



# Balance in resource allocation problems: a changing reference approach

Özlem Karsu<sup>1</sup> · Hale Erkan<sup>2</sup>

Received: 6 March 2019 / Accepted: 5 February 2020 / Published online: 17 February 2020  
© Springer-Verlag GmbH Germany, part of Springer Nature 2020

## Abstract

Fairness is one of the primary concerns in resource allocation problems, especially in settings which are associated with public welfare. Using a total benefit-maximizing approach may not be applicable while distributing resources among entities, and hence we propose a novel structure for integrating balance into the allocation process. In the proposed approach, imbalance is defined and measured as the deviation from a reference distribution determined by the decision-maker. What is considered balanced by the decision-maker might change with respect to the level of total output distributed. To provide an allocation policy that is in line with this changing structure of balance, we allow the decision-maker to change her reference distribution depending on the total amount of output (benefit). We illustrate our approach using a project portfolio selection problem. We formulate mixed integer mathematical programming models for the problem with maximizing total benefit and minimizing imbalance objectives. The bi-objective models are solved with both the epsilon-constraint method and an interactive algorithm.

**Keywords** Bi-objective resource allocation problem · Fairness · Knapsack problem · Balance · Equity · Convex cones approach · Epsilon-constraint approach

## 1 Introduction

Allocating resources across multiple entities is a problem encountered in many real-life settings, hence resource allocation problems are widely studied in the operational research (OR) literature (Luss 2012). Various decision support systems have

---

✉ Özlem Karsu  
ozlemkarsu@bilkent.edu.tr

Hale Erkan  
hale.erkan@Mcombs.utexas.edu

<sup>1</sup> Industrial Engineering Department, Bilkent University, Ankara, Turkey

<sup>2</sup> McCombs School of Business, The University of Texas at Austin, Texas, USA

been proposed to help the decision-makers allocate resources so that the total output of the system is maximized.

Typically, resources are allocated so that the entities will enjoy benefits (outputs), i.e., any resource allocation to entities is associated with an output allocation. Resources and outputs can change based on the problem type: For example, in a healthcare resource allocation problem, resource can be budget and benefit can be measured with the number of people who benefit from a healthcare project or it can be measured by the quality adjusted life years<sup>1</sup> (QALYs) gained by the target population. Even if in some situations where there could be projects which have some negative impacts (e.g., inadvertent adverse consequences due to treatments in healthcare projects), we assume that the projects generate a single output, and this represents benefits. Hence, throughout this paper, we use the terms output and benefit and also resource and input interchangeably.

The one widely used approach in resource allocation is total output maximization, also known as the *utilitarian* approach. This method focuses on maximizing the total output regardless of how it is distributed across the entities. It achieves the best result in terms of the total output; however, it may fail to sustain fairness (balance) in the allocation. This is because this approach may ignore allocating resources to entities that are not as good as the others at converting resources to output. Consider a healthcare project selection problem, where projects are categorized by target patients' ages; resource usage is measured with the cost of a project and benefit is the resulting QALYs for a patient group. In this case, the utilitarian approach can result in a highly unfair resource distribution among categories. Elderly patients may not receive any resource, since these patient groups obtain relatively less benefit per unit resource devoted to them. Hence, using a utilitarian approach could be considered a highly controversial decision.

One approach to ensure fairness can be equally dividing the resources among entities, if possible. However, such an allocation may also be inapplicable, since it may result in a high efficiency loss (sacrifice from the best level of the total output). In the same healthcare resource allocation problem, a policy maker may also find a completely equal resource allocation to infants and elderly unfair or unacceptable on the grounds that the society should provide more resources to younger generations. Between these two extremes (a total output maximizing allocation and fairness maximizing allocation), there are other allocations, i.e., the problem is a bi-objective decision-making problem.

Moreover, as the example shows, fairness itself is a subjective concept, and hence it does not have a de facto definition. Depending on the problem setting and decision-maker, several different criteria can be listed to describe a "fair" allocation. In this study, we suggest an approach that will accommodate these differences while incorporating balance concerns into the resource allocation process. The proposed method enables decision-makers to define a perfectly balanced allocation (reference allocation) differently at different levels of output. In order to achieve this structure,

---

<sup>1</sup> QALY is a measure of health combining length and quality of life. It is widely used in studies focusing on evaluations of healthcare programs (Drummond et al. 2015)

a bi-objective mixed integer mathematical programming formulation is used. In addition, we engage both a posteriori and interactive solution approaches, which differ in terms of decision-maker's participation throughout the solution process.

In recent years, fairness has become an important criterion in many OR applications. It is almost a mandatory criterion in decisions affecting social welfare as in public sector decision-making (see, e.g., (Butler and Williams 2002; Kelly et al. 1998; Smith et al. 2013; Mestre et al. 2012; Heitmann and Brüggemann 2014; Karsu and Morton 2015 and the references therein for a review of such applications).

Equity-related concerns can be categorized as equitability and balance (Karsu and Morton 2015). Equitability aims to achieve an even distribution of the resources or outputs, since it assumes that the entities are indistinguishable, i.e., the identities of the entities do not affect the decision. Balance concerns, however, are relevant in cases, where the entities have different characteristics. They may, for example, differ in terms of productivity, need or claims. In such cases, a completely equal allocation may be undesirable and the decision-maker may want to allocate the resource in different proportions.

For the case where resource allocation across entities with different characteristics is considered, Karsu and Morton (2014) propose a bi-criteria framework to handle balance and total benefit maximization concerns. They suggest using a reference distribution approach to measure how balanced a given distribution is. Similarly, Stewart (2016) uses a reference point approach to solve a real-life multi-objective resource allocation problem where quantity, quality and balance are the main concerns.

In this paper, we follow this line of research and provide a methodology to help decision-makers to trade balance off against total benefit in resource allocation settings. We assume that the decision-maker has balance concerns for the output distribution and extend the idea of balance measurement in Karsu and Morton (2014) to a piecewise linear concept, which allows the decision-maker to change her reference proportion depending on the total output. The main contribution of this work in comparison with the previous work, especially the one by Karsu and Morton (2014), is the changing reference approach. This extension is a generalization of the previous work and would provide more flexibility in real-life applications. This generalization significantly changes the structure of the bi-objective programming models solved, as will be discussed in Sect. 2.

The rest of the paper is as follows: In Sect. 2 we first introduce the imbalance measurement approach with changing reference proportions at different total output intervals. We exemplify the use of the approach on knapsack-type discrete resource allocation problems and provide the corresponding mathematical models. In Sect. 3, we discuss a further extension of the approach, which uses interpolation of threshold reference proportion vectors to assign a unique reference proportion vector to each total output value as opposed to using a fixed reference proportion for the whole interval. In Sect. 4, we demonstrate the feasibility of the approach by solving example problems using the epsilon-constraint method and providing the results of our computational experiments. For settings where the decision-maker (DM) is willing to provide preference information, we discuss an interactive approach and report on

its performance. We conclude the discussion in Sect. 5 by summarizing our results and pointing out some further research directions that could be pursued.

## 2 Problem definition

### 2.1 Changing reference proportions in resource allocation

We measure balance with respect to desired proportions of the decision-maker and allow her to change these proportions as the total output or the total input changes. In this section, we discuss this idea of changing reference proportions in detail.

The underlying idea is similar to that of the well-known allocation rule from Babylonian Talmud, which divides a given resource in different proportions, changing with respect to the total amount of resource and demand (see Young 1995 for a detailed description). The story is as follows: A man dies leaving a heritage and credits to three different creditors with amounts of 100, 200, and 300 units. According to Talmud's rule, if the heritage is 100 units, each creditor gets 33.3 units (equal amounts); if heritage is 200 units then creditors receive 50, 75, 75 units, respectively, and if it is 300 units, they receive 50, 100, 150 units, respectively. In this allocation policy the received shares change depending on the total amount of heritage. When heritage is small, each creditor receives equal proportions; however, as the amount on hand increases their shares become proportional to their credits. If the decision-maker keeps the equal proportion policy in the third scenario, 100% of creditor one's credits will be paid, while only 33.3% of creditor three's credit will be covered. From creditor three's perspective, this allocation may not be acceptable, and therefore in such situations, shifting the distribution from equal amounts may improve fairness.

A similar situation may also be seen in investment planning. Armbruster and Delage (2015) and Eeckhoudt and Schlesinger (2006) state that investors may become more risk tolerant when the potential output is higher. As the expected benefit gets higher, the investor becomes more prudent to invest in more risky projects, and hence the reference allocation changes.

We will explain the novel approach that we suggest for imbalance measurement in resource allocation settings using a small example as follows:

Assume that a local healthcare provider is planning the annual budget allotment for a set of proposed projects, each of which targets a different population group. For simplicity, consider a case with two population groups, which are significantly different with respect to their life styles. Let us assume that the first and second groups represent people who are negligent and cautious about their health, respectively. (Such behavioral differences could emerge due to other attributes of the groups such as income.) The provider measures health gain in terms of the total scores based on QALYs.

If the total score that would result from the allocation decisions is relatively low, the policy makers may prefer to use the resources such that benefit is evenly distributed across population groups. Making the allocation in favor of the group

that returns more benefit per unit resource (in our example this corresponds to the second group) would create a significant gap between their healthcare status; thus it may not be appropriate for ethical reasons. Conversely, if total score is relatively high (e.g., between 70 and 100), the overall community health may be considered to be above standards. Since the overall health state is already good, having a more uneven distribution between population groups may be more acceptable. Both groups already have above satisfactory health status; hence, when benefit distribution shifts toward one of the groups, the difference would not be as striking as the first case. Therefore the policy maker may prefer an allocation where the second population group has higher returns in order to increase total health score. To summarize, distributing a good in a fair manner may be more important when the total welfare is already low. As the total welfare increases, the allocator may want to shift resources to more productive entities.

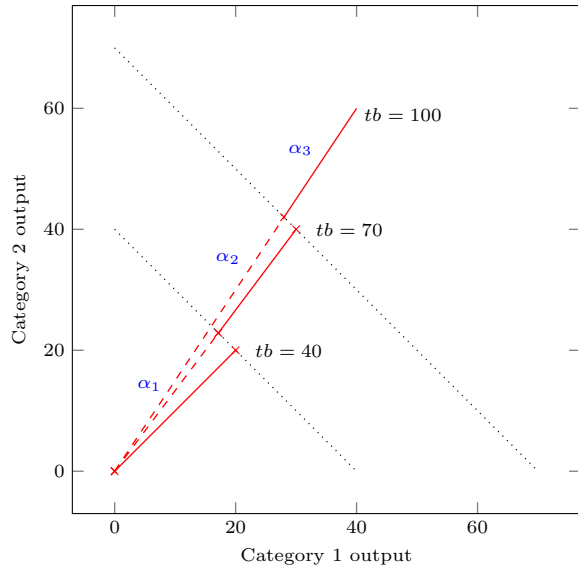
We illustrate the changing reference proportions in the example in Figure 1a, where there are two categories (population one and population two). The two axes represent the total benefit enjoyed by the categories, i.e., the total health score of each population. As long as the total benefit is below 40 units, decision-maker uses reference proportion  $\alpha_1$ . This reference proportion is a vector showing the desired benefit percentages for each group, in this example  $\alpha_1 = (0.5, 0.5)$ , i.e., if the total benefit is below 40 units, the policy makers would like to distribute it evenly across the two groups. When the total benefit exceeds 40 units, he/she changes the reference distribution vector from  $\alpha_1$  to  $\alpha_2 = (0.4, 0.6)$ , and in this representation, he/she prefers to see that the bigger portion of the benefit is in category 2. As the output increases, reference distribution becomes less even between categories.

This methodology can be applied to incorporate balance concerns in various resource allocation environments which can be formulated as: *continuous knapsack* problems, where benefits are obtained with respect to some production function, *discrete knapsack* problems, where costs and benefit parameters are explicitly given, or in *assignment problems* with capacity limitations. We illustrate the proposed approach using a discrete knapsack problem. Knapsack problem is a well-known combinatorial optimization problem, which has various real-life applications and is in the class of NP-hard problems (Kellerer et al. 2004). The problems that we formulate will trade balance off against total benefit, and hence they are bi-objective knapsack problems (see Carazo 2015 for a discussion on multi-criteria knapsack problems).

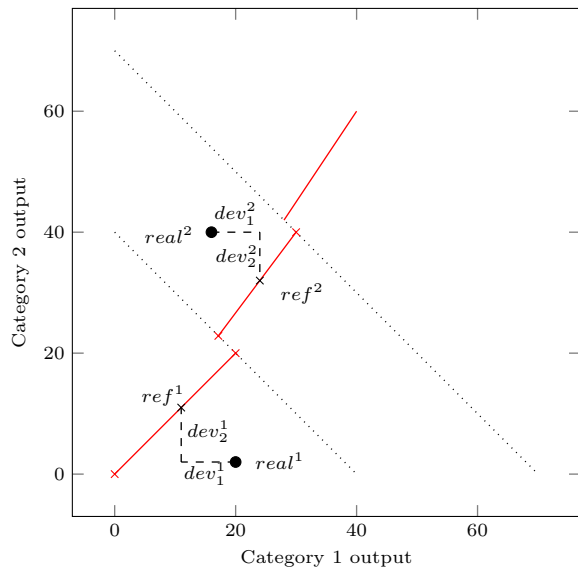
## 2.2 Mathematical model with interval-based reference proportions

We consider a bi-objective discrete knapsack problem to illustrate our resource allocation mechanism. In the problem setting, suppose there are  $N$  proposed projects and each project incurs a cost of  $c_i$  units and returns a benefit of  $b_i$  units. Projects are divided in different categories such that each project belongs to one (and only one) category. Let  $K$  be the number of categories and let  $K \leq N$ . We assume that the decision-maker has a limited budget  $B$ , which is not enough to fund all of the

**Fig. 1** Changing Reference Proportion Setting



**a** Changing reference proportions



**b** Distances from reference distributions

projects ( $\sum_{i=1}^N c_i > B$ ). Benefit and cost values of each project are assumed to be known, and hence productivity of each project is known a priori to the investments decision.

We define the binary variables  $\mathbf{x} = (x_1, \dots, x_N)$  as follows:  $x_i = 1$  if project  $i$  is selected in the portfolio and 0 otherwise. We assume that the degree of imbalance

of any distribution is measured by its distance to a reference allocation, which distributes the benefit according to DM's desired (reference) proportions. We allow the DM to determine different reference proportions for different levels of total benefit distributed. Assume that there are  $M$  threshold values defining  $(M - 1)$  intervals (interval  $m$  is defined by threshold levels  $T_m$  and  $T_{m+1}$ ). Reference proportion  $\alpha_{mk}$  is the desired proportion of total benefit allocated to category  $k = 1, \dots, K$  if the total benefit is in interval  $m = 1, \dots, (M - 1)$ . Reference proportion can increase/decrease as we move along on the intervals to shift the allocation in favor of particular categories.

Recall our health benefit example with two population groups. Assume that the groups are constructed based on the individuals' smoking habits (group 1: smokers and group 2: nonsmokers). For simplicity, assume that there are two healthcare project portfolios: Portfolio 1 would result in 50 units of benefit to group 1 and 10 units of benefit to group 2 while Portfolio 2 provides 5 and 15 units to these two groups, respectively. Consider the reference proportions of (0.5, 0.5) when the total benefit is less than 40 and (0.4, 0.6) when it is more than 40 (for higher total benefit levels the DM would rather encourage the nonsmokers by allocating more benefit to that group). Then, at portfolio 1's total benefit level, the DM would rather have  $(60 \cdot 0.4, 60 \cdot 0.6) = (24, 36)$ . We call this *reference allocation* and actual allocation of (50, 10) the *realized allocation*. Similarly, at Portfolio 2's total benefit level (20), the DM would rather distribute the benefit equally so its reference allocation vector is  $(20 \cdot 0.5, 20 \cdot 0.5) = (10, 10)$  while the realized allocation is (5, 15).

Imbalance of a realized allocation is measured as the total component-wise deviation from reference distribution of that allocation. That would be  $(|50 - 24| + |10 - 36|) = 52$  for Portfolio 1 and  $(|5 - 10| + |15 - 10|) = 10$  for Portfolio 2. As seen, Portfolio 2 is more in line with the DM's allocation preferences in terms of balance but it has less total benefit than the first one.

Figure 1b shows the component-wise deviations of two realized allocations from their reference allocation vectors. Realized allocations are represented with  $\text{real}^1$ ,  $\text{real}^2$ , and  $\mathbf{x}^1$ ,  $\mathbf{x}^2$  are the corresponding decision variable vectors. These two realized allocations have total benefits lying in different intervals, and their imbalance level will be calculated with respect to their own reference proportion vectors. By using the corresponding  $\alpha$  values, we calculate reference distributions as follows;  $\text{ref}_k^1 = \alpha_{1k} \sum_{i=1}^N b_i x_i^1$  and  $\text{ref}_k^2 = \alpha_{2k} \sum_{i=1}^N b_i x_i^2$ ,  $k = 1, 2$ . Component-wise deviation from reference distribution can be measured as  $\text{dev}_k^j = |\text{real}_k^j - \text{ref}_k^j|$   $j = 1, 2$   $k = 1, 2$ . Then,  $\text{imbalance}^j = \sum_{k=1}^K \text{dev}_k^j$   $j = 1, 2$ .

Notations used throughout the paper for the mathematical formulations are introduced below:

### Problem parameters

- $N$ : Number of projects
- $K$ : Number of categories
- $M$ : Number of thresholds
- $B$ : Total budget
- $c_i$ : Cost of project  $i = 1, \dots, N$

- $b_i$ : Benefit of project  $i = 1, \dots, N$   
 $T_m$ : Threshold values defining intervals for total benefit values  $m = 1, \dots, M$   
 $\alpha_{mk}$ : Reference (desired) proportion for category  $k$  in interval  $m$   
 $k = 1, \dots, K \quad m = 1, \dots, (M-1) \quad (\alpha_m \in R^K, \sum_{k=1}^K \alpha_{mk} = 1)$   
 $g_{ik} = \begin{cases} 1, & \text{if project } i \text{ is accepted} \\ 0, & \text{otherwise} \end{cases}$   
 $TB$ : Total benefit of all the proposed projects  $\sum_{i=1}^N b_i$   
 $\Delta T_m$ : Difference between two consecutive threshold values,  $T_{m+1} - T_m$ ,  
 $m = 1, \dots, (M-1)$

### Decision variables

- $x_i = \begin{cases} 1, & \text{if project } i \text{ is accepted, } i = 1, \dots, N \\ 0, & \text{otherwise} \end{cases}$   
 $\bar{X}_m$ : Amount of benefit gained within the interval  $m = 1, \dots, (M-1)$   
 $y_m = \begin{cases} 1, & \text{if} \\ 0, & \text{otherwise} \end{cases}$   
 $\alpha_k$ : Selected reference proportion for each category  $k = 1, \dots, K$ ,  
 $(\sum_{k=1}^K \alpha_k = 1)$   
 $Imbalance$ : Imbalance indicator of the benefit distribution

The proposed bi-objective interval-based reference proportion model (IRPM) is provided below:

$$\max \sum_{i=1}^N b_i x_i, \quad \min \quad Imbalance \quad (1)$$

$$\sum_{i=1}^N c_i x_i \leq B \quad (2)$$

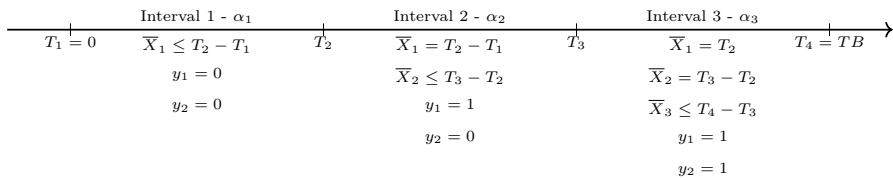
$$\sum_{i=1}^N b_i x_i = \sum_{m=1}^{(M-1)} \bar{X}_m \quad (3)$$

$$\bar{X}_1 \leq \Delta T_1 \quad (4)$$

$$\bar{X}_m \geq \Delta T_m y_m \quad m = 1, \dots, (M-2) \quad (5)$$

$$\bar{X}_m \leq \Delta T_m y_{m-1} \quad m = 2, \dots, (M-1) \quad (6)$$

$$\alpha_k = \alpha_{1k}(1 - y_1) + \sum_{m=1}^{M-3} \alpha_{m+1k}(y_m - y_{m+1}) + \alpha_{M-1k} y_{M-2} \quad k = 1, \dots, K \quad (7)$$



**Fig. 2** Indicating intervals

$$\sum_{k=1}^K \alpha_k = 1 \quad (8)$$

$$\text{Imbalance} = \sum_{k=1}^K \left| \alpha_k \sum_{i=1}^N b_i x_i - \sum_{i=1}^N g_{ik} b_i x_i \right| \quad (9)$$

$$x_i \in \{0, 1\} \quad i = 1, \dots, N \quad (10)$$

$$\bar{X}_m \geq 0, \quad m = 1, \dots, (M-1) \quad (11)$$

$$y_m \in \{0, 1\} \quad m = 1, \dots, (M-2) \quad (12)$$

Constraint (2) ensures that the budget is not exceeded. Constraint (3) is used to calculate the total benefit of the projects which are selected in the portfolio. Constraints (4)–(6) are used to identify the interval of total benefit. If total benefit corresponds to interval  $m$  then  $\{y_j = 1 \quad \forall j < m\}$  and  $\{y_j = 0 \quad \forall j \geq m\}$ . An example setting with three intervals and four threshold points is presented in Fig. 2. Initial and last threshold values are set to the lower and upper bounds of total benefit value, respectively. In solutions where total benefit value ( $\sum_{i=1}^N b_i x_i$ ) is smaller than  $T_2$ ,  $\bar{X}_1 < \Delta T_1$ ,  $y_1 = y_2 = 0$  and reference proportion vector  $\alpha_1$  is used. When total benefit distributed exceeds  $T_2$ ,  $\bar{X}_1$  is exactly equal to  $\Delta T_1$ ,  $y_1 = 1$  and reference proportion vector  $\alpha_2$  is used. Lastly, if the solution has a total benefit value greater than  $T_3$ , then both interval indicators;  $y_1$  and  $y_2$  take value 1, and reference proportion vector  $\alpha_3$  is assigned to the allocation.

The reference proportions change at the thresholds, and at each threshold, one of the two consecutive proportion vectors should be used. Without loss of generality, we assume that at each threshold  $T_m$ ,  $\alpha_m$  will be used. If total benefit can only have integer values, subtracting a small value,  $0 < \epsilon < 1$ , from the threshold values restrains any solution from achieving a threshold value. If the total benefit does not necessarily have integer values, one can introduce binary variables to control whether total benefit distributed is equal to a threshold value and additional constraints to indicate which reference proportion vector should be assigned in such cases.

Constraint (7) makes sure that  $\alpha$  is equal to the reference proportion vector of the corresponding interval. Imbalance is measured in (9), and it is calculated as the total deviation from the reference benefit distribution. The desired (reference) and actual benefits in category  $k$  are  $\alpha_k \sum_{i=1}^N b_i x_i$  and  $\sum_{i=1}^N g_{ik} b_i x_i$ , respectively, for  $k = 1, \dots, K$ . We add up the absolute difference between desired and realized benefits over all categories.

Constraint (9) involves nonlinear terms. We linearize them using additional auxiliary variables:  $w_{ki}$  and  $d_k$ . We delete constraint (9) and add the following:

$$w_{ki} \leq x_i \quad k = 1, \dots, K \quad i = 1, \dots, N \quad (13)$$

$$w_{ki} \leq \alpha_k \quad k = 1, \dots, K \quad i = 1, \dots, N \quad (14)$$

$$w_{ki} \geq \alpha_k - (1 - x_i) \quad k = 1, \dots, K \quad i = 1, \dots, N \quad (15)$$

$$\sum_{i=1}^N b_i w_{ki} - \sum_{i=1}^N g_{ik} b_i x_i \leq d_k \quad k = 1, \dots, K \quad (16)$$

$$\sum_{i=1}^N g_{ik} b_i x_i - \sum_{i=1}^N b_i w_{ki} \leq d_k \quad k = 1, \dots, K \quad (17)$$

$$\text{Imbalance} = \sum_k d_k \quad (18)$$

$$w_{ki} \geq 0 \quad k = 1, \dots, K \quad i = 1, \dots, N \quad (19)$$

Constraints (13)–(15) are used to linearize the product of  $\alpha_k$  (continuous) and  $x_i$  (binary) decision variables such that  $w_{ki}$  is equal to this product ( $w_{ki} = \alpha_k x_i$ ).

With constraints (16)–(17) and the auxiliary variables  $d_k$ , we rewrite the absolute deviation value term with linear inequalities. New variable  $d_k$  is the deviation in category  $k$  from the reference distribution. This linearization works when  $d_k$ s are minimized in the objective function. If one has another model in which this is not the case, alternative linearization techniques should be used.

In the linearized model, there are  $3KN + 2M + 2K + 2$  constraints (excluding the set constraints) and  $N + 2M + 2K + NK - 2$  variables,  $N + M - 2$  of which are binary.

There is a well-known alternative approach to model such piecewise problems (Williams 2013). However, we use an incremental formulation that is computationally more efficient. We provide the alternative formulations and compare these formulations in terms of solution times in Appendix A. The results also justify our modeling choice.

In this formulation, each project is assumed to yield benefits that are relevant to one category only. The proposed approach can be easily extended for the more

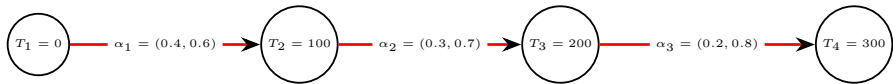


Fig. 3 Example Setting

general case in which each project generates benefits in several categories. This is possible, for example, by apportioning the projects' total benefits to different categories by using fractional amounts. Such an extension could be built by relaxing the binary parameter  $g_{ik}$  such that  $g_{ik} \in [0, 1]$  and  $\sum_k g_{ik} = 1, \forall i \in \{1, \dots, N\}$ .

### 3 Moving reference proportion model (MRPM)

In IRPM, there are multiple reference proportion vectors and the model assigns an  $\alpha_k$  based on the interval of the total benefit. The same reference distribution is used for the entire interval between two thresholds, regardless of its distance from the threshold points. Although this may seem reasonable, in some cases, it may result in abrupt changes in the solutions, especially for the ones with total benefit closer to the thresholds.

Let us consider the allocation policy in Fig. 3 with two categories. Until total benefit reaches  $T_2 = 100$  units,  $\alpha_1$  is  $(0.4, 0.6)$  and afterward it changes to  $(0.3, 0.7)$ . Suppose that there are two Pareto solutions with total benefit values 96 and 102 units. In the first solution, with 96 units of benefit, benefits are distributed as  $(38, 58)$  units between categories. In the second solution, total benefit is in the second interval, and hence reference proportion vector shifts to  $(0.3, 0.7)$  and benefit distribution becomes  $(31, 71)$  units. In the second solution, the benefit of category 1 decreases by 18% while benefit of category 2 increases by 22%. Even though total benefits are close to each other, these two distributions have significant differences. It may be expected that as the total benefit increases, each categories' benefit will also increase; however, in the neighborhood of thresholds, results do not coincide with this expectation. For a fact, these increases/decreases are inevitable due to the selected reference distribution coefficients, while one category's share is increasing, other's have to decrease. Thus in the threshold neighborhoods, there are sudden jumps from one distribution to the other, which may be undesired.

In order to address this drawback, we enhanced our model to move  $\alpha_k$  values from one threshold point to the other by using interpolation. For each total output value, the reference proportion vector is calculated uniquely by taking convex combination of two adjacent reference proportion vectors. This way, benefits of entities will gradually increase/decrease as the total benefit changes. In the previous example; where we have 96 units of benefit the corresponding reference proportion will be calculated using equation  $\alpha_k = \bar{X}_1 \frac{(\alpha_{2k} - \alpha_{1k})}{T_2 - T_1} + \alpha_{1k}$ , in this example it is equal to  $(0.304, 0.696)$ . In the second solution, with 102 units of benefit, reference proportion

becomes (0.298, 0.702). Consecutively, benefit distributions are (29, 71) and (30, 70), respectively. By moving the reference proportion vector from one threshold to the other, we can achieve a more moderate change in distributions.

We modify IRPM to formulate this change. Reference proportion constraint (7) is reformulated as below, and this adjusted model is referred as the moving reference proportion model (MRPM).

$$\alpha_k = (1 - y_1) \left[ \frac{\bar{X}_1(\alpha_{2k} - \alpha_{1k})}{\Delta T_1} + \alpha_{1k} \right] + \sum_{m=2}^{M-2} (y_{m-1} - y_m) \left[ \frac{\bar{X}_m(\alpha_{m+1k} - \alpha_{mk})}{\Delta T_m} + \alpha_{mk} \right] + y_{M-2} \alpha_{M-1k} \quad k = 1, \dots, K$$

This term is nonlinear due to the product of variables  $y_m$  (binary) and  $\bar{X}_m$  (continuous). We linearize it with the additional decision variable  $t_{mm}$ , and the new formulation contains the constraints of the IRPM models except constraint (7) with the following:

$$\begin{aligned} \alpha_k = & \left[ \bar{X}_1 \frac{(\alpha_{2k} - \alpha_{1k})}{\Delta T_1} + \alpha_{1k} \right] - \left[ t_{11} \frac{(\alpha_{2k} - \alpha_{1k})}{\Delta T_1} + y_1 \alpha_{1k} \right] \\ & + \sum_{m=2}^{M-2} \left[ \left( \frac{t_{m-1m}(\alpha_{m+1k} - \alpha_{mk})}{\Delta T_m} + y_{m-1} \alpha_{mk} \right) \right. \\ & \left. - \left( t_{mm} \frac{(\alpha_{m+1k} - \alpha_{mk})}{\Delta T_m} + y_m \alpha_{mk} \right) \right] + y_{M-2} \alpha_{M-1k} \quad k = 1, \dots, K \end{aligned} \quad (20)$$

$$t_{mm} \leq \bar{X}_m^{\text{UB}} y_m \quad m = 1, \dots, (M-2) \quad (21)$$

$$t_{mm} \leq \bar{X}_m \quad m = 1, \dots, (M-2) \quad (22)$$

$$\bar{X}_m - \bar{X}_m^{\text{UB}} (1 - y_m) \leq t_{mm} \quad m = 1, \dots, (M-2) \quad (23)$$

$$t_{m-1m} \leq \bar{X}_m^{\text{UB}} y_{m-1} \quad m = 2, \dots, (M-2) \quad (24)$$

$$t_{m-1m} \leq \bar{X}_m \quad m = 2, \dots, (M-2) \quad (25)$$

$$\bar{X}_m - \bar{X}_m^{\text{UB}} (1 - y_{m-1}) \leq t_{m-1m} \quad m = 2, \dots, (M-2) \quad (26)$$

$$t_{mm} \geq 0 \quad m = 1, \dots, (M-2) \quad (27)$$

$$t_{m-1m} \geq 0 \quad m = 2, \dots, (M-2) \quad (28)$$

In linearization, upper and lower bounds on  $\bar{X}_m$  are set to  $\Delta T_m$  and 0, respectively (The upper bound is denoted as  $\bar{X}_m^{\text{UB}}$ ). This model has  $3KN + 8M + 2K - 13$  constraints and  $N + 4M + 2K + NK - 7$  decision variables.

It is possible to make improvements to the formulations by introducing simple valid inequalities. Recall the binary variable  $y_m$ , which is used as the interval indicator as follows: When total benefit is in interval  $m$ ,  $\{y_j = 1 \quad \forall j < m\}$  and  $\{y_j = 0 \quad \forall j \geq m\}$ . Hence, whenever  $y_m = 1$ , we have  $y_j = 1 \forall j \leq m-1$  and we tighten the models by adding the following constraints:

$$y_m \geq y_{m+1} \quad m = 1, \dots, M-3 \quad (29)$$

Models IRP and MRP represent the general formulation of the proposed bi-objective resource allocation problem. We conducted preliminary experiments to analyze the effect of improvements suggested. We have observed that the percentage improvement in solution time is positive in most instances, and hence we use the model variants with these improvements in our main experiments for both IRPM and MRPM. From now on, IRPM and MRPM will refer to their variants with the improvements.

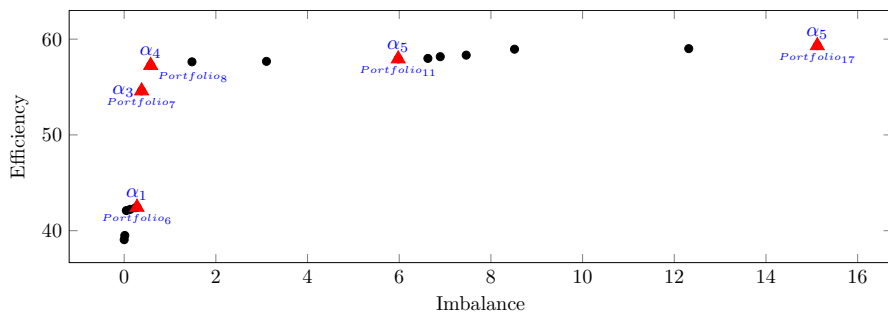
In Appendix A, we provide alternative formulations for this problem as discussed before and show that the formulation we suggest outperforms these alternative formulations in most instances.

## 4 Computational experiments

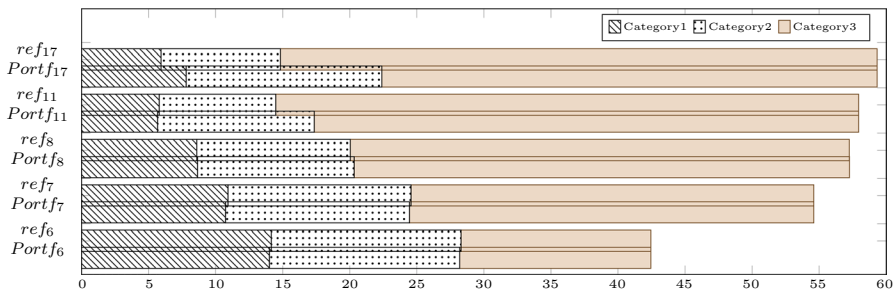
### 4.1 An illustrative example

As an example, we solve IRPM for a real-life R&D project selection problem of a public sector agency, where the projects belong one of the three sector-based categories (this problem is considered from an input-oriented aspect and with a single reference allocation in Karsu and Morton 2014). There are 39 projects, whose cost and benefit (output) values are assumed to be known. The budget is taken as 45% of the total cost of all the projects available. We determine four thresholds and solve the bi-objective problem using the epsilon-constraint approach (see Appendix B for a pseudocode of the approach). In the example, without loss of generality, it is assumed that as total benefit distributed increases DM prefers more uneven distributions that encourage more productive categories (Category 3). Figure 4 shows the set of all nondominated solutions, and Fig. 5 shows the realized and the reference benefit distributions of five example solutions.

Figure 4 shows some example portfolios from different parts of the frontier and the  $\alpha$  vectors used to assess imbalance for these portfolios. In Fig. 5 we also see how the output would be distributed if it were possible to realize the desired proportions. On the left hand side of the Pareto front, we have an example balanced allocation, Portfolio<sub>6</sub>, with 42.45 units of total benefit which uses reference



**Fig. 4** All Pareto Solutions



**Fig. 5** Benefit distribution among categories

proportion vector  $\alpha = \alpha_1 = (0.33, 0.33, 0.33)$ . This solution ensures an even benefit allocation across the three categories, as desired. Two solutions in the middle, Portfolio<sub>7</sub> and Portfolio<sub>8</sub> have higher total benefit than Portfolio<sub>6</sub>. The realized distributions in these portfolios slightly deviate from their reference distributions, but they are still close. Portfolio<sub>11</sub> is another example that has higher total benefit than the previous portfolios. On the other extreme, there is the total benefit-maximizing solution, Portfolio<sub>17</sub>, with 59.32 units of benefit and a higher imbalance value. In this portfolio, one can see that the realized distribution notably deviates from the reference. However, it is the best solution given this total benefit level since the model ensures that for any total benefit level, a maximally balanced solution is returned. As we move toward the maximum total benefit extreme, outputs of category 3 are significantly higher than those of the first two categories in the reference distributions and hence in the realized distributions. By presenting such Pareto solutions, we allow DMs to extensively understand the trade-off between the total benefit maximizing and most balanced solutions and to which extent their desired (reference) allocations can be realized in each interval. The DM can choose from the given set of alternatives based on her preferences. Moreover, such visualizations help to structure discussions and negotiations between different parties that might be affected by allocation decisions.

Throughout the paper, we make the analysis assuming that DM encourages more productive categories in higher output levels. However, it is also possible to

implement a reversed policy meaning that, in certain problem settings, DM may prefer more even distributions when total benefit is higher. Such a change in the policy would not require any adjustments in the model other than changing  $\alpha_{mk}$  parameters.

## 4.2 Data generation

In this part, we discuss the computational experiments that we performed to demonstrate the feasibility of the approach. We generated problem instances and engaged the epsilon-constraint method to solve the resulting bi-objective programming problems. We first discuss the details of our data generation scheme and then report the results of our computational tests.

In different decision-making environments, parameters may have certain characteristics; for instance, relation between cost and benefit values can vary or productivity of projects may differ depending on the category of each project. In order to reflect special properties of different problem settings we performed computational experiments on different problem instances.

In order to mimic different types of project portfolio selection problems that could be encountered in real life and to analyze the effect of parameters on solution times, we considered three different problem types. In all problem types, cost values have discrete uniform distribution with integer parameters 1 and  $R$ ,  $R \in \{30, 100\}$ , ( $c_i \sim \text{discrete uniform}(1, R) \ i = 1, \dots, N$ ). Benefit values are generated with a different policy in each problem type. In type A instances, benefit is generated randomly, while in types B and C, it is generated using a linear function of cost. Problem types are described below.

**Uncorrelated Instances (Type A):** Benefit values are also randomly distributed between  $[5, R]$ , i.e.  $b_i \sim \text{discrete uniform}(5, R) \ i = 1, \dots, N$ . In this type, there is no correlation between cost and benefit values, i.e., productivity of the projects are randomly determined.

**Strongly Correlated Instances (Type B):** In this type, benefit values are positively correlated with the cost values.  $b_i = tc_i + a \ i = 1, \dots, N$  where  $t \in \mathbb{R}^+$  and  $a \sim \text{discrete uniform}(0, R/10)$ .  $t$  is an indicator of productivity and  $a$  is used to have some diversity among projects. In our experiments, we set  $t$  as 1.2.

**Category-Based Strongly Correlated Instances (Type C):** In this type, each category has a different level of productivity, and hence  $t_k$  is used instead of  $t$ .  $b_i = \sum_{k=1}^K g_{ik} t_k c_i + a \ i = 1, \dots, N$  (Recall that  $g_{ik}$  is 1 if project  $i$  belongs to category  $k$  and 0 otherwise). In our experiments we set  $t_1 = 1.2, t_2 = 1.5, t_3 = 1.8$  when  $K = 3$  and additionally  $t_4 = 2.0$  when  $K = 4$ .

In all of the generated instances budget,  $B$ , is set to  $\lfloor \frac{\sum_{i=1}^N c_i}{2} \rfloor$ . We generated problem instances with 3 and 4 categories and assumed there are 5 intervals defined by 6 threshold points. The threshold levels defining the intervals are determined as a function of  $TB = \sum_{i=1}^N b_i$ . The  $\alpha_m$  and  $T$  values are provided in Table 1. The threshold levels and  $\alpha_m$  values used are for illustrative purposes. In real life, other levels could be used for these parameters.

**Table 1** Reference  $\alpha_m$  and corresponding  $T_m$  values

Interval ( $m$ )	$\alpha_m$		$T_m$
	$K = 3$	$K = 4$	
1	(0.33, 0.33, 0.33)	(0.25, 0.25, 0.25, 0.25)	0–0.3TB
2	(0.30, 0.30, 0.40)	(0.20, 0.21, 0.26, 0.33)	0.3TB–0.45TB
3	(0.20, 0.25, 0.55)	(0.15, 0.18, 0.27, 0.40)	0.45TB–0.6TB
4	(0.15, 0.20, 0.65)	(0.12, 0.15, 0.30, 0.43)	0.6TB–0.75TB
5	(0.10, 0.15, 0.75)	(0.10, 0.12, 0.33, 0.45)	0.75TB–TB

The algorithms are coded in Java and the computational studies are performed on Eclipse. All mathematical models are solved using CPLEX 12.8. All of the instances are solved in a computer with Intel Xeon CPU E5-1650 3.6 GHz processor and 32 GB RAM. Computation times are given in central processing unit seconds (CPU).

### 4.3 Epsilon-constraint method results

We first used the epsilon-constraint method (Haimes et al. 1971) to find the set of nondominated solutions (A solution is nondominated if it is not possible to find another solution which is at least as good as it with respect to both total benefit and balance). Starting from the nondominated solution that has the minimum imbalance value, in each iteration it finds a new solution by minimizing imbalance and restricting total benefit by a constraint. In each iteration, two single objective models are solved (min imbalance and  $\max \sum_{i=1}^N b_i x_i$ ) to ensure that a nondominated solution is found. At each iteration, the solution of the first model is feasible for the second model, which maximizes total benefit fixing the imbalance to the value found in the first one. We use this solution as an initial solution to the second model. Moreover, some of the interval-defining variables ( $y_m$ s) can be fixed since the total benefit in the second model will be at least as much as the total benefit found by solving the first model.

Similarly, in the models where the total benefit level is restricted via an epsilon-constraint, some information on potential intervals is also available, e.g. some of the lower intervals are not possible anymore. Therefore, we fix the values of several variables in the model that are used to select the interval (see the algorithm description in Appendix B). We performed our main computational experiments exploiting this problem specific property when using the epsilon-constraint method.

Any algorithm that is designed for bi-objective integer programming problems can be used to find the set of nondominated points (see, e.g., Chalmet et al. 1986; Ralphs et al. 2006 for algorithms based on weighted sum and Tchebycheff scalarizations, respectively. See also Boland et al. 2015 for a recent solution algorithm and the references therein for further examples.)

Details of the epsilon-constraint algorithm are provided in Appendix B. All benefit parameters are integer in our problem instances, therefore it is possible to

**Table 2** IRP and MRP model results

<i>N</i>	<i>R</i>	<i>K</i>	Type	IRPM				MRPM			
				CPU time		<i>NS</i>		CPU time		<i>NS</i>	
				Mean	Max	Mean	Max	Mean	Max	Mean	Max
30	30	3	A	9.2	16	84.8	152	448.8	844	100.4	164
			B	6.8	13	9.2	13	326.6	589	16.4	20
			C	7.4	10	29	37	643.4	1247	30.8	50
		4	A	28.4	58	62	99	49.6	71	79	135
			B	14	21	17.4	31	55.8	93	24	41
			C	22.6	30	46.4	65	67.4	133	42.8	59
	100	3	A	7.2	12	75	101	450	823	97.2	122
			B	14.2	22	31.6	79	350.8	504	51	104
			C	15.8	22	59.4	97	628.4	1128	59.6	99
		4	A	26.4	51	57.4	99	50.6	66	77.4	140
			B	36.2	74	36	66	241.6	969	71.6	247
			C	80.8	169	129.4	197	284.2	649	113.4	199
50	30	3	A	911	2431	130	177	–	–	–	–
			B	9.8	14	8.6	12	–	–	–	–
			C	61.8	128	45.4	68	–	–	–	–
		4	A	–	–	–	–	–	–	–	–
			B	–	–	–	–	–	–	–	–
			C	941.6	3672	69.4	76	–	–	–	–
	100	3	A	1266.8	4291	129.6	188	–	–	–	–
			B	318.6	993	61.2	224	–	–	–	–
			C	97.25 (1)	120	138.25	200	–	–	–	–
		4	A	–	–	–	–	–	–	–	–
			B	2984.2	7703	63.8	213	–	–	–	–
			C	1164.25	2983	178.25	223	–	–	–	–

generate the whole set of nondominated solutions. Presenting the set of all nondominated solutions is useful for the DM to see the trade-off between the two criteria considered. It informs the DM on the effect of changing the portfolio from one efficient solution to the other.

We solve five instances for each problem type, *N*, *K* and *R* combinations, and report the average and maximum values of the solution time and number of exact nondominated solutions obtained, |*NS*|, for the IRP and MRP models, respectively.

Results of the IRPM and MRPM are provided in Table 2. We use a time limit of 3 h as follows: If an instance cannot be solved within 3 h, that is, if the whole set of Pareto solutions cannot be returned, we terminate that instance and its values are not included in the performance metrics. Number of instances that cannot be completed within the time limit is reported in parenthesis in the column reporting the average CPU time. If, for a set of instances, at least 3 out of 5 instances could not be solved

within the time limit, we do not report the results of the set. Such cases are indicated with “-” in the table.

Due to the additional constraints and binary variables, MRPM’s solution times are significantly higher than IRPM.  $|NSI|$  values also tend to increase in this approach, with exceptions in Type C problems. None of the larger MRPM instances with  $N = 50$  could be solved within the time limit, and hence we do not provide results for these cases.

Type A problem instances lead to relatively larger solution sets, making the total solution time higher than the other types in most cases. On the other hand, Type B problems have the least number of nondominated solutions, indicating a relation between the correlation structure and the number of nondominated solutions obtained.

There are two factors that affect the total computational time: the number of nondominated points (  $|NSI|$  ), which dictates the number of single objective models solved in the algorithm, and the time required for solving each single objective model. Knapsack problems’ solution times highly depend on the characteristic of parameter set. Some parameter sets are observed to be harder to solve. Pisinger (2005) analyses the performance of different algorithms on a variety of randomly generated knapsack problem instances, which differ in terms of the correlation between cost and benefit values of projects. He concludes that the easiest problem instances usually have uncorrelated benefit and cost values and the hardest problem instances have strongly correlated benefit and cost values. Our observations are in line with these conclusions: the time required per model is smallest in Type A instances and highest in Type B instances. Again, this seems to be because in Type B problems, cost and benefit values are correlated with a constant parameter  $t$ . When we set  $t$  category-specific as in Type C instances, we change the correlation structure between cost and benefit and obtain an intermediate setting between the two extremes of Type A (uncorrelated) and Type B (strongly correlated) instances. With a few exceptions, the number of nondominated solutions and time per model nondominated solution (time per single objective model) of Type C instances are between those of Type A and Type B instances.

Increasing the problem size ( $N$ ) makes the corresponding problems significantly more difficult to solve, as expected. Increasing the number of categories ( $K$ ) however has opposite effects in IRPM and MRPM. The MRPM becomes easier to solve when the number of categories is increased from 3 to 4, *ceteris paribus*. Moreover, it is observed that problems with a higher range parameter ( $R$ ) tend to have larger Pareto sets and higher solution times.

#### 4.4 Interactive approach

In the epsilon-constraint method, two models are solved to find each nondominated solution and many instances cannot be solved in reasonable time due to the high number of iterations required and rapidly increasing solution times. One way of handling this challenge is using metaheuristic algorithms that return a set of

approximate Pareto solutions. Such algorithms are computationally very efficient; however, the solutions are neither exact nor with a quality guarantee.

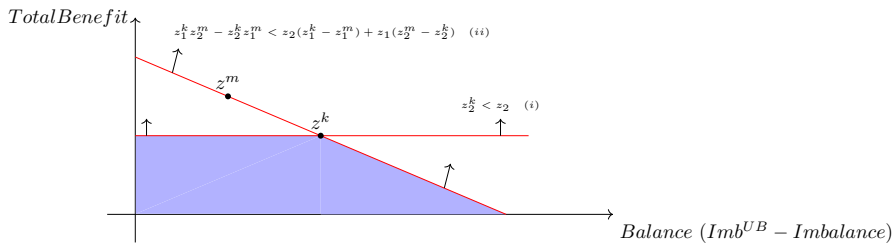
Whenever possible, one can use preference information from the DM and focus only on nondominated solutions that could be of interest to him/her. We now discuss such an interactive approach that could be used to guide the DM to his/her most preferred solution, without generating all the nondominated solutions beforehand. This approach requires less time as it does not generate the nondominated solutions that would not be of interest to the DM. It also helps the DM by guiding him/her to the most preferred solution as opposed to just presenting him/her the set of nondominated solutions, which may be too large.

The interactive algorithm we use is based on the approach proposed in Lokman et al. (2016) to find the most preferred alternative of a DM for multi-objective integer programming settings. We adjust this method to our bi-objective mixed integer programming problems.

The fundamental assumption of the algorithm is that the DM has a non-decreasing quasiconcave utility function, which is unknown. Given this assumption and preference information from the DM (mostly in terms of pairwise comparisons of alternative solutions), one can infer that some parts of the feasible criteria space would not contain the most preferred solution and hence can eliminate them from further consideration. The method is called the convex cone method and such areas are called (convex) cone-dominated areas (Korhonen et al. 1984).

The algorithm keeps an incumbent, which is the most preferred nondominated solution so far. At each iteration, a challenger is generated for the incumbent by solving scalarization model (a weighted sum model) and preference information between a new pair of solutions is obtained from the DM. Let  $z^m$  and  $z^k$  be such two nondominated solutions in the Pareto front. Then either  $z^m \succ z^k$  or  $z^k \succ z^m$ , where  $\succ$  denotes the binary relation “is preferred to”. Given this pairwise comparison, a 2-point cone is defined as follows:  $C(z^m, z^k) = \{z : z = z^k + \mu(z^m - z^k), \mu \geq 0\}$  if  $z^m \geq z^k$ . Korhonen et al. (1984) prove that if the DM has a quasiconcave utility function, a point in the cone cannot be better than the cone generators, which are  $z^m$  and  $z^k$ , in our example. Moreover, the corresponding cone-dominated region  $CD(z^m, z^k) = \{z' : z' \leq z \text{ for some } z \in C(z^m, z^k)\}$  (assuming both objectives are of a maximization type) cannot contain the most preferred solution for the DM and hence is eliminated from the criterion space. When there are multiple such pairwise comparisons, all the corresponding cone-dominated regions are eliminated. These steps are repeated until the most preferred solution is the only feasible solution to the problem. For the sake of simplicity, we formulate the problem such that both objectives are of maximization type by defining  $\text{Balance} = \text{Imb}^{\text{UB}} - \text{Imbalance}$ , where  $\text{Imb}^{\text{UB}}$  is an upper bound on imbalance. Let  $z_1(x)$  be the balance value of a solution  $x$  and  $z_2(x)$  be the total benefit value of a solution  $x$  (both to be maximized).

Figure 6 demonstrates a two-point cone and the corresponding cone-dominated region for our bi-objective problems. Assuming  $z^m \succ z^k$  and  $z_2^m > z_2^k$ , it directly follows that  $z_1^m < z_1^k$  (since  $z^k$  is nondominated). As shown by Lokman et al. (2016), any point  $z$  that is not cone-dominated has to satisfy at least one of the conditions (i), (ii), as otherwise it is in the cone-dominated region (shaded). The algorithm guarantees that a newly found nondominated solution is not dominated by any of the



**Fig. 6** Cone-dominated region when  $z^m > z^k$  (shaded)

cones generated so far, by using binary variables in the scalarization model. These variables enforce any new nondominated solution found by the scalarization model to satisfy at least one of the two conditions (i) and (ii). The pseudocode of the algorithm is provided in Appendix B.

We re-solved the problem instances with the interactive algorithm, using a quadratic function to represent DM's underlying utility function,  $\max \sum_{i=1}^2 -w_i^2 (z_i^f - z_i)^2$  where  $w = (0.7, 0.3)$  and  $z_i$  denotes the ideal point  $z_i^f = \max_{x \in X} z_i(x)$ ,  $i = 1, 2$ . This utility function is only used to simulate the answers of the DM to the pairwise comparison questions asked by the algorithm. The algorithm would work for any DM whose preference model is in line with a quasiconcave utility function. The most preferred solution and the performance of the interactive algorithm, in terms of the number of questions asked, is expected to change as the underlying preference model changes.

The results of these experiments are summarized in Tables 3 and 4. Instead of |NS|, we report the average values for the percentage of nondominated solutions found until determining the most preferred alternative. This percentage is calculated as follows:

$$\%NS = \frac{\text{Number of solutions found in the interactive algorithm}}{|NS|} \times 100$$

The number of solutions found is equal to the number of pairwise questions asked to the DM. Compared to the epsilon-constraint approach, in most of the cases this approach solves the problems in considerably less CPU time and number of iterations. This effect is even more prominent in Type C problems, where only a small portion of the set of nondominated solutions needs to be generated before identifying the most preferred solution. There are, however, exceptions to this observation: for example, for IRPM, in most problems with  $N = 50$ , the interactive algorithm requires more time than the epsilon-constraint approach. A similar situation occurs in MRPM for Type B instances with  $R = 100, K = 4$ . This may be due to the extra constraints preventing any newly found solution to be cone-dominated, making the models harder to solve.

We observe the effect of problem size in the performance of the interactive algorithm, too. As  $N$  increases, the solution time increases, making some of the problems

**Table 3** IRPM-interactive approach performance comparison

<i>N</i>	<i>R</i>	<i>K</i>	Type	Epsilon-con- straint approach		Interactive approach		
				CPU time		CPU time		%NS
				Mean	Max	Mean	Max	
30	30	3	A	9.2	16	1.4	6	11.79
			B	6.8	13	0.8	1	21.74
			C	7.4	10	1.4	2	10.34
		4	A	28.4	58	4.4	16	19.35
			B	14	21	3.8	13	26.44
			C	22.6	30	1.6	2	6.90
	100	3	A	7.2	12	0.6	2	12.53
			B	14.2	22	2.8	7	32.28
			C	15.8	22	1.8	3	7.74
		4	A	26.4	51	6.6	18	24.74
			B	36.2	74	3.2	7	13.33
			C	80.8	169	41.2	180	20.40
50	30	3	A	911	2431	4796.25	12,106	49.04
			B	9.8	14	30	93	18.29
			C	61.8	128	2022	9961	7.27
		4	A	–	–	712.75	1164	–
			B	–	–	105.66 (2)	205	–
			C	941.6	3672	52	122	5.67
	100	3	A	1266.8	4291	1159.33 (3)	2780	45.88
			B	318.6	993	25	42	4.24
			C	97.25 (1)	120	102	223	5.91
		4	A	–	–	2006.66 (2)	4347	–
			B	2984.2	7703	48.75	86	6.40
			C	1164.25	2983	215	718	3.45

unsolvable within the time limit of 3 h. “–” sign indicates that 3 out of 5 instances could not be solved within the time limit. Nevertheless, the interactive algorithm is able to solve more of the problem instances that could not be solved using the epsilon-constraint approach. This shows the value of using preference information, whenever possible.

## 5 Conclusion

We consider resource allocation problems, in which an indivisible resource is allocated across multiple entities so that they will enjoy benefits. Motivated by the fact that a pure total benefit-maximizing approach may not be appropriate in terms of balance, we suggest a bi-objective framework that trades balance

**Table 4** MRPM-interactive approach performance comparison,  $N = 30$ 

$R$	$K$	Type	Epsilon-constraint method		Interactive approach		
			CPU time		CPU time		%NS
			Mean	Max	Mean	Max	Mean
30	3	A	448.8	844	9.2	37	10.96
		B	326.6	589	5.4	16	12.20
		C	643.4	1247	2.4	4	6.49
	4	A	49.6	71	5.2	15	12.15
		B	55.8	93	21.2	56	30.00
		C	67.4	133	21	86	14.02
100	3	A	450	823	3.2	10	9.05
		B	350.8	504	9.6	38	6.67
		C	628.4	1128	56.4	253	15.77
	4	A	50.6	66	5.2	24	8.53
		B	241.6	969	–	–	–
		C	284.2	649	5.8	19	2.12

off against total benefit. We capture the cases where the decision-maker's understanding of balance changes as the total amount distributed changes. We exemplify the proposed method on project portfolio selection problems, where the projects belong to different categories and there is a concern for ensuring a balanced allocation of the benefit as well as maximizing total benefit. We solve the resulting bi-objective optimization problems using the epsilon-constraint method and also implement an interactive algorithm that finds the most preferred solution of a DM.

In real-life decision-making settings, some pragmatic challenges may be encountered in implementing the approach. We list some of these and potential ways to address these challenges: The proposed recommendation depends on the threshold levels. If these are not already available, they can be defined as a function of some problem dependent parameter (such as proportions of total benefit of the considered projects as we did in the computational experiments or proportions of the maximum attainable total benefit). If specific threshold values are difficult to obtain, one can perform sensitivity analysis by solving the problem multiple times, each time with a different choice of these parameters. One can then determine the winner portfolios in the sense that they are nondominated in most scenarios (different choices of parameters).

In some real-life problems arising in public domain, the benefits can be hard to ascertain in ways that are incontestable (given that there may be a need to make scoping choices concerning direct vs. indirect benefits, or benefits which are realized in the short vs. long term); thus, the elicitation of reference proportions could prove challenging in practice. Some of these challenges can still be addressed using the same structure by extending the problem dimensions. When multiple types of

quantifiable benefits exist (such as long term versus short term benefits), a multi-objective programming extension of the structure could be used, showing the trade-off between total benefit and balance in the allocations of multiple benefits. This would, of course, make the mathematical models more difficult to solve. In a problem where there are  $p$  benefits (e.g. short term and long term), the resulting models will have  $2p$  objective functions.

In this study, we suggested formulations and solution approaches for a setting where balance on the output distribution is sought. Balance could also be defined with respect to the resource devoted to the entities. Although it is very relevant and desirable to define balance with respect to the input in some settings, in some others it may not provide an agreeable solution. Especially in cases where the entities' productivities (i.e., their ability to convert resources to outputs) are very different, indirectly controlling the output allocation by managing the resource allocation may not be straightforward. However, whether one should ensure balance in the resource allocation or output allocation is context-specific. Observing and reporting on the difference between these two approaches is an interesting direction for future research and applications.

Considering balance concerns explicitly in the decision support systems will provide useful insights to the decision-makers on the alternative allocation options and the trade-off between total benefit and balance concerns. Further research can be performed on the multi-objective version of the problem, in which relationship among different balance and efficiency measures can be examined.

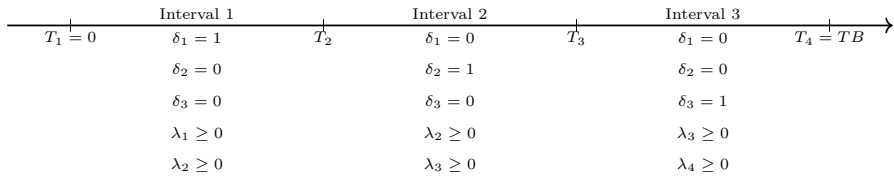
**Acknowledgements** This study was supported by TUBITAK (The Scientific and Technological Research Council of Turkey) under Grant Number: 215M713. We thank the anonymous reviewers for their thorough feedback and constructive comments, which led to substantial improvement of the manuscript.

## Appendix A: Alternative formulations

Recall the alternative formulations mentioned in Sect. 2.2. These are provided in this section.

### IRP alternative formulations

For IRP, alternative formulation 1 is a standard formulation that writes any total benefit value as a convex combination of the threshold values with the help of continuous variables ( $\lambda$ ) showing these coefficients and a binary variable vector  $\delta$  indicating the interval that the total benefit value belongs to. Alternative formulation 2 is similar to Alternative 1 but without the  $\lambda$  variables. Finally, we provide alternative formulation 3, which is similar to the proposed variant, but handles the interval fixing variables ( $y_m s$ ) in a different way.



**Fig. 7** Indicating intervals in alternative models 1 and 2

### IRPM alternative formulation 1

$$\max \sum_{i=1}^N b_i x_i, \quad \min \quad \text{Imbalance} \quad (2), (8), (10), (13) - (19) \quad (30)$$

$$\sum_{i=1}^N b_i x_i = \sum_{m=1}^M \lambda_m T_m \quad (31)$$

$$\lambda_m \leq \delta_{m-1} + \delta_m \quad m = 2, \dots, M-1 \quad (32)$$

$$\lambda_M \leq \delta_{M-1} \quad (33)$$

$$\sum_{m=1}^{M-1} \delta_m = 1 \quad (34)$$

$$\sum_{m=1}^M \lambda_m = 1 \quad (35)$$

$$\alpha_k = \sum_{m=1}^{M-1} \delta_m \alpha_{mk} \quad \forall k \in K \quad (36)$$

$$\delta_m \in \{0, 1\} \quad m = 1, \dots, (M-1) \quad (37)$$

$$0 \leq \lambda_m \leq 1 \quad m = 1, \dots, M \quad (38)$$

Figure 7 shows how the decision variables are set in these formulations.

**IRPM alternative formulation 2:** This formulation is the similar to Alternative 1 but without the  $\lambda$  variables.

$$\begin{aligned}
 & \max \sum_{i=1}^N b_i x_i, \quad \min \quad \text{Imbalance} \\
 & (2), (8), (10), (13) - (19), (34), (36), (37) \\
 & \sum_{m=1}^{M-1} \delta_m T_m \leq \sum_{i=1}^N b_i x_i
 \end{aligned} \tag{39}$$

$$\sum_{i=1}^N b_i x_i \leq \sum_{m=1}^{M-1} \delta_m T_{m+1} \tag{40}$$

**IRPM alternative formulation 3:** This formulation is a variant of the proposed formulation, where the  $y_m$  variables are set differently. We replace constraints (3)–(6) and 11 in the proposed formulation with constraints (41) and (42) and obtain the following model:

$$\begin{aligned}
 & \max \sum_{i=1}^N b_i x_i, \quad \min \quad \text{Imbalance} \\
 & (2), (7), (8), (10), (12) - (19)
 \end{aligned} \tag{41}$$

$$y_m \leq 1 - \frac{T_{m+1} - \sum_{i=1}^N b_i x_i}{\text{Big}M} \quad m = 1, \dots, (M-2)$$

$$\frac{\sum_{i=1}^N b_i x_i - T_{m+1}}{\text{Big}M} \leq y_m \quad m = 1, \dots, (M-2) \tag{42}$$

where  $\text{Big}M = \max_m \{T_{m+1}, TB - T_{m+1}\}$ .

## MRP alternative formulations

### MRPM alternative formulation 1

MRP alternative formulation 1 is the modified version of Alternative 1 for the IRPM, for the moving reference case. For MRP,  $\alpha$  is also a convex combination of threshold proportions. Hence the same coefficient variables ( $\lambda$ ) are used to determine  $\alpha$ . Alternatives 2 and 3 are the modified versions of IRP Alternatives 2 and 3 for the moving reference setting, respectively.

$$\begin{aligned}
& \max \sum_{i=1}^N b_i x_i, \quad \min \quad \text{Imbalance} \\
& (2), (8), (10), (13) - (19), (30) - (35), (37), (38) \\
& \alpha_k = \sum_{m=1}^{M-1} (\lambda_m \alpha_{mk}) + \lambda_M \alpha_{M-1k} \quad \forall k = 1, \dots, K
\end{aligned} \tag{43}$$

### MRPM alternative formulation 2

$$\begin{aligned}
& \max \sum_{i=1}^N b_i x_i, \quad \min \quad \text{Imbalance} \\
& (2), (8), (10), (13) - (19), (34), (37), (39), (40) \\
& \alpha_k = \sum_{m=1}^{M-2} \delta_m \left[ \frac{(\sum_i b_i x_i - T_m)(\alpha_{m+1k} - \alpha_{mk})}{\Delta T_m} + \alpha_{mk} \right] + \delta_{M-1} \alpha_{M-1k} \quad \forall k \in K
\end{aligned}$$

To linearize the multiplication: Define new continuous variables:  $h_{mi} = \delta_m \times x_i$  for all  $m = 1, \dots, M-2$  and  $i = 1, \dots, N$ . Add the following constraints:

$$h_{mi} \geq x_i + \delta_m - 1 \quad m = 1, \dots, M-2, i = 1, \dots, N \tag{44}$$

$$h_{mi} \leq \delta_m \quad m = 1, \dots, M-2, i = 1, \dots, N \tag{45}$$

$$h_{mi} \leq x_i \quad m = 1, \dots, M-2, i = 1, \dots, N \tag{46}$$

$$h_{mi} \geq 0 \quad m = 1, \dots, M-2, i = 1, \dots, N \tag{47}$$

### MRPM Alternative Formulation 3

$$\begin{aligned}
& \max \sum_{i=1}^N b_i x_i, \quad \min \quad \text{Imbalance} \\
& (2), (8), (10), (12) - (28), (41), (42)
\end{aligned}$$

Tables 5 and 6 show the comparisons of alternative formulations in terms of average and maximum solution times, for IRP and MRP models, respectively. In Table 5, it is observed that all formulations have very close solution times for smaller instances of IRP. As the problems get larger, the proposed formulation performs notably better in most settings but there is no clear winner. For  $N = 50$ ,  $R = 100$ ,  $K = 4$  instances, the alternative formulations failed to return solutions to some of the type B and C instances, while with the proposed formulation it was possible to solve them within the time limit. It is also clearly seen in Table 6 that alternative formulations 1 and 2 perform significantly worse than MRPM. The performances of MRPM and alternative 3, which is actually quite similar to MRPM, are comparable.

**Table 5** CPU time comparison among alternative IRP models

<i>N</i>	<i>R</i>	<i>K</i>	Type	IRPM		Alternative 1		Alternative 2		Alternative 3	
				Mean	Max	Mean	Max	Mean	Max	Mean	Max
30	30	3	A	9.2	16	9.75	21	8.8	16	8.6	19
			B	6.8	13	5	8	4.8	7	7.4	17
			C	7.4	10	7.75	11	7.2	10	7.4	10
		4	A	28.4	58	25.5	37	25.8	36	28.8	46
			B	14	21	15.5	22	17.2	26	16	23
			C	22.6	30	23.5	31	23.6	27	23.2	28
		100	A	7.2	12	7	12	6.6	12	6.6	12
			B	14.2	22	15.5	25	12.6	21	12.4	18
			C	15.8	22	18.5	25	16.8	22	16.2	22
50	30	4	A	26.4	51	26.25	43	26.4	43	28.8	60
			B	36.2	74	38	73	36.8	84	39.6	85
			C	80.8	169	77.25	163	74	148	81.6	163
		3	A	911	2431	731.25	2793	1183	4184	890.4	3673
			B	9.8	14	56	149	173.4	645	33	97
			C	61.8	128	677.25	2163	238.8	495	381.6	964
		4	A	–	–	–	–	–	–	–	–
			B	–	–	–	–	–	–	–	–
			C	941.6	3672	1059.5	3173	486.6	1372	2420.8	6374
		100	A	1266.8	4291	584.5	1781	1285.2	2609	2451.6	8756
			B	318.6	993	1517.5	5971	1882.8	8750	547.2	2491
			C	97.25 (1)	120	117 (1)	151	93.5 (3)	110	103 (1)	118
		4	A	–	–	–	–	–	–	–	–
			B	2984.2	7703	147.25 (1)	278	143.5 (1)	229	2467.25 (1)	9470
			C	1164.25	2983	615.25 (1)	1435	400.75 (1)	10,834	1051 (1)	3286

## Appendix B: Solution methods

We used the well-known epsilon-constraint method to obtain exact Pareto front. Additionally, we used an interactive approach to find the most preferred solution. Both methods are provided below.

Let the generic model be:

$$\min_{x \in X} \text{ Imbalance, } \max \sum_{i=1}^N b_i x_i$$

where  $x$  and  $X$  denote the decision variable vector and the feasible region, respectively.

**Table 6** CPU time comparison among alternative MRP models,  $N = 30$ 

$R$	$K$	Type	MRPM		Alternative 1		Alternative 2		Alternative 3	
			Mean	Max	Mean	Max	Mean	Max	Mean	Max
30	3	A	448.8	844	1049.6	2377	2582.2	4956	349.4	751
		B	326.6	589	668.2	1243	1811	2714	227.4	381
		C	643.4	1247	1159.4	2329	3370.4	6865	596.8	718
	4	A	49.6	71	83	119	215	338	54.8	84
		B	55.8	93	84	152	292.6	558	55.4	112
		C	67.4	133	89.4	152	344	844	69.8	124
100	3	A	450	823	1080.2	2300	3229.2	6679	381.8	810
		B	350.8	504	638.8	915	2106.8	3282	501.6	749
		C	628.4	1128	1082.4	2210	4393.4	5924	820.2	1560
	4	A	50.6	66	81.4	123	233.8	314	56.4	79
		B	241.6	969	321	1267	1591.4	6721	255	998
		C	284.2	649	369.6	796	1432.2	3369	292.2	691

**Algorithm 1** Epsilon Constraint Method

---

**Initialize:** Set  $\epsilon = 0$ ,  $f = 1$ ,  $step\_size = 1$ ,  $NS = \emptyset$ .  $NS$  is the set of nondominated vectors.  $ML = 1$ ;  
**while**  $f=1$  **do**  
     $min$  Imbalance  
    s.t.  $\sum_{i=1}^N b_i x_i \geq \epsilon$   
     $x \in X$   
     $y_m = 1 \quad m = 1, \dots, ML - 1$   
    **if** the problem is infeasible **then**  
         $f = 0$   
    **else**  
        objective value vector =  $(Imbalance^*, \sum_{i=1}^N b_i x_i^*)$ .  
         $ML = \max m : \sum_{i=1}^N b_i x_i^* > T_m$ . Solve  
         $max \sum_{i=1}^N b_i x_i$   
        s.t.  $Imbalance \leq Imbalance^*$   
         $x \in X$   
         $y_m = 1 \quad m = 1, \dots, ML - 1$   
        objective value vector =  $(Imbalance^{**}, \sum_{i=1}^N b_i x_i^{**})$   
         $NS = NS \cup (Imbalance^{**}, \sum_{i=1}^N b_i x_i^{**})$ .  
        Set  $\epsilon = \sum_{i=1}^N b_i x_i^{**} + step\_size$ .  
         $ML = \max m : \sum_{i=1}^N b_i x_i^{**} + step\_size > T_m$   
    **end if**  
**end while**

---

We use variable fixing rules depending on the information that we obtain at each iteration on the possible total benefit levels. We provide these rules for the original formulation. Using the same idea on alternative formulations using  $\delta_m$  variables instead of  $y_m$  variables is straightforward.

As another variant of the epsilon-constraint algorithm, one could fix the intervals fully at each iteration, by setting an upper bound on the total benefit defined by the next interval threshold. Then all  $\alpha_k$  variables can be fixed in the model. In

this case, the resulting solution may not be Pareto-optimal. Hence, at the end of the algorithm, a solution set guaranteed to include the Pareto set is found. This issue, however, could be easily handled by post-processing.

For the interactive algorithm we formulate both objectives as maximization type. Let the generic model be:

$$\begin{aligned} \max \quad & z_1(x) : \text{Balance}, \quad \max z_2(x) : \sum_{i=1}^N b_i x_i \\ & x \in X \\ & \text{Balance} = \text{Imb}^{\text{UB}} - \text{Imbalance} \end{aligned}$$

where  $x$  and  $X$  denote the decision variable vector and the feasible region, respectively.  $\text{Imb}^{\text{UB}}$  is an upper bound on the imbalance value, we set  $\text{Imb}^{\text{UB}} = \sum_i b_i$ .

---

**Algorithm 2** Interactive Algorithm
 

---

**Initialize:**  $T = \emptyset$ ,  $C = \emptyset$ ,  $f = 1$  and  $h = 1$ . (Arbitrarily) select weights  $\lambda_i > 0$ ,  $i = 1, 2$ , ( $\sum_{i=1}^2 \lambda_i = 1$ ). Solve the initial model.  $T$  is a set representing available preferences of DM and  $C$  is the set of nondominated points found so far except the current incumbent.

$\max \quad \lambda_1 z_1(x) + \lambda_2 z_2(x)$

$x \in X$ , where  $z_1(x)$  and  $z_2(x)$  are the balance and efficiency objectives, respectively.

**if** the problem is infeasible **then**

Stop.

**else**

$z^1 = (z_1^1, z_2^1)$  is the initial incumbent,  $z^{\text{inc}} = z^1$ .

**end if**

**while**  $f = 1$  **do**

Set  $h = h + 1$ . Select new arbitrary weights  $\lambda_i$  and solve the model with additional cone domination constraints defined over set  $T$ .

**if** the problem is infeasible **then**

Set  $f = 0$ . The incumbent point,  $z^{\text{inc}} = (z_1^{\text{inc}}, z_2^{\text{inc}})$ , is the most preferred solution of the DM.

**else**

Denote optimal solution as challenger,  $z^h = (z_1^h, z_2^h)$ .

Compare the incumbent  $z^{\text{inc}}$  and the challenger  $z^h$ .

**if**  $z^{\text{inc}} \prec z^h$  **then**

$T = T \cup (z^{\text{inc}}, z^h)$

$C = C \cup z^h$ .

**else**

$C = C \cup z^{\text{inc}}$ .

**for**  $z^k : z^k \in C$  **do**

$T = T \cup (z^h, z^k)$

**end for**

$z^{\text{inc}} = z^h$

**end if**

**end if**

**end while**

---

## References

- Armbruster B, Delage E (2015) Decision making under uncertainty when preference information is incomplete. *Manag Sci* 61(1):111–128
- Boland N, Charkhgard H, Savelsbergh M (2015) A criterion space search algorithm for biobjective integer programming: the balanced box method. *INFORMS J Comput* 27(4):735–754
- Butler M, Williams HP (2002) Fairness versus efficiency in charging for the use of common facilities. *J Oper Res Soc* 53(12):1324–1329. <https://doi.org/10.1057/palgrave.jors.2601456>
- Carazo AF (2015) Multi-criteria project portfolio selection. In: *Handbook on project management and scheduling*, Vol. 2, Springer, pp 709–728
- Chalmet L, Lemonidis L, Elzinga D (1986) An algorithm for the bi-criterion integer programming problem. *Eur J Oper Res* 25(2):292–300
- Drummond MF, Sculpher MJ, Claxton K, Stoddart GL, Torrance GW (2015) *Methods for the economic evaluation of health care programmes*. Oxford University Press, Oxford
- Eeckhoudt L, Schlesinger H (2006) Putting risk in its proper place. *Am Economic Rev* 96(1):280–289. <http://www.jstor.org/stable/30034365>
- Haimes Y, Lasdon L, Wismer D (1971) On a bicriterion formation of the problems of integrated system identification and system optimization. *IEEE Trans Syst Man Cybern* 1(3):296–297. <https://doi.org/10.1109/TSMC.1971.4308298>
- Heitmann H, Brüggemann W (2014) Preference-based assignment of university students to multiple teaching groups. *OR Spectrum* 36(3):607–629. <https://doi.org/10.1007/s00291-013-0332-9>
- Karsu Ö, Morton A (2014) Incorporating balance concerns in resource allocation decisions: a bi-criteria modelling approach. *Omega* 44:70–82
- Karsu Ö, Morton A (2015) Inequity averse optimization in operational research. *Eur J Oper Res* 245(2):343–359
- Kellerer H, Pferschy U, Pisinger D (2004) Introduction to NP-completeness of knapsack problems. In: *Knapsack problems*. Springer, Berlin, pp 483–493
- Kelly FP, Maulloo AK, Tan DKH (1998) Rate control for communication networks: shadow prices, proportional fairness and stability. *J Oper Res Soc* 49(3):237–252. <https://doi.org/10.1057/palgrave.jors.2600523>
- Korhonen P, Wallenius J, Zions S (1984) Solving the discrete multiple criteria problem using convex cones. *Manag Sci* 30(11):1336–1345
- Lokman B, Köksalan M, Korhonen PJ, Wallenius J (2016) An interactive algorithm to find the most preferred solution of multi-objective integer programs. *Ann Oper Res* 245(1–2):67–95
- Luss H (2012) *Equitable resource allocation: models, algorithms and applications*, vol 101. Wiley, New York
- Mestre AM, Oliveira MD, Barbosa-Póvoa A (2012) Organizing hospitals into networks: a hierarchical and multiservice model to define location, supply and referrals in planned hospital systems. *OR Spectr* 34(2):319–348. <https://doi.org/10.1007/s00291-011-0272-1>
- Pisinger D (2005) Where are the hard knapsack problems? *Comput Oper Res* 32(9):2271–2284
- Ralphs T, Saltzman M, Wiecek M (2006) An improved algorithm for solving biobjective integer programs. *Ann Oper Res* 147:43–70
- Smith HK, Harper PR, Potts CN (2013) Bicriteria efficiency/equity hierarchical location models for public service application. *J Oper Res Soc* 64(4):500–512. <https://doi.org/10.1057/jors.2012.68>
- Stewart TJ (2016) Multiple objective project portfolio selection based on reference points. *J Bus Econ* 86(1–2):23–33
- Williams HP (2013) *Model building in mathematical programming*. Wiley, London
- Young HP (1995) *Equity: in theory and practice*. Princeton University Press, Princeton

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.