Uncertainty Feature Optimization: an implicit paradigm for problems with noisy data

Niklaus Eggenberg, Matteo Salani, Michel Bierlaire [*]

August 18, 2009

Abstract

Optimization problems due to noisy data solved using stochastic programming or robust optimization approaches require the explicit characterization of an *uncertainty set* U that models the nature of the noise. Such approaches depend on the modeling of the uncertainty set and suffer from an erroneous estimation of the noise.

In this paper, we introduce a framework that considers the uncertain data implicitly. We define the concept of *Uncertainty Features* (UF), which are problem-specific structural properties of a solution. We show how to formulate an uncertain problem using the *Uncertainty Feature Optimization* (UFO) framework as a multi-objective problem. We show that stochastic programming and robust optimization are particular cases of the UFO framework. We present computational results for the Multi-Dimensional Knapsack Problem (MDKP) and discuss the application of the framework to the airline scheduling problem.

1 Introduction

Nowadays, Operations Research tools are widely used to optimize real world problems. The major difficulty the modelers are faced with is the noisy nature of the data most of the problems are due to. As shown by Birge and Louveaux (1997), Herroelen and Leus (2005) and Sahinidis (2004), a deterministic approach, i.e. an approach neglecting the uncertain nature of the data, leads to unstable solutions: either feasibility is lost or the solution's performance is poor when data is revealed. The ability of a solution to remain feasible with respect to data changes is called the *robustness* of the solution. In the case the solution is not robust, we define the *recoverability* of the solution as the average performance of the solution including both

[*] TRANSP-OR laboratory, École Polytechnique Fédérale de Lausanne, CH-1015 Lausanne, Switzerland

original costs and the costs incurred when modifying the solution to retrieve feasibility, which are called the *recovery costs*; by convention, we assume that a solution that cannot be recovered has infinite recovery costs. The operations of repairing a solution are computed by a *recovery* algorithm.

We distinguish two classes of methods to solve noisy problems: *reactive* and *proactive* methods. The former are also called *on-line* algorithms, and re-compute solutions whenever data changes. The latter compute an *a priori* solution before data is deterministically revealed, which requires predictions on the future data outcome modeled by an *uncertainty set*, denoted by U. Proactive methods are sub-divided into two sub-classes: *expected-mean* and *worst-case* methods. On the one hand, expected-mean methods seek the solution performing best in average over an explicit set U provided with a probabilistic distribution. On the other hand, worst-case based methods seek the most conservative solution, i.e. the one performing best in the worst possible scenario. For both methods, an appropriate model of U is the key to the solution's quality; unfortunately, this is a difficult task and, as we show in this paper, an erroneous estimation of U might have significant consequences in terms of solution quality.

The concept of *Uncertainty Feature Optimization* (UFO) is different from on-line, expected-mean and worst-case approaches: it aims at finding a proactive solution without the explicit characterization of an uncertainty set. The fact that the problem is due to noisy data is considered *implicitly* using *Uncertainty Features* (UF), which are structural properties of the solution improving its robustness or recoverability. The structural properties are problem specific and, in case recoverability is considered, also depend on the chosen recovery strategy. As uncertainty features do not provide any a priori information on the quality of the solution, a reasonable way to validate an UF's efficiency is to evaluate the outcomes by simulation, as we do in this paper.

The initial motivation for UFO comes from the airline scheduling problem, see Kohl et al. (2007) for a general survey. Airline scheduling requires a proactive method, because of the early publication deadlines of the schedule. In addition, due to many unpredictable influencing factors, modeling an explicit uncertainty set is a difficult task. Several contributions in the literature attempt to model an uncertainty set, see for example Lan et al. (2006), Shebalov and Klabjan (2006), Policella (2004). The main conclusions of the works adopting robust approaches are that the obtained solutions exhibit a particular property such as the number of plane crossings (Klabjan et al. (2002), Bian et al. (2005)), a reduced length of plane rotations (Rosenberger et al., 2004) or increased idle time (Al-Fawzana and Haouari, 2005). Remarkably, models aiming at an increase of the solution's recoverability draw the same conclusion: the stochastic model with recourse of Yen and Brige (2006) addresses the crew scheduling problem. Their solutions exhibit pairings with a reduced number of plane changes. The UFO framework is meant to directly optimize these structural properties, modeling them as UFs.

In this work we show that UFO is a general framework for optimization under uncertainty, which does not require the explicit modeling of the uncertainty set. The consequence is a gain of stability of the solution with respect to changes in the noise's nature. Furthermore, we prove that UFO is a generalization of existing proactive methods for a particular choice of UFs. Finally, we present the validation of the framework by simulation.

The structure of the paper is as follows: section 2 summarizes the literature on methods for optimization under uncertainty and discusses their benefits and drawbacks. Section 3 presents the Uncertainty Feature Optimization (UFO) framework and section 4 demonstrates how to derive existing proactive methods from the UFO framework. In section 5, we show practical examples of UFO: we present simulation results on the Multi-Dimensional Knapsack Problem (MDKP) and we discuss the application of UFO to airline scheduling in section 6. Finally, section 7 concludes the paper with some future research issues.

2 Optimization under Uncertainty

For general surveys on optimization under uncertainty we refer to Herroelen and Leus (2005) and Sahinidis (2004) and references therein.

In the literature, we identify three classes of approaches to problems due to noisy data: reactive, stochastic and worst-case (or robust).

Reactive Algorithms Reactive or *on-line* algorithms are based on the wait-and-see strategy, i.e. the solution is updated whenever required. The benefit of such methods is that it eventually provides, if it exists, a globally feasible solution. However, the solution is unstable, since it depends on the data realization. Furthermore, as most often the solution has to be updated in real time, the decision policy limits most often to fast greedy heuristics. Finally, it is hard to estimate the quality of a solution, which is usually measured by the *competitivity ratio*, comparing the obtained solution to the deterministic optimal solution. In real world applications, on-line algorithms perform at acceptable ranges in terms of optimality deviation, but one can usually find scenarios for which the algorithms perform poorly. For a survey on reactive algorithms, we refer to Albers (2003).

Stochastic Programming Stochastic optimization is a widely studied field and a standard approach to deal with uncertainty, see Birge and Louveaux (1997). The main objective is to optimize the *expected* value of the objective over the whole set of uncertain data, i.e. the uncertainty set U: this implies the knowledge of a probabilistic measure on U. The clear benefit of the approach is that the obtained solution is the one that performs best in average: if the solution is carried out many times, then the average cost tends to the expected cost. The drawback is the requirement of an explicit uncertainty set provided with a probabilistic measure. In addition, the approach must evaluate a solution on the whole set U to determine its expected cost, which is, in general, computationally hard. Finally, the computed expected cost is only an estimator on the possible solution's outcome: one cannot guarantee that the real cost matches the expected cost on a single scenario realization. The expected cost is a good indicator only when the obtained solution is implemented many times under the same conditions.

In *stochastic optimization with recourse* or *multi-stage stochastic optimization* (Birge and Louveaux (1997), Kall and Wallace (1994), Herroelen and Leus (2005)), a *recourse* strategy that defines the reaction to take when information on a scenario is revealed is considered. The major advantages of this approach is that two levels of information are considered, namely the a priory knowledge and the possible data outcomes along time: the solution thus also provides the action to take in case of significant information gain. The benefit of the approach is that the two decisional levels lead to the best expected solution, *including recourse costs*, which is a much better approximation on the real cost than the only expected cost (without the recourse costs). The drawbacks are the needs of the probabilistic uncertainty set and the additional computational complexity : the recourse problem has to be solved for each realization in U in order to get a single solution's expected recourse cost, and all solutions must be considered to determine the one minimizing the total expected cost (the sum of first level and recourse costs). For large scale problems where individual evaluation for each scenario is not realistic, the method requires either a closed form for the recourse costs or a formulation of the recourse problem as an underlying problem. In the case of a discrete uncertainty set for which the recourse problem can be expressed as a set of m linear functions given a solution, we get a problem with at least $\mathsf{n} \times \mathsf{m}$ constraints, where n is the number of decisional stages at which recourse has to be taken. In such cases, sampling techniques are necessary tools to deal with the dimension of the deterministic equivalent of the stochastic formulation, resulting in approximate approaches (Linderoth et al., 2006).

Worst-Case Based Approaches The class of worst-case based approaches is mainly composed of methods leading to *robust* solutions, i.e. solutions that are feasible even in the worst possible scenario. Many contributions deal with robust optimization; Soyster (1973) was the first to introduce a formal approach of robustness, and Bertsimas and Sim (2004) and Ben-Tal and Nemirovski (2001) give a more formal framework for different classes of problems. The main advantage of a robust solution is that, if the uncertainty set is exhaustive and a robust solution exists, then the methodology provides an upper bound on the cost. Moreover, as it is a worst case based method, it does not require a probability distribution on the uncertainty set (although its characterization is still needed). The considered uncertainty set plays a crucial role, since it determines the level of protection of the solution. But this is a major drawback, since if all scenarios are considered, the solution might be way too conservative and lead to a solution with high costs for most of the possible outcomes; neglecting part of the possible outcomes leaves the possibility for the solution to become unfeasible. In this case, the cost of the solution is no longer an upper bound, and then the question arises whether the additional costs on the considered outcomes are worth it. The trade-off between conservatism and performance is addressed in Bertsimas and Sim (2004): the solution is ensured to be feasible for a *bounded worst-case*, as opposed to the *unbounded worst-case*; the authors show some bounds on the probability of the solution to be infeasible given a worst-case bound, but do not specify how to set the bounds on the worst case.

This leads to another type of worst-case based approach, namely the *risk management* methods, see Kall and Mayer (2005). For these methods, a probabilistic measure on the uncertainty set is required, and the optimal solution is the one that has the best trade-off between expected cost and probability to be infeasible. The probability to be infeasible is modeled using quantile functions, which return bounds on variables ensuring that the probability of these variables to have values lower or equal to that bound is a chosen constant. The optimal solution is the one with lowest expected cost given a specific value of the probability bound, that is then called the *protection level* of the solution. The benefit of the approach is to find the solution with lowest expected cost and provide a probabilistic measure of infeasibility. The method suffers, however, from the needs of a probabilistic uncertainty set as does stochastic programming. Moreover, the obtained problem is computationally hard, such that only particular problems are solvable. Note that risk management also fits into the class of stochastic methods.

Lately, Fischetti and Monaci (2008) introduce the concept of *light robustness*, which can be seen as an extension of Bertsimas and Sim (2004). The aim of a light robust solution is to minimize the constraint violation within a determined maximal deviation from the deterministic optimal solution. The quality of a solution is defined as the worst violation in the basic Light Robustness (LR) and the deviation from the average violation in the Heuristic Light Robustness (HLR) approach. In this work, the authors fix a maximal optimality deviation from the deterministic optimum within which the LR or HLR measures of robustness have to be optimized. The study limits to integer linear problems with the uncertainty set defined by Bertsimas and Sim (2004).

In both the (light) robust and the risk management methods, the user invests some additional costs in order to gain feasibility within a determined set of outcomes. Bertsimas and Sim (2004) call it the *price of robustness.*

We learn from the literature that all existing methods have some drawbacks: deriving an uncertainty set is a difficult problem; erroneous uncertainty sets may significantly impact the solution's performance in reality; only few a priori information is known about the real outcome. Additionally, stochastic programming approaches lead to computationally hard problems (Birge and Louveaux, 1997) and robust solutions might be too conservative.

The Uncertainty Feature Optimization (UFO) framework do not need the estimation of the uncertainty set which is the main drawback of other a priori approaches. This reduces the modeling effort of the characterization of the uncertainty set U, makes the approach more stable against errors in the noise's nature estimation and does not significantly increase the complexity of the original problem. The inconvenience is that no a priori guarantee about future outcome is possible: only simulation allows to test the performance of the solution. Moreover, the problem of determining the UFs is problem specific.

3 UFO Framework

The main idea of Uncertainty Feature Optimization (UFO) is to save the modeling effort to derive an uncertainty set U by considering the uncertainty implicitly with Uncertainty Features (UF)s. An UF is a structural property of the solution that is proven to ameliorate the solution's *robustness* (capacity to remain feasible) or *recoverability* (reduction of recovery costs when solution is infeasible). Without loss of generality, we suppose the UF has to be maximized in order to increase the solution's robustness or recoverability.

Consider the general deterministic optimization problem (P):

$$\begin{aligned} z_{\mathsf{P}} = &\min f(\mathbf{x}) && (1)\\ &\alpha(\mathbf{x}) \leq \mathbf{b} && (2)\\ &\mathbf{x} \in X && (3) \end{aligned}$$

Additionally, we suppose that (P) is prone to noise in the data, whose nature is unknown and is neglected in formulation (P). Indeed, the optimal solution of (P) might be unfeasible when exposed to the realization of the data.

An *Uncertainty Feature* (UF) is a function $\mu : \mathbb{R}^n \to \mathbb{R}$ that maps $\mathbf{x}$ into a scalar $\mu(\mathbf{x})$. Let $\mathbb{M}$ be the number of considered uncertainty features.

We reformulate (P) as a multi-objective optimization problem by adding the uncertainty features $\mu(\mathbf{x})$. Objective (1) becomes:

$$[z_{\mathsf{P}}, z_1, \ldots, z_{\mathbb{M}}] = [\min f(\mathbf{x}), \max \mu_1(\mathbf{x}), \ldots, \max \mu_{\mathbb{M}}(\mathbf{x})]. \qquad (4)$$

The obtained problem is then transformed into the following problem (P'):

$$\begin{aligned} z_{\mathsf{P}'} = &[\max \mu_1(\mathbf{x}), \ldots, \max \mu_{\mathbb{M}}(\mathbf{x})] && (5)\\ &\alpha(\mathbf{x}) \leq \mathbf{b} && (6)\\ &f(\mathbf{x}) \leq (1+\rho) f^* && (7)\\ &\mathbf{x} \in X && (8) \end{aligned}$$

where f^* is the optimal solution of the deterministic problem (P), and ρ is a scalar of the same sign than f^* and is called the *budget ratio.* We call constraint (7) the *budget constraint.* It limits the optimality gap with respect to the deterministic optimal solution f^*.

Remarkably, the feasibility of solution $\mathbf{x}$ according to (P) remains: any feasible solution of (P') is also feasible for (P). The noisy data the problem is prone to is implicitly considered when maximizing the UFs, since the UF is chosen such that solutions with a high UF value are performing better in the noisy environment.

We solve the multi-objective optimization problem (4) using the relaxation of the initial objective in a budget constraint. Other possibilities of solving such a problem are the exploration of the Pareto frontier or to optimize a weighted combination of the different objectives. Although the choice seems arbitrary at this point, we show in the next section that the budget constraint is particularly convenient: first of all, it is an intuitive approach for practitioners as it allows a clear understanding of the additional costs, and it allows us to derive existing a priori methods as particular cases of the UFO framework.

We assume that an increase of $\mu(\mathbf{x})$ implies a better performance of the solution $\mathbf{x}$ under noisy data: there is a significant (inverse) correlation between $\mu(\mathbf{x})$ and $f(\mathbf{x})$ which is a guideline to identify potential efficient UFs. When using several UFs, then (P') is still a multi-objective optimization problem. We suggest, at this stage, to solve a weighted combination of the UFs, normalizing them according to their respective correlation with the original objective.

The methodology to find UFs is to consider a specific problem's structure, the practitioner's knowledge and, if any, the recovery policy to determine intuitive UFs, and then use trial and error simulations to measure this correlation.

4 UFO as a Generalization

In this section we show that stochastic and robust optimization formulations can be derived from the UFO framework using appropriate uncertainty features. We assume, for this section, that the uncertainty set U is provided.

4.1 Stochastic Programming

Consider the following uncertainty feature:

$$\mu_{\mathtt{Stoc}}(\mathbf{x}) = -\mathbb{E}_U(f(\mathbf{x})),$$

where $\mathbb{E}_U(f(\mathbf{x}))$ is the expected value of $f(\mathbf{x})$ over the uncertainty set U. Applying the UFO framework, we get the following problem:

$$\begin{aligned} z_{\mathtt{Stoc}} = &\min \mathbb{E}_U(f(\mathbf{x})) && (9)\\ &\alpha(\mathbf{x}) \leq \mathbf{b} && (10)\\ &f(\mathbf{x}) \leq (1+\rho)f^* && (11)\\ &\mathbf{x} \in X && (12) \end{aligned}$$

When $\rho = 0$ and a feasible solution exists, the solution space reduces to the deterministic optimal solutions only, and the value $z^*_{\mathtt{Stoch}}$ is the expected cost of the deterministic solution. When $\rho \to \infty$, all feasible solutions are considered: the solution is the one minimizing the expected cost, i.e. the solution of the corresponding stochastic expected cost minimization problem.

Suppose that we are provided with a recovery (or *recourse*) strategy: for each solution $\mathbf{x}$, let $g(\mathbf{x}, \xi)$ be the recovery (fixed recourse) costs for solution $\mathbf{x}$ when the observed data outcome is $\xi \in U$. The corresponding Deterministic Equivalent Program (D.E.P.) (Birge and Louveaux, 1997) formulation of a two-stage stochastic program with fixed recourse is:

$$z_{\mathsf{Rec}} = \min f(\mathbf{x}) + \mathbb{E}_{U}(g(\mathbf{x}, \xi)) \tag{13}$$

$$\alpha(\mathbf{x}) \leq \mathbf{b} \tag{14}$$

$$\mathbf{x} \in X \tag{15}$$

We define the following UF:

$$\mu_{\mathsf{Rec}}(\mathbf{x}) = -\left[f(\mathbf{x}) + \mathbb{E}_{U}(g(\mathbf{x}, \xi))\right],$$

Applying the UFO framework, we obtain formulation (13)-(15) with the additional budget constraint $f(\mathbf{x}) \leq (1+\rho)f^*$. Again in presence of a feasible solution, $\rho = 0$ means only deterministic optimal solutions are considered, whereas $\rho \to \infty$ finds the solution of the D.E.P. (Birge and Louveaux, 1997).

4.2 Robust Optimization

Consider the following problem for linear robust optimization:

$$z^*_{\mathsf{ROB}} = \min \mathbf{c}^T \mathbf{x} \tag{16}$$

$$\sum_{j=1,\ldots,m} a_{ij} x_j + \beta_i(\mathbf{x}, \Gamma_i) \leq b_i \qquad \forall i = 1, \ldots, n \tag{17}$$

$$\mathbf{l} \leq \mathbf{x} \leq \mathbf{u} \tag{18}$$

In this problem, only the matrix coefficients A vary; the vectors $\mathbf{l}$ and $\mathbf{u}$ are lower and upper bounds for the variables, respectively. The uncertainty set U is characterized by the sets J_i containing the indexes of the uncertain coefficients for each row $i = 1, \ldots, n$. Each coefficient satisfies $a_{ij} \in [a_{ij} - \hat{a}_{ij}, a_{ij} + \hat{a}_{ij}]$.

Given a solution $\mathbf{x}$, the worst coefficient realization at row i is given by

$$\beta_i(x, \Gamma_i) = \max_{\{S_i \cup \{t_i\} | S_i \subseteq J_i, |S_i| = \lfloor \Gamma_i \rfloor, t_i \in J_i \setminus S_i\}} \left\{ \sum_{j \in S_i} \hat{a}_{ij} \mid x_j \mid + (\Gamma_i - \lfloor \Gamma_i \rfloor) \hat{a}_{it_i} \mid x_{t_i} \mid \right\}, \tag{19}$$

where Γ_i is an upper bound on the number of coefficient allowed to vary simultaneously. With this notation, formulation (16)-(18) is equivalent to the problem considered in Bertsimas and Sim (2004)[1]

We define the complementary function of $\beta_i(x, \Gamma_i)$ as

$$\overline{\beta}_i(\mathbf{x}, \Gamma_i) = \min_{\{S_i \cup \{t_i\} | S_i \subseteq J_i, |S_i| = \lfloor J_i - \Gamma_i \rfloor, t_i \in J_i \setminus S_i\}} \left\{ \sum_{j \in S_i} \hat{a}_{ij} \mid x_j \mid + (J_i - \Gamma_i - \lfloor J_i - \Gamma_i \rfloor) \hat{a}_{it_i} \mid x_{t_i} \mid \right\},$$

which, given a solution $\mathbf{x}$, corresponds to the value of the $J_i - \Gamma_i$ coefficients that contribute least to the total deviation $\sum_{j=1,\ldots,m} \hat{a}_{ij} \mid x_j \mid$.

[1] Bertsimas and Sim (2004) use a maximization problem; we transform it to a minimization problem to match our framework, and replace the y_j variables, $-y_i \leq x_j \leq y_j$ by $\mid x_j \mid$, $\forall j = 1, \ldots, m$

To illustrate the two functions, consider the example with a unique constraint ($n = 1$), 3 variables ($m = 3$) and $J_1 = \{1, 2, 3\}$, i.e. all coefficients are changing. Now, suppose $\Gamma_1 = 1.4$, $\mathbf{x} = (2, 3, 1)^T$ and that $\hat{a}_{1j} = (2, 0.5, 1.2)^T$. The worst scenario for the current solution and Γ_1 is $\beta(\mathbf{x}, 1.4) = 4.6$ (using $S_1 = \{1\}$ and $t_1 = \{2\}$) and the *best* scenario when $J_1 - \Gamma_1 = 1.6$ coefficients take value $a_{ij} + \hat{a}_{ij}$ is $\overline{\beta}(\mathbf{x}, 1.4) = 2.1$ (using $S_1 = \{3\}$ and $t_1 = \{2\}$). We see that $\overline{\beta}(\mathbf{x}, 1.4) + \beta(\mathbf{x}, 1.4) = 6.7 = \beta(\mathbf{x}, 3) = \overline{\beta}(\mathbf{x}, 0)$. This example illustrates the following complementarity theorem.

Theorem (Complementarity)
Given β_i and $\overline{\beta}_i$ defined as above, then

$$\beta_i(\mathbf{x}, J_i) = \overline{\beta}_i(\mathbf{x}, \Gamma_i) + \beta_i(\mathbf{x}, \Gamma_i).$$

The proof of the theorem is in Appendix A. Note also that both β_i and $\overline{\beta}_i$ are positive valued functions.

In the next two paragraphs we address two cases when using formulation (16)-(18). The former considers the problem of finding a robust solution with given values of the parameters Γ_i, and we show that a tighter formulation can be obtained using the UFO framework. The latter is to determine values of the parameters Γ_i such that a robust solution exists, and we show that, using the framework, we are able to derive an algorithm to determine a lower bound for Γ_i that guarantee the existence of a robust solution.

Equivalent UFO formulation We suppose for now that the parameters Γ_i are given and fixed and that a robust solution satisfying (16)-(18) exists. We focus first on the feasibility of the solution for all possible scenarios in U, we apply the framework to this feasibility problem and show that an appropriate choice of UF and ρ lead to an equivalent formulation to (13)-(15) .

Let (F) be the following *feasibility problem*:

$$\begin{aligned}(F) \quad z_F^* &= \min\nolimits_{\mathbf{x}\in X}\{g(\mathbf{x})\}\\ &= \min\nolimits_{\mathbf{x}\in X}\{\max\nolimits_{i=1,\dots,n}(g_i(\mathbf{x}))\}\\ &= \min\nolimits_{\mathbf{x}\in X}\left\{\max\nolimits_{i=1,\dots,n}\left(\sum\nolimits_{j=1}^m a_{ij}x_j + \beta_i(\mathbf{x}, J_i) - b_i\right)\right\}\end{aligned}$$

$g(\mathbf{x})$ is the value of the most violated constraint in the worst scenario when all the J_i coefficients of row i vary, i.e. the unbounded worst-case; the set of feasible solutions X describes the set of solutions defined by (17)-(18). A solution with $g(\mathbf{x}) \leq 0$ is a solution that is feasible on the whole uncertainty set U.

If $z_F^* \leq 0$, i.e. at least one robust solution exists, we set the budget constraint as $g(\mathbf{x}) \leq 0$ and UFO leads to the robust solution that has lowest cost, which is what is sought. If $z_F^* < 0$, it does not make sense to restrict the search space to a subset of the set of all robust solutions, i.e. by setting the budget constraint to be strictly negative, and we still adopt the budget constraint $g(\mathbf{x}) \leq 0$.

We assume that $z_F^* > 0$, i.e. no robust solution exist on U and we denote the optimal solution of (F) by $\mathbf{x}^*$. We apply the UFO framework using $\mu(\mathbf{x}) = -\mathbf{c}^T\mathbf{x}$, i.e. the original cost function with negative sign as UF. As required, μ and g are inversely correlated because of the *price of robustness* (Bertsimas and Sim, 2004). Additionally, maximizing μ increases the performance of the solution: the cost is decreased. We then apply the budget constraint on each function $g_i(\mathbf{x})$ individually, i.e. adding constraints

$$g_i(\mathbf{x}) \leq (1+\rho_i)z_F^* \quad \forall i = 1, \dots, n$$

where $(1+\rho_i)z_F^*$ depends on the value of $\overline{\beta}_i(\mathbf{x}^*, \Gamma_i)$:

$$(1+\rho_i)z_F^* = \begin{cases} \min_{\{k|\overline{\beta}_k(\mathbf{x}^*,\Gamma_k)>0\}} \{\overline{\beta}_k(\mathbf{x}, \Gamma_k)\} & \text{if } \overline{\beta}_i(\mathbf{x}^*, \Gamma_i) > 0 \\ 0 & \text{if } \overline{\beta}_i(\mathbf{x}^*, \Gamma_i) = 0 \end{cases}$$

Note that there is at least one i such that $\overline{\beta}_i(\mathbf{x}^*, \Gamma_i) > 0$: indeed, if $\overline{\beta}_i(\mathbf{x}^*, \Gamma_i) = 0 \ \forall i$, we obtain from the complementarity theorem:

$$g_i(\mathbf{x}^*) = \sum_{j=1}^{m} a_{ij}x_j + \beta_i(\mathbf{x}, J_i) - b_i = \sum_{j=1}^{m} a_{ij}x_j + \beta_i(\mathbf{x}, \Gamma_i) - b_i.$$

Now, as we supposed a solution to (16)-(18) exists, there is at least one solution such that $\sum_{j=1}^{m} a_{ij}x_j + \beta_i(\mathbf{x}^*, \Gamma_i) - b_i \leq 0$, which contradicts the optimality of $\mathbf{x}^*$ as $z_F^* > 0$.

If non-zero, $(1+\rho_i)z_F^*$ corresponds to the lowest constraint violation due to the unbounded worst case with respect to the bounded one, i.e. the additional $J_i - \Gamma_i$ simultaneously varying coefficients. Concretely, adding these budget constraints extends the solution space for all constraints that do not satisfy the bounded worst-case, i.e. they restrict the worst case by limiting the number of simultaneously varying coefficients.

We thus obtain the UFO formulation (F'):

$$z_{F'}^* = \min \mathbf{c}^T\mathbf{x} \tag{20}$$

$$\sum_{j=1,\dots,m} a_{ij}x_j + \beta_i(\mathbf{x}, J_i) - (1+\rho_i)z_F^* \leq b_i \qquad \forall i = 1, \dots, n \tag{21}$$

$$\mathbf{x} \in X \tag{22}$$

For all i such that $\overline{\beta}_i(\mathbf{x}^*, \Gamma_i) = 0$, it is clear that $\beta_i(\mathbf{x}, J_i) \geq \beta_i(\mathbf{x}, \Gamma_i)$ as $\Gamma_i \leq J_i$; therefore, solutions satisfying constraint (21) always satisfy constraints (17). Furthermore, for constraints i such that $\overline{\beta}_i(\mathbf{x}^*, \Gamma_i) > 0$, using the theorem, we get:

$$\beta_i(\mathbf{x}, J_i) - \min_{\{k|\overline{\beta}_k(\mathbf{x}^*,\Gamma_k)>0\}} \overline{\beta}_k(\mathbf{x}, \Gamma_k) \geq \beta_i(\mathbf{x}, J_i) - \overline{\beta}_i(\mathbf{x}, \Gamma_i) = \beta_i(\mathbf{x}, \Gamma_i).$$

(F') is thus a tighter formulation than problem (16)-(18) since it is robust for a larger uncertainty set.

Determining parameters Γ_i The approach of Bertsimas and Sim (2004) assumes values Γ_i as given parameters. It is however not easy to determine values that provide a sufficient protection level while guaranteeing the existence of a solution. Using UFO allows to derive an algorithm to determine, in a finite number of iterations, a lower bound on values of Γ_i such that a bounded robust solution exists or leads to the proof that the solution set X defined by constraints (17)-(18) is empty.

At each iteration k of the algorithm, we consider the problem

$$\begin{aligned} z_F^{(k)^*} &= \min g(\mathbf{x}) \\ & g(\mathbf{x}) \leq (1+\rho^{(k-1)}) z_F^{(k-1)^*} \\ & \mathbf{x} \in X \end{aligned}$$

We denote the optimal solution at iteration k by $\mathbf{x}_k^*$ (or $(x_j)_k^*$ for a single variable).

For the algorithm, we use a different definition for $1+\rho^{(k-1)}$ than previously: we replace min by max to get the following definition:

$$1+\rho^{(k-1)} = \begin{cases} \frac{\max_{i=1,\ldots,n}\left\{\overline{\beta}_i^{(k-1)}(\mathbf{x},\Gamma_i^{(k-1)})\right\}}{z_F^{(k-1)^*}} & \text{if } z_F^{(k-1)^*} > 0 \\ 0 & \text{otherwise} \end{cases}$$

The value of $(1+\rho^{(k)})$ is then determined using the following parameters:

$\Gamma_i^{(0)} = 0;$

$\Gamma_i^{(k)} = \sup\left\{0 \leq \Gamma \leq \Gamma_i^{(k-1)} \mid \overline{\beta}_i(\mathbf{x}_k^*, \Gamma) \geq z_F^{(k)^*}\right\}, \quad k \neq 0.$

We then iterate over k.

This algorithm leads either to the proof that $X = \emptyset$ or to values of $\Gamma_i^{(k)}$, $i = 1, \ldots, n$ such that at least one robust solution exists. The proof of convergence is reported in Appendix B. In the case Γ_i are integers, the method converges in at most $m \times n$ iterations.

When used to derive the approach of Bertsimas and Sim (2004), the UFO framework has similarities with the light robustness of Fischetti and Monaci (2008): both methods adopt a budget constraint. The objective of light robustness, however, is based on an uncertainty characterization: it aims at finding the solution with lowest constraint violation in the worst case. UFO is a generalization of the approach, as the LR and HLR violation methods proposed in the paper can be formulated as UFs. The main difference is that Fischetti and Monaci (2008) start from the original cost minimization problem, aiming at minimizing the constraint violation, and characterize the worst violation, i.e. the worst scenario, according to the optimal solution of the deterministic problem. We believe that this is not correct in general, as the characterization of the worst scenario depends on a solution and should be evaluated for each of them independently, as it is the case in Bertsimas and Sim (2004): the worst scenario is characterized by the function $\beta_i(\mathbf{x}, \Gamma_i)$, which clearly depends on solution $\mathbf{x}$; with this notation, Fischetti and Monaci (2008) characterize the worst scenario as $\beta_i(\mathbf{x}^*, \Gamma_i)$ for each solution $\mathbf{x} \in X$, $\mathbf{x}^*$ being the optimal solution of the deterministic problem; their worst scenario is constant with respect to changing solutions $\mathbf{x}$. The methodology leads to a heuristic way to compute maximal values of Γ_i such that, if it exists, a bounded worst-case robust solution exists.

5 Illustration on the Multi-Dimensional Knapsack Problem

In this section, we show the complete process of applying the UFO framework to a commonly used benchmark problem: the Multi-Dimensional Knapsack Problem (MDPK). We first briefly describe the problem and its equivalent UFO formulation using different UFs. We then describe the performed simulations, which compare the deterministic optimal solution, robust solution using the formulation of Bertsimas and Sim (2004) and four different UFs; we also test solutions obtained when using different UFs (including the robust formulation) simultaneously. The tests are set up to highlight the effects of erroneous uncertainty estimation and the consequences of additional budget. We finally perform a validation of the different UFs according to the obtained results, i.e. we show that some UFs are indeed improving the solutions.

5.1 UFO applied to the Multi-Dimensional Knapsack Problem (MDKP)

We apply the UFO framework to the MDKP, which is commonly used as a benchmark problem for stochastic and robust optimization. The problem is formulated as:

$$
\begin{aligned}
z^*_{\mathtt{MDKP}} = & \max \mathbf{p}^T \mathbf{x} & \\
& \sum_{j=1}^{m} a_{ij} x_j \leq b_i & \forall i = 1, \dots, n \\
& x_j \in \mathbb{Z}_+ & \forall j = 1, \dots, m
\end{aligned}
$$

The x_j variable corresponds to the number of times object j is taken in the solution, $p_j \geq 0$ is its revenue and a_{ij}, $i = 1, \dots, n$ are the different volumes of the object j. The optimal solution of (MDKP) is denoted by $\mathbf{x}^* = \{x^*_j\}_{j=1}^m$, and has value z^*; we also refer to $\mathbf{x}^*$ as the *deterministic* solution of the problem.

We suppose that all coefficients a_{ij} may vary, i.e. the set varying coefficients is $J_i = \{1, \dots, m\}$, $\forall i = 1, \dots, n$.

Using a general UF $\mu(\mathbf{x})$ and applying the UFO framework, we obtain the following problem (MDKP'):

$$
\begin{aligned}
z^*_{\mathtt{MDKP'}} = & \max \mu(\mathbf{x}) & \\
& \sum_{j=1}^{m} a_{ij} x_j \leq b_i & \forall i = 1, \dots, n \\
& \mathbf{p}^T \mathbf{x} \geq (1 - \rho) z^* & \\
& x_j \in \mathbb{Z}_+ & \forall j = 1, \dots, m
\end{aligned}
$$

MDKP is a maximization problem and consequently the budget constraint require a greater or equal sign. Additionally, z^* is multiplied by $(1 - \rho)$, i.e. the budget constraint bounds a maximal loss of revenue.

We derive four different UFs for the problem:

$\mu_{\mathtt{MTK}}(\mathbf{x}) = 1 - \max_{j=1,\dots,m}\{\frac{x_j}{\max_{k=1,\dots,m}\{x_k^*\}}\}$ the Maximal Taken object;

$\mu_{\mathtt{DIV}}(\mathbf{x}) = \sum_{j=1,\dots,m}(\frac{\min\{x_j,1\}}{m})$ the Diversification of the taken objects;

$\mu_{\mathtt{IR}}(\mathbf{x}) = 1 - \max_{j=1,\dots,m}\{\frac{a_{ij}x_j}{b_i}\}$ the maximal Impact Ratio of a taken object;

$\mu_{\mathtt{2SUM}}(\mathbf{x}) = 1 - \max_{i,j\neq k}\{\frac{a_{ij}x_j + a_{ik}x_k}{b_i}\}$ the maximal *size* of two objects in a same constraint;

The UFs' definitions ensure that they all have unconstrained optimal (i.e. maximal) values at 1.0.

The derived UFs follow intuition: taking many times the same object j is *risky*, as if any of its coefficient a_{ij} increases, the solution becomes more likely to be infeasible. The negative sign of $\mu_{\mathtt{MTK}}$ ensures the maximal taken object is minimized. Having a *diversified* solution, which is what $\mu_{\mathtt{DIV}}(\mathbf{x})$ focuses on, is another potentially improving property: we do not expect that all coefficients increase simultaneously, and the increase of some coefficients might be compensated by the decrease of some others. Finally, the $\mu_{\mathtt{IR}}$ and $\mu_{\mathtt{2SUM}}$ capture an aversion to select objects with higher coefficients.

Notation For simplicity, we refer to each UF using its name only; for example MTK is used instead of $\mu_{\mathtt{MTK}}$; DET is the solution of the deterministic problem (MDKP); ROB refers to the robust MDKP derived from the robust formulation of Bertsimas and Sim (2004), with a normalized objective such that its optimal value is 1.

We call a *model* the combination of one or more UFs, including potentially ROB, and a budget ratio ρ. For example MTK_0.1 is the solution of (MDKP') with $\mu_{\mathtt{MTK}}(\mathbf{x})$ and a budget ratio $\rho = 0.1$; IR_DIV_0.2 is the solution of (MDKP') when using as objective function the arithmetic mean of the UFs, i.e. $\frac{\mu_{\mathtt{IR}}(\mathbf{x})+\mu_{\mathtt{DIV}}(\mathbf{x})}{2}$. When $\rho = 1$, the budget constraint is trivially satisfied, as all revenues are positive and the value of $\rho = 1$ is not displayed in a model's name.

Note that when the model contains ROB, then the budget ratio is always set to $\rho = 1$ to avoid conflicts between the budget constraint and the objective.

5.2 Simulation Description

The simulations are performed using a tool developed in Java and using the COIN-OR CBC[2] library, called Multi-dimensional Knapsack Creator (MKC)[3]. We generate a total of 150 *instances* with the following properties, and solve each instance with a total of 56 models, as described by Table (1).

[2] www.coin-or.org

[3] MKC is a package which includes an instance generator, a solver and a simulator for the MDKP. It is available upon request to niklaus.eggenberg@epfl.ch

Model	DET	ROB50	ROB10	ROBU10	MTK	DIV	IR	2SUM
DET	X							
ROB50		X			X	X	X	X
ROB10			X		X	X	X	X
ROBU10				X	X	X	X	X
MTK					R		R	R
DIV						R	R	R
IR							R	
2SUM								R

Table 1: Summary of the 56 different models solved for each instance; X means the model is uniquely solved with budget ratio $\rho = 1.0$; R means the model is solved with budget ratio $\rho \in \{0.05, 0.1, 0.15, 0.2, 0.25\}$.

The instance set is as follows:

number of constraints	$n \in \{1, 5, 10\}$
number of objects	$m = 50$
right hand side	$b_i = 4000r$, $i = 1, \dots, n$ and $r \sim U[0.8, 1.2]$
profit	$p_j = 100r$, $j = 1, \dots, 50$ and $r \sim U[0.8, 1.2]$
matrix coefficient	$a_{ij} = r \times 0.25 p_j$, $i = 1, \dots, n$, $j = 1, \dots, 50$ and $r \sim U[0.8, 1.2]$

The parameters we use for the instance generation are based on the simulations of Pisinger (1995), who shows that profit-weight correlation is computationally harder because of a higher degeneration of the optimal solution. The magnitude of the average values $\mathbf{p}$, $\mathbf{b}$, and $\hat{a}_{ij}$ are matching the values in Bertsimas and Sim (2004); note that in this study, the authors solve problems with $m = 200$, but only for a single constraint; we choose $m = 50$ to reduce complexity, as we solve problems with up to $n = 10$ constraints.

Furthermore, when using ROB, the values of the parameters Γ_i are influencing a solution's performance. We thus use three different set of values, as described below:

ROB50	$\Gamma_i = 50 =\mid J_i \mid$, $i = 1, \dots, n$
ROB10	$\Gamma_i = 10$, $i = 1, \dots, n$
ROBU10	$\Gamma_i \sim U[0, 10]$, $i = 1, \dots, n$

ROB50 considers 50 simultaneously varying coefficients, and corresponds to the unbounded worst case when all coefficients are varying simultaneously. ROB10 and ROBU10 are restricted to 10 simultaneously varying coefficients and a random number between 0 and 10 simultaneously varying coefficients, respectively. With ROB10, we test the effect on underestimating the worst-case by limiting the worst-case according to intuition: the optimal solution of (MDKP) rarely uses more than 10 objects, thus taking $\Gamma_i = 10$ is supposed to be sufficient. The random number of varying coefficients of ROBU10 corresponds to a random protection level for the different constraints. The combined model ROB10_2SUM is the robust solution restricted to 10 simultaneously varying coefficients, and maximizing the following objective function:

$$\frac{\mathbf{p}^T\mathbf{x} + \mu_{\mathtt{2SUM}}(\mathbf{x})}{2z^*}.$$

Thus, ROB50 is the unbounded robust model, whereas ROB10 and ROBU10 are bounded cases limiting to 10 simultaneously varying coefficients and a random number between 0 and 10 simultaneously varying coefficients, respectively.

For each instance, we set the standard deviation matrix $\hat{A}$ used by the robust models ROB50, ROB10 and ROBU10 to be $\hat{a}_{ij} = 0.1 a_{ij}$, as done in Bertsimas and Sim (2004). The uncertainty set U is thus modeled by matrix $\hat{A}$.

For the simulation, we generate, for each instance, a certain number of *scenarios*. Each of them corresponds to a realization of the coefficient matrix, which is denoted $\tilde{A}$. We then test, for each model, if the computed solution is feasible with the new coefficients $\tilde{A}$, i.e. if $\tilde{A}\mathbf{x} \leq \mathbf{b}$ when $\mathbf{x}$ is the solution of a given model. If the solution is infeasible, we count a failure; if it is feasible, we compute the optimality gap with respect to the (deterministic) optimal solution $\tilde{z}^*$ of the scenario, i.e. the solution of problem (MDKP) where $A = \tilde{A}$.

We use 11 different methods to generate $\tilde{A}$ as reported below:

DÂ_r $\tilde{a}_{ij} = a_{ij} + r\hat{a}_{ij}$, $r \in \{0.75, 1.0\}$ (1 scenario per instance)

UA_r $\tilde{a}_{ij} = a_{ij} + s$, $s \sim U[-ra_{ij}, ra_{ij}]$, $r \in \{0.2, 0.25, 0.3\}$ (10 scenarios per instance)

GA_r $\tilde{a}_{ij} = s$, $s \sim N(-\hat{a}_{ij}, r)$, $r \in \{0.2, 0.25, 0.3\}$ (10 scenarios per instance)

RU_r $\tilde{a}_{ij} = s$, $s \sim U[(1-r)25, (1+r)25]$, $r \in \{0.1, 0.15\}$ (10 scenarios per instance)

RG $\tilde{a}_{ij} = s$, $s \sim N(25, 0.1)$ (10 scenarios per instance)

The names of the simulations come from the uncertainty set characterization (A, $\hat{A}$ or purely random) and the used generation type: “D for deterministic, “U for uniform and “G for Gaussian.

For notational simplicity and brevity, we do not detail the individual simulations, but we aggregate results for a particular simulation type: DÂ refers to simulations DÂ_1.0 and DÂ_0.75, UA for UA_r, $r \in \{0.2, 0.25, 0.3\}$, GA for GA_r, $r \in \{0.2, 0.25, 0.3\}$ and RUG for RU_r, $r \in \{0.1, 0.15\}$ and RG_0.1.

The DÂ scenarios are deterministic; in this case, $\hat{A}$ is exactly or slightly overestimating the coefficient's variability for $r = 1$ and $r = 0.75$, respectively. In UA, we randomly generate the coefficients in $[a_{ij} - ra_{ij}, a_{ij} + ra_{ij}]$ with $r \in \{0.2, 0.25, 0.3\}$; in this case, $\hat{A}$ describes correctly the nature of the noise, but underestimates its magnitude; these simulations are meant to show the effect of an underestimation of the noise, which is, actually, what the bounded robust models do.

For GA and RUG, $\hat{A}$ is an erroneous characterization of the uncertain coefficients; in those cases, we simulate the fact that the uncertainty characterization $\hat{A}$ is incorrect both in magnitude and distribution. The aim of these simulations is to show the sensitivity of the different models with respect to erroneous noise characterization. Note that, as ROB50 is the unbounded robust model, we expect all scenarios to be feasible as long as

$\tilde{a}_{ij} \in [a_{ij} - \hat{a}_{ij}, a_{ij} + \hat{a}_{ij}]$, i.e. for simulations DÂ; for the bounded robust models ROB10 and ROBU10, feasibility cannot be guaranteed on any of the simulations, nor can it for any robust model, bounded or not, for any other simulations.

In total, we generate 13,800 scenarios, 4600 for each set of instances with 1, 5 and 10 constraints, respectively.

5.3 Simulation Results

This section summarizes the simulation results; the complete results are available on request[4].

Global Results Tables 2, 3 and 4 report results for representative models for all 150 instances, with a total of 4600 generated scenarios for 1, 5 and 10 constraints, respectively. The tables show, for each model, the UF value (when pertinent), the number of scenarios for which the solution is infeasible, the percentage of infeasible scenarios, the average optimality gap of feasible scenarios compared to the scenario's optimum $\tilde{z}^*$, the maximal observed optimality gap and the average computation time on the 150 solved instances.

Model	DET	ROB50	ROB10	ROBU10	MTK_0.25	DIV_0.25	IR_0.25	2SUM_0.25	DIV_2SUM_0.25	MTK_IR_0.25	ROB50_MTK
UF Value	-	-	-	-	0.9820	1.0	0.9784	0.9569	0.9785	0.9799	0.7895
# infeasible scenarios	2346	1095	1082	1119	1883	1322	34	43	81	106	78
Infeasiblilty Percentage	51.00	23.80	23.52	24.33	40.93	28.74	0.74	0.93	1.76	2.30	1.86
Average gap [%]	12.62	22.37	22.53	21.93	23.42	30.05	44.50	44.04	42.48	42.03	35.18
Maximal gap [%]	99.5	99.55	99.55	99.55	99.60	99.62	99.63	99.64	99.63	99.62	99.59
Average CPU time [s]	0.1	0.7	8.5	6.0	0.1	0.1	0.5	13.2	16.2	0.8	22.1

Table 2: Simulation results for selected models for instances with $n = 1$ constraint with 4600 scenarios.

[4] Please contact the corresponding author for the complete results by email: niklaus.eggenberg@epfl.ch

Model	DET	ROB50	ROB10	ROBU10	MTK_0.25	DIV_0.25	IR_0.25
Average UF Value	-	-	-	-	0.9670	1.0	0.9774
# infeasible scenarios	4151	1823	1648	1969	2146	1158	4
Infeasiblilty Percentage	90.24	39.63	35.83	42.80	46.65	25.17	0.09
Average gap [%]	1.08	9.27	10.21	8.64	12.77	22.91	33.96
Maximal gap [%]	38.05	52.21	52.55	50.42	56.34	60.2	62.95
Average CPU time [s]	0.1	2.1	24.8	14.7	0.1	0.1	1.4

Model	2SUM_0.25	DIV_2SUM_0.25	MTK_IR_0.25	ROB50_MTK
UF Value	0.9554	0.9776	0.9721	0.7772
# infeasible scenarios	2	6	8	54
Infeasiblilty Percentage	0.04	0.13	0.17	1.17
Average gap [%]	34.11	32.26	32.14	24.01
Maximal gap [%]	62.77	61.18	61.13	56.44
Average CPU time [s]	238.9	241.0	1.1	72.8

Table 3: Simulation results for selected models for instances with $n = 5$ constraints with 4600 scenarios.

Model	DET	ROB50	ROB10	ROBU10	MTK_0.25	DIV_0.25	IR_0.25
Average UF Value	-	-	-	-	0.9580	1.0	0.9770
# infeasible scenarios	4469	2061	1640	2285	2598	1192	0
Infeasiblilty Percentage	97.15	44.80	35.65	46.67	56.48	25.91	0.00
Average gap [%]	0.21	6.80	8.70	6.02	8.22	20.71	32.07
Maximal gap [%]	19.04	32.51	33.05	28.79	36.63	44.95	48.54
Average CPU time [s]	0.3	9.1	98.3	75.0	0.2	0.1	6.5

Model	2SUM_0.25	DIV_2SUM_0.25	MTK_IR_0.25	ROB50_MTK
UF Value	0.9550	0.9774	0.9674	0.7716
# infeasible scenarios	0	3	2	57
Infeasiblilty Percentage	0.00	0.07	0.04	1.24
Average gap [%]	32.22	29.7	29.64	21.62
Maximal gap [%]	48.09	49.54	49.53	40.66
Average CPU time [s]	519.9	598.6	3.3	165.3

Table 4: Simulation results for selected models for instances with $n = 10$ constraints with 4600 scenarios.

Looking at Tables 2, 3 and 4, we clearly see that some of the UFs are outperforming even the unbounded robust solution ROB50 in terms of feasibility. Interestingly, we observe that the deterministic solution DET is increasingly bad with an increasing number of constraints, which is actually also the case for the robust models, but not necessarily the UF models. This shows that the more the problem is constrained, the more sensitive it becomes to perturbations. In the case of the MDKP this is intuitive: with more constraints, a same object is more likely to have one of its coefficients increased.

Similarly to the deterministic solutions, the robust solutions tend to take many times a same object. Instead of taking the object with highest marginal benefit, it grabs the one with lowest variability in $\hat{A}$. In that case, if $\hat{A}$ is a poor approximation, there is a higher likelihood that at least one of the coefficients exceeds the upper bound characterized by $\hat{A}$.

We also observe that the optimality gaps are significantly smaller for models with high failure rates; the reason is that for such models, only scenarios being close to the original instance are feasible, meaning that the optimal solution is similar to the model's solution. Looking at the average computation times, we see that the models using 2SUM are the most time consuming, which comes from the quadratic number of variables required to linearize the UF. The robust models are the second most time-consuming in average.

Sensitivity to noise changes Let us consider the results for the different simulations for some of the models, especially the robust ones. Table 5 shows the percentages of failures for a selection of models for 1, 5 and 10 constraints.

n	1 Constraint				5 Constraints				10 Constraints			
Simul Type	DÂ	UA	GA	RUG	DÂ	UA	GA	RUG	DÂ	UA	GA	RUG
ROB50	0.00	30.07	31.93	11.00	0.00	42.60	68.20	10.73	0.00	47.93	80.20	9.27
ROB10	0.00	29.60	31.47	11.07	0.00	37.80	62.27	9.20	26.00	33.60	68.67	5.33
ROBU10	52.00	35.60	33.00	12.53	85.00	41.73	61.13	17.73	93.00	50.20	75.40	20.53
ROB50_MTK	0.00	0.33	4.87	0.00	0.00	0.13	3.47	0.00	0.00	0.00	3.80	0.00
ROB10_IR	83.00	3.40	11.87	0.33	95.00	5.80	1.40	0.27	88.00	6.53	29.00	0.40
ROBU10_DIV	53.00	12.07	21.07	3.60	43.00	9.93	33.33	2.07	51.00	11.60	33.80	3.53
IR_0.25	19.00	29.93	0.93	0.00	3.00	0.00	0.13	0.00	0.00	0.00	0.00	0.00
2SUM_0.25	24.00	21.33	1.20	0.00	1.00	0.00	0.07	0.00	0.00	0.00	0.00	0.00
MTK_IR_0.25	55.00	23.33	2.93	0.00	5.00	0.00	0.20	0.00	1.00	0.00	0.07	0.00

Table 5: Percentage of infeasible solutions obtained by different models for four simulations on problems with $n = 1$, 5 and 10 constraints.

Looking at Tables 5, we see that, as expected, the unbounded robust model ROB50 achieves 100% feasibility for the simulations DÂ, but this is not true for the bounded robust methods ROB10 and ROBU10. Interestingly, the UF solutions increase in performance for increasing number of constraints in the DÂ simulations. Now, in the case $\hat{A}$ is an underestimation of the noise's magnitude, we see that the performance of the robust models is dramatically decreased. Remarkably, it appears that ROB10 is increasingly better in terms of feasibility than ROB50: in the 10 constraint case with simulations GA, model ROB50 is clearly the worst, with more than 80% of infeasible scenarios. This implies an important fact, namely that focusing uniquely on robustness when using an erroneous uncertainty characterization might actually be worse than using a bounded worst-case approach. In other words, the absolute robustness (the solutions seeking robustness at all costs) should be used as objective if and only the used uncertainty characterization is reliable.

Tables 5 show two additional properties of UFO. The first is that the UFO solutions are able to compete with the robust models even when the robust model disposes of the perfect information, and are clearly able to outperform the robust models when these use an erroneous uncertainty characterization. However, the robust solutions, when disposing of an accurate uncertainty characterization, are much better in terms of optimality gaps than the UF models (ROB50 has an optimality gap of 1.2% in DÂ with 5 constraints instances, whereas, for example, IR_0.25 has an optimality gap of 20.72%). Additionally, when combining the robust solution with an UF, we significantly reduce the sensitivity of the solution to the erroneous uncertainty characterization, although the performance of the solution is decreased in the perfect information cases `DÂ`.

Sensitivity to the budget ratio We report on the influence of the budget ratio on a solution's performance. Figure 1 shows the evolutions of the percentage of infeasible solutions and the UF value for increasing budget ratios, using the values of the 13,800 scenarios. Figure 2 shows the evolution of the optimality gap with respect to the deterministic optimum of each scenario for increasing budget ratio ρ.

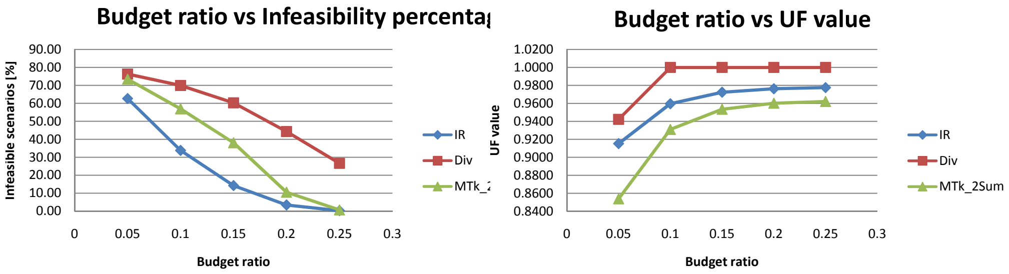

Figure 1: Evolution of the percentage of infeasible scenarios and the UF value for increasing values of the budget ratio ρ on the entire set of 13,800 scenarios.

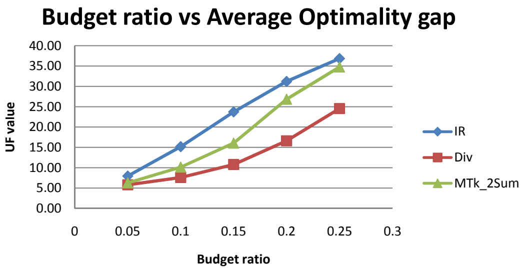

Figure 2: Evolution of the optimality gap between the models' solution and the deterministic optimum $\tilde{z}^*$ of each scenario for increasing values of the budget ratio ρ on the entire set of 13,800 scenarios.

Figure 1 clearly shows the increasing performance of the UF models for increasing budget ρ; however, we also see that the optimality gap increases as shown in Figure 2. Interestingly, the budget ratio and the optimality gap increase at similar rates, which allows to consider the budget ratio as a rough estimator on the average loss of revenue.

Another remarkable fact is that, due to the high degeneration of the near-optimal solutions, the model `DIV` reaches the optimum 1.0 even with a small budget of 0.1. This means that all the solutions with higher budget ratio are actually equivalent, however, the failure in feasibility percentage still decreases. There are two possible explanations for this: either the model `DIV` is not an efficient UF, or the cost-reduction is, in itself, an UF that increases the robustness of the solution. We show in the next paragraph that DIV is improving the robustness of the solution, which means that suboptimality is, in itself, increasing robustness; this observation corresponds to the price of robustness as observed by Bertsimas and Sim (2004).

5.4 UF Validation

Our results show that there is an evident gain in terms of feasibility for some of our UFs. Figure 3 shows the histograms of feasible and infeasible solutions for the different UFs independently for the 10 constraint scenarios: we discretize the UF value interval $[0, 1]$ into 100 intervals, and display, for each interval, the number of scenarios for solutions with UF value within the interval. The histograms show the total number of feasibility tests performed in our experiments, that is a total of 257,600 observations. The infeasible solutions are displayed in black and the feasible ones in white. For visual evidence, we discard the last 5 intervals; Table 6 summarizes, for each UF, the cumulated number of infeasible and feasible scenarios observed in the discarded interval, i.e. in UF values within $[0.96, 1.0]$. Note that no solution has UF value greater than 0.95 for the 2Sum model.

Model	MTK		DIV		IR	
	Feasible	Infeasible	Feasible	Infeasible	Feasible	Infeasible
# observations	27,643	6,653	86,672	26,948	108,737	26,319

Table 6: Cumulated number of feasible and infeasible observations for the UF intervals $[0.95, 1.0]$ for the four UFs.

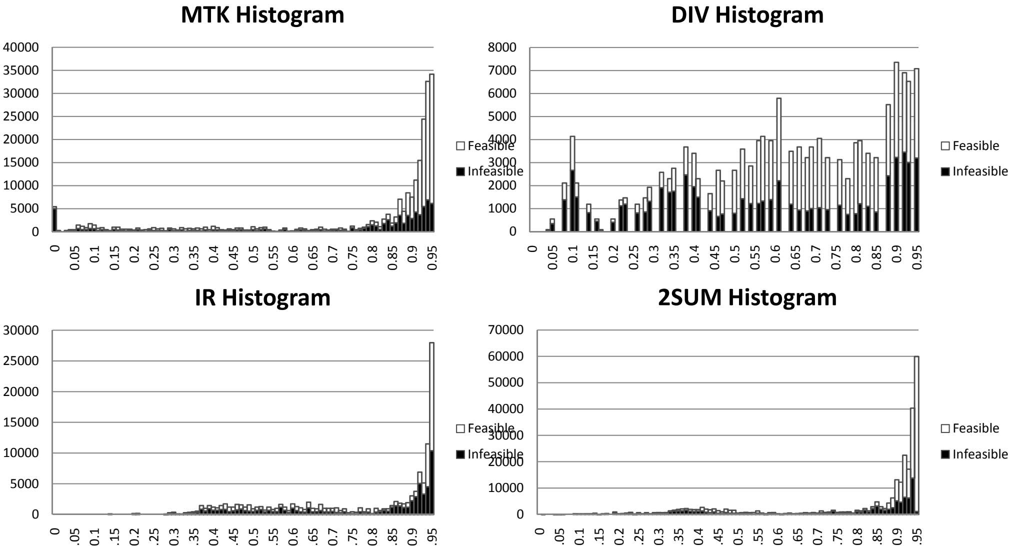

Figure 3: Histogram showing the repartition of feasible and infeasible solutions for the four used UFs.

First, note the large infeasibility peak for MTK for UF value 0. Actually, model DET has always UF value 0, and it might also occur that a solution has a negative value if the most taken object in that model has higher value than the deterministic model (in this case, the UF value is mapped into the interval $[0, 0.01)$). Next note that both number of feasible and number of infeasible solutions grow for large UF values, which is due to the fact that we are optimizing these values with the models. Clearly, the UFs are inter-correlated, as the largest number of observations always occur for the highest UF values. Note that in the DIV histogram, this is not evident. The reason is that we discarded the solutions with value 1.0, i.e. for which all objects are used, and the total number of observations with value 1.0 is over 100,000, i.e. almost 40% of the observations.

Looking at the histograms in Figure 3, our intuition is that the feasibility rate significantly increases for all UFs but the DIV one. In Table 7 we compare the feasibility rates for the cumulative number of observations in UF interval $[0, 0.80]$ and compare it with the value in interval $(0.80, 1.0]$.

Model	MTK		DIV		IR		2SUM	
	≤ 0.8	$(0.8, 1.0]$	≤ 0.8	$(0.8, 1.0]$	≤ 0.8	$(0.8, 1.0]$	≤ 0.8	$(0.8, 1.0]$
# observations	60,216	194,609	100,004	157,596	51,612	205,988	63,756	193,844
feasibility [%]	45.74	72.06	54.95	71.16	47.10	69.32	43.99	59.70

Table 7: Cumulative feasibility rates for the different UFs and the number of observations.

The cumulative feasibility rates show that there is indeed a correlation between the robustness of the solution and the four proposed UFs. However, it appears that some of the UFs, mainly DIV and MTK, are not efficient when used alone. Their combination with another UF or even the robust formulation often increases the performance of each model individually, sometimes in an impressive way, as, for example, the combination of ROB50 and MTK_0.25. The individual is feasible in 64% and 60% of the scenarios, respectively, and when combined, model ROB50_MTK is feasible in 98% of the scenarios.

6 Extension to Airline Scheduling

The Airline Scheduling Problem (ASP) is a complex planning problem involving many regulations, see Kohl et al. (2007) for a survey. The many facets of the problem (route choice, fleet assignment, tail assignment, crew pairing and crew roistering) represent a combinatorial challenge for operations research scientists (Clausen et al., 2009). The additional problem is that the computed schedules have to be carried out in a rapidly varying environment influenced by many factors such as weather, human factors (strikes, illness, . . .) and economical factors. The complexity of the environment makes it extremely difficult, if not impossible, to derive a complete and correct characterization of its behavior.

Being already a hard problem in its deterministic form, ASP is extremely challenging when tackled with proactive methods: it is a good candidate for the UFO framework. This does not hold uniquely for the ASP problem: Fischetti and Monaci (2008) successfully apply light robustness, which is computationally similar to UFO, to the train timetabling problem, showing impressing computational time savings, in addition to competitive solutions in terms of robustness.

As discussed in section 1, some possible uncertainty features to increase robustness of an airline schedule are idle time, plane crossings or number of plane routes matching the worker's union constraints. As it is unlikely that a robust solution exists, it is appealing to search for increased recoverability as well.

In Eggenberg et al. (2008), the authors present a Column Generation (CG) algorithm to solve the Aircraft Recovery Problem. The advantage of the technique is that it is flexible enough to be applied for either aircraft, crew or passenger recovery problems. The algorithm is based on *recovery networks*, encoding each *unit*'s (aircrafts, crew or passengers) feasible route. The performance of the recovery algorithm is directly linked with the structure of the recovery networks. This can be exploited at the ASP phase, using UFs based on the recovery networks' structure in order to increase recoverability.

UFO is a promising framework for computationally hard problems due to uncertain data such as the ASP for two reasons: the first is that, as long as the used UFs are of the same nature than the original objective function, then the computational difficulty is equivalent to solving twice a problem of same difficulty than the initial problem: once to get the lower bound f^* and once to solve the UFO problem (5)-(8). The second reason is that the characterization of uncertainty sets for such problems is a crucial but hard problem that is not required for general UFs.

7 Conclusion

In this paper, we address the problem of optimization of problems prone to noisy data. Unlike most of the existing methods, the Uncertainty Feature Optimization framework does not require the explicit characterization of an uncertainty set, i.e. the possible outcomes of the data: the UFO framework considers the uncertainty implicitly. An UF is *any* feature expected to improve the solution's performance in reality, and it is left to the user to decide the complexity and computational effort to invest in the estimation of the future outcome. Additionally, we demonstrate how to combine different UFs, resulting in a general multi-optimization problem.

We show that existing methods such as stochastic optimization or robust optimization are special cases of UFs, supposing that an uncertainty set is provided. The proof of the generalization for the robust approach of Bertsimas and Sim (2004) leads to an algorithm computing upper bounds on the method's parameters to guarantee a robust solution exists; to our knowledge, the only approach providing such bounds is given by the heuristic method of Fischetti and Monaci (2008).

Computational results on the Multi Dimensional Knapsack Problem (MDKP) show that the UFO approach is competitive against the robust approach. The results show that the UFO approach is more stable to variations in the nature of the noise a problem's data is due to, unlike the robust approach of Bertsimas and Sim (2004): the exact knowledge of the noise's nature is beneficial, but when the nature is erroneously approximated, it might annihilate the method's efficiency. Thus, methods using uncertainty sets should be applied only when the used characterization is sufficiently reliable. Additionally, as show our results, the only knowledge of the noise's nature is not sufficient for the robust approach: the parameters of the method clearly influence the performance of a robust solution. We also observe that the budget ratio used in the UFO framework is a rough estimate of the average solution's optimality gap compared to each scenario's optimum, and that the obtained results are consistent with the principle of the *price of robustness* of Bertsimas and Sim (2004), i.e. that feasibility comes at a certain cost. Furthermore, we demonstrate how to combine different UFs (and even the robust model) using a normalization procedure; the results show that the solutions of the combined models are globally better than the solutions of the individual ones. When combining an UF with robust optimization, we observe that the sensitivity issues related to erroneous uncertainty characterization is reduced. Finally, we show that the UFs are indeed correlated with the solution's robustness, i.e. solution with higher UF values are more robust.

References

Al-Fawzana, M. and Haouari, M. (2005). A bi-objective model for robust resource-constrained project scheduling, *International Journal of Production Economics* **96**: 175187.

Albers, S. (2003). Online algorithms: A survey, *Mathematical Programming* **97**: 3–26. Invited paper at ISMP 2003.

Ben-Tal, A. and Nemirovski, A. (2001). *Lectures on Modern Convex Optimization, Analysis, Algorithma, and Engineering Applications*, MPS-SIAM Series on Optimization.

Bertsimas, D. and Sim, M. (2004). The price of robustness, *Operations Research* **52**: 35–53.

Bian, F., Burke, E., Jain, S., Kendall, G., Koole, G., Mulder, J. L. S. J., Paelinck, M., Reeves, C. and Suleman, I. R. M. (2005). Measuring the robustness of airline fleet schedules, *Multidisciplinary Scheduling: Theory and Applications*, Springer US, chapter 10.1, pp. 381–392.

Birge, J. R. and Louveaux, F. (1997). *Introduction to Stochastic Programming*, Springer.

Clausen, J., Larsen, A. and Larsen, J. (2009). Disruption management in the airline industry – review of models and methods, *Computers & Operations Research* **doi:10.1016/j.cor.2009.03.027.**

Eggenberg, N., Salani, M. and Bierlaire, M. (2008). Constraint specific recovery networks for solving airline recovery problems, *Technical Report TRANSP-OR 080828*, Ecole Polytechnique Fdrale de Lausanne, Switzerland.

Fischetti, M. and Monaci, M. (2008). Light robustness, *Technical report*, DEI, Universit di Padova, Italy.
URL: *http://www.dei.unipd.it/~fisch/papers/light_robustness.pdf*

Herroelen, W. and Leus, R. (2005). Project scheduling under uncertainty: Survey and research potentialy, *EJOR* **165**: 289–306.

Kall, P. and Mayer, J. (2005). *Stochastic Linear Programming, Models, Theory and Computation*, Springer.

Kall, P. and Wallace, S. (eds) (1994). *Stochastic Programming*, John Wiley & Sons, New York, N.Y.

Klabjan, D., Johnson, E., Nemhauser, G., Gelman, E. and Ramaswamy, S. (2002). Airline crew scheduling with time windows and plane-count constraints, *Transportation Science* **36**(3): 337–348.

Kohl, N., Larsen, A., Larsen, J., Ross, A. and Tiourine, S. (2007). Airline disruption management - perspectives, experiences and outlook, *Journal of Air Transport Management* **13**(3): 149–162.

Lan, S., Clarke, J.-P. and Barnhart, C. (2006). Planning for robust airline operations: Optimizing aircraft routings and flight departure times to minimize passenger disruptions, *Transportation Science* **40**: 15–28.

Linderoth, J., Shapiro, A. and Wright, S. (2006). The empirical behavior of sampling methods for stochastic programming, *Annals of Operations Research* **142**(1): 215–241.

Pisinger, D. (1995). An expanding-core algorithm for the exact 0-1 knapsack problem, *EJOR* **87**: 175–187.

Policella, N. (2004). *Robust Scheduling: Analysis and Synthesis of Flexible Solutions*, PhD thesis, Universita di Roma "la Sapienza".

Rosenberger, J., Johnson, E. and Nemhauser, G. (2004). A robust fleet assignment model with hub isolation and short cycles, *Transportation Science* **38**(3): 357–368.

Sahinidis, N. V. (2004). Optimization under uncertainty: state-of-the-art and opportunities, *Computers and Chemical Engineering* **28**: 971–983.

Shebalov, S. and Klabjan, D. (2006). Robust airline scheduling: Move-up crews, *Tranportation Science* **40**(3): 300–312.

Soyster, A. (1973). Convex programming with set-inclusive constraints and application to inexact linear programming, *Operations Research* **21**(5): 1154–1157.

Yen, J. W. and Brige, J. R. (2006). A stochastic programming approach to the airline crew scheduling problem, *Transportation Science* **40**: 3–14.

A Complementarity Theorem

Theorem (Complementarity)
Let

$$\beta_i(x,\Gamma_i) = \max_{\{S_i\cup\{t_i\}|S_i\subseteq J_i,|S_i|=\lfloor\Gamma_i\rfloor,t_i\in J_i\setminus S_i\}} \left\{ \sum_{j\in S_i} \hat{a}_{ij} \mid x_j \mid + (\Gamma_i - \lfloor\Gamma_i\rfloor)\hat{a}_{it_i} \mid x_{t_i} \mid \right\},$$

and

$$\overline{\beta}_i(\mathbf{x}, \Gamma_i) = \min_{\{S_i \cup \{t_i\} | S_i \subseteq J_i, |S_i| = \lfloor J_i - \Gamma_i \rfloor, t_i \in J_i \setminus S_i\}} \left\{ \sum_{j \in S_i} \hat{a}_{ij} \mid x_j \mid + (J_i - \Gamma_i - \lfloor J_i - \Gamma_i \rfloor) \hat{a}_{it_i} \mid x_{t_i} \mid \right\}.$$

Then the following relation holds:

$$\beta_i(\mathbf{x}, J_i) = \overline{\beta}_i(\mathbf{x}, \Gamma_i) + \beta_i(\mathbf{x}, \Gamma_i).$$

Proof:

For a fixed vector $\mathbf{x}$ and a fixed constraint $i \in \{1, \dots, n\}$, let $S_i^* \cup \{t_i^*\}$ be the optimal set maximizing $\beta_i(\mathbf{x}, \Gamma_i)$ and $\overline{S}_i^* \cup \{\overline{t}_i^*\}$ the optimal set minimizing $\overline{\beta}_i(\mathbf{x}, \Gamma_i)$.

We assume, w.l.o.g. that the $|J_i|$ changing coefficients are ordered with respect to $\hat{a}_{ij} \mid x_j \mid$. Then, the $\lfloor J_i - \Gamma_i \rfloor$ first ones are in $\overline{S}_i^*$ and, similarly, the $\lfloor \Gamma_i \rfloor$ last ones are in S_i^* and $\overline{S}_i^* \cap S_i^* = \emptyset$.

When Γ_i is integer, then $\overline{S}_i^* \cup S_i^* = J_i$ and the theorem is trivially proved. We thus assume that Γ_i is non-integer. In this case, $\overline{S}_i^* \cap S_i^* = \emptyset$ implies that both indexes of the coefficients considered only as fractionally varying are the same, i.e. $\overline{t}_i^* = t_i^*$; the fractionally varying coefficient in both cases is the one at position $\lfloor J_i - \Gamma_i \rfloor + 1$. Additionally, as Γ_i is non-integer, the following holds:

$$\lfloor J_i - \Gamma_i \rfloor = J_i - \lfloor \Gamma_i \rfloor - 1$$

Let us sum all terms of $\overline{S}_i^*$ and S_i^* using the previous equality and the fact that $\overline{t}_i^* = t_i^*$ and $\overline{S}_i^* \cap S_i^* = \emptyset$:

$$\begin{aligned}
\overline{\beta}_i(\mathbf{x}, \Gamma_i) + \beta_i(\mathbf{x}, \Gamma_i) &= \left(\sum_{j \in \overline{S}_i^*} \hat{a}_{ij} \mid x_j \mid \right) + (J_i - \Gamma_i - \lfloor J_i - \Gamma_i \rfloor) \hat{a}_{i\overline{t}_i^*} \mid x_{\overline{t}_i^*} \mid \\
&\quad + \left(\sum_{j \in S_i^*} \hat{a}_{ij} \mid x_j \mid \right) + (\Gamma_i - \lfloor \Gamma_i \rfloor) \hat{a}_{it_i^*} \mid x_{t_i^*} \mid \\
&= \left(\sum_{j \in J_i, j \neq t_i^*} \hat{a}_{ij} \mid x_j \mid \right) + (J_i - \Gamma_i - \lfloor J_i - \Gamma_i \rfloor + \Gamma_i - \lfloor \Gamma_i \rfloor) \hat{a}_{it_i^*} \mid x_{t_i^*} \mid \\
&= \left(\sum_{j \in J_i, j \neq t_i^*} \hat{a}_{ij} \mid x_j \mid \right) + (J_i - \Gamma_i - (J_i - \lfloor \Gamma_i \rfloor - 1) + \Gamma_i - \lfloor \Gamma_i \rfloor) \hat{a}_{it_i^*} \mid x_{t_i^*} \mid \\
&= \left(\sum_{j \in J_i, j \neq t_i^*} \hat{a}_{ij} \mid x_j \mid \right) + \hat{a}_{it_i^*} \mid x_{t_i^*} \mid \\
&= \sum_{j \in J_i} \hat{a}_{ij} \mid x_j \mid \\
&= \beta_i(\mathbf{x}, J_i)
\end{aligned}$$

□

B Convergence Proof

Theorem (Convergence)
Consider the iterative process defined in section 4. With the provided notation, consider the following algorithm:

```
1) set $\Gamma_i^{(0)} = J_i$;
2) Solve the problem finding the value of $z_F^{(k)^*}$;
3) IF $z_F^{(k)^*} \leq 0$ STOP:
   there exists a robust solution for the given sets of $\Gamma_i^{(k)}$, $i = 1, \ldots, n$;
4) at iteration $k$, let $i^* \in \{1, \ldots, n\}$ be the index of a function $g_i(\mathbf{x}_k^*)$
   with value $z_F^{(k)^*}$, then:
   find $\Gamma_{i^*}^{(k)} = \sup\{0 \leq \Gamma \leq \Gamma_{i^*}^{(k-1)} \mid \overline{\beta}(\mathbf{x}_k^*, \Gamma) \geq z_F^{(k)^*}\}$;
5) IF no such $\Gamma$ exists STOP: the set $X = \emptyset$;
   ELSE set $k = k + 1$, and go back to 2).
```

The proposed algorithm converges in a finite number of iterations.

Proof:

From the complementarity theorem, we get that $\overline{\beta}_i(\mathbf{x}, \Gamma)$ is a decreasing function for increasing Γ (using the theorem with $\beta_i(\mathbf{x}_{\underline{\mathbf{k}}}^*, \Gamma)$ being an increasing function for increasing Γ). Moreover, $\beta_i(\mathbf{x}, J_i) = \overline{\beta}_i(\mathbf{x}, 0)$, as by definition $\beta_i(\mathbf{x}, 0) = 0$.

We assume that the algorithm did not converge after iteration $k-1$, i.e. that $z_F^{(k-1)^*} > 0$, and $i^* \in \{1, \ldots, n\}$ is an index such that $g_{i^*}(\mathbf{x}_k^*) = z_F^{(k)^*}$ and at least $\Gamma_{i^*}^{(k-1)} > 0$.

If no solution exists for

$$\overline{\beta}_{i^*}(\mathbf{x}_k^*, \Gamma) \geq z_F^{(k)^*}, 0 \leq \Gamma \leq \Gamma_{i^*}^{(k-1)},$$

this holds in particular for the largest possible value of the left-hand-side, obtained with $\Gamma = 0$ as $\overline{\beta}_{i^*}(\mathbf{x}_k^*, \Gamma)$ decreases for increasing Γ. Therefore, $\overline{\beta}_{i^*}(\mathbf{x}_{\mathbf{k}}^*, 0) < z_F^{(k)^*}$, which implies that

$$\beta_{i^*}(\mathbf{x}_{\mathbf{k}}^*, J_{i^*}) < z_F^{(k)^*} = \sum_{j=1}^{m} a_{i^*j}(x_j)_k^* + \beta_{i^*}(\mathbf{x}_{\mathbf{k}}^*, J_{i^*}) - b_{i^*}.$$

Reordering the previous result, we obtain

$$\sum_{j=1}^{m} a_{i^*j}(x_j)_k^* > b_{i^*},$$

which contradicts $\mathbf{x} \in X$, since X is defined by constraints $A\mathbf{x} \leq \mathbf{b}$, and we proved that $X = \emptyset$.

Let us now prove that the sequence is not stationary by contradiction: We assume that that $\Gamma_i^{(k)} = \Gamma_i^{(k-1)}$ for all i: in particular, this is also true for i^*. As $\Gamma_{i^*}^{(k)}$ is obtained by $\Gamma_{i^*}^{(k)} = \sup\{0 \leq \Gamma \leq \Gamma_{i^*}^{(k-1)} \mid \overline{\beta}(\mathbf{x}_k^*, \Gamma) \geq z_F^{(k)^*}\}$, then the following holds:

$$\overline{\beta}_{i^*}(\mathbf{x}_k^*, \Gamma_{i^*}^{(k)}) = \overline{\beta}_{i^*}(\mathbf{x}_k^*, \Gamma_{i^*}^{(k-1)}) \geq g_{i^*}(\mathbf{x}_k^*),$$

and by reordering the previous inequality:

$$\sum_{j=1}^{m} a_{i^*j}(x_j)^*_k + \beta_{i^*}(\mathbf{x}^*_k, J_{i^*}) - \overline{\beta}_{i^*}(\mathbf{x}^*_k, \Gamma^{(k)}_{i^*}) \le b_{i^*}.$$

Using the complementarity problem, this leads to

$$\sum_{j=1}^{m} a_{i^*j}(x_j)^*_k + \beta_{i^*}(\mathbf{x}^*_k, \Gamma^{(k)}_{i^*}) \le b_{i^*},$$

i.e. the solution is robust for $\Gamma^{(k)}_i$ simultaneously changing coefficients and $z_F^{(k)^*} \le 0$, and the iterative process converged.

For a non stationary solution, $\Gamma^{(k)}_i < \Gamma^{(k-1)}_i$ for at least $i = i^*$. Moreover, we know that a solution of $\Gamma^{(k)}_{i^*}$ exists, otherwise we would have proved that $X = \emptyset$.

Finally, at iteration $k+1$, all functions satisfy $f_i(\mathbf{x}) \le \beta_{i^*}(\mathbf{x}^*_k, \Gamma^{(k)}_{i^*})$, for all $\mathbf{x} \in X$, the inequality being strict at least for $i = i^*$. The values of $z_F^{(k)^*}$ are thus strictly decreasing as well, for increasing k.

We proved that the method eventually converges either to a solution with $z_F^{(k)^*} \le 0$, or we obtain $\Gamma^{(k)}_i = 0$ for all $i = 1, \ldots, n$, meaning no solution for $A\mathbf{x} \le \mathbf{b}$ exists.

□