Operational Research (2021) 21:2459–2487
https://doi.org/10.1007/s12351-019-00512-1

ORIGINAL PAPER

Multi-objective optimisation under deep uncertainty

Babooshka Shavazipour[1] · Theodor J. Stewart[1]

Received: 13 December 2018 / Revised: 17 July 2019 / Accepted: 31 July 2019 /
Published online: 14 September 2019

Abstract

This paper presents a scenario-based Multi-Objective structure to handle decision problems under deep uncertainty. Most of the decisions in real-life problems need to be made in the absence of complete knowledge about the consequences of the decision and/or are characterised by uncertainties about the future which is unpredictable. These uncertainties are almost impossible to reduce by gathering more information and are not statistical in nature. Therefore, classical probability-based approaches, such as stochastic programming, do not address these problems; as they require a correctly-defined complete sample space, strong assumptions (e.g. normality), or both. The proposed method extends the concept of two-stage stochastic programming with recourse to address the capability of dealing with deep uncertainty through the use of scenario planning rather than statistical expectation. In this research, scenarios are used as a dimension of preference to avoid problems relating to the assessment and use of probabilities under deep uncertainty. Such scenario-based thinking involved a multi-objective representation of performance under different future conditions as an alternative to expectation. To the best of our knowledge, this is the first attempt of performing a multi-criteria evaluation under deep uncertainty through a structured optimisation model. The proposed structure replacing probabilities (in dynamic systems with deep uncertainties) by aspirations within a goal programming structure. In fact, this paper also proposes an extension of the goal programming paradigm to deal with deep uncertainty. Furthermore, we will explain how this structure can be modelled, implemented, and solved by Goal Programming using some simple, but not trivial, examples. Further discussion and comparisons with some popular existing methods will also provided to highlight the superiorities of the proposed structure.

Keywords Deep uncertainty · Goal programming · Multi-objective optimisation · MCDM · Scenario planning · Dynamic-robust optimisation

✉ Babooshka Shavazipour
b.shavazipour@gmail.com

[1] Department of Statistical Sciences, University of Cape Town, Rondebosch, Cape Town 7701, South Africa

1 Introduction

In most of the real-world optimisation problems, decision- or policy-makers are faced with several (possibly conflicting) criteria/objectives. Satisfying these conflicting criteria/objectives is not simply possible and may need an additional decision analysis/support process. Multi-Criteria Decision Analysis (MCDA) is mainly designed to help and support the Decision Makers (DMs) to find a more robust alternative/solution, while taking into account these multiple (conflicting) criteria, by evaluating the performances of each alternative in terms of different criteria (see, for example, Belton and Stewart 2002; Goodwin et al. 2004 for more details).

Moreover, on many occasions, we have to make a decision and choose the one alternative while we do not have enough information or knowledge about the consequences of our decision or we may not even know about all other options. Many other unpredictable variables and/or events could affect the outcomes of the decisions (such as policy-makers' and stakeholders' risk-taking, habits, relations, behaviours, religious beliefs, previous decisions, power, and money). This uncertainty also will be more complex when it is related to the future.

Uncertainty can be studied from different dimensions such as nature, location and degrees/levels of uncertainty (Walker et al. 2003). Among them, the where the uncertainty can appear, i.e. the depth or level of uncertainty plays a key role in handling uncertainty and, therefore, different definitions and classifications have been proposed for various levels/degrees of uncertainty (see for example Knight 1921; Quade 1989; Morgan et al. 1992; Courtney 2001; Kwakkel et al. 2010; Walker et al. 2003, 2013). Knight (1921) were the first researchers to make the distinction that the calculable and controllable part of the unknown is referred as *risk*, while the incalculable and uncontrollable part of the unknown refers to *uncertainty*. Generally, statistical methods and/or probabilities have been utilised and developed by many authors to handle the controllable and calculable parts (Quade 1989; Morgan et al. 1992), while the latter part is often ignored for many years. The term *deep uncertainty*, which refers to the latter part, has been used recently by several authors in various contexts, especially in climate change studies. However, to the best of our knowledge, there is not any particular definition or guideline with which we can classify and model the optimisation problems under uncertainty. To reach this goal, in this paper, we classified the depth of uncertainty based on a suitable definition which could be used as a guideline in optimisation problems under uncertainty.

The primary motivation for writing this paper was that we were concerned that many optimisation models do not treat the higher degrees of uncertainty and risk observing many real-world problems, especially, with long-term decision-making processes such as strategic planning problems. These uncertainties are almost impossible to reduce by gathering more information and are not statistical in nature. Therefore, there is no statistical (probability-based) approach nor predictive modelling that can help to deal with this kind of uncertainty. In this paper, we were exclusively interested in the multi-objective optimisation problems under deep uncertainty. In dealing with such a problem, a decision is not only one that can seize different conflicting objectives; it also relies sustainable planning that should be *robust* and *adaptable*. This means that the decision should perform satisfactorily under a broad variety of futures (it is robust) and it can be adapted to changing variable future conditions (Haasnoot et al. 2011). Most of the early works and applications in Multi-Criteria Decision-Making(MCDM)/Multi-Objective Optimisatio(MOO) have ignored deep uncertainty because of its complexity. Many of these studies have been limited to lower levels of uncertainty by assuming some probability distributions (mostly the Normal distribution). To the best of our knowledge, none of the existing methodologies in MCDM/MOO literature deal substantially with deep uncertainty. Therefore, the primary aim of this study is to address this important gap in MOO and provide a way forward for these complex problems. In fact, this is the first attempt of performing a multi-criteria evaluation under deep uncertainty through a structured optimisation model.

Under conditions of the deep uncertainty, qualitative components in complex problems are incompletely understood and potential outcomes not enumerable. In such situations, it becomes useful to utilise the concept of scenarios as representations of coherent futures, as a framework for thought and critical conversation (Durbach and Stewart 2012). Therefore, utilising the Scenario Planning (SP) methodology is meaningful and also helpful to treat these kinds of uncertainty. The term *scenario* is used in this study as "logical descriptions of plausible futures in which the outcomes of decisions will emerge".

In this research, we aimed to introduce a two-stage scenario-based structure to dealing with deep uncertainty in MCDM/MOO problems. The main idea was to extend the concept of two-stage stochastic programming with recourse to address the capability of dealing with deep uncertainty through the use of scenario planning rather than statistical expectation. Such scenario-based thinking involved a multi-objective representation of performance under different future conditions as an alternative to expectation, which fitted naturally into the broader multi-objective problem context.

Therefore, we first define and classify different levels of uncertainty which could be utilised in optimisation problems under uncertainty in Sect. 2. Then, after a quick review of some basic concepts of Goal Programming and its regular models under conditions of certain and stochastic parameters in Sect. 3; the proposed *dynamic-robust multi-objective optimisation structure* which could pave the way for us to handle deep uncertainty in complex long-term strategic planning is introduced and formulated in Sect. 4. Section 5 applied the Wierzbicki reference point model to solve the linear form of the proposed two-stage model. In the Following Sect. 6, a simple farming example is presented and followed by discussion on results, contribute to planning and understanding. Section 7 utilised another example to further illustrate the proposed two-stage approach. This section also provided a comparison between the proposed framework and popular existing methods, such as single-scenario optimisation and robust optimisation. It further highlighted some superiorities of the proposed framework. Finally, this study will be summarised and concluded in Sect. 8.

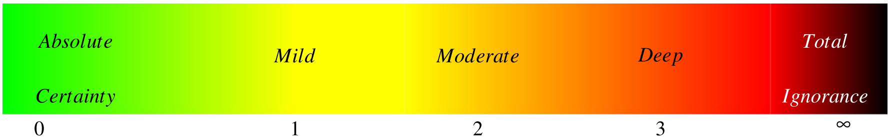

Fig. 1 Different degrees of uncertainty

2 Deep uncertainty

The term deep uncertainty has been used recently by several authors in various contexts, especially in climate change studies. For example, Bankes (2002) uses the term deep uncertainty to explain the phenomena that could not well be modelled by the probability and statistic's tools.

We use the term *degree of uncertainty* as depth or level of uncertainty and define uncertainty of degree **'0'** as absolute certainty or deterministic knowledge, and uncertainty of **infinity degree** as total ignorance. We also propose three major levels of uncertainty: the **mild** uncertainty with an uncertainty of **first-degree**, the **moderate** uncertainty with an uncertainty of **second-degree**, and **deep** uncertainty which can be labelled as the **third-degree** of uncertainty. Figure 1 shows how they stand on the spectrum while the definitions of these three intermediate levels have been described as follows:

- **Mild** uncertainty (First-degree): Outcomes can be enumerated and probabilities (or probability distribution) are specified.
- **Moderate** uncertainty (Second-degree): Outcomes can be enumerated but probabilities (or probability distribution) are difficult to specify generally.
- **Deep** uncertainty (Third-degree): Outcomes cannot be completely enumerated, so that, probabilities are not definable.

When different degrees of uncertainty are simultaneously observed in a problem/model, the highest degree of existing uncertainty determines the degree of uncertainty in that problem/model.

3 Goal programming background

In the literature dedicated to MCDM, the Goal Programming (GP) approach, developed by Charnes et al. (1955) and Charnes and Cooper (1959), is one of the most popular and oldest techniques. The GP model is a distance function; it can be a vector or a weighted sum dependent on the goal programming variant used, where the deviations (δ_i) between the achievement and aspiration levels are to be minimised. The most generic form of the GP model is as follows (Jones and Tamiz 2010):

$$\begin{array}{lll} Min & \phi = h(\delta^-, \delta^+) & \\ s.t. & z_i(\mathbf{x}) + \delta_i^- - \delta_i^+ = g_i, & i = 1, \dots, m; \\ & \mathbf{x} \in \mathbf{X}, & \\ & \delta_i^-, \delta_i^+ \geq 0, & i = 1, \dots, m. \end{array} \tag{1}$$

where δ_i^- and δ_i^+ describe the negative and the positive deviations[1] with respect to the aspiration level (goals) g_i, respectively. $\mathbf{x} = \{x_1, \dots, x_n\}$ denote the decision variables, $z_i(\mathbf{x})$ represents the ith objective function and $\mathbf{x} \in \mathbf{X}$ are the initial problem constraints which indicate the feasible region of our problem.

Since then, GP has been frequently developed and applied by many authors in a wide variety of real-world applications such as financial management, human resources, marketing, agriculture, transport, quality control, allocation problems, and production. More information about the GP and its applications can be found in Tamiz et al. (1998), Jones and Tamiz (2002, 2010, 2016) and Aouni et al. (2014).

The literature on goal programming under uncertainty is less popular than non-stochastic GP but several approaches could be found in the literature which developed GP models in an uncertain environment. To contrast the GP model with uncertain parameters, it is sufficient to consider model 1 with some uncertain parameters in objective and/or constraints (e.g. Stochastic Goal Programming (SGP), Scenario-Based GP, Fuzzy GP, Robust GP, and Dynamic GP). We believe that the limitation of GP to modelling uncertainty relating to future states of the world can be rectified by scenario planning. Integrating the scenario planning and goal programming, if well structured and defined, could be helpful in this complex multi-criteria decision-making problems. Furthermore, amongst different kinds of SGP, the SGP with recourse seems able to match with the concept of dynamic robustness and deep uncertainty if we can find a way to avoid using the probabilities and expectations in Stochastic Programming. A simple framework for thinking about decision making with future uncertainties is suggested by that of *stochastic programming with recourse*, first introduced for stochastic linear programming (Beale 1955; Dantzig 1955, 1963; Dantzig and Madansky 1961). The framework allows certain "*here-and-now*" decisions to be made before states or parameters are fully known, after which *corrective actions*, subject to *penalties* will need to be taken to adjust for shortfalls or deviations from goals in our context. The basic formulation of a two-stage stochastic linear programming problem can be found in Shapiro et al. (2009). More information about the stochastic programming, its stability and solving approaches can be found in Van Slyke and Wets (1969), Dupa cová (1986), Martinez and Aguado (1998) and Shapiro and Homem-de Mello (1998). Moreover, several applications of Stochastic Programming with recourse can be found in the literature such as petroleum refinery planning (Khor et al. 2008), transportation planning

[1] In most cases, a one-sided deviation may be important in which the positive deviations in Maximization and the negative deviations in Minimization problems would be an advantage. However, for general formulation here, the two-side deviation is considered and both positive and negative deviations are undesirable.

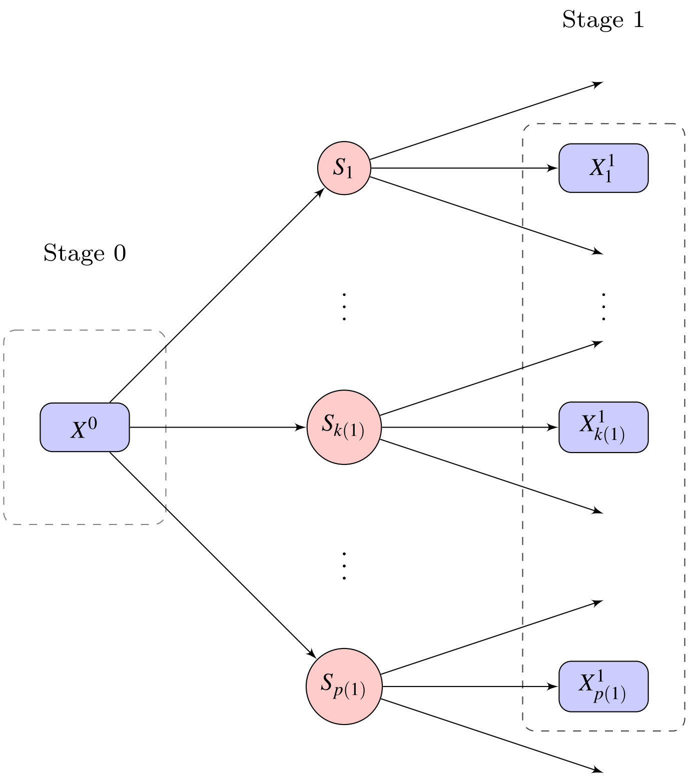

Fig. 2 Two-stage decision-making process with p scenarios

(Barbarosoglu and Arda 2004), hospital bed planning Ben Abdelaziz and Masmoudi (2012), and portfolio selection Masmoudi and Abdelaziz (2012).

Overall, although it seems that Stochastic Goal Programming with recourse, gives the DM an opportunity to react in an intelligent/optimal way. We shall now argue that this basic framework can be applied to deeper levels of uncertainily, provided that alternatives to statistical expectation can be found. To the best of our knowledge, no such approach has been proposed in the literature and we thus develop a structure based on integrating scenario planning and goal programming.

4 Proposed two-stage approach

The simplest recourse model is two-stage (i.e. a set of recourse variables to be applied after that scenario is fully revealed only). Consider a two-stage decision-making process as shown in Fig. 2. Suppose that the decision maker (DM) makes and implements an initial decision ($x^0 \in \mathbf{X}^0$, $\mathbf{X}^0$ is a feasible set of the initial decisions) at stage '0' which is a scenario free decision. Then we will wait and see what happens in the future. Suppose p scenarios ($s_k \in \{s_1, \dots, s_p\}$) are defined to describe the plausible future space. The subsequent decision will be taken by DM, as a recourse, if scenario k occurs at stage '1' ($x_k^1 \in \mathbf{X}^1(\mathbf{x}^0, k), k = 1, \dots, p;$). Therefore, p contingent decisions will be identified, one for each scenario. The union of the initial decision and the relevant contingent decision will constitute the full decision of the problem. In the proposed structure, we suggest p different sets of decisions—one for each plausible scenario. However, only one contingent decision set will be implemented after the realisation of the scenario.

The primary distinction between the above structure and that of SGP is that no expectation operator is invoked. Robustness will be besought treating goal achievements in terms of each criterion-scenario combination as separate goals to be explained in the next section. We may need to emphasis that, in deep uncertainty, scenarios are a representation of plausible futures but not, necessarily, complete specifications (or even approximations often) of all plausible futures (Goodwin and Wright 2001; Stewart 1997). They do not constitute a complete probability space so that probabilities are arguable. Therefore, under these conditions, the standard form of the decision trees, with the attached probabilities to the possible states and calculable expected outcomes, cannot be utilized to identify the best alternatives (Clemen and Reilly 2013).

In the proposed two-stage structure, we optimise the aggregation of the initial decision (x^0) and dependent (recourse) decisions ($x_k^1, k = 1, \dots, p$) which should be defined for each and every plausible scenario. In fact, we are looking for a robust initial decision which does not need to be feasible for every single scenario, but the aggregation of the initial decision (x^0) and the subsequent scenario-dependent decision ($x_k^1, k = 1, \dots, p$) must be feasible for every scenario k ($k = 1, \dots, p$). In other words, DM makes an initial scenario-free decision. Then, after scenario revelation, he/she has a chance to adapt his/her decision, if it is necessary, by a subsequent scenario-dependent decision (recourse) while may well pay some penalties for any deviation from initial goals. In this structure, we do not utilise models to produce forecasts but try instead to identify the reasonable initial decision (which is compensable, whatever happens in the future) and a suitable contingent scenario-dependent decision for every single plausible scenario, implemented after scenario revelation. This process can be summarised as follows:

- **Stage 0**: An **initial decision** is made before any scenario revelation (Scenario free decision).
- **Stage 1**: A **contingent decision** to be taken if the scenario k is revealed (Scenario dependent decision).

4.1 Mathematical formulation

As mentioned earlier, in the realm of multi-criteria decision making, decisions in a scenario-based model needs to be evaluated in terms of all criteria under conditions of each scenario as a dimension of preferences. Stewart et al. (2013) introduced the term "meta-criteria" and used this to assess and compare different alternatives regarding a two-dimensional ($m \times p$) array of performance measures (all criterion-scenario combinations). Moreover, to avoid challenges of considering probabilities under deep uncertainty, we utilise scenarios as a dimension of preference (a component of the meta-criteria) and, to some extent, extend the meta-criteria concept to find a robust decision and adopt a scenario-based multi-criteria structure under deep uncertainty that can provide us with a robust decision.

Moreover, the decisions/solutions need to be evaluated and compared (on the basis of performance) in terms of each criterion under conditions of an uncertain scenario. Therefore, the main purpose of this section is to introduce a model to find the best possible decision/solution by which the performance measures regarding each criterion i ($i = 1, \dots, m$) under conditions of an uncertain scenario k ($k = 1, \dots, p$) must be optimised. These performance measures are now modelled as objective functions/meta-criteria which represent different dimension of preferences. In other words, each objective function/meta-criterion represents preferences in terms of a criterion under conditions of a scenario. In fact, each meta-criterion (performance measure under a particular scenario) is treated as an objective or dimension of preference.

In the proposed two-stage structure, we are looking one step ahead, postpone part of the decision and leave room for possible adaptation later (after scenario realisation). We identified suitable adaptive plans for every plausible scenario in advance. A robust decision in our philosophy (let us call it a *dynamic-robust decision*) is a split decision containing two subgroups of decisions: the initial decision(s) followed by recourse decision(s), in which the full decision is robust across scenarios. Unlike the regular robust solution, the initial decision does not need to be optimal or even feasible under conditions of all scenarios.[2] Instead, it must be good enough, with beneficial foresight, to lead us to the optimal aggregations of this initial decision(s) and the following recourse decisions.

As mentioned in the previous section, we need to formulate an optimisation model that evaluates the overall performances for all m criteria in the conditions of all p scenarios. Each overall performance must include the performances of the initial decision (x^0) and one recourse decision ($x_k^1, k = 1, \dots, p$) related to scenario k. The initial decision is common amongst all scenarios while recourse decisions particularly identify for a relevant scenario. In other words, by applying the concept of meta-criteria, we are seeking an aggregation of the above decisions that provides us with the best performance measure in all $m \times p$ meta-criteria. Therefore, a multi-criteria two-stage model includes optimisation of the $m \times p$ meta-criteria/objectives under conditions of some problem constraints. Hence, generally, the two-stage multi-objective optimisation problem under uncertainty, in general form, can be formulated as follows:

$$
\begin{array}{lll}
Opt_{(\mathbf{x}^0, \mathbf{x}_k^1)} & \mathbf{F} = f_{ik}^1(\mathbf{x}^0, \mathbf{x}_k^1); & k = 1, \dots, p; \quad i = 1, \dots, m; \\
s.t. & u_r^0(\mathbf{x}^0) \le 0, & r = 1, \dots, R_0; \\
 & u_r^1(\mathbf{x}^0, \mathbf{x}_k^1) \le 0, & k = 1, \dots, p; \\
 & & r = R_0 + 1, \dots, R_0 + \cdots + R_{k-1} + R_k;
\end{array}
\tag{2}
$$

[2] In other words, the intersection of the constraints of all scenarios can be empty because there will be a contingent decision in the next stage and the aggregated decision will describe the outcomes, not only the first initial decision (see Example 7.2 for further illustration).

where $\mathbf{x}^0 = (x_1^0, \ldots, x_n^0) \in \mathbf{X}^0$ is an n-dimensional initial scenario-free decision variable vector which was made in stage '0' before scenario k happens and $\mathbf{X}_0$ is an initial decision space.

$\mathbf{x}_k^1 = (x_{1k}^1, \ldots, x_{nk}^1) \in \mathbf{X}^1(\mathbf{x}^0, k), (k = 1, \ldots, p)$; is an n-dimensional contingent scenario-dependent decision vector which is taken in stage '1' if scenario k is revealed and $\mathbf{X}^1(\mathbf{x}^0, k)$ is a contingent decision space when scenario k is unfolded.

$f_{ik}^1(\mathbf{x}^0, \mathbf{x}_k^1), (k = 1, \ldots, p; i = 1, \ldots, m)$, is ith meta-criterion/objective includes scenario-free performances, if scenario k is revealed. In fact, they indicate *preferences regarding criterion i under conditions of scenario k.*

$u_r^0(\mathbf{x}^0)$ is the set of inequality constraints in stage '0'.

$u_r^1(\mathbf{x}^0, \mathbf{x}_k^1)$ is the set of inequality constraints in stage '1'.

The problem consists in optimising $m \times p$ objectives under $(R_0 + R_1 + \cdots + R_p)$ constraints. Both objectives and constraints can be linear or non-linear. However, this paper focuses on the linear problems.

Moreover, to make the model more general, some initial objectives could be considered for stage '0' (i.e. before any uncertainty$(f_{i_0}^0(\mathbf{x}^0), (i_0 = 1, \ldots, m_0))$. In this case, the model contains optimisation of $\mathbf{F} = [f_{i_0}^0(\mathbf{x}^0); f_{ik}^1(\mathbf{x}^0, \mathbf{x}_k^1)]$. Then, the total number of objectives equals to $m_0 + (m \times p)$ (if $m_0 = m$, then the total is equal to $m \times (p + 1)$).

It is also possible that a different number of objectives is considered in each scenario which could lead us to model the higher levels of uncertainty. In this situation, the problem consists in Optimising $(m_0 + (m_1 + \cdots + m_p))$ objectives.

4.1.1 Two-stage multi-objective linear programming (MOLP)

Without loss of generality, It can be assumed that the objectives are to be minimised (maximisation of $\mathbf{f(x)}$ is equivalent to minimising $-\mathbf{f}(\mathbf{x})$). If the objectives and constraints are linear and each decision evaluated on m criteria denoted by $C = \{C_1, \ldots, C_m\}$, problem 2 is a Multi-Objective Linear Programming (MOLP) with $m \times p$ objectives which can be formulated as follows:

$$
\begin{aligned}
&Min\ Z_{ik} = \sum_{j=1}^{n} c_{ij}^0 x_j^0 + \left(\sum_{j=1}^{n} c_{ijk}^0 x_j^0 + \sum_{j=1}^{n} c_{ijk}^1 x_{jk}^1 \right); && i = 1, \ldots, m; k = 1, \ldots, p; \\
&s.t. \quad \sum_{j=1}^{n} a_{rj}^0 x_j^0 \le b_r^0, && r = 1, \ldots, R_0; \\
&\qquad \sum_{j=1}^{n} a_{rjk}^0 x_j^0 + \sum_{j=1}^{n} a_{rjk}^1 x_{jk}^1 \le b_{rk}^1, && k = 1, \ldots, p; \\
& && r = R_0 + 1, \ldots, R_0 + \cdots + R_k; \\
&\qquad x_j^0, x_{jk}^1 \ge 0. && j = 1, \ldots, n;\ k = 1, \ldots, p.
\end{aligned}
\tag{3}
$$

where Z_{ik} describe the ith linear meta-criterion/objective function representing *preferences in terms of criterion i under conditions pertaining to scenario k*, and includes some initial goals (scenario-free) before any uncertainties $(\sum_{j=1}^{n} c_{ij}^0 x_j^0)$, consequences of the initial decision in the second stage if scenario k revealed and it's outcomes ($\sum_{j=1}^{n} c_{ijk}^{0} x_{j}^{0} + \sum_{j=1}^{n} c_{ijk}^{1} x_{jk}^{1}$). The sets of linear inequalities for the first and the second stage are respectively described in the constraints as well as the non-negativity constraints.

5 Solving by goal programming (the reference point method)

The Goal Programming (GP) approach, introduced by Charnes et al. (1955, 1961), is one of the most popular and oldest Multi-Criteria Decision Analysis (MCDA) techniques. The most appealing idea about GP is that, firstly, the goals are set by DM in objective space then try to come close to it through minimising the deviation or distance measure (this deviation represents by a norm) between this specific goal vector (or aspiration levels) and the decision outcome (obtainable objective vector).

However, without some extra assumptions, there is no norm minimization can generate efficient/vector-optimal solutions. Then, the above-mentioned idea(coming close to a goal) is mathematically contradictory to the concept of efficiency (vector optimality) (Wierzbicki 1998). Therefore, this idea can lead to dominated solutions in the standard form of goal programming approach. This represents the main shortcoming of the typical goal programming approach (Wierzbicki 1998; Ogryczak 1994). Hence, in regular goal programming, a distance function is utilised to evaluate the performances. If the distance function is zero, then there is no more optimisation possible. That is mainly due to the fact that we cannot get a negative value for a distance function and so the optimisation process is stopped (Wierzbicki 1998).

Amongst different GP models, it has been shown by Ogryczak (1994) that the solutions of the *Reference Point Method* (RPM) are always Pareto efficient. Although methods exist to correct standard GP for non-efficiency (see Jones and Tamiz 2010 and Romero 1991), the RPM is a direct and simple approach to achieve this end. The RPM uses the so-called quasi-satisficing decisions introduced by Wierzbicki (1982). In this approach, optimisation is carried out (even after reaching the desirable goals) and the goals/reference points are interpreted consistently, based on the vector optimisation (Pareto efficiency) concept. Therefore, the solutions/decisions can be improved beyond the primary targets, and aspiration levels/reference points are obtainable with some surplus (i.e. the achievements are beyond our ideal expectations, surplus achievements compared to the ideal). Therefore, unlike the standard goal programming, RPM continues searching for better points even if the predetermined goals/reference points have already been obtained. In fact, the RPM uses the underlying optimisation philosophy instead of the underlying satisfying philosophy which is utilised in typical GP.

The efficient/Pareto solutions in RPM are generated by minimising some so-called scalarization achievement functions (which are a kind of utility or value functions) instead of minimising the distance measures (norm) which is a strategy used in regular GP methods. In other words, the RPM utilises the reference levels (instead of aspiration levels in GP) to control and lead the parameters. This change in philosophy provides the ability for the model to accept negative weights regarding the negative deviations in the models. Therefore, the RPM continues searching for better points (even if the predetermined reference levels have already been obtained) and, therefore, generates the efficient/optimal solution.

These scalarization functions utilise an arbitrary small positive scalar ϵ (also called regularisation/augmentation term) to keep searching for better solutions (even after obtaining the reference levels). This scalar (ϵ) guarantee efficiency/optimality in the case of multiple optimal solutions which is a normal outcome in MCDM problems. Different scalarization functions have been introduced in the literature. One of the simplest forms of scalarization achievement function is formulated as follows (Ogryczak 1994):

$$Max_{1\le i\le m}\{\omega_i(f_i(x)-g_i)\}+\epsilon\sum_{i=1}^{m}(\omega_i(f_i(x)-g_i)), \tag{4}$$

where $\omega_i > 0\,(i = 1, \dots, m)$ are scaling factors, g_i explain reference levels, and ϵ is an arbitrary small positive number.

Minimising this scalarization achievement function (4) over the feasible set of a decision problem can produce an efficient solution. By defining $\delta_i\ (i = 1, \dots, m)$ for representing the goal deviations, we will have $f_i(x) - g_i = \delta_i$. Therefore, the corresponding RPM model can be formulated as follows:

$$\begin{aligned} &Min\ \psi = \varphi + \epsilon\sum_{i=1}^{m}(\omega_i\delta_i) && \\ &s.t.\quad \omega_i\delta_i \le \varphi && i = 1, \dots, m, \\ &\qquad\ f_i(x) - \delta_i = g_i && i = 1, \dots, m, \\ &\qquad\ \mathbf{x} \in \mathbf{X} && \end{aligned} \tag{5}$$

where $\varphi = Max_{1\le i\le m}\{\omega_i\delta_i\}$, $\mathbf{x}$ is the vector of decision variables, and $\mathbf{X}$ is the set of feasible solutions of the problem.

The RPM is an interactive technique and the DM preferences/expectations will be reflected by the reference levels which are determined by the DM (similar to the typical GP). The optimal/efficient solution could be compared with previous/other solutions by the DM, and could be modified (via some changes in the reference levels) if necessary. Furthermore, the RPM (and GP in general) is significantly useful in the proposed framework as it can easily handle large numbers of objective functions/crit, variables and constraints (Tamiz et al. 1998). Identifying the goals for each criterion is the only action that the DM needs to carry out in the GP approach. This, as pointed out by Belton and Stewart (2002), "facilitates the formal incorporation of the treatment of uncertainty through use of scenarios". The two-stage structure needs to evaluate and compare alternatives/solutions taking into account all meta-criteria (all criteria-scenario combinations) that, especially in a real-world problem, may give rise to dealing with very large numbers of meta-criteria. Therefore, the DMs need only to set their desirable goals for each meta-criterion without trade-off assessments or further calculations. This can markedly reduce computation which would make the process of setting goals easier and help us gain robust solutions which perform satisfactorily on all meta-criteria. Of course, they still able to compare the trade-offs between the meta-criteria if they want to interactively be involved.

The RPM sets into a GP framework by adding an L_1 restoration stage to the L_∞ based GP model (Jones and Tamiz 2010). Although utilising the L_1 based model is more common in GP, it requires consideration of trade-offs, whereas the L_∞ based model focuses more on worst-case performance which is more balanced/robust in our context of including deep not always measurable uncertainties. Nonetheless, a different variant of GP can be chosen regarding the type of problem and/or the structure of the DMs' preferences. This needs a mindful analysis both before and after the solution process to avoid the modelling pitfalls such as normalisation, restoration and Pareto efficiency as discussed in Tamiz et al. (1998).

In the two-stage structure proposed in the current study, reference points represent aspiration levels/goals for each meta-criterion by which the DMs' preferences apply in the evaluation process and lead the model to construct an efficient solution which is the "closest or better" to the preferred reference points. Moreover, each meta-criterion associates with a measurable attribute value(goal/reference point) which represents a natural interplay between meta-criteria and the reference point mode of value elicitation.

Accordingly, all the above reasons motivate us to utilise the RPM to solve the multi-objective models. In such models, we would seek to minimise all $m \times p$ objective functions (or performance measures (Z_{ik})). If we define $m \times p$ goals g_{ik} for all meta-criteria corresponding to deviation variables δ_{ik} and, then consider the RPM, the associated constraints can be formulated as follows:

$$\begin{aligned} Z_{ik} - \delta_{ik} = \sum_{j=1}^{n} c_{ij}^0 x_j^0 + \left(\sum_{j=1}^{n} c_{ijk}^0 x_j^0 + \sum_{j=1}^{n} c_{ijk}^1 x_{jk}^1 \right) \\ - \delta_{ik} = g_{ik}, \;\; i = 1, \dots, m, k = 1, \dots, p; \end{aligned} \tag{6}$$

So, the equivalent RPM model can be formulated as follows:

$$\begin{array}{lll} Min \;\; \psi = \phi + \epsilon \sum_{k=1}^{p} \sum_{i=1}^{m} (\omega_{ik} \delta_{ik}) & & \\ s.t. \;\; \omega_{ik}\delta_{ik} - \phi \leq 0, & & i = 1, \dots, m, k = 1, \dots, p; \\ \quad \sum_{j=1}^{n} (c_{ij}^0 + c_{ijk}^0) x_j^0 + \sum_{j=1}^{n} c_{ijk}^1 x_{jk}^1 - \delta_{ik} = g_{ik}, & & i = 1, \dots, m, k = 1, \dots, p; \\ \quad \sum_{j=1}^{n} a_{rj}^0 x_j^0 \leq b_r^0, & & r = 1, \dots, R_0; \\ \quad \sum_{j=1}^{n} a_{rjk}^0 x_j^0 + \sum_{j=1}^{n} a_{rjk}^1 x_{jk}^1 \leq b_{rk}^1, & & k = 1, \dots, p; \\ & & r = R_0 + 1, \dots, R_0 + \cdots + R_k; \\ \quad x_j^0, x_{jk}^1 \geq 0, & & \forall i, j, k, r, \\ \quad \phi, \delta_{ik} \, free \; of \; sign. & & \forall i, k. \end{array} \tag{7}$$

Table 1 Farming data set

	Wheat			Corn			Soy-beans		
	s_1	s_2	s_3	s_1	s_2	s_3	s_1	s_2	s_3
Yield (Tonnes/ha)	4	3.6	3.3	9	6	5	2.2	2	1.5
Planting cost (R/ha)	2400	2500	2750	4300	4500	4750	2900	3400	3900
Selling price (R/tonnes)	1600	1850	2100	1250	1450	1650	3050	3300	3500
Environmental benefit coefficient (%)	14	13	12	12	13	14	22	24	25
Purchase price (R/tonnes)	1800	2000	2300	1400	1600	1800	N/A	N/A	N/A
Minimum requirement (tonnes)	400	400	400	440	440	440	N/A	N/A	N/A

where $\omega_{ik} \geq 0, (i = 1, \ldots, m, k = 1, \ldots, p)$; are the *importance weighting* of deviations which set by DM. ϵ is an arbitrarily small positive number, and $\phi = Max_{i,k}\{\omega_{ik}\delta_{ik}\}$.

6 Simple farming example

In this section, we show that how the proposed two-stage model works under deep uncertainty using a simple farming example.[3] We will explain how this structure can be modelled, implemented, and solved by Goal Programming. Further discussion will be presented at the final part of this section. We may need to note that, this example is only an illustration that how the proposed framework and the extended GP can work but it is not truly a case study in deep uncertainty. Applying the proposed framework in some real-life problems that face deep uncertainty lies in our future directions.

Example 6.1 Assume that a farmer can organically grow wheat, corn or soy-beans on the entirety of his 1000 ha plot. From this land, he needs to produce 400 tonnes of wheat and 440 tonnes of corn to feed his cattle. If the farmer fails to produce enough feed, the shortfall must be bought from a wholesaler at the cost of **R**2000/tonne of wheat and **R**1600/tonne of corn. However, if he produces an excess of feed, this can be sold for **R** 1850/tonne of wheat and **R** 1450/tonne of corn.

The farmer can also grow soy-beans which sell for **R**3300/tonne. He knows well enough that his yields are not always precise. As the yields are highly dependant on weather, he decides to consider three different scenarios for weather conditions. Assume three scenarios: good (s_1), average (s_2) and bad (s_3) weather. All data have been described in Table 1.

[3] The idea has been taken from Linderoth (2003).

Furthermore, the farmer is an advocate for organic farming practices which promote certain environmental benefits such as sustainable soil and climate, as well as negligible impact on ecosystems. As a result of this, the farmer benefits from reduced taxes which are calculated by an environmental benefit coefficient that is specific each product and affected by a range of different factors. Related environmental benefit coefficients for each product under various scenarios are described in Table 1. The farmer also needs to consider planting costs which for wheat, corn and soy-beans are $R2400$, $R4300$, and $R2900$, respectively. He has to pay these minimum costs at the beginning of the year. In addition these costs, extra charges may need to paid depending on scenario revelation-that is, if scenario 2 or 3 happen. To further clarify, this means that the farmer will have to pay extra if (and only if) one of the scenarios 2 or 3 unfold. These extra costs would be payable after scenario realisation. It must be noted that there is a recourse/adaptation option at the second stage of the decision-making process. This means that, just after observing weather conditions (and in the second stage of decision-making), the farmer would have a chance to decide how much of each crop to purchase or sell. It may need to be noted that in this stage (second stage) the farmer has enough knowledge about the current weather conditions (the scenario that unfolded). This information can help him to estimate his yields at the end of the period and adapt his initial decision(s).

To formulate the model, let us set the following variables:

Setting variables	
$x_i^0(i = 1, 2, 3)$:	Hectares of wheat, corn,and soy-beans planted (initial decision), respectively
$x_{ik}^1(i = 1, 2, 3)$:	Tonnes of wheat, corn, and soy-beans sold if scenario k ($k = 1, 2, 3$) unfolded, respectively
$x_{4k}^1; x_{5k}^1$:	Tonnes of wheat and corn purchased if scenario k ($k = 1, 2, 3$) revealed, respectively

Suppose that there are three criteria ($C_i, i = 1, 2, 3;$) which define as follows:

- $\mathbf{C_1}$: **Min**imising the sum of current investment costs.
- $\mathbf{C_2}$: **Max**imising the liquidity at the end of the year which earn from selling crops.
- $\mathbf{C_3}$: **Max**imising the environmental benefits.

Then, the two-stage multi-objective optimisation model of this problem would be formulated as follows:

$$
\begin{aligned}
& MinZ_{11} = 2400x_1^0 + 4300x_2^0 + 2900x_3^0, \quad k = 1; \\
& MinZ_{12} = 2500x_1^0 + 4500x_2^0 + 3400x_3^0, \quad k = 2; \\
& MinZ_{13} = 2750x_1^0 + 4750x_2^0 + 3900x_3^0, \quad k = 3. \\
& MaxZ_{21} = 1600x_{11}^1 + 1250x_{21}^1 + 3050x_{31}^1 - 1800x_{41}^1 - 1400x_{51}^1, \quad k = 1; \\
& MaxZ_{22} = 1850x_{12}^1 + 1450x_{22}^1 + 3300x_{32}^1 - 2000x_{42}^1 - 1600x_{52}^1, \quad k = 2; \\
& MaxZ_{23} = 2100x_{13}^1 + 1650x_{23}^1 + 3500x_{33}^1 - 2300x_{43}^1 - 1800x_{53}^1, \quad k = 3. \\
& MaxZ_{31} = (0.14)x_1^0 + (0.12)x_2^0 + (0.22)x_3^0, \quad k = 1; \\
& MaxZ_{32} = (0.13)x_1^0 + (0.13)x_2^0 + (0.24)x_3^0, \quad k = 2; \\
& MaxZ_{33} = (0.12)x_1^0 + (0.14)x_2^0 + (0.25)x_3^0, \quad k = 3. \\
& s.t. \\
& x_1^0 + x_2^0 + x_3^0 = 1000 \quad (\textit{Land limitation}) \\
& 4x_1^0 - x_{11}^1 + x_{41}^1 = 400 \\
& (3.6)x_1^0 - x_{12}^1 + x_{42}^1 = 400 \quad (\textit{Wheat requirement}) \\
& (3.3)x_1^0 - x_{13}^1 + x_{43}^1 = 400 \\
& 9x_2^0 - x_{21}^1 + x_{51}^1 = 440 \\
& 6x_2^0 - x_{22}^1 + x_{52}^1 = 440 \quad (\textit{Corn requirement}) \\
& 5x_2^0 - x_{23}^1 + x_{53}^1 = 440 \\
& (2.2)x_3^0 - x_{31}^1 = 0 \\
& 2x_3^0 - x_{32}^1 = 0 \quad (\textit{Soybean balance}) \\
& (1.5)x_3^0 - x_{33}^1 = 0 \\
& x_1^0, x_2^0, x_3^0, x_{1k}^1, x_{2k}^1, x_{3k}^1, x_{4k}^1, x_{5k}^1 \geq 0 \quad k = 1, 2, 3.
\end{aligned}
\tag{8}
$$

It is important to note at this stage that the model is not unit-invariant. Therefore, if various units exist for different objectives, to getting the best result, it is better to normalise the data before running the model. For instance, dividing by the maximum or utilising some weights can make the model unit-invariant (a review of some normalisation methods can be found in Tamiz et al. 1998). To solve this problem using the RPM, the farmer is asked to set his goals for each objective, as well as some weights for deviations from these goals. If $\forall i, k; \omega_{ik} = 1$, $g_1 = (1{,}400{,}000, 10{,}000{,}000, 160)$, $g_2 = (1{,}550{,}000, 8{,}000{,}000, 165)$, and $g_3 = (1{,}750{,}000, 6{,}500{,}000, 170)$. After normalising units, the equivalent linear model of (6) can be formulated which would provide us with a solution and the optimal values of objectives. One solution can be generated in every run. More solutions can be gained by changing the aspiration levels and/or the importance weights. Table 2 describes the result reaching by solving the equivalent linear model of (6), after applying the goals, weights, and normalising the units.

Table 2 The result of the proposed two-stage MOLP for farming example

Scenario	$X^0 = (x_1^0, x_2^0, x_3^0)$ $X_k^1 = (x_{1k}^1, x_{2k}^1, x_{3k}^1, x_{4k}^1, x_{5k}^1)$	Current costs (Z_{1k})	Liquidity (Z_{2k})	Environmental benefits (Z_{3k})	$\Delta = (\delta_{1k}, \delta_{2k}, \delta_{3k})$
Scenario free	$X^0 = (446.7, 553.3, 0)$	–	–	–	–
$k = 1$	$X_1^1 = (1386.9, 4539.6, 0, 0, 0)$	3,452,241	7,893,431	128.9	(431.84, 443.49, 124.26)
$k = 2$	$X_2^1 = (1208.2, 2879.7, 0, 0, 0)$	3,606,569	6,409,805	130	(432.96, 334.78, 140)
$k = 3$	$X_3^1 = (1074.2, 2326.5, 0, 0, 0)$	3,856,693	6,093,716	131	(443.49, 85.53, 155.74)

6.1 Discussion of contributions to planning and understanding

This example includes three criteria/objectives and three scenarios. Therefore, the decisions must be evaluated and compared in all 9 criterion-scenario combinations, called meta-criteria. Here, these combinations are represented by $Z_{ik}, i = 1, 2, 3; k = 1, 2, 3;$ followed by land limitation as a scenario-free (fundamental limitation) and product requirements as the scenario-dependent constraints (uncertain upcoming restrictions) that completely constitute the proposed two-stage scenario-based multi-objective optimisation model for this example. To solve this multi-objective model with RPM, the DM defined his/her goals and importance weights, and the final results have been described in Table 2.

The DM (farmer here) has to make his initial decision ($\mathbf{X}^0$, that is how to allocate different parts of the land to planting various types of crops) at the beginning of the period/year before he has any information about the future events (uncertain weather in this particular example). This initial decision is common in all scenarios and lies in the common decision space, then, it needs to be feasible in all scenarios. In fact, the initial solution is a robust solution. The uncertain events will have an effect on the outcomes of the initial decision. For example, effects could be in terms of yield, expenses, selling prices, and environmental benefits in the current example. Clearly, it is not possible to find a perfect decision that keeps its optimality in all scenarios, even in this simple example.

Accordingly, as seen in Table 2, there will be different achievements in various scenarios (even though no recourse decision is needed); i.e. there will be a different objective space for each scenario. We may need to note that, In a scenario-based multi-objective optimisation problem (SBMOOP), there is no single Pareto frontier. Instead, there exist several Pareto frontiers (in different objective spaces), one related to each scenario. Moreover, because of various constraints, the decision space could also vary in different scenarios. This concept will be discussed later in Sect. 7.2, where we compare the proposed framework with the robust GP.

Moreover, the DM (farmer) only can make the recourse decisions ($\mathbf{X}_k^1$, sales and purchase decisions) after scenario realisations (seeing his yield). Note that these recourse decisions add more decision variables and then append further dimensions to the initial common decision space which construct the aggregated decisions which are different in each scenario. The proposed two-stage framework looks for the best robust initial decision (in a lower dimensional space) that, together with the subsequent recourse decisions (in different dimensions), provide us with the best possible performances under all circumstances. Indeed, the initial solution to the proposed two-stage model is a robust solution (comparable with the solution to the robust methods), however, in the proposed model there is a chance for further improvement in every single scenario which can address the essential shortcoming of typical robust solutions which are too conservative. This would be the most important advantage of the proposed two-stage model compared to the robust multi-objective optimisation methods. Furthermore, It will be shown (later in Sects. 7.1 and 7.2) that the proposed two-stage model can provide us with an optimal solution when no common feasible solution can be found by other models, such as the single-scenario and robust models. In fact, the aggregation of the solutions/decisions which is provided by the proposed two-stage model guarantees the feasibility and overall optimality if the feasible region under conditions of any scenario is not empty (i.e. for each scenario, at least one point can be found satisfying all restrictions).

As mentioned earlier, solving the proposed two-stage framework by RPM (or GP) provide us with only a single solution from the Pareto fronts related to different scenarios which place it as a priori method. However, depends on the DM, it also can be used as an interactive or a posteriori method which allows comparing the trade-offs between the meta-criteria. For example, the DMs can be involved in solution generation and compare different solutions by applying their preferences as aspiration levels/reference points. Different solutions can be produced by changing the reference points and/or the weights. Although,in GP, the weights are primarily used to achieve a comparative scaling on each objective (normalising the goals in the model), they can also utilise to demonstrate the DM's preferences concerning to each goal (Ringuest 2012 discussed several methods developing for determining the values of weights in GP). We may need to note that, in this study, we do not focus on the system of goal and weight choice. As we believe, because of too many goals/weights, elicitation of the DM preferences is hardly possible in practice and needs a separate study which lies in our future directions of research.

7 Example illustrating scenario effects

The aim of this section is to provide comparison with some popular existing scenario-based goal programming methods, specifically the single-scenario models and the robust GP. The purpose is to highlight the superiorities of the proposed two-stage structure in comparison with these approaches. Especially, this superiority is crucial when the intersection of the constraints of all scenarios is empty (i.e. the integrated problem has no feasible solution). Furthermore, it will be shown that, even if (after the revelation of scenarios) we have a chance to adapt the decision in corresponding single-scenario optimisation models, their solutions could not be as good as the solutions to the proposed two-stage approach which considers the

Table 3 The amount of the coefficients of the model in the case of five different scenarios

Scenario/ coefficients	c^0_{11k}	c^0_{12k}	g_{1k}	c^0_{22k}	c^0_{21k}	g_{2k}	a^0_{11k}	a^0_{12k}	b^1_{1k}	a^0_{21k}	a^0_{22k}	b^1_{2k}	a^0_{31k}	a^0_{32k}	b^1_{3k}
$K = 1$	− 5	4	− 15	1	2	0	1	1	4	1	0	3	0	1	3
$K = 2$	− 3	1.5	− 9	2	3	2	− 1	− 1	− 1	1	0	3	0	1	3
$K = 3$	0.5	9	28	2	1	7	1	1	5	− 1	0	− 2	0	− 1	− 3
$K = 4$	0.7	7	16.1	2.2	1.3	9.2	− 1	− 1	− 2	− 1	0	− 3	0	− 1	− 2
$K = 5$	− 7	4	− 12	1.5	2.5	13	− 1	− 1	− 6	1	0	4	0	− 1	− 4

required adaptations in advance (before scenario realisation). It was possible to use the previous example here and all the aspects analysed could be combined in a single problem. However, for the sake of simplicity, we prefer to discuss the above-mentioned concepts in a very simple example.

Example 7.1 Let us consider the following model:

$$
\begin{aligned}
&Min\ \mathbf{F} = [f_{1k}, f_{2k}] = \sum_{k=1}^{5}\sum_{j=1}^{2}\left(c^0_{ijk}x^0_j + c^1_{ijk}x^1_k\right); && i = 1,2; k = 1,\dots,5;\\
&s.t.\quad \sum_{j=1}^{2} a^0_{rjk}x^0_j + a^1_{rk}x^1_k \le b^1_{rk}, && r = 1,2,3; k = 1,\dots,5;\\
&\qquad x^0_j, x^1_k \ge 0. && j = 1,2; k = 1,\dots,5.
\end{aligned}
\tag{9}
$$

where the coefficients of the model in the case of five different scenarios have been shown in Table 3.

To solve a problem using the goal programming approach, we need to set some goals (g_{ik}) for each objective which could be established by DM or we may use the optimum solution of the specific problem when there is not any uncertainty in the model. For example, if it is known that scenario ‘1’ will unfold we could solve two simple single objective models, related to two objectives, and subject to specific constraints of scenario ‘1’. These goals, which are obtained by solving two single objective problems subjected to specific constraints of scenario $k, (k = 1,\dots,5)$, have also been demonstrated in Table 3.

If $c^1_{ijk} = 1$, and $a^1_{rk} = -1$, for all $i, j, r,$ and k, then the proposed two-stage model for Example 7.1 can be formulated. By solving the equivalent linear model the following solution, which shown in Table 4, can be found.

As shown in Table 4, the proposed 2-stage model suggests that $X^0 = (x^0_1, x^0_2) = (2.62, 1.32)$ (the solutions have been rounded off to two decimal places) as an initial decision which is feasible (robust feasible solution) in all scenarios. The contingent (recourse) decisions $(x^1_k, k = 1,\dots,5)$, which will be implemented after scenario realisation, have been presented in column three of Table 4. Number ‘0’ for scenarios ‘1’ and ‘2’ describe that there is no need for any more

Table 4 The result of the proposed two-stage MOLP for Example 7.1

Scenario/solutions	$X^0 = (x_1^0, x_2^0)$	x_k^1	$F = (f_{1k}, f_{2k})$	$G = (g_{1k}, g_{2k})$	$\Delta = (\delta_{1k}, \delta_{2k})$	φ
$k = 1$	(2.62, 1.32)	0	(− 7.8, 5.25)	(− 15,0)	(7.2, 5.25)	7.2
$k = 2$	(2.62, 1.32)	0	(− 5.87, 9.2)	(− 9, 2)	(3.13, 7.2)	7.2
$k = 3$	(2.62, 1.32)	7.64	(20.8, 14.2)	(28, 7)	(− 7.2, 7.2)	7.2
$k = 4$	(2.62, 1.32)	3.94	(15, 11.4)	(16.1, 9.2)	(− 1.1, 2.2)	2.2
$k = 5$	(2.62, 1.32)	4.55	(− 8.5, 11.78)	(− 12, 13)	(3.5, − 1.22)	3.5

contingent decisions, and that the DM needs to do nothing if one of these two scenarios is revealed. It means that no further improvement possible if these scenarios revealed. There would be a corrective decision for any other scenario realisation. The value of objective functions (achievements) ($F = (f_{1k}, f_{2k})$) and their goals have been presented in columns 4 and 5, respectively. The column 6 of Table 4 shows the deviations(δ_{ik}) from the initial goals. Positive and negative signs represent the positive and negative deviations from the goals.[4] Finally, the maximum deviations from the goals (φ) are demonstrated in the last column of the Table. For example, consider the last row where scenario '5' occurred. If the suggested initial decision ($X^0 = (x_1^0, x_2^0) = (2.62, 1.32)$) was applied, now and after the revelation of scenario '5', the contingent decision ($x_5^1 = 4.55$) needs to be implemented as a recourse in which the values of '−8.5' and '11.78' could be reached for the first and the second objectives, respectively. The goals were '−12' and '13' so, in comparison with our achievements ('−8.5' and '11.78'), these show '3.5' units positive deviation from the first goal and '1.22' units negative deviation from g_{25} that represent over achievement in this meta-criteria. Obviously, the maximum deviation in this scenario is '$\varphi = 3.5$'. More discussion on results of Example 7.1 and comparison with other approaches can be found in the next subsection.

7.1 Comparison with single-scenario models

In this section, Example 7.1 is revisited and solved by some single-scenario models. In these models, we assume that the future scenario is known and the problem is solved for this specific scenario. Therefore, the problem needs to be solved under the conditions of every plausible scenario.

Example 7.2 Consider Example 7.1 for which five single-scenario models need to be run (each model includes the specific objective functions and constraints related to the considered scenario) By solving these five specific models, five different

[4] Sometimes, a one-side deviation may be considered in which the positive deviations in Maximisation problem, and the negative deviations in Minimisation problem would be an advantage. However, for this specific example, the two-sided deviation is considered and both positive and negative deviations are undesirable.

Table 5 Comparing different solutions of various models

Scenario/goals	S_1	S_2	S_3	S_4	S_5	Proposed 2-stage model
$X^0 = (x_1^0, x_2^0)$	(2.5, 0)	(2.2, 0)	(2, 3)	(3, 2)	(3.65, 4)	$(2.62, 1.32) + x_k^1$

Table 6 Comparing the feasibility of each solution in different scenario realisations

Scenario/ models	S_1	S_2	S_3	S_4	S_5	Proposed 2-stage model
s_1	✓	✓	×	×	×	✓
s_2	✓	✓	✓	✓	×	✓
s_3	×	×	✓	×	×	✓
s_4	×	×	×	✓	✓	✓
s_5	×	×	×	×	✓	✓

solutions can be reached which have been portrayed in Table 5 together with the proposed two-stage solution.

At first, the feasibility of these solutions in every single scenario has been checked, and the result has been presented in Table 6. Check marks indicate that the solution is feasible for that specific scenario realisation, and the cross marks represent the infeasibility of the related solutions under conditions of that scenario.

It can be clearly seen that, except for the solution of the proposed two-stage model, none of the solutions of the five single-scenario models is feasible in all scenarios. In fact, the intersection of the constraints of these five scenarios is empty, and it is impossible to find a feasible solution that satisfies all constraints. However, by utilising the proposed two-stage approach the initial solution together with a recourse action could solve this problem. These results present the power of our proposed approach in comparison with existing single scenario-based goal programming methods, especially when the intersection of the constraints of all scenarios is empty, and the integrated problem has no feasible solution (viz. there is no robust (feasible) solution). The question that may raise here is: what if it is possible to correct the solutions of these five single scenario models after scenario realisation? It should be quite clear that the solution of the proposed two-stage approach must perform equally or better than the other approaches because it is an optimal solution. However, to further illustration this, these solutions have been applied to a two-stage structure (Tables 7, 8, 9, 10, 11).

In this case, suppose that there is a possibility of adapting the first decision after scenario revelation.[5] This means that one of the solutions of these five single

[5] More often, in real situations, there is not enough time to correct the decisions if there is not any preparation.

Table 7 Achievements which could be reached by applying the solution of scenario 1 model as an initial solution together with the possible recourse action in Example 7.1

Scenario/ solutions	$X^0 = (x_1^0, x_2^0)$	x_k^1	$F = (f_{1k}, f_{2k})$	$G = (g_{1k}, g_{2k})$	$\Delta = (\delta_{1k}, \delta_{2k})$	φ
$k = 1$	(2.5, 0)	0	(– 12.5, 2.5)	(– 15,0)	(2.5, 2.5)	2.5
$k = 2$	(2.5, 0)	0	(– 7.5, 5)	(– 9, 2)	(1.5, 3)	3
$k = 3$	(2.5, 0)	14.375	(15.625, 19.375)	(28, 7)	(– 12.375, 12.375)	12.375
$k = 4$	(2.5, 0)	10.743	(12.49, 16.68)	(16.1, 9.2)	(– 3.61, 7.48)	7.48
$k = 5$	(2.5, 0)	8.031	(– 9.47, 11.78)	(– 12, 13)	(2.53, – 1.22)	2.53

Table 8 Achievements which could be reached by applying the solution of scenario 2 model as an initial solution together with the possible recourse action in Example 7.1

Scenario/solutions	$X^0 = (x_1^0, x_2^0)$	x_k^1	$F = (f_{1k}, f_{2k})$	$G = (g_{1k}, g_{2k})$	$\Delta = (\delta_{1k}, \delta_{2k})$	φ
$k = 1$	(2.2, 0)	0	(– 11, 2.2)	(– 15,0)	(4, 2.2)	4
$k = 2$	(2.2, 0)	0	(– 6.6, 4.4)	(– 9, 2)	(2.4, 2.4)	2.4
$k = 3$	(2.2, 0)	14.75	(15.85, 19.15)	(28, 7)	(– 12.15, 12.15)	12.15
$k = 4$	(2.2, 0)	10.93	(12.47, 15.77)	(16.1, 9.2)	(– 3.63, 6.57)	6.57
$k = 5$	(2.2, 0)	8.03	(– 7.1, 11.6)	(– 12, 13)	(4.9, – 1.4)	4.9

Table 9 Achievements which could be reached by applying the solution of scenario 3 model as an initial solution together with the possible recourse action in Example 7.1

Scenario/solutions	$X^0 = (x_1^0, x_2^0)$	x_k^1	$F = (f_{1k}, f_{2k})$	$G = (g_{1k}, g_{2k})$	$\Delta = (\delta_{1k}, \delta_{2k})$	φ
$k = 1$	(2, 3)	1	(3, 9)	(– 15,0)	(18, 9)	18
$k = 2$	(2, 3)	0	(– 1.5, 13)	(– 9, 2)	(7.5, 11)	11
$k = 3$	(2, 3)	0	(28, 7)	(28, 7)	(0, 0)	0
$k = 4$	(2, 3)	1	(23.4, 9.3)	(16.1, 9.2)	(7.3, 0.1)	7.3
$k = 5$	(2, 3)	2.04	(0.04, 12.54)	(– 12, 13)	(12.04, – 0.46)	12.04

Table 10 Achievements which could be reached by applying the solution of scenario 4 model as an initial solution together with the possible recourse action in Example 7.1

Scenario/solutions	$X^0 = (x_1^0, x_2^0)$	x_k^1	$F = (f_{1k}, f_{2k})$	$G = (g_{1k}, g_{2k})$	$\Delta = (\delta_{1k}, \delta_{2k})$	φ
$k = 1$	(3, 2)	1	(– 6, 8)	(– 15,0)	(9, 8)	9
$k = 2$	(3, 2)	0	(– 6, 12)	(– 9, 2)	(3, 10)	10
$k = 3$	(3, 2)	6.36	(25.86, 14.36)	(28, 7)	(– 2.14, 7.36)	7.36
$k = 4$	(3, 2)	0	(16.1, 9.2)	(16.1, 9.2)	(0, 0)	0
$k = 5$	(3, 2)	2.84	(– 10.16, 12.44)	(– 12, 13)	(1.84, – 0.66)	1.84

scenario optimisation models is applied and then, after scenario revelation, we will try to adapt the decision to the extent possible, if there is a need for such a correction. Therefore, we consider a two-stage process in which the initial decision

Table 11 Achievements which could be reached by applying the solution of scenario 5 model as an initial solution together with the possible recourse action in Example 7.1

Scenario/solutions	$X^0=(x_1^0,x_2^0)$	x_k^1	$F=(f_{1k},f_{2k})$	$G=(g_{1k},g_{2k})$	$\Delta=(\delta_{1k},\delta_{2k})$	φ
$k=1$	(3.65, 4)	3.65	(8, 12)	(− 15,0)	(16.4, 15.3)	16.4
$k=2$	(3.65, 4)	1	(1, 17)	(− 9, 2)	(5.05, 18.03)	18.03
$k=3$	(3.65, 4)	2.65	(38, 9)	(28, 7)	(12.475, 6.96)	12.475
$k=4$	(3.65, 4)	0	(30.4, 10.6)	(16.1, 9.2)	(14.46, 4.03)	14.46
$k=5$	(3.65, 4)	0	(2, 13)	(− 12, 13)	(2.45, 2.475)	2.475

referred to the solution of the single-scenario optimisation models, and the contingent solutions for every single scenario revelation are what we are looking for. Tables 7, 8, 9, 10 and 11 in page 21 describe and compare the contingent solutions, objective functions, goals, deviations and maximum deviation from the goals for every single scenario realisation.

In all these Tables 7, 8, 9, 10 and 11, the initial and contingent decisions have been presented in the second and third columns, respectively. Columns 4–6 have respectively described the achievements (values of the objective functions), goals and the deviations from the goals. Finally, the last column shows the maximum deviations from the goals that would be obtained if scenario $k(k=1,2,3,4,5;)$ is revealed. By taking a glance over these five tables and comparing them to Table 4, which indicates the result of our proposed two-stage model, it is clear that the proposed two-stage approach has the minimum of maximum deviations from goals ($Min-Max\ \delta_{(2stage)}=7.2$) in comparison with single-scenario models ($Min-Max\ \delta_{(single-scenarios)}=$ 12.375, 12.15, 18, 10 and 18.03, respectively). So, it vividly demonstrates the optimality of the proposed model as it is expected, mainly due to the fact it is looking ahead, and must, therefore, find better goal achievement than the single-scenario models. For example, let us compare the proposed solution to the solution which was obtained from the single-scenario model related to the forth scenario, which has the least difference compared to the proposed model. This solution is the best solution, if and only if, scenario ‘4’ manifested in which all goals would be reached completely. Otherwise, except for scenario ‘5’, the proposed two-stage model provides us with better performance than this solution.

A summary of differences between the objective function (φ) of the proposed two-stage model and the others in various scenarios has been recorded in Table 12. Negative numbers describe the advantages of the objective function (φ) of the proposed model in comparison with the single-scenario models. As shown in the last row of this Table, the aggregation of these differences over all scenarios is a negative number in all columns which highlights the optimality of the solutions of the proposed two-stage model.

As shown in the above example, the proposed two-stage model can provide us with an optimal solution when no common feasible solution can be found by the single-scenario models and the intersection of the combined constraints is empty. In fact, by applying the proposed two-stage structure, we are always able to find a feasible solution if, at least, one feasible solution can be found for every single-scenario

Table 12 A summary of differences between the objective function (φ) of the proposed two-stage model and the others in various scenarios

Scenario	Differences of φ's				
	$2stage - s_1$	$2stage - s_2$	$2stage - s_3$	$2stage - s_4$	$2stage - s_5$
$k = 1$	4.7	3.2	– 10.8	– 1.8	– 9.2
$k = 2$	4.2	4.8	– 3.8	– 2.8	– 10.83
$k = 3$	– 5.175	– 4.95	7.2	– 0.16	– 5.275
$k = 4$	– 5.28	– 4.38	– 5.1	2.2	– 12.26
$k = 5$	0.97	– 1.9	– 8.54	1.66	1.025
Total	– 0.585	– 3.22	– 21.04	– 0.9	– 36.54

model. We need to note that this feasible solution is not necessarily feasible in all single-scenario models. In other words, the necessary and sufficient condition for the feasibility of the full solution(union of initial and recourse solutions) in the proposed two-stage model is the feasibility of the particular single-scenario models(i.e. each single-scenario model is feasible and can provide at least one feasible solution that satisfies its constraints). Then, if that decision could not meet the limitations of the other models, the scenario-related recourse arrangements, with appropriate coefficients, help to compensate for the differences.

Obviously, if the feasible region under conditions of any scenario is not empty(i.e. for each scenario, at least one point can be found satisfying all restrictions), we are able to construct some feasible solutions for the proposed two-stage model by setting appropriate coefficients for the recourse decisions. This feature indicates the natural superiority of generating the dynamic-robust solutions in the proposed two-stage structure that highly motivated our study in complex MCDA problems. This superiority will be further highlighted in the next section, where it is compared to the regular robust goal programming approaches.

7.2 Comparison with robust goal programming

As seen in the previous section, the intersection of the constraints of plausible scenarios is empty, in Example 7.2, and it is impossible to find a feasible solution nor a robust one, which satisfies all constraints. Therefore, none of the single-scenario or robust approaches can solve the generated example but the proposed two-stage framework can. In fact, in the proposed two-stage framework, there is no need to find a common feasible initial solution that works for all scenarios, including the worst scenario. Rather, it is sufficient to be good enough as a starting point that enables the construction of the best possible combined decisions, together with scenario-related recourse decisions, in every single plausible scenario. This feature expands the robust decisions philosophy in our context and methodology by introducing and applying the concept of *dynamic-robust decisions* by which the limitation of the regular robust strategy (i.e. a risk-averse solution because of a bad scenario) is also reduced, its advantages are developed, and an extra motivation is appended to our study.

Therefore, for further illustration, let us make a comparison between the proposed two-stage structure and the robust goal programming approach, which was introduced by Kuchta (2004). Accordingly, we changed some coefficients in this example to find a non-empty intersection in which we can find a robust solution. Now we can compare this robust decision with the decision suggested by the proposed Two-stage structure.

Example 7.3 To reach this goal, it is sufficient to change b^1_{11} from '4' to '6' and b^1_{13} from '5' to '6' in Example 7.1. Then, $X^0 = (x^0_1, x^0_2) = (3, 3)$ would be a robust solution, and the solution of the proposed two-stage model remained Pareto optimum with no change ((2.61, 1.32)).

Results and comparison between the two, have been shown in Table 13. It can be clearly seen that except for the realisation of scenario '3' in which the robust solution will provide a better performance, implementing the proposed two-stage solution has a minimum deviations ($Min\ \varphi$) from the goals if scenario '1', '2' or '4' unfolded. In the case of the last scenario, both approaches have almost the same results with slight advantages for the robust solution in comparison with the proposed two-stage solution. Overall, it is evident that the proposed two-stage solution can provide us with better performance for this example.

Note that the solution of the robust model is dominated by the solution of the proposed two-stage model mainly due to the fact it is searching in the same feasible region as the robust model, plus it leaves room for the possibility of adaptation. If there is no more improvement possible by any contingent decisions in any scenario, the solutions of both approaches are expected to be equal (as occurs in Example 6). Indeed, the initial solutions of the proposed two-stage model will be equal or better than the Pareto optimal solutions to the robust GP. If there is no improvement possible in any meta-criterion after the scenario realisation, then, the solutions to both methods are equal. In this case, the decision spaces in all scenarios are analogous. Nonetheless, in the case of different decision spaces in various scenarios, the two-stage framework finds an under optimal initial solution (from the common decision space) while considering possible improvements, for every meta-criterion, in aggregation with contingent/recourse solutions. As a result, the proposed two-stage

Table 13 The proposed two-stage approach in comparison with the robust GP

Scenario/ solutions	Robust Solution			Goals $G = (g_{1k}, g_{2k})$	Proposed Two-Stage Solution		
	$F = (f_{1k}, f_{2k})$	$\Delta = (\delta_{1k}, \delta_{2k})$	φ		$F = (f_{1k}, f_{2k})$	$\Delta = (\delta_{1k}, \delta_{2k})$	φ
$k = 1$	(− 3, 9)	(12, 9)	12	(− 15,0)	(− 7.8, 5.25)	(7.2, 5.25)	7.2
$k = 2$	(− 4.5, 15)	(4.5, 13)	13	(− 9, 2)	(− 5.87, 9.2)	(3.13, 7.2)	7.2
$k = 3$	(28.5, 9)	(0.5, 2)	2	(28, 7)	(20.8, 14.2)	(− 7.2, 7.2)	7.2
$k = 4$	(23.1, 10.5)	(7, 1.3)	7	(16.1, 9.2)	(15, 11.4)	(− 1.1, 2.2)	2.2
$k = 5$	(− 9, 12)	(3, 1)	3	(− 12, 13)	(− 8.5, 11.78)	(3.5, − 1.22)	3.5

methods provide us with scenario-specific dynamic robust solutions choosing from relevant decision spaces. Moreover, remember that the two-stage structure can provide a meaningful solution even if no “robust” solution exists. In this case, there is no common decision space (i.e. the intersection of the constraints of all scenarios is empty), however, the combined solutions ($\mathbf{x} = (\mathbf{x^0}, \mathbf{x^1_k})$) to scenario k lies in the particular decision space of that scenario.

8 Conclusion

The focus and primary motivation throughout this paper have been on dealing with deep uncertainty and risk observing in many multi-criteria decision-making problems, especially, with long-term decision-making processes such as strategic planning problems in which the decision needs to be made (and one alternative must be chosen) before getting enough knowledge about consequences of the decision or even sufficient awareness of all other options.

Therefore, this paper introduced a novel robust-dynamic optimisation approach, named the scenario-based two-stage framework, to deal with multi-objective optimisation problems under deep uncertainty. The main idea was extended by the two-stage stochastic programming with recourse to address the capability of dealing with deep uncertainty through the use of scenario planning rather than statistical expectation. Scenarios are utilised as a dimension of preference (a component of the meta-criteria) to avoid problems of evaluating probabilities under deep uncertainty. The proposed approach is to consider rather the consequences to the decision maker(s) of various actions under the conditions represented by each scenario. The importance of the consideration depends not only on the “*likelihood*” of the scenarios but on repercussions from stakeholders or the public, in order to evaluate decision robustness in a deep sense. In this sense, the meta-criterion concept is a richer structure than an (arguable) expected value. This *dynamic-robust multi-objective optimisation structure* can help us handle the higher degrees of uncertainty in complex MCDA problems. A dynamic-robust decision in our philosophy is a split decision and contains two subgroups of decisions, the initial decision(s) followed by recourse decision(s), in which the full decision is robust across scenarios.

It is shown that the proposed two-stage structure not only helps us cope with deep uncertainty but also can contribute to solving the scenario-based problems without a feasible nor robust solution. This approach does not look for a solution that could be feasible in every single scenario. Such a solution, if it exists, may well cost too much as it needs to satisfy all constraints of all plausible scenarios, while only one scenario will happen. In general, a few scenarios, which may never unfold, may promote a conservative solution that often does not satisfy DM when another possible scenario manifested. Furthermore, waiting for scenario realisation without suitable preparations may neglect the opportunity for possible adaptations. In the proposed method, any specific cost would be postponed to the second stage and after scenario realisation. Thus, there is no need to pay some costs for some scenarios that would never be revealed.

In comparison with the Robust Optimisation (RO) approaches, the proposed two-stage approach avoids over conservatism by providing a chance for modification after the revelation of scenario. Furthermore, because of the complexity of multi-objective optimisation problems with deep uncertainty, identifying the worst case is hardly possible (if ever possible). The proposed methodology introduces the concept of *dynamic-robust decisions* by extending the robust decision philosophy. These kinds of decisions, reduce the shortcoming of the regular robust strategies such as generating a risk-averse solution because of a bad scenario. Also, the advantages of the robust solutions are developed.

The concepts, structures, and methodology that were introduced in this paper were generic and there will be various opportunities for further examination and improvements. Moreover, some challenges in theory and/or applications still need to be considered and addressed. Some of these challenges and the most interesting topics for future research can be listed as follows:

1. Extend to multi-stage problems: To be able to deal with the deep uncertainty in problems with more than two stages, the multi-stage structure or moving horizon algorithms could also be introduced as extensions of the two-stage framework. However, more uncertain statements bring more complexity to the problem. Therefore, we will need to indicate whether the better performance using a multi-stage model is significant enough to warrant the added computation.
2. Handling a large number of scenarios: The most challenging issue on this methodology is the limited number of scenarios that can be handled. Further study on this issue seems to be necessary and would improve the structure. Some scenario reduction algorithm or utilising a novel solution method may help to achieve this purpose.
3. Surprises are out of our expectations: This paper considers the surprises as plausible scenarios; however, it is possible that some surprises come from outside of the considered scenarios. Dealing with these kinds of uncertainty could be challenging and may need some changes in structuring. This could also be an introduction to integrating anti-fragility concept with MCDA and scenario planning.
4. Non-linear Problems: In this paper, the proposed structure is applied in only Linear Problems (LP), to avoid more complexity. Non-linear problems would be an important direction for future research. Different solution methods may also need to be examined to solve the non-linear problems.
5. Examine different solution methods: The *RPM* is used, in this study, to solve the MOLP problems. There are some other interesting methods, such as NAUTILUS and NIMBUS, that could contribute to the proposed methodology, and also some multi-objective evolutionary algorithms, such as NSGA.
6. Comparing with other approaches: Using other GP variants such as extended, meta or multi-choice GP as a comparison with RPM would be an interesting study. Moreover, comparing these models with stochastic (probability-based) models to examine effect/benefit of knowing the probability would also be interesting.

7. Setting too many goals and weights by the DM: As mentioned earlier, this study does not concentrate on the system of goal (and weight) choice. Considering all the meta-criteria in the proposed framework may lead us to solve problems with too many objectives. Therefore, the DM needs to set too many goals (and importance weights). In practice, eliciting much information from the DM is hardly possible, if ever. Accordingly, further study on this issue seems to be necessary and very interesting.
8. Real-life applications: As the proposed methodology has a very generic structure, it can be easily utilised in various real-world problems that face deep uncertainty and which include several conflicting criteria/objectives, such as socio-economic risks, environmental risk, climate change, portfolio management, scheduling, energy saving, power systems, water management, supply chain network design, energy retrofits, and decision support systems. However, some new application areas could be challenging and could require some restructuring of the model.
9. Application for basic OR problems: Expand and examine the proposed methodology in some basic OR problems such as multi-objective knapsack, assignment, and minimum cost/maximum flow problems under conditions of deep uncertainty. Applying the proposed methodology in these basic problems could be an interesting direction that may be challenging in a different way.

Acknowledgements We would like to thank the Department of Statistical Sciences and Faculty of Science for funding this work via the SCIENCE FACULTY Ph.D FELLOWSHIP 2014–2016.

References

Aouni B, Colapinto C, Torre DL (2014) Financial portfolio management through the goal programming model: current state-of-the-art. Eur J Oper Res 234(2):536–545. 60 years following Harry Markowitz's contribution to portfolio theory and operations research

Bankes SC (2002) Tools and techniques for developing policies for complex and uncertain systems. In: PNAS, vol 99, pp 7263–7266

Barbarosoglu G, Arda Y (2004) A two-stage stochastic programming framework for transportation planning in disaster response. J Oper Res Soc 55(1):43–53

Beale EML (1955) On minimizing a convex function subject to linear inequalities. J R Stat Soc Ser B (Methodological) 17(2):173–184

Belton V, Stewart T (2002) Multiple criteria decision analysis: an integrated approach. Springer, Berlin

Ben Abdelaziz F, Masmoudi M (2012) A multiobjective stochastic program for hospital bed planning. J Oper Res Soc 63(4):530–538

Charnes A, Cooper W (1959) Chance-constrained programming. Manag Sci 6:73–80

Charnes A, Cooper W, Ferguson R (1955) Optimal estimation of executive compensation by linear programming. Manag Sci 1:138–351

Charnes AC et al (1961) Management models and industrial applications of linear programming. Technical report. Wiley, New York

Clemen RT, Reilly T (2013) Making hard decisions with DecisionTools. 3rd edn, South-Western, Cengage learning

Courtney H (2001) 20/20 Foresight: crafting strategy in an uncertain world. Harvard Business School Press, Boston

Dantzig GB (1955) Linear programming under uncertainty. Manag Sci 1(3–4):197–206

Dantzig GB (1963) Linear programming and extensions. Princeton University Press, Princeton

Dantzig GB, Madansky A (1961) On the solution of two-stage linear programs under uncertainty. In: Proceedings of the fourth berkeley symposium on mathematical statistics and probability, vol. 1. University of California Press Berkeley, CA, pp 165–176
Dupa cová J (1986) Stability in stochastic programming with recourse. contaminated distributions. In: Prékopa A, Wets R-B (eds) Stochastic programming 84 part I, volume 27 of mathematical programming studies. Springer, Berlin, Heidelberg, pp 133–144
Durbach IN, Stewart TJ (2012) Modeling uncertainty in multi-criteria decision analysis. Eur J Oper Res 223(1):1–14
Goodwin P, Wright G (2001) Enhancing strategy evaluation in scenario planning: a role for decision analysis. J Manag Stud 38(1):1–16
Goodwin P, Wright G, Phillips LD (2004) Decision analysis for management judgment. Wiley, New York
Haasnoot M, Middelkoop H, van Beek E, van Deursen WPA (2011) A method to develop sustainable water management strategies for an uncertain future. Sustain Dev 19(6):369–381
Jones D, Tamiz M (2002) Goal programming in the period 1990–2000. In: Ehrgott M, Gandibleux X (eds) Multiple criteria optimization: state of the art annotated bibliographic surveys, vol 52. International series in operations research and management science. Springer, New York, pp 129–170
Jones D, Tamiz M (2010) Practical goal programming. International series in operations research and management science. Springer, Berlin
Jones D, Tamiz M (2016) A review of goal programming. In: Greco S, Ehrgott M, Figueira J (eds) Multiple criteria decision analysis. Springer, Berlin, pp 903–926
Khor CS, Elkamel A, Ponnambalam K, Douglas PL (2008) Two-stage stochastic programming with fixed recourse via scenario planning with economic and operational risk management for petroleum refinery planning under uncertainty. Chem Eng Process 47(9–10):1744–1764
Knight FH (1921) Risk, uncertainty and profit. Dover Publication, New York
Kuchta D (2004) Robust goal programming. Control Cybern 33(3):501–510
Kwakkel J, Walker W, Marchau V (2010) From predictive modeling to exploratory modeling: how to use non- predictive models for decisionmaking under deep uncertainty. In: 25th mini-EURO conference on uncertainty and robustness in planning and decision making (URPDM 2010), Coimbra-Portugal
Linderoth J (2003) Stochastic programming. Lecture Notes. Lehigh University, IE495
Martinez AJH, Aguado AG (1998) Stochastic goal programming wth recourse. Revista de la Real Academia de Ciencias Exactas, Fisicasy Naturales 92(4):409–414
Masmoudi M, Abdelaziz FB (2012) A recourse goal programming approach for the portfolio selection problem. INFOR Inf Syst Oper Res 50(3):134–139
Morgan M, Henrion M, Small M (1992) Uncertainty: a guide to dealing with uncertainty in quantitative risk and policy analysis. Cambridge University Press, Cambridge
Ogryczak W (1994) A goal programming model of the reference point method. Ann Oper Res 51(1):33–44
Quade ES (1989) Analysis for public decisions. Elsevier Science, New York
Ringuest JL (2012) Multiobjective optimization: behavioral and computational considerations. Springer, Berlin
Romero C (1991) Handbook of critical issues in goal programming. Pergamon Press, Oxford
Shapiro A, Homem-de Mello T (1998) A simulation-based approach to two-stage stochastic programming with recourse. Math Program 81(3):301–325
Shapiro A, Dentcheva D, Ruszczyński A (2009) Lectures on stochastic programming: modeling and theory. MOS-SIAM series on optimization. Society for Industrial and Applied Mathematics, SIAM, Philadelphia
Stewart TJ (1997) Scenario analysis and multicriteria decision making. In: Climaco J (ed) Multicriteria analysis. Springer, Berlin, pp 519–528
Stewart TJ, French S, Rios J (2013) Integrating multicriteria decision analysis and scenario planning-review and extension. Omega 41(4):679–688
Tamiz M, Jones D, Romero C (1998) Goal programming for decision making: an overview of the current state-of-the-art. Eur J Oper Res 111(3):569–581
Van Slyke R, Wets R (1969) L-shaped linear programs with applications to optimal control and stochastic programming. SIAM J Appl Math 17(4):638–663
Walker W, Harremoes P, Rotmans J, van der Sluijs J, van Asselt M, Janssen P, von Krauss MK (2003) Defining uncertainty: a conceptual basis for uncertainty management in model-based decision support. Integr Assess J 4(1):5–17

Walker WE, Haasnoot M, Kwakkel JH (2013) Adapt or perish: a review of planning approaches for adaptation under deep uncertainty. Sustainability 5(3):955–979

Wierzbicki AP (1982) A mathematical basis for satisficing decision making. Math Model 3(5):391–405 (Special {IIASA} Issue)

Wierzbicki AP (1998) Reference point methods in vector optimization and decision support. Technical report. International Institute for Applied Systems Analysis, Laxenburg