedure, where in the first phase the set of all efficient solutions is computed, and in the second phase the neighborhood of each one of the solutions is determined. The indicators we propose are based on the knowledge of these neighborhoods. They are mainly of two types, cardinality indicators and quality indicators. We discuss consistency properties for the indicators and present some numerical evaluations for specific problem classes.

*Keywords:* multi-objective integer linear programming, decision space robustness, connectedness of efficient solutions, representation, decision analysis.

## 1. Introduction

In industrial or economical applications, a computed optimal or efficient solution may be subject to small changes or deviations during its implementation. These deviations may be due to technical, political, and/or strategic reasons, which are generally not known beforehand and which are thus not included in the optimization model. As a consequence, a solution obtained with an optimization method may, in practice, not be implementable, or may be affected with severe drawbacks.
In this paper, we focus on multi-objective integer programming problems that have a

*Email addresses:* stiglmayr@math.uni-wuppertal.de (Michael Stiglmayr), figueira@tecnico.ulisboa.pt (José Rui Figueira), klamroth@math.uni-wuppertal.de (Kathrin Klamroth), paquete@dei.uc.pt (Luís Paquete), schulze@math.uni-wuppertal.de (Britta Schulze)

discrete solution set. Implementation uncertainties then relate to specific variables or, for example, items that were selected for a knapsack solution, but that become unavailable when the solution is to be implemented. In this situation, we can try to anticipate the potential failure of (parts of) solutions and incorporate appropriate robustness measures into the optimization process.

Dealing with uncertainty or, more general, with the imperfect knowledge of data in single and multi-objective optimization is not new. *Imperfect knowledge* about data and models in general is mainly due to four aspects: the *arbitrariness* when, for example, we opted for a specific objective function or constraint among several others that would be also adequate; the *uncertainty* since we are using methods, tools, and approaches for modeling and anticipating an unknown future; the *imprecision* related with the tools we are using to measure objects; and, the *ill-determination* related with the fact that we are modeling some aspects that are possibly not well determined and defined (for more details see, for example, Roy et al., 2014).

Dealing with the imperfect knowledge about the whole model in single and multi-objective optimization is rather a Herculean task. Researchers have been concentrating their attention mainly on the imperfect knowledge of some type of data, i.e., the model parameters. For single objective optimization we can mention three fundamental references: Ben-Tal et al. (2009), Kouvelis and Yu (1997), and Birge and Louveaux (1997).

In multi-objective optimization there is a long tradition for dealing with the imperfect knowledge of data and several approaches have been proposed in the literature. For a survey about different concepts of robustness in multi-objective optimization see, for example, Ide and Schöbel (2016). The following is a brief and non-exhaustive summary with main approaches found in the literature: stochastic programming (Inuiguchi et al., 2016; Słowiński and Teghem, 1990); fuzzy/possibilistic programming (Adeyefa and Luhandjula, 2011; Inuiguchi et al., 2016; Słowiński and Teghem, 1990); fuzzy-stochastic programming (Inuiguchi et al., 2016; Sakawa et al., 2011); interval programming (Oliveira and Henggeler-Antunes, 2007); parametric programming (Dellnitz and Witting, 2009; Witting et al., 2013); minimax like programming (Aissi et al., 2009; Ehrgott et al., 2014); set valued optimization (Ide and Köbis, 2014); and, Monte Carlo simulation (Mavrotas et al., 2015). The different approaches have in common to find solutions which are more or less robust with respect to changes in some parameters which occur in the constraints and/or objectives of an optimization problem. We refer to this as *parameter robustness* in the following. In this context, a solution that is more or less immune to parameter changes is a robust solution.

If, however, the uncertainty is an intrinsic property of the variables, we talk about *decision robustness* or *variable robustness* (see, for example, Beyer and Sendhoff, 2007). Very recently, Eichfelder et al. (2015, 2017, 2019) introduced and made some studies about the concept of uncertainty with respect to the implementation of decision variable values in continuous multi-objective optimization.

Despite the vast literature in the field there is a missing link: Multi-objective integer linear programs (MOILP) that have a combinatorial nature are rarely addressed, and robustness measures with respect to implementation uncertainties are unavailable. Nev-

ertheless, the number of practical situations is wide, and the following examples can give a flavor about our positions with respect to the development of new robustness measures.
**Bi-objective $\{0,1\}-$knapsack problem:** Consider a set of technology projects $P = \{P_1, \ldots, P_j, \ldots, P_n\}$. These projects were assessed according to two different criteria: the net present value and the dual use potential with respect to already implemented technology projects. The two criteria are to be maximized: we want to maximize the overall net present value and to maximize the overall dual use potential. Each project was also assessed in terms of its investment cost and there is a budgetary constraint, which avoids the selection of the whole set of projects. Assume now that a decision maker selected the best compromise set of projects to be implemented, say $S$. At the right moment of the implementation, suppose that one of the projects, say project $P_j \in S$, is interdicted, i.e., it becomes unavailable. This may happen because there was an uncertainty with respect to the future, i.e., in between the moment the decision maker selected the subset of projects and the moment of their implementation. One major question may arise: What is the most robust selection of projects $S$ (if any) so that even if $S$ can not be implemented as planned, there is a viable alternative set of projects $S'$ in the neighborhood of $S$ that may be used instead?
**Multi-objective shortest path problem:** Consider a multi-objective routing problem form source $s$ to destination $t$ minimizing, e.g., travel time, $CO_2$-emission and the total cost of toll fees. An important prerequisite for a practicable route is the possibility to avoid traffic jams and road closures by a local adaptation of the route. Since the information on those obstructions of traffic may get available during driving time, one searches for a local detour rather than a second best $s$-$t$ path. Depending on the chosen route, the number and quality of route variants with local detours may vary significantly. Similar situations, where a part of a chosen solution fails, can be considered, e.g., in the context of job assignments (multi-objective linear or bottleneck assignment problems, failure of one assignment: one worker cannot do a certain job) or in the cable trench problem which is a variant of the multi-objective spanning tree problem (failure of one edge in the graph). All these applications in common is that one is not willing to completely discard a once chosen solution only because one single component of it is not available/blocked/fails. The repair of a solution can be considered as replacing the chosen solution by a neighboring one.
The robustness measures that we discuss in this paper can be best understood with the help of the following questions.

1. Are there enough *feasible neighbors* around the selected efficient solution that may be used to replace it in case an impossibility occurs at the moment of its implementation?

2. Are there enough *efficient neighbors* around the selected efficient solution that may be used to replace it in case an impossibility occurs at the moment of its implementation?

3. Are there enough *high quality neighbors* around the selected efficient solution that may be used to replace it in case an impossibility occurs at the moment of its

implementation? This leads to a second question: How good should be the quality of the neighbors? Among several possibilities we are interested in two of them: The worst case, and the average case.

With respect to these questions, in our study we consider two specific cases of neighborhood: $\mathcal{N}(x)$ is the set of all immediate neighbors of a feasible solution $x$, and $\mathcal{N}^j(x)$ is the set of all immediate neighbors of a feasible solution $x$ in which a specific component, say $x_j$, is interdicted (i.e., it becomes unavailable and/or cannot be implemented). The purpose of this paper is to answer the questions by presenting ways of measuring the robustness in multi-objective discrete optimization problems. In this attempt, the paper covers a variety of MOILP problems with a combinatorial structure with applications in a vast range of areas.
This paper is organized as follows. Section 2 summarizes the fundamental concepts, their definitions, and the adopted notation. Section 3 presents several robustness indicators for MOILP problems, some numerical results, theoretical properties of the indicators and a fundamental distinction between robustness and representation. In Section 4 the proposed robustness concepts are applied in the context of representation problems and decision making using Electre Tri-C. Finally, some concluding remarks and avenues for future research are provided.

## 2. Concepts, Definitions, and Notation

This section is devoted to the presentation of the problem, the main concepts of dominance and efficiency, as well as the concepts of neighborhood and adjacency. It ends with the introduction of an illustrative example.

### *2.1. Problem Formulation*

A general multi-objective optimization problem can be stated as follows.

$$\max_{x \in X} z(x) = \big(z_1(x), \ldots, z_p(x)\big)^\top \tag{1}$$

where $z$ is a vector-valued function such that $z : X \to \mathbb{R}^p$ with $X \subseteq \mathbb{R}^n$. In this formulation $\mathbb{R}^n$ is called the *decision space*, while its subset $X$ represents the *feasible region*. Each function $z_k$, for $k = 1, \ldots, p$, is a real-valued function such that $z_k : X \to \mathbb{R}$, and $\mathbb{R}^p$ is referred to as the *objective space*.
In what follows we will restrict ourselves to discrete multi-objective optimization problems where the feasible region $X$ can be defined as $X = P \cap \mathbb{Z}^n$, with $P = \{x \in \mathbb{R}^n \colon A\,x \leq b,\ x \geqq 0\}$ being a polyhedron, i.e., the intersection of finitely many halfspaces.
The functions $z_k$, for $k = 1, \ldots, p$, are linear. They can be defined as follows: $z_k = c^{k\top} x$, where $c^k = (c^k_1, \ldots, c^k_n)^\top \in \mathbb{R}^n$ is the vector of coefficients of the objective function $z_k$ for $k = 1, \ldots, p$. Within such a framework we are placed in the context of multi-objective integer linear programming. This problem can be presented in a more compact way, as follows.

$$\max_{x \in X} z(x) = C\,x, \tag{2}$$

where $C$ is a $p \times n$ matrix in which each rows corresponds to a vector $c^{k\top}$, for $k = 1, \ldots, p$. The *outcome vector* of a feasible solution $x \in X$ is the image of $x$ under the vector-valued function, i.e., $z(x)$. The set $Z = \{z(x) \colon x \in X\} = z(X) \subseteq \mathbb{R}^p$ is the *feasible set* in the *objective or outcome space* $\mathbb{R}^p$.
It is well-known that whenever $p \geqslant 2$ there is no canonical order in the objective space. The following *vector componentwise order* can thus be used.

$$\begin{aligned}
z' \geqq z'' &\iff z'_k \geqslant z''_k, && k = 1, \ldots, p, \\
z' \geq z'' &\iff z'_k \geqslant z''_k, && k = 1, \ldots, p, \text{ with } z' \neq z'', \\
z' > z'' &\iff z'_k > z''_k, && k = 1, \ldots, p.
\end{aligned}$$

In multi-objective optimization dominance relations replace the canonical ordering structure of real numbers. Based on the componentwise order the concept of dominance allows to define a partial order in the objective space of a multi-objective problem and make a distinction between dominated and non-dominated vectors or points as it will be shown in the next section. It will also be possible to see the particularities of a MOILP problem with respect to the definition of its non-dominated vectors (for more details, see Ehrgott, 2005; Steuer, 1986).

### *2.2. Dominance and Efficiency*

Dominance plays a central role in multi-objective optimization. From the componentwise order defined in the previous section a straightforward definition can be stated as follows.

**Definition 2.1** (Dominance)**.** *Let $z', z'' \in \mathbb{R}^p$ denote two outcome vectors. Then, $z'$ dominates $z''$ if $z' \geq z''$.*

Let $\bar{z} \in Z$ denote a feasible outcome vector. Then, $\bar{z}$ is called a *non-dominated vector* if and only if there does not exist another $z \in Z$ such that $z \geq \bar{z}$. Otherwise, $\bar{z}$ is a dominated outcome vector. Let $Z_{\mathrm{N}}$ denote the set of all non-dominated outcome vectors or points.
A feasible solution $\bar{x} \in X$ is called an *efficient solution* if and only if there does not exist another $x \in X$ such that $z(x) = Cx \geq z(\bar{x}) = C\bar{x}$. Otherwise, $\bar{x}$ is called *inefficient*. Let $X_{\mathrm{E}}$ denote the set of all efficient solutions. For MOILPs it holds that $X_{\mathrm{E}} \neq \varnothing$ if $X \neq \varnothing$.
A non-dominated outcome vector $\bar{z} \in Z_{\mathrm{N}}$ is said to be *supported non-dominated* if there is a real vector $\lambda \in \mathbb{R}^p$ with $\lambda \geq 0$ such that $\lambda^\top \bar{z} \geqslant \lambda^\top z$ for all other feasible outcome vectors $z \in Z$. Otherwise, the non-dominated outcome vector is called *unsupported non-dominated*. Analogously, the preimage of a supported non-dominated vector is called *supported efficient solution*, while the preimage of an unsupported non-dominated vector is called *unsupported efficient solution*. Let the sets of supported non-dominated vectors, unsupported non-dominated vectors, supported efficient solutions, and unsupported efficient solutions be denoted, respectively by $Z_{\mathrm{sN}}$, $Z_{\mathrm{uN}}$, $X_{\mathrm{sE}}$, and $X_{\mathrm{uE}}$.

*2.3. Adjacency and Neighborhood*

In order to investigate the relations and distances between feasible solutions in the decision space, we need to introduce three concepts: adjacency, adjacency graph, and neighborhood structure. We adopt two concepts of adjacency as suggested in Gorski et al. (2011): An *LP-based definition of adjacency* and a *combinatorial definition of adjacency*. For the first, we have to assume that *all* feasible solutions in $X$ correspond to extreme points of the linear programming relaxation of the problem, i.e., $X = \text{ext}\{x \in \mathbb{R}^n : Ax \leq b\}$ with some integer constraint matrix $A$ and right-hand-side $b$. Then two feasible solutions $x', x'' \in X$ are called *adjacent* if and only if $x''$ can be obtained from $x'$ by applying a single pivot operation. Note that the resulting concept of adjacency depends on the given ILP formulation of the problem and is in general not unique.

A combinatorial definition of adjacency, on the other hand, operates directly on the combinatorial structure of the considered problem. It is thus formulated for multi-objective combinatorial optimization problems (MOCO), i.e., for MOIP problems that possess a combinatorial structure. Examples are knapsack and assignment problems as well as shortest path, spanning tree, and network flow problems. Adjacency in this sense is always problem specific. For example, in an instance of the knapsack problem two binary solutions $x', x''$ can be considered adjacent if they differ in at most two variable entries, where at most one of these entries is equal to 1 in each of the two adjacent solutions. Similarly, two spanning trees $x', x''$ of a minimum spanning tree problem are usually called adjacent if $x''$ can be obtained from $x'$ by adding an edge and removing another edge from the obtained cycle. In some of these cases, a combinatorial definition of adjacency may in fact be equivalent to an appropriate LP-based definition of adjacency. We refer to Gorski et al. (2011) for more details. In both cases, we refer to the operation of moving from one feasible solution $x'$ to an adjacent feasible solution $x''$ as an *elementary move*.

The notion of adjacency imposes an interrelation between the feasible solutions in $X$ which can be represented in terms of a so-called adjacency graph.

**Definition 2.2** (Adjacency graph)**.** *The adjacency graph for a set of solutions $X$ is a graph $G = (V, A)$ where each node in $V$ represents one solution in $X$ and vice versa. Two vertices $v_{x'}$ and $v_{x''}$ are connected by an edge $[v_{x'}, v_{x''}] \in A$ if and only if the corresponding solutions $x'$ and $x''$ are adjacent.*

**Definition 2.3** (Neighborhood)**.** *The* neighborhood *of a feasible solutions $x$, denoted by $\mathcal{N}(x)$, is a subset of $X$ containing all feasible solutions $x'$ that are adjacent to $x$. The solutions $x' \in \mathcal{N}(x)$ are also called* neighbors *of $x$.*

Solutions in the neighborhood of a feasible solutions $x$ are also denoted as immediate or first-order neighbors, i.e., those solutions that are at most a single elementary move away from $x$. Analogously, it is possible to define a $k$-order neighborhood, where the neighbors of $x$ are at most $k$ elementary moves away from $x$. In what follows only immediate neighbors are used to form the neighborhood structure.

In the next section all the concepts will be illustrated through the use of a bi-objective cardinality constrained problem.

*2.4. Illustrative Example*

To illustrate the neighborhood structure of a combinatorial optimization problem we use a bi-objective cardinality constrained optimization problem (CCP):

$$\begin{aligned} \max \;\; z_1(x) \;&=\; \sum_{j=1}^{n} c_j^1 \, x_j \\ \max \;\; z_2(x) \;&=\; \sum_{j=1}^{n} c_j^2 \, x_j \\ \text{subject to:} \;\; &\sum_{j=1}^{n} x_j = \ell \\ & x_j \in \{0,1\}, \quad j = 1, \ldots, n \end{aligned} \tag{CCP}$$

The feasible set of problem (CCP) is denoted by $X := \{x \in \{0,1\}^n : \sum_{j=1}^{n} x_j = \ell\}$. We define one elementary move as swapping two variables, i.e., two feasible solutions $x$ and $x'$ are adjacent if there exist $p, q \in \{1, \ldots, n\}$ such that

$$\begin{cases} x_p = 0 \;\text{ and }\; x'_p = 1 & \text{for } \; p \in \{1, \ldots, n\}, \\ x_q = 1 \;\text{ and }\; x'_q = 0 & \text{for } \; q \in \{1, \ldots, n\} \setminus \{p\}, \text{ and} \\ x_j = x'_j & \text{for all } \; j \in \{1, \ldots, n\} \setminus \{p, q\} \end{cases}$$

Note that every vertex $v_x$ in $G$ has the same degree $|\mathcal{N}(x)| = \ell(n - \ell)$, i.e., the same number of immediate neighbors.

Consider the following numerical example of a bi-objective cardinality constrained problem with 10 items:

$$\begin{aligned} \max \quad z_1(x) \;&=\; (9,5,3,5,10,5,9,9,8,4)\, x \\ \max \quad z_2(x) \;&=\; (6,10,10,3,8,5,3,2,4,4)\, x \\ \text{s.t.} \quad \sum_{i=1}^{10} x_i \;&=\; 5 \\ x_i \;&\in\; \{0,1\} \quad \forall i \in \{1, \ldots, 10\}. \end{aligned} \tag{3}$$

The set of non-dominated solutions for the previous problem is the following. They are lexicographically ordered with respect to the first objective function value.

$$\begin{aligned} Z_{\mathrm{N}} = \; \Big\{ & z^1 = (45,23),\, z^2 = (42,29),\, z^3 = (41,31),\, z^4 = (38,32), \\ & z^5 = (37,33),\, z^6 = (36,37),\, z^7 = (35,38),\, z^8 = (32,39) \Big\} \end{aligned}$$

The sets of supported non-dominated and unsupported non-dominated outcome vectors are, respectively, the following, $Z_{\mathrm{sN}} = \{z^1, z^3, z^6, z^7, z^8\}$ and $Z_{\mathrm{uN}} = \{z^2, z^4, z^5\}$ (see Figure 1).

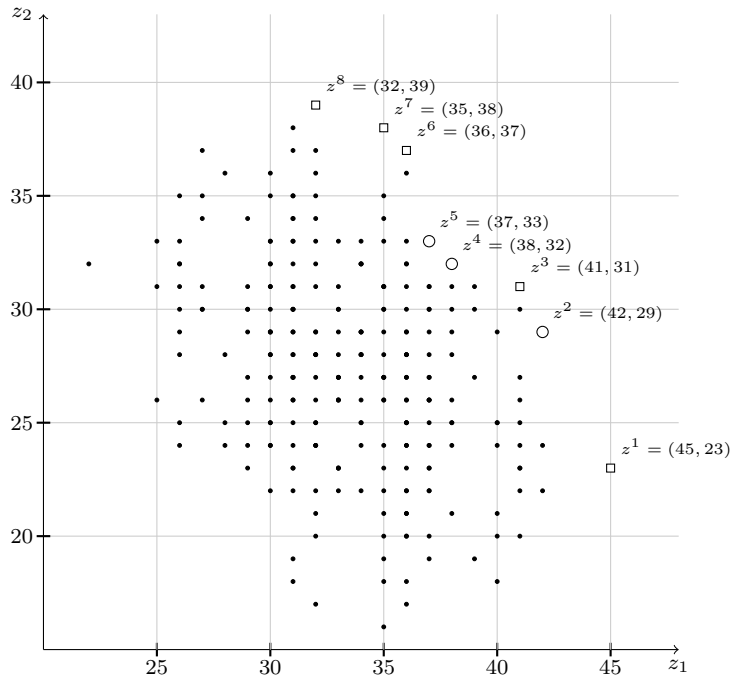



Figure 1: The set of feasible, supported non-dominated (squares) and unsupported non-dominated (circles) outcome vectors for the illustrative example in Section 2.4.

The set of corresponding efficient solutions is as follows.

$$X_{\mathrm{E}} = \Big\{x^1 = (1000101110),\, x^2 = (1100101100),\, x^3 = (1100101010),\, x^4 = (1100111000),\\ x^5 = (1100110010),\, x^6 = (1110101000),\, x^7 = (1110100010),\, x^8 = (1110110000)\Big\}$$

Analogously, the sets of supported efficient and unsupported efficient solutions are, respectively, the following, $X_{\mathrm{sE}} = \{x^1, x^3, x^6, x^7, x^8\}$ and $X_{\mathrm{uE}} = \{x^2, x^4, x^5\}$.
Let us now consider an efficient solution, say $x^3 \in X_{\mathrm{E}}$ and determine the neighborhood $\mathcal{N}(x^3)$. It consists of the following $\ell(n-\ell) = 5 \cdot (10-5) = 25$ solutions:

$$\begin{aligned}
&x^{3(1)} = (0110101010),\, x^{3(2)} = (0101101010),\; x^{3(3)} = (0100111010),\, x^{3(4)} = (0100101110),\\
&x^{3(5)} = (0100101011),\, x^{3(6)} = (1010101010),\; x^{3(7)} = (1001101010),\, x^{3(8)} = (1000111010),\\
&x^{3(9)} = (1000101110)\;\; x^{3(10)} = (1000101011),\, x^{3(11)} = (1110001010),\, x^{3(12)} = (1101001010),\\
&x^{3(13)} = (1100011010),\, x^{3(14)} = (1100001110),\, x^{3(15)} = (1100001011),\, x^{3(16)} = (1110100010),\\
&x^{3(17)} = (1101100010),\, x^{3(18)} = (1100110010),\, x^{3(19)} = (1100100110),\, x^{3(20)} = (1100100011),\\
&x^{3(21)} = (1110101000),\, x^{3(22)} = (1101101000),\, x^{3(23)} = (1100111000),\, x^{3(24)} = (1100101100),\\
&x^{3(25)} = (1100101001).
\end{aligned}$$

And, the image of $\mathcal{N}(x^3)$ in the objective space consists of the following outcome vectors (plotted in Figure 2):

$$z^{3(1)} = (35, 35),\ z^{3(2)} = (37, 28),\ \ z^{3(3)} = (37, 30),\ z^{3(4)} = (41, 27),$$
$$z^{3(5)} = (36, 29),\ z^{3(6)} = (39, 31),\ \ z^{3(7)} = (41, 24),\ z^{3(8)} = (41, 26),$$
$$z^{3(9)} = (45, 23),\ z^{3(10)} = (40, 25),\ z^{3(11)} = (34, 33),\ z^{3(12)} = (36, 26),$$
$$z^{3(13)} = (36, 28),\ z^{3(14)} = (40, 25),\ z^{3(15)} = (35, 27),\ z^{3(16)} = (35, 38),$$
$$z^{3(17)} = (37, 31),\ z^{3(18)} = (37, 33),\ z^{3(19)} = (41, 30),\ z^{3(20)} = (36, 32),$$
$$z^{3(21)} = (36, 37),\ z^{3(22)} = (38, 30),\ z^{3(23)} = (38, 32),\ z^{3(24)} = (42, 29),$$
$$z^{3(25)} = (37, 31).$$

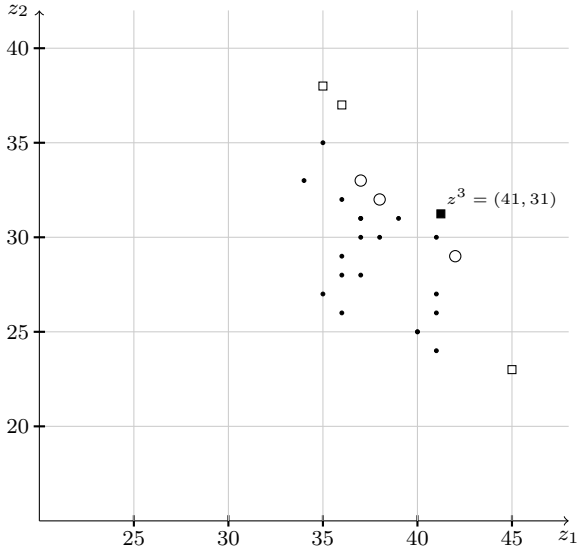



Figure 2: The neighborhood of solution $x^3$ in the objective space (illustrative example in Section 2.4).

The neighborhoods of all efficient solutions $x \in X_{\mathrm{E}}$ of the example given in (3) are illustrated in Figure 5 in the Appendix.

## 3. Robustness Indicators for MOILP Problems

In this section we introduce two main classes of robustness indicators for MOILP problems. The first subsection is devoted to a feasibility based robustness indicator, while the second subsections presents some efficiency based indicators. The section ends with a continuation of the example (3) introduced in Section 2.4.

### *3.1. Feasibility Based Indicators*

In case of failure of an efficient solution, the feasibility based robustness indicator answers the question on the number of alternative neighboring solutions, which could be used

to substitute. In that sense, counting the number of feasible neighbors is a measure of robustness for a given efficient solution. The following definition presents these concepts in a more formal and concrete way.

**Definition 3.1** (Feasibility robustness indicator)**.** *Let $x \in X_{\mathrm{E}}$ denote an efficient solution. Then the number of feasible neighbors of $x$*

$$I^{cf}\Big(x, \mathcal{N}(x)\Big) = |\mathcal{N}(x)|,$$

*is called* feasibility robustness indicator *of $x$ w.r.t. $\mathcal{N}(x)$.*

In the case of problem (CCP) we have that $I^{cf}(x) = |\mathcal{N}(x)| = (n-\ell)\ell$ is constant, for all $x \in X$. Thus, feasibility robustness yields no meaningful result for (CCP). Let us consider feasibility robustness for other MOILPs with a neighborhood structure based on swaps of items. Given an efficient solution, $x \in X_{\mathrm{E}}$, where $x$ is composed of several items, assume we cannot implement one of its items, say item $j$, i.e., $x_j = 1$. This item will be replaced by another item, say item $i$, with $x_i = 1$. An immediate neighbor is a solution of the form, $x' = x + e^i - e^j$, where $e^i$ and $e^j$ are unit vectors. More formally, the set of *immediate neighbors* of $x$ is given by:

$$\mathcal{N}(x) = \{x' \in X \ : \ x' = x + e^i - e^j, \ i, j = 1, \ldots, n, \ i \neq j, \ x_i = 0, \ x_j = 1\}.$$

The motivation for our indicator comes from the fact that if, for some reason, we cannot implement exactly the solution $x$, we should be able to implement a solution $x' \in \mathcal{N}(x)$. Thus, the higher the cardinality of the neighborhood of a solution the better the solution is (the larger the number of alternatives available for replacing it).

### *3.2. Efficiency Based Indicators*

Efficiency based indicators go beyond feasibility based robustness indicators and take into account not only the number of feasible solutions in the neighborhood but also their quality. Counting the number of efficient neighbors is a measure of robustness for a given efficient solution and allows to define a class of indicators. Another general class of indicators is defined by taking into account the changes in the components of the outcome vectors.

#### *3.2.1. Cardinality Based Efficiency Indicator*

Let $x \in X_{\mathrm{E}}$ denote an efficient solution, and let $\mathcal{N}(x)$ denote the set of neighbors of $x$. The number of neighbors that are also efficient is an indicator of how well the solutions in the immediate neighborhood of $x$ perform with respect to the dominance ordering. The following formal definition can now be introduced.

**Definition 3.2** (Efficiency robustness indicator)**.** *Let $x \in X_{\mathrm{E}}$. Then the number of efficient immediate neighbors of $x$*

$$I^{ce}\Big(x, \mathcal{N}(x), X_{\mathrm{E}}\Big) = |\mathcal{N}(x) \cap X_{\mathrm{E}}|,$$

*is called* efficiency robustness indicator *of* $x$ *w.r.t.* $\mathcal{N}(x)$ *and* $X_{\mathrm{E}}$.

Let $\bar{x}$ be an efficient solution of problem (CCP) containing item $i$, and suppose item $i$ becomes unavailable. Consider the same instance of (CCP) but without item $i$, denoted by (CCP$_{-i}$). Then, all neighbors of $\bar{x}$ without item $i$ that are efficient in (CCP) are efficient solutions of (CCP$_{-i}$).

*3.2.2. $\varepsilon$-robustness Indicator*

Let $x \in X_{\mathrm{E}}$ and let $\mathcal{N}(x)$ denote the set of the neighbors of $x$. The $I^{\varepsilon}(x)$ indicator measures the highest outcome criterion-wise degradation with respect to $Z_{\mathrm{N}}$ over the complete set of neighbors of $x$. The smaller the value of $I^{\varepsilon}(x)$ is the more robust $x$ is with respect to this indicator.

**Definition 3.3** ($\varepsilon$-robustness indicator)**.** *Let* $x \in X_{\mathrm{E}}$. *Then*

$$I^{\varepsilon}\Big(x, \mathcal{N}(x), Z_{\mathrm{N}}\Big) = \max_{\hat{x}\in\mathcal{N}(x)} \left\{ \min_{z\in Z_{\mathrm{N}}} \left\{ \max_{k=1,\dots,p} \left\{ \frac{z_k}{c^{k\top}\hat{x}} \right\} \right\} \right\},$$

*is called* $\varepsilon$-robustness indicator *of* $x$ *w.r.t.* $\mathcal{N}(x)$ *and* $Z_{\mathrm{N}}$.

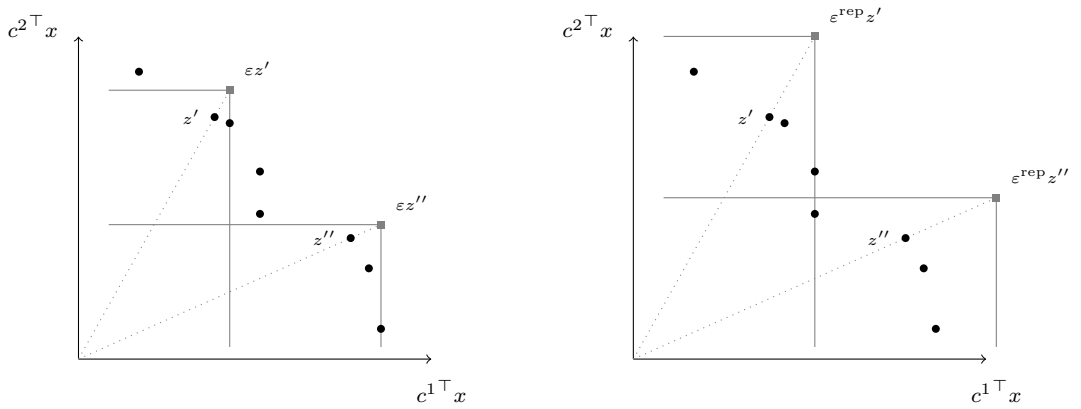
The value of the $\varepsilon$-robustness indicator of an efficient solution $x$ is the smallest scalar such that each image of a neighboring solution which is stretched by this scalar dominates at least one non-dominated point. One might wonder why we are not considering the stretching factor to make any neighboring point non-dominated, which would be reasonable from an application point of view. However, the determination of this factor is difficult since it would involve a disjunctive formulation (better in at least one objective function).

The $\varepsilon$-robustness indicator is defined in analogy to the *$\varepsilon$-indicator* as used in representation and approximation algorithms (see, e.g., Zitzler et al., 2003). However, $\varepsilon$-indicator follows a different paradigm:

$$\min_{\substack{R\subseteq Z_{\mathrm{N}}\\ |R|=k}} \max_{z\in Z_{\mathrm{N}}} \left\{ \min_{\hat{x}\in R} \left\{ \max_{k=1,\dots,p} \left\{ \frac{z_k}{c^{k\top}\hat{x}} \right\} \right\} \right\}$$

The value of the $\varepsilon$-indicator is the scalar stretching factor such that each point in the non-dominated set is dominated by at least one stretched non-dominated point in the representation. For a graphical comparison of $\varepsilon$-robustness indicator and $\varepsilon$-indicator see Figure 3.

Instead of considering the worst case, i.e., the maximal stretching factor required for a solution in the neighborhood, the $\varepsilon$-average-robustness indicator takes the average stretching factor into account. This can compensate neighboring solutions quite far away form the Pareto front if the majority of neighbors is efficient or close to the Pareto front.




(a) $\varepsilon$-robustness indicator (b) $\varepsilon$-indicator

Figure 3: Comparison of $\varepsilon$-robustness indicator $\varepsilon$ and $\varepsilon$-indicator $\varepsilon^{\text{rep}}$. In both cases the same two points $z'$ and $z''$ are selected as an illustrative example for images of neighboring solutions or for a representative subset, respectively. In the illustration both points $z'$ and $z''$ are streched by the same factor such that at least one point is dominated by each or all points are dominated by at least one, respectively.

**Definition 3.4** ($\varepsilon$-average-robustness indicator)**.** *Let $x \in X_{\mathrm{E}}$. Then*

$$I^{\Sigma\varepsilon}\Big(x, \mathcal{N}(x), Z_{\mathrm{N}}\Big) = \frac{1}{|\mathcal{N}(x)|} \sum_{\hat{x}\in\mathcal{N}(x)} \min_{z\in Z_{\mathrm{N}}} \left\{ \max_{k=1,\dots,p} \left\{ \frac{z_k}{c^{k\top}\hat{x}} \right\} \right\},$$

*is called* $\varepsilon$-average-robustness indicator *of $x$ w.r.t. $\mathcal{N}(x)$ and $Z_{\mathrm{N}}$.*

In some applications, it may be meaningful to assume that, whenever an item becomes unavailable in an efficient solution $x$, it is replaced by the *best possible* alternative. In this case, we can consider a criterion-wise $\varepsilon$-robustness indicator as follows. Let $\mathcal{N}^j(x)$ denote the set of neighbors of $x$, in which the $j$-th variable with value 1 in $x$ has value 0 in those neighbors. The $I_D^{\varepsilon}(x)$ indicator measures the highest outcome criterion-wise degradation with respect to $Z_{\mathrm{N}}$, assuming that whenever a variable should be replaced, the best possible replacement will be used.

**Definition 3.5** (Interdiction $\varepsilon$-robustness indicator)**.** *Let $x \in X_{\mathrm{E}}$. Then*

$$I_D^{\varepsilon}\Big(x, \mathcal{N}^j(x), Z_{\mathrm{N}}\Big) = \max_{j=1,\dots,n} \left\{ \min_{\hat{x}\in\mathcal{N}^j(x)} \left\{ \min_{z\in Z_{\mathrm{N}}} \left\{ \max_{k=1,\dots,p} \left\{ \frac{z_k}{c^{k\top}\hat{x}} \right\} \right\} \right\} \right\},$$

*is called* interdiction $\varepsilon$-robustness indicator *of $x$ w.r.t. $\mathcal{N}(x)$ and $Z_{\mathrm{N}}$.*

Also in the case of interdiction robust indicators we take an average case analysis into account by defining the interdiction $\varepsilon$-average-robustness indicator.

**Definition 3.6** (interdiction $\varepsilon$-average-robustness indicator)**.** *Let $x \in X_{\mathrm{E}}$, and let $\varphi_j \in [0,1]$ ($\sum_{j=1}^{n} \varphi_j = 1$) denote a multiplier factor that, if an item selected in $x$ gets*

*unavailable, it will be item* $j$, $j = 1, \dots, n$. *Then*

$$I_D^{\Sigma\varepsilon}\Big(x, \mathcal{N}^j(x), Z_{\mathrm{N}}\Big) = \sum_{j=1}^{n} \varphi_j \left( \min_{\hat{x} \in \mathcal{N}^j(x)} \left\{ \min_{z \in Z_{\mathrm{N}}} \left\{ \max_{k=1,\dots,p} \left\{ \frac{z_k}{c^{k\top} \hat{x}} \right\} \right\} \right\} \right)$$

*is called* interdiction $\varepsilon$-average-robustness indicator *of* $x$ *w.r.t.* $\mathcal{N}(x)$ *and* $Z_{\mathrm{N}}$.

Note that the multiplier factor $\varphi_j$ maybe be a probability, a weight, or in the case of our cardinality constrained knapsack problem $\varphi_j = \ell$, for $j = 1, \dots, n$.
Before we investigate the properties of the robustness indicators we evaluate all of them on the example introduced in Section 2.4. The computation of the robustness indicators is straight forward based on the determination of the neighborhood of the respective solution. The results are presented in Table 1.

### *3.3. Comparison to Decision Robustness in the Continuous Case*

In Eichfelder et al. (2017) decision robustness is considered for continuous multi-objective optimization problems. Since our robustness indicators significantly differ from the set-valued optimization approach suggested therein, we will show the differences and highlight why our indicators are better suited for the discrete structure of MOILPs. In the continuous case the set of realizations of a solution $x$ is assumed to be a compact set containing $x$, which we can identify with an appropriately chosen neighborhood $\mathcal{U}(x)$. According to Eichfelder et al. (2017), a solution $x$ is called decision robust feasible if all of its realizations are feasible, i.e. $\mathcal{U}(x) \subset X$. Furthermore, a solution $x^* \in \hat{X} = \{x \in X : \mathcal{U}(x) \subset X\}$ is called decision robust efficient if there is no $x' \in \hat{X} \setminus \{x^*\}$ such that

$$z(\mathcal{U}(x')) \subset z(\mathcal{U}(x^*)) + \mathbb{R}^p_+ \iff \forall x \in \mathcal{U}(x^*)\ \exists x'' \in \mathcal{U}(x') \colon z(x) \leqslant z(x'')$$

A solution is decision robust, if there is no other solution, which is better in all realizations.
Applying this definition to discrete optimization problems is problematic, since discrete neighborhoods are not local like in the continuous case. Depending on the definition of neighborhood and the problem structure all neighborhoods of feasible solutions might contain infeasible solutions. Moreover, the discrete neighborhood of a solution often covers a significantly large part of the feasible set. In the assignment problem, for example, the distance between any pair of solutions is at most two elementary swaps, i.e., changes along alternating paths (Gorski et al., 2011). Thus, the images of the solutions in the neighborhood might be spread considerably in objective space. Consequently, the majority of solutions would be decision robust w.r.t. to the set-valued definition, because not all realizations. However, in the discrete case it is not possible to consider smaller neighborhoods than the ones defined by basic swaps. Consequently, we relaxed both concepts decision feasibility robustness and decision efficiency robustness to gradual indicators.

| $x$ | $I^{cf}(\cdot)$ | $I^{ce}(\cdot)$ | $I^{\varepsilon}(\cdot)$ | $I^{\Sigma\varepsilon}(\cdot)$ | $I_D^{\varepsilon}(\cdot)$ | $I_D^{\Sigma\varepsilon}(\cdot)$ |
|---|---|---|---|---|---|---|
| $x^1$ | 25 | 2 | 1.277778 | 1.099418 | 1.125000 | 1.044693 |
| $x^2$ | 25 | 4 | 1.208333 | 1.071064 | **1.064516** | **1.023337** |
| $x^3$ | 25 | **6** | 1.171429 | 1.065554 | 1.088235 | 1.029076 |
| $x^4$ | 25 | 5 | 1.185185 | 1.083877 | 1.129032 | 1.045390 |
| $x^5$ | 25 | 4 | 1.193548 | 1.085609 | 1.114286 | 1.045097 |
| $x^6$ | 25 | 5 | **1.166667** | **1.058705** | **1.064516** | 1.024709 |
| $x^7$ | 25 | 4 | **1.166667** | **1.058705** | 1.088235 | 1.034909 |
| $x^8$ | 25 | 4 | 1.230769 | 1.087099 | 1.114286 | 1.039918 |

Table 1: Robustness indicators for the example (3) introduced in Section 2.4.

### *3.4. Properties of Robustness Indicators*

This section provides some theoretical properties of the proposed indicators and the relationship between the three types of indicators. The most well-known unary requirements an indicator must fulfill are quite natural and easy to check. Despite these two aspects, it is important to consider them as properties of our indicators in this paper.

**Proposition 3.7** (Existence)**.** *All the proposed indicators always exist and are well-defined, but they are not unique with respect to the feasible solutions $x \in X$ (i.e., two or more solutions can have the same indicator value).*

*Proof.* It is easy to see from the definitions that all the indicators are well-defined formulas. They are not unique, i.e., they are non-injective, since several solutions can have the same indicator value as it can be seen in Table 1. □

**Proposition 3.8** (Symmetry)**.** *All the proposed indicators are symmetric with respect to any permutation of the components and/or elements of their input.*

*Proof.* The proof is also trivial for all indicators. Let us take for example indicator $I_D^{\varepsilon}$, which has as input the solution $x = (x_1, \ldots, x_j, \ldots, x_n)$, the neighborhoods $\mathcal{N}^j(x)$, for $j = 1, \ldots, n$, and the set of non-dominated points $Z_{\mathrm{N}}$. It is easy to see that for any $x$ re-ordered as $x^{\pi} = (x_{\pi(1)}, \ldots, x_{\pi(j)}, \ldots, x_{\pi(n)})$, where $\pi : \{1, \ldots, n\} \to \{1, \ldots, n\}$ is an arbitrary permutation of the indices, the value of the indicator does not change. The same applies to any permutation of the elements of the sets $\mathcal{N}^j(x)$, for $j = 1, \ldots, n$, and $Z_{\mathrm{N}}$. □

**Proposition 3.9** (Idempotency)**.** *Indicators $I^{\varepsilon}$, $I^{\Sigma\varepsilon}$, $I_D^{\varepsilon}$, and $I_D^{\Sigma\varepsilon}$ are idempotent with respect to the arguments of their aggregation operators.*

*Proof.* This proof is also trivial for all the indicators. Let us consider again indicator $I_D^{\varepsilon}$. This indicator is composed of four min / max aggregation operators. The result min / max of a set where all the arguments have the same value, e.g., $\min\{a, \ldots, a, \ldots, a\}$, is always the same value $a$. □

**Proposition 3.10** (Scale invariance)**.** *Indicators $I^{\varepsilon}$, $I^{\Sigma\varepsilon}$, $I^{\varepsilon}_{D}$, and $I^{\Sigma\varepsilon}_{D}$ are positively homogeneous with respect to the coefficients of the matrix $C$, i.e., any multiplication with a positive scalar of the coefficients of the matrix, $\alpha C$, does not have impact on the values of any of these indicators. However, these indicators are not stable with respect to translations of the coefficients of $C$ (i.e., linear affine transformations).*

*Proof.* Positive homogeneity is only related with the aggregation operator:

$$\max\left\{\frac{z_k}{c^{k\top}\hat{x}}\right\}$$

It is very easy to see that multiplying both $z_k$ and $c^{k\top}$ by $\alpha > 0$ does not change the aggregation value since the operator is still the same:

$$\frac{\alpha z_k}{\alpha c^{k\top}\hat{x}} = \frac{z_k}{c^{k\top}\hat{x}}.$$

The indicators are not stable with respect to linear affine transformations, which can be easily seen from the definition. □

**Proposition 3.11.** *If $I^{\varepsilon}(x) = 1$ then $|\mathcal{N}(x)| = |\mathcal{N}(x) \cap X_{\mathrm{E}}|$.*

*Proof.* This proof is trivial. □

**Proposition 3.12.** *The relations between the following pairs of indicators hold.*

1. $I^{ce}(x) \leqslant I^{cf}(x)$
2. $I^{\varepsilon}_{D}(x) \leqslant I^{\varepsilon}(x)$
3. $I^{\Sigma\varepsilon}(x) \leqslant I^{\varepsilon}(x)$
4. $I^{\Sigma\varepsilon}_{D}(x) \leqslant I^{\varepsilon}_{D}(x)$

*Proof.* The proofs are trivial. □

## 4. Fields of Applications for the Robustness Indicators

The proposed robustness indicators can be considered as a tool to evaluate the quality of efficient solutions in an a-posteriori analysis. This additional quality measure can be utilized in different ways. Two possible approaches will be covered in this section.
Due to the possibly large number of non-dominated points (and efficient solutions, respectively) decision makers are confronted with huge amount of mathematically incomparable alternatives. The burden of the decision maker can be significantly reduced by the selection of a representative subset of non-dominated points/efficient solutions (see e.g. Sayın, 2000; Vaz et al., 2015). In an a-posteriori algorithm to determine a representation, decision robustness can be integrated as an additional quality criterion, ensuring robustness of the representation.

Instead of leaving the decision to a human decision maker it is also possible to use automatic tools, so called decision support systems, to select the most preferred efficient solutions. Since the robustness of a solution can be considered as a quality criterion, it should be integrated in the process of decision making.

### *4.1. Decisions Robust Representations*

We aim at the computation of a representative subset $R \subseteq X_{\mathrm{E}}$ of efficient solutions, which has a fixed cardinality of $|R| = k$, optimizes some representation quality measure and is "decision robust". However, the decision robustness indicators proposed in Section 3 gradually measure the robustness of a single solution. We thus extend this solution-wise definition to sets of solutions. Since every single solution in a representative subset should be a valuable choice for the decision maker, each one should satisfy a certain robustness level. This consideration leads us to a worst case analysis:

**Definition 4.1** (Decision Robustness of solution sets)**.** *Let $R \subseteq X_{\mathrm{E}}$ be a subset of efficient solutions and let $I \in \{I^{cf}, I^{ce}, I^{\varepsilon}, I^{\Sigma\varepsilon}, I_D^{\varepsilon}\}$ denote a decision robustness indicator. Then the respective decision robustness of a subset of efficient solutions $R \in X_{\mathrm{E}}$ is*

$$I(R) = \max_{r \in R} I(r)$$

Most representation quality measures are based on points in the objective space. Since we are considering representative sets of the efficient set $R \subseteq X_{\mathrm{E}}$ in the decision space, we adapt the definitions of uniformity, coverage, and $\varepsilon$-indicator (Sayın, 2000; Zitzler et al., 2003) to our notation. Note that the representation quality is increasing with increasing values of uniformity, thus we will use (contrary to the original definition) their negative value to obtain three minimization quality measures $Q^U$, $Q^C$, and $Q^{\varepsilon}$.

$$\begin{aligned}
&\text{Uniformity} && Q^U(R) = - \min_{\substack{r^i, r^j \in R \\ r^i \neq r^j}} \|z(r^i) - z(r^j)\| \\
&\text{Coverage} && Q^C(R) = \max_{z \in Z_{\mathrm{N}}} \min_{r \in R} \|z(r) - z'\| \\
&\varepsilon\text{-Indicator} && Q^{\varepsilon}(R) = \max_{z \in Z_{\mathrm{N}}} \min_{r \in R} \max_{i \in \{1,\dots,q\}} \frac{z_i}{(z(r))_i}
\end{aligned}$$

**Definition 4.2** (Biobjective robust representation problem)**.** *Let $Q \in \{Q^U, Q^C, Q^{\varepsilon}\}$ denote one of the representation quality measures uniformity, coverage, or $\varepsilon$-indicator, and let $I \in \{I^{cf}, I^{ce}, I^{\varepsilon}, I^{\Sigma\varepsilon}, I_D^{\varepsilon}\}$ denote a decision robustness indicator. Then, the* biobjective robust representation problem *is defined as*

$$\begin{aligned}
\min \quad & I(R) \\
\min \quad & Q(z(R)) \\
\text{s.t.} \quad & |R| = k \\
& R \subseteq X_{\mathrm{E}}
\end{aligned} \tag{RRP}$$

Since the worst case robustness value $I(R) = \max_{x \in R} I(x)$ is attained by (at least) one efficient solution in the representation, it is possible to solve (RRP) by a sequence of $|X_{\mathrm{E}}|$ $\varepsilon$-constraint scalarizations, which can be rewritten by considering only sufficiently robust solutions:

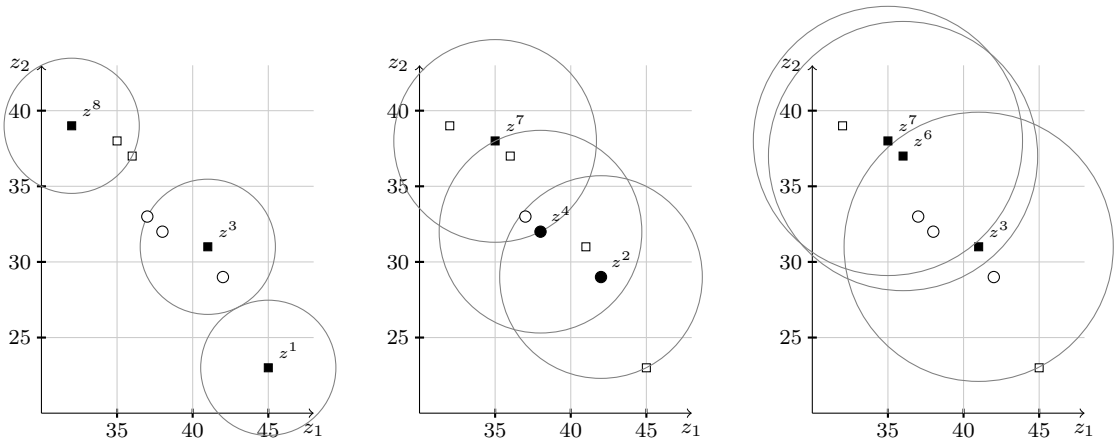
$$\begin{aligned} \min \quad & Q(z(R)) \\ \text{s.t.} \quad & |R| = k \\ & R \subseteq \{r \in X_{\mathrm{E}} \colon I(r) \leq \varepsilon\} \end{aligned} \tag{RRP-$\varepsilon$}$$

In case of biobjective integer programming problems (RRP-$\varepsilon$) can be solved in a running time polynomial in the number of efficient solutions using dynamic programming (Vaz et al., 2015).

We use the numerical example (3) introduced in Section 2.4 to evaluate the biobjective representation problem (RRP) with respect to coverage and $\varepsilon$-robustness.

$$\min_{\substack{R \subseteq X_{\mathrm{E}} \\ |R|=3}} \left( I^{\varepsilon}(R),\ Q^{C}(z(R)) \right) \tag{4}$$

This example (4) has eleven efficient representations mapping to three nondominated points (w.r.t. coverage and $\varepsilon$-robustness), see Figure 4. Since the solution $z^1$ which is important to yield a small coverage radius (c.f. Figure 4(a)), has a rather high $\varepsilon$-robustness indicator value, the two quality measures are conflicting. On the other hand, the three most robust solutions $(z^3, z^6, z^7)$ do not cover the non-dominated set well (c.f. Figure 4(c)).




(a) $R = \{x_1, x_3, x_8\}$, $Q^C(R) = 4.472$, $I^{\varepsilon}(R) = 1.278$

(b) $R = \{x_2, x_4, x_7\}$, $Q^C(R) = 6.708$, $I^{\varepsilon}(R) = 1.208$

(c) $R = \{x_3, x_6, x_7\}$, $Q^C(R) = 8.944$, $I^{\varepsilon}(R) = 1.171$

Figure 4: Three nondominated representation w.r.t. coverage and $\varepsilon$-robustness indicator. The supported non-dominated points are depicted as squares, unsupported non-dominated as circles, which are filled black if the respective is in the robust representative subset. The grey circles indicate the coverage radii.

*4.2. A Composite Qualitative Robustness Index*

This section is devoted to composite qualitative robustness indices that can be used for assessing the robustness of each efficient solution, $x \in X_{\mathrm{E}}$, with respect to (most of) the indicators presented in this paper. We will exemplify this by applying a simplified version of Electre Tri-C (Almeida-Dias et al., 2010). In this simplified version we consider one criterion per indicator and we do not make use of discriminating and veto thresholds. The method can be illustrated through the data in Table 1, by removing index $I^{cf}(\cdot)$ since its value is the same for all the solutions. The remaining indices are thus our criteria. The notation $g_1, \ldots, g_5$ will be used to represent them in this context, where $g_j(x)$ denotes the performance of an efficient solution $x \in X_{\mathrm{E}}$ on criterion $g_j$, for $j = 1, \ldots, 5$ (we will use the subscript $j$ to avoid confusion with components of the vector of variables $x$). For the sake of simplicity we assume without loss of generality that all the criteria are to be minimized. In the context of our illustrative problem (3), we thus multiply the data of the first column by $-1$. The performance table can be presented in the following Table 2 (see also Table 1).

| $x$ | $g_1(x)$ | $g_2(x)$ | $g_3(x)$ | $g_4(x)$ | $g_5(x)$ |
|---|---|---|---|---|---|
| min | $-6$ | 1.166667 | 1.017986 | 1.064516 | 1.023337 |
| max | $-2$ | 1.277778 | 1.057133 | 1.129032 | 1.045390 |
| $x^1$ | $-2$ | 1.277778 | 1.057133 | 1.125000 | 1.044693 |
| $x^2$ | $-4$ | 1.208333 | 1.029872 | 1.064516 | 1.023337 |
| $x^3$ | $-6$ | 1.171429 | 1.024572 | 1.088235 | 1.029076 |
| $x^4$ | $-5$ | 1.185185 | 1.042190 | 1.129032 | 1.045390 |
| $x^5$ | $-4$ | 1.193548 | 1.043855 | 1.114286 | 1.045097 |
| $x^6$ | $-5$ | 1.166667 | 1.017986 | 1.064516 | 1.024709 |
| $x^7$ | $-4$ | 1.166667 | 1.017918 | 1.088235 | 1.034909 |
| $x^8$ | $-4$ | 1.230769 | 1.045288 | 1.114286 | 1.039918 |

Table 2: Performance for the example problem (3) introduced in Section 2.4.

The Electre Tri-C method is a pairwise comparison method with the objective to form an outranking relation for all ordered pairs of efficient solutions (hereafter we use the term actions instead of solutions), and then explore this relation by assigning the actions to categories. In the following, we will discuss the two main steps of the Electre Tri-C method, i.e., *outranking* and *classification*, in more detail and illustrate them using the data of Table 2.

*4.2.1. Pairwise Comparison and Outranking*

In the simplified version of Electre Tri-C, there are only three possibilities when comparing an ordered pair of actions $(x, \hat{x}) \in X_{\mathrm{E}} \times X_{\mathrm{E}}$ on each criterion $g_j$, for $j = 1, \ldots, n$ (recall that all the criteria are to be minimized):

$x$ is at least as good as (outranks) $\hat{x}$ ($x \succsim_j \hat{x}$), iff $g_j(x) \leqslant g_j(\hat{x})$;

$x$ is strictly preferred to $\hat{x}$ ($x \succ_j \hat{x}$), iff $g_j(x) < g_j(\hat{x})$;

$x$ is indifferent to $\hat{x}$ ($x \sim_j \hat{x}$), iff $g_j(x) = g_j(\hat{x})$.

The weight or relative importance (weights can also be interpreted as the voting power of the criterion) of each criterion $g_j$, denoted by $w_j \in [0,1]$, with $\sum_{j=1}^{n} w_j = 1$ is a fundamental preference parameter that impacts the outranking relation. In what follows, we consider the same normalized weight for each criterion, i.e., $w_j = 1/5 = 0.2$, for $j = 1, \ldots, 5$ in the example problem.

*4.2.2. Categorization*

In our example, we consider only three categories: "low robustness ($C^1$)", "medium robustness ($C^2$)", and "high robustness ($C^3$)" in order to qualitatively classify the eight efficient actions by taking into account all the criteria (indices) in a composite (aggregated) way.

Each category is represented by a representative/central element, say $\hat{x}^1$ for the category $C^1$ of the lowest robustness level; $\hat{x}^2$ for the category $C^2$ with the medium robustness level; and $\hat{x}^3$ for the category $C^3$ of the highest robustness level. In the example problem, we can consider the following data for each one of these central actions (let us assume that the actions are "well" separated according to the separability properties of the method). These values are obtained by taking into account the range of each criterion in Table 2, but possibly also some additional information on the quality of the given actions in the respective criterion. For example, the range of the values of $g_1$ are given by $g_1(x) \in [-6, -2]$ for $x \in X_{\mathrm{E}}$, where actually the worst attained value of $-2$ might still be quite good as compared to other non-listed actions. This information is used for defining the first components of the representative actions of each category. Since we are minimizing, the value $-1$ is a rather bad value for an action, $-3$ is a medium value, and $-5$ is a good value for $g_1$. We proceed in the same way for the other components and set

$$
\begin{aligned}
g(\hat{x}^1) &= (-1, 1.260, 1.047, 1.120, 1.041) \\
g(\hat{x}^2) &= (-3, 1.230, 1.037, 1.100, 1.033) \\
g(\hat{x}^3) &= (-5, 1.200, 1.027, 1.080, 1.028).
\end{aligned}
$$

*4.2.3. Implementation*

The first step in the method is to construct a comprehensive outranking relation by taking into account all the criteria at the same time (i.e, by aggregating them). For such a purpose a degree of credibility between an ordered pair of actions, $(x, \hat{x})$, denoted by $\sigma(x, \hat{x})$, is computed. In the simplified version of the method, this degree of credibility is computed only by the power of the coalition of criteria for which $x$ outranks $\hat{x}$, i.e,

$$
\sigma(x, \hat{x}) = \sum_{\{j \,:\, g_j(x) \leqslant g_j(\hat{x})\}} w_j.
$$

This is a fuzzy number since $\sigma(x, \hat{x}) \in [0, 1]$, which represents the degree to which $x$ outranks $\hat{x}$. In other words, it specifies the fraction of votes favorable to $x$ over $\hat{x}$. The

degrees of credibility between the solutions of our example are shown in Table 3.
For example, we compare the performance lists of $x^4$ and $\hat{x}^2$, taking into account their performances $g(x^4) = (\mathbf{-5}, \mathbf{1.185185}, 1.042190, 1.129032, 1.045390)$ and $g(\hat{x}^2) = (-3, 1.230, \mathbf{1.037}, \mathbf{1.100}, \mathbf{1.033})$, respectively. Action $x^4$ is better in the first two criteria and worse in the other three. Thus, $\sigma(x^4, \hat{x}^2) = 0.2 + 0.2 = 0.4$ (the fraction of votes favorable to $x^4$ against $\hat{x}^2$ is 0.4) and $\sigma(\hat{x}^2, x^4) = 0.2 + 0.2 + 0.2 = 0.6$ (with similar interpretation).

| | $\hat{x}^1$ | $\hat{x}^2$ | $\hat{x}^3$ | $\hat{x}^1$ | $\hat{x}^2$ | $\hat{x}^3$ |
|---|---|---|---|---|---|---|
| $x^1$ | 0.2 | 0.0 | 0.0 | 0.8 | 1.0 | 1.0 |
| $x^2$ | 1.0 | 1.0 | 0.4 | 0.0 | 0.0 | 0.6 |
| $x^3$ | 1.0 | 1.0 | 0.6 | 0.0 | 0.0 | 0.4 |
| $x^4$ | 0.6 | 0.4 | 0.4 | 0.4 | 0.6 | 0.8 |
| $x^5$ | 0.8 | 0.4 | 0.2 | 0.2 | 0.6 | 0.8 |
| $x^6$ | 1.0 | 1.0 | 1.0 | 0.0 | 0.0 | 0.2 |
| $x^7$ | 1.0 | 0.8 | 0.4 | 0.0 | 0.2 | 0.6 |
| $x^8$ | 1.0 | 0.2 | 0.0 | 0.0 | 0.8 | 1.0 |

Table 3: Credibility degrees $\sigma(x, \hat{x})$ and $\sigma(\hat{x}, x)$.

In order to transform the fuzzy numbers into crispy ones, we can use a kind of cutting or majority level, denoted by $\lambda$, which may be interpreted as a majority level like in voting theory. For example, if $\lambda \geqslant 0.55$ we only accept an outranking where the coalition of criteria has a majority over 55% (in our example, $x^4$ does not outrank $\hat{x}^2$ since $\sigma(x^4, \hat{x}^2) < 0.55$, but $\hat{x}^2$ outranks $x^4$ since $\sigma(\hat{x}^2, x^4) \geqslant 0.55$). After the application of such a $\lambda$ cutting level we can devise the following comprehensive relations for an ordered pair of alternatives (these relation will further be used to compare the solutions against the characteristic central action and in the assignment procedures in order to assign them to the most adequate category(ies)):

$x$ outranks $\hat{x}$ ($x \succsim_\lambda \hat{x}$) iff $\sigma(x, \hat{x}) \geqslant \lambda$;

$x$ is preferred to $\hat{x}$ ($x \succ_\lambda \hat{x}$) iff $\sigma(x, \hat{x}) \geqslant \lambda$ and $\sigma(\hat{x}, x) < \lambda$;

$\hat{x}$ is preferred to $x$ ($\hat{x} \succ_\lambda x$) iff $\sigma(\hat{x}, x) \geqslant \lambda$ and $\sigma(x, \hat{x}) < \lambda$;

$\hat{x}$ is indifferent to $x$ ($\hat{x} \sim_\lambda x$) iff $\sigma(x, \hat{x}) \geqslant \lambda$ and $\sigma(\hat{x}, x) \geqslant \lambda$;

$\hat{x}$ is incomparable to $x$ ($\hat{x} \|_\lambda x$) iff $\sigma(x, \hat{x}) < \lambda$ and $\sigma(\hat{x}, x) < \lambda$.

Back to our example, since the central action $\hat{x}^2$ is preferred to $x^4$, we use the (reverse strict preference) notation $x^4 \prec_\lambda \hat{x}^2$. Table 4 presents the preference relations between solutions and central actions and it will be useful to understand the mechanism of the assignment procedures (the relations depend on the chosen $\lambda$; in this case we choose $\lambda = 0.55$).

| | $\hat{x}^1$ | $\hat{x}^2$ | $\hat{x}^3$ |
|---|---|---|---|
| $x^1$ | $\prec_{0.55}$ | $\prec_{0.55}$ | $\prec_{0.55}$ |
| $x^2$ | $\succ_{0.55}$ | $\succ_{0.55}$ | $\prec_{0.55}$ |
| $x^3$ | $\succ_{0.55}$ | $\succ_{0.55}$ | $\succ_{0.55}$ |
| $x^4$ | $\succ_{0.55}$ | $\prec_{0.55}$ | $\prec_{0.55}$ |
| $x^5$ | $\succ_{0.55}$ | $\prec_{0.55}$ | $\prec_{0.55}$ |
| $x^6$ | $\succ_{0.55}$ | $\succ_{0.55}$ | $\succ_{0.55}$ |
| $x^7$ | $\succ_{0.55}$ | $\succ_{0.55}$ | $\prec_{0.55}$ |
| $x^8$ | $\succ_{0.55}$ | $\prec_{0.55}$ | $\prec_{0.55}$ |

Table 4: Preference relations with central actions.

From these relations ELECTRE TRI-C makes use of two assignment procedures conjointly to assign the actions to an interval of categories (the best situation occurs when the two procedures produce the same assignment). The categories are, in our case, $C^1 \prec C^2 \prec C^3$: they are ordered from the worst to the best and they are characterized by $\hat{x}^1$, $\hat{x}^2$, and $\hat{x}^3$, respectively. The two procedures need the definition of a selection function of the form

$$\rho(x, \hat{x}) = \min\{\sigma(x, \hat{x}), \sigma(\hat{x}, x)\}.$$

In our example, $\rho(x^4, \hat{x}^2) = \min\{\sigma(x^4, \hat{x}^2), \sigma(\hat{x}^2, x^4)\} = \{0.4, 0.6\} = 0.4$. Table 5 below presents the computations for all the solutions.

| | $\hat{x}^1$ | $\hat{x}^2$ | $\hat{x}^3$ |
|---|---|---|---|
| $x^1$ | 0.2 | 0.0 | 0.0 |
| $x^2$ | 0.0 | 0.0 | 0.4 |
| $x^3$ | 0.0 | 0.0 | 0.4 |
| $x^4$ | 0.4 | 0.4 | 0.4 |
| $x^5$ | 0.2 | 0.4 | 0.2 |
| $x^6$ | 0.0 | 0.0 | 0.2 |
| $x^7$ | 0.0 | 0.2 | 0.4 |
| $x^8$ | 0.0 | 0.2 | 0.0 |

Table 5: The values of $\rho(x, \hat{x})$.

The two procedures can be presented as follows.

**Algorithm 4.3** (Descending procedure)**.** *Choose $\lambda \in [0.5, 1]$. Decrease $k$ from 3 to the first value $k$ such that $\sigma(x, \hat{x}^k) \geqslant \lambda$ (i.e, $x \succsim_\lambda \hat{x}^k$), or set $k$ to 0 if such a value does not exist.*

1. *For $k = 3$, select $C^3$ as a possible category to assign action $x$.*
2. *For $0 < k < 3$, if $\rho(x, \hat{x}^k) > \rho(x, \hat{x}^{k+1})$, then select $C^k$ as a possible category to assign $x$; otherwise, select $C^{k+1}$.*

3. *For $k = 0$, select $C^1$ as a possible category to assign $x$.*

If we continue with our example for $\lambda = 0.55$, the first $k$ for which $\sigma(x^4, \hat{x}^k) \geqslant 0.55$ is for $k = 1$. We are in Case 2 of Algorithm 4.3, but since we do not have $\rho(x^4, \hat{x}^1) = 0.4 > \rho(x, \hat{x}^2) = 0.4$, then $x^4$ will be assigned to $C^2$.

**Algorithm 4.4** (Ascending procedure)**.** *Choose $\lambda \in [0.5, 1]$. Increase $k$ from 1 to the first value $k$ such that $\sigma(\hat{x}^k, x) \geqslant \lambda$ (i.e, $\hat{x}^k \succsim_\lambda x$), or set $k$ to 4 if such a value does not exist.*

1. *For $k = 1$, select $C^1$ as a possible category to assign action $x$.*
2. *For $1 < k < 4$, if $\rho(x, \hat{x}^k) > \rho(x, \hat{x}^{k-1})$, then select $C^k$ as a possible category to assign $x$; otherwise, select $C^{k-1}$.*
3. *For $k = 4$, select $C^3$ as a possible category to assign $x$.*

If we follow the procedure for our example for $\lambda = 0.55$, the first $k$ for which $\sigma(\hat{x}^k, x^4) \geqslant 0.5$ is for $k = 2$. We are in Step 2, but since we do not have $\rho(x^4, \hat{x}^2) = 0.4 > \rho(x, \hat{x}^1) = 0.4$, then $x^4$ will be assigned to $C^1$.
The results are as follows (note that the previous tables change when changing the values of $\lambda$), when using the two procedures conjointly:

- Assignments with $\lambda = 0.55$:

  $$C^1 = \{x^1, x^4\}, \;\; C^2 = \{x^4, x^5, x^8\}, \;\; C^3 = \{x^2, x^3, x^6, x^7\}.$$

  All the actions are precisely assigned, except $x^4$ which can be of low and medium robustness.

- Assignments with $\lambda = 0.65$:

  $$C^1 = \{x^1, x^4\}, \;\; C^2 = \{x^4, x^5, x^8\}, \;\; C^3 = \{x^2, x^3, x^6, x^7\}.$$

  The assignments are the same, even with a higher cutting level.

- To show the impact of the cutting level, we also computed the assignments for $\lambda = 0.85$ (even though very high cutting levels are of little practical relevance):

  $$C^1 = \{x^1, x^4, x^5\}, \;\; C^2 = \{x^7, x^8\}, \;\; C^3 = \{x^2, x^3, x^4, x^5, x^6, x^7\}.$$

  Note that at this very high cutting level, the actions $x^4$ and $x^5$ can not be clearly assigned, i.e., the method provides uncertainty about the robustness of these solutions at this cutting level.

If we look at the indicator values for the actions, the assignments for $\lambda = 0.55$ (and also for $\lambda = 0.65$) make sense. We can say, that $x^2$, $x^3$, $x^6$, and $x^7$ are quite robust actions, while $x^1$ is an action with low robustness.

## 5. Concluding Remarks

In this paper we adopt the concept of decision space robustness to multiobjective integer linear programming problems. In many practical applications, the computed optimal solutions can not be exactly implemented in reality. By identifying possible sets of alternative realizations in appropriately chosen neighborhoods, we propose robustness indicators that assess the quality of the considered solution under slight deviations. This can be applied to support decision making and the selection of a most preferred and simultaneously robust solution. We exemplify such an approach using the ELECTRE TRI-C method. Moreover, robustness indicators can be extended to sets of solutions, such that the robustness of different representations of the efficient set can be evaluated and optimized.

From a computational perspective, the computation of neighborhoods of solutions usually requires the a priori computation of the complete efficient set or feasible set, respectively, which is usually very costly. However, depending on the problem structure, this neighborhood information can be obtained already during the solution process, for example, in dynamic programming type algorithms Correia et al. (see, e.g., 2018). Future research could also address possible advantages of using neighborhood search techniques, despite the fact that the efficient set of multiobjective integer linear programming problems is not connected in general Gorski et al. (2011).

*Acknowledgments.* This work was supported by DAAD-CRUP Luso-German bilateral cooperation under the 2017-2018 research project MONO-EMC (Multi-Objective Network Optimization for Engineering and Management Support). José Rui Figueira also acknowledges the support from the FCT grant SFRH/BSAB/139892/2018 during his stay at the Department of Mathematics of the University of Wuppertal, Germany.

## Appendix

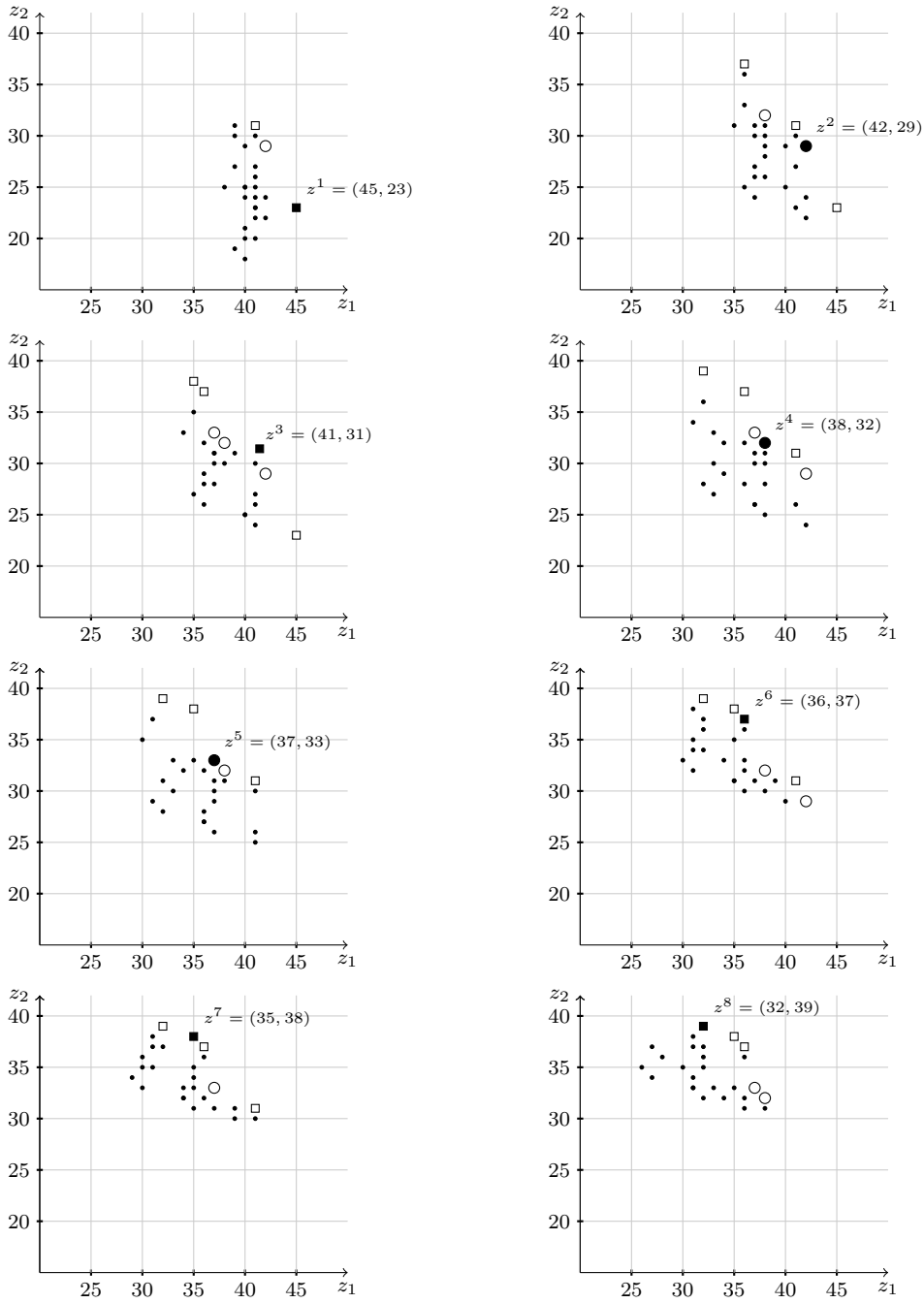



Figure 5: Neighborhoods $\mathcal{N}(x)$ for all efficient solutions $x \in X_{\mathrm{E}}$ of the example given in (3).

## References

A. Adeyefa and M. Luhandjula. Multiobjective stochastic linear programming: An overview. *American Journal of Operations Research*, 1:203–213, 2011.

H. Aissi, C. Bazgan, and D. Vanderpooten. Min–max and min–max regret versions of combinatorial optimization problems: A survey. *European Journal of Operational Research*, 197(2):427–438, 2009.

J. Almeida-Dias, J.-R. Figueira, and B. Roy. Electre Tri-C: A multiple criteria sorting method based on characteristic reference actions. *European Journal of Operational Research*, 204(3):565–580, 2010.

A. Ben-Tal, L. El Ghaoui, and A. Nemirovski. *Robust Optimization*. Princeton University Press, Princeton, USA, 2009.

H.-G. Beyer and B. Sendhoff. Robust optimization – A comprehensive survey. *Computer Methods in Applied Mechanics and Engineering*, 196:3190–3218, 07 2007. doi: 10.1016/j.cma.2007.03.003.

J. Birge and F. Louveaux. *Introduction to Stochastic Programming*. Springer, New York, NY, USA, 1997.

P. Correia, L. Paquete, and J. R. Figueira. Compressed data structures for bi-objective 0,1-knapsack problems. *Computer and Operations Research*, 89:82–93, 2018. doi: doi:10.1016/j.cor.2017.08.008.

M. Dellnitz and K. Witting. Computation of robust Pareto points. *International Journal of Computing Science and Mathematics*, 2:243–266, 2009.

M. Ehrgott. *Multicriteria Optimization*. Springer, Berlin, Germany, 2nd edition, 2005.

M. Ehrgott, J. Ide, and A. Schöbel. Minimax robustness for multi-objective optimization problems. *European Journal of Operational Research*, 239(1):17–31, 2014.

G. Eichfelder, C. Krüger, and A. Schöbel. Multi-objective regularization robustness. Technical Report 2015-13, Preprint-Reihe, Institut für Numerische und Angewandte Mathematik, Georg-August Universität Göttingen, 2015.

G. Eichfelder, C. Krüger, and A. Schöbel. Decision uncertainty in multiobjective optimization. *Journal of Global Optimization*, 69(2):485–510, apr 2017. doi: 10.1007/s10898-017-0518-9.

G. Eichfelder, J. Niebling, and S. Rocktäschel. An algorithmic approach to multiobjective optimization with decision uncertainty. *Journal of Global Optimization*, jul 2019. doi: 10.1007/s10898-019-00815-9.

J. Gorski, K. Klamroth, and S. Ruzika. Connectedness of efficient solutions in multiple objective combinatorial optimization. *Journal of Optimization Theory and Applications*, 150(3):475–497, 2011.

J. Ide and E. Köbis. Concepts of efficiency for uncertain multi-objective optimization problems based on set order relations. *Mathematical Methods of Operations Research*, 80(1):99–127, 2014.

J. Ide and A. Schöbel. Robustness for uncertain multi-objective optimization: A survey and analysis of different concepts. *OR Spectrum*, 38:235–271, 2016.

M. Inuiguchi, K. Kato, and H. Katagiri. Fuzzy multi-criteria optimization: Possibilistic and fuzzy/stochastic approaches. In S. Greco, M. Ehrgott, and J.-R. Figueira, editors, *Multiple Criteria Decision Analysis: State of the Art Surveys*, pages 851–902. Springer Science + Business Media, Inc., New York, NY, USA, 2016.

P. Kouvelis and G. Yu. *Robust Discrete Optimization and Its Applications*. Kluwer Academic Publishers, Norwell, MA, USA, 1997.

G. Mavrotas, J.-R. Figueira, and E. Siskos. Robustness analysis methodology for multi-objective combinatorial optimization problems and application to project selection. *Omega: The International Journal of Management Science*, 52:142–155, 2015.

C. Oliveira and C. Henggeler-Antunes. Multiple objective linear programming models with interval coefficients – An illustrated overview. *European Journal of Operational Research*, 181(3):1434–1463, 2007.

B. Roy, J. R. Figueira, and J. Almeida-Dias. Discriminating thresholds as a tool to cope with imperfect knowledge in multiple criteria decision aiding: Theoretical results and practical issues. *Omega: The International Journal of Management Science*, 43:9–20, 2014.

M. Sakawa, I. Nishizaki, and H. Katagiri. *Fuzzy Stochastic Multiobjective Programming*. Springer Publishing Company, Inc., New York, NY, USA, 2011.

S. Sayın. Measuring the quality of discrete representations of efficient sets in multiple objective mathematical programming. *Mathematical Programming*, 87(3):543–560, 2000.

R. Słowiński and J. Teghem, editors. *Stochastic vs. Fuzzy Approaches to Multiobjective Mathematical Programming Under Uncertainty*. Kluwer Academic Publishers, Norwell, MA, USA, 1990.

R. Steuer. *Multiple Criteria Optimization: Theory, Computation, and Application*. John Wiley & Sons, New York, USA, 1986.

D. Vaz, L. Paquete, C. M. Fonseca, K. Klamroth, and M. Stiglmayr. Representation of the non-dominated set in biobjective combinatorial optimization. *Computers & Operations Research*, 63:172–186, 2015.

K. Witting, S. Ober-Blöbaum, and M. Dellnitz. A variational approach to define robustness for parametric multiobjective optimization problems. *Journal of Global Optimization*, 57:331–345, 2013.

E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. Grunert da Fonseca. Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Transactions on Evolutionary Computation*, 7(2):117–132, 2003.