Bi-criterion procedures to support logistics decision making: cost and uncertainty

May 5, 2015

Abstract

In practical decision making, one often is interested in solutions that balance multiple objectives. In this study we focus on generating efficient solutions for optimization problems with two objectives and a large but finite number of feasible solutions. Two classical approaches exist, being the constraint method and the weighting method, for which a specific implementation is required for this problem class. This paper elaborates specific straightforward implementations and applies them to a practical allocation problem, in which transportation cost and risk of shortage in supplied livestock quality are balanced. The variability in delivered quality is modelled using a scenario-based model that exploits historical farmer quality delivery data. The behaviour of both implementations is illustrated on this specific case, providing insight in i.) the obtained solutions, ii.) their computational efficiency. Our results indicate how efficient trade-offs in bi-criterion problems can be found in practical problems.

Keywords: Multi-objective optimization, Allocation problems, Scenario-based modelling, Constraint method, Weighting method, Stochastic programming

1 Problem description

In the optimization of business and logistic processes one is frequently faced with conflicting objectives. Specifically, in the meat processing chain, we are not only dealing with logistics costs, but also with large variety in product quality, market segments, and processing options. Differences in farmer production and breeding systems result in variation in quality features such as carcass weight, fat layer thickness, and lean meat percentage [Perez et al., 2009], whereas market segments vary with respect to preferred quality features [Grunert, 2006] (e.g. Japan prefers fat meat, Greece prefers light and lean carcasses). Rijpkema et al. [2013] describe an allocation problem where a decision maker is confronted with variability in delivered quality. For this problem, a model is presented where one of the objectives is to minimize the expected shortage with respect to a demanded product quality, whereas the other objective is the minimization of transportation cost. The structure of the model implies a bi-criterion model with a finite, but large number of decision alternatives. In so-called bi-criterion problems, the set of non-dominated solutions presents the optimal trade-offs between both objectives, and is called the Pareto set or trade-off curve. These non-dominated solution sets are typically known to be quasiconcave in their objectives, i.e. there is a diminishing marginal rate of substitution between both objectives, see

Geoffrion [1967].

There are basically three approaches to find non-dominated solutions for this problem type, being i.) the multiobjective simplex method, ii.) the constraint method, and iii.) the weighting method [Romero and Rehman, 2003]. As the applicability of the multiobjective simplex method is limited to very small-sized problems, its practical usefulness is restricted [Romero and Rehman, 2003]. We therefore do not consider this method here. In the constraint method, first introduced by Marglin [1967], non-dominated solutions are obtained by optimizing a single objective while restricting the other. In the weighting method, first introduced by Zadeh [1963], non-dominated solutions are obtained by optimizing multiple objectives simultaneously (e.g. [Lofti et al., 2011]). The weighting method will find extreme efficient points [Romero and Rehman, 2003], sometimes called supported efficient solutions Hansen [1979]. The constraint method may also find so-called interior efficient points.

The bi-criterion planning problem under consideration has a specific characteristic: the number of feasible solutions is large but finite. The corresponding research question is: how to generate sets of solutions that represent the efficient trade-off between two objectives for this type of bi-criterion problem? Recently several procedures have been described in literature to generate efficient solutions of integer programming problems for a larger number of objectives where all coefficients have integer values (or can be transformed to them), see Lokman and Köksalan [2013], Mavrotas and Florios [2013], Ozlen et al. [2014], Zhang and Reimann [2014]. In the problem we consider this is not necessarily the case. To find efficient solutions for this problem

type we present two simple and efficient procedures based on the original ideas of weigthing and the constraint method. The first is based on the constraint method and iteratively changes which objective is constrained and which objective is optimized. A similar approach for integer problems has originally been investigated in Chalmet et al. [1986]. The second procedure is based on the weighting method, using a systematic iterative procedure to vary the adopted weight. The procedure is similar to the approach presented by Cohon et al. [1979]. For a comparison of other procedures on a practical case, we refer to Tóth et al. [2006]. The effectiveness of the two procedures is showcased by the practical allocation problem and differences in computational efficiency and obtained solution sets are discussed.

This paper is organised as follows. In Section 2, a practical bi-criterion problem that has the typical characteristics we are interested in is presented. Section 3 introduces the two procedures to generate sets of efficient solutions for this problem type. They are elaborated with instances of the practical case in Section 4. Section 5 summarizes the findings.

2 A livestock allocation model

The type of problem we study in this paper has a large but finite number of feasible solutions and two conflicting objectives for a single stakeholder. The practical case we are interested in concerns an allocation problem faced by a large meat processing company. For this planning problem, Rijpkema et al. [2013] present a stochastic programming model to reduce service level violations in supplied livestock quality. In this paper we do not consider

service levels, but instead focus on minimizing the expected shortfall from demanded quality. The adopted model indices, data, and variables used in this allocation model can be found in Table 1.

A meat processing company allocates livestock batches from I farms (with $i = 1, \ldots, I$) to J slaughterhouses (with $j = 1, \ldots, J$) and tries to limit logistics costs f_1 of allocation plan X, determined by

$$f_1(X) = \sum_{i=1}^{I} \sum_{j=1}^{J} X_{ij} a_i d_{ij}, \tag{1}$$

subject to

$$\sum_{i=1}^{I} X_{ij} a_i \leq c_j \qquad j = 1, \ldots, J, \tag{2}$$

$$\sum_{j=1}^{J} X_{ij} = 1 \qquad i = 1, \ldots, I, \tag{3}$$

$$X_{ij} \in \{0, 1\} \qquad i = 1, \ldots, I, j = 1, \ldots, J. \tag{4}$$

Equation 2 sets a limit on slaughterhouse capacity, and Equation 3 and 4 ensure that each livestock batch is allocated to a single slaughterhouse.

In the practical problem, the meat processor distinguishes 12 carcass quality classes. For the illustration of the procedures, we consider only the demand for a single carcass quality class at slaughterhouse j, and we use h_j to denote the number of demanded carcasses in this quality class at slaughterhouse j. Although using more classes for the ilustration does not complicate the model, it allows us to systematically vary the demanded amount. The meat processor would like to ensure that demand for this quality class is fulfilled. However, the fraction of animals that individual farmer i delivers

in the specific quality class reveals a certain variability, which we denote by random variable ζ_i. We define $S_j(X)$ as the expected shortfall from demand h_j at slaughterhouse j given allocation plan X, with

$$S_j(X) = E_\zeta[(h_j - \sum_{i=1}^{I} X_{ij}\zeta_i a_i)^+]. \tag{5}$$

The decision maker would like to limit both logistics costs f_1 and the sum of the expected shortfall $S_j(X)$ from demand h_j at all J locations. The expected shortfall $S_j(X)$ is based on modelling possible outcomes of uncertain fraction ζ_i using historical farmer delivery data. This historical information consists of quality data available from L previous deliveries, for each of the I farmers. For the model of stochastic fraction ζ_i the support is the set of observed fractions, where each element has the same probability of $1/L$ on occurrence. Assuming independence between the farmers defines an outcome space with $N = L^I$ possible outcomes ξ_{in}, where ξ_{in} denotes the fraction of animals farmer i delivers in the quality class under scenario n. Each of these possible outcomes has a probability $1/N = (1/L)^I$. Using this stochastic model, the expected shortfall $S_j(X)$ is defined by

$$S_j(X) = \frac{\sum_{n=1}^{N}(h_j - \sum_{i=1}^{I} X_{ij}\xi_{in}a_i)^+}{N}. \tag{6}$$

The decision maker would like to find an allocation plan X with a limited total expected shortage f_2 at all J slaughterhouses, defined by

$$f_2(X) = \sum_{j=1}^{J} S_j(X). \tag{7}$$

As the number of possible outcomes N is, even for relatively small number of farmers I and historical observations L, very large, this MILP model only has a theoretical meaning. In scenario-based modeling (see e.g. Birge and Louveaux [1997]) it is usual to consider a finite subset with a limited number (e.g. $N = 5000$) of samples $\Xi_n = (\xi_{1n}, \xi_{2n}, \ldots, \xi_{In})$ of the possible outcomes of the random variable. This allows us to approximate the performance of solution X in objective f_2.

We now have a scenario-based allocation model that has two objectives ($f_1(X)$ and $f_2(X)$) and a large but finite number of feasible solutions X in the allocation problem defined by Equations 2, 3, and 4. As the reader may imagine, there is a trade-off between the logistics costs f_1 and expected shortfall of demand f_2. It is therefore important to find solutions that balance levels of both objective f_1 and f_2, for which two procedures are elaborated in Section 3.

Table 1: Model indices, data and variables of livestock allocation problem

Indices	
i	farm index, $i = 1, \ldots, I$
j	slaughterhouse index, $j = 1, \ldots, J$
n	scenario-number, $n = 1, \ldots, N$
Data	
a_i	animals delivered by farmer i in number of animals.
d_{ij}	transportation costs from farm i to slaughterhouse j in € per animal
c_j	processing capacity of slaughterhouse j in number of animals
h_j	demand for carcass quality class at slaughterhouse j in number of animals
ζ_i	fraction of animals from farmer i in the quality class
ξ_{in}	fraction of animals from farmer i in the quality class under scenario n
Ξ_n	realisation of farmer fraction deliveries under scenario n
Variables	
X_{ij}	allocation of animals from farm i to slaughterhouse j

3 Bi-criterion algorithms

In bi-criterion problems, a decision maker would like to determine the trade-off between two objectives by obtaining a complete overview of solutions that are non-dominated with respect to objectives f_1 and f_2. This can be done by generating a subset $\mathbb{X}$ of non-dominated solutions of the finite set of feasible plans. The corresponding objective values $f_1(X), f_2(X)$ for $X \in \mathbb{X}$ form a Pareto front. We present two procedures to efficiently generate a Pareto front for this problem type, i.e. a problem with two objectives and a large but finite number of feasible solutions. Section 3.1 presents a strategy based on the constraint method, whereas the procedure presented in Section 3.2 is based on the weighting method. Section 3.3 discusses the difference of these straightforward implementations with recent studies on specific multi-objective integer programming from literature literature.

3.1 A constraint method procedure

Algorithm 1 (in: problem data, K. out: Set $\mathbb{X}$ of efficient solutions in $f_1 f_2$.

$\varphi_2 = \min\{f_2(X)\}$	*lowest value* f_2
$\mathbb{X} = \operatorname{argmin}\{f_1(X), f_2(X) \leq \varphi_2\}$	*store first solution*
$F_1 = \min\{f_1(X)\}$	*lowest value* f_1
$\mathbb{X} = \mathbb{X} \cup \operatorname{argmin}\{f_2(X), f_1(X) \leq F_1\}$	*add second solution*
$F_2 = f_2(\mathbb{X})$	*value objective* f_2
$\epsilon = \frac{F_2 - \varphi_2}{K}$	*min gap objective* f_2
while $F_2 > \varphi_2 + \epsilon$	
$F_1 = \min\{f_1(X), f_2(X) \leq F_2 - \epsilon\}$	*Pareto objective value* f_1
$Y = \operatorname{argmin}\{f_2(X), f_1(X) \leq F_1\}, \mathbb{X} = \mathbb{X} \cup Y$	*store new solution*
$F_2 = f_2(Y)$	*Pareto value objective* f_2

In this section, a procedure based on the constraint method is presented to find non-dominated solutions X in $f_1 f_2$ for problems with a large but finite number of feasible solutions. In the constraint method, one objective is minimized (e.g. $\min\{f_1(X)\}$) whereas the other objective is restricted (e.g. $f_2(X) \leq F_2 - \epsilon$). By iteratively changing the objective that is constrained and the objective that is minimized, a set of non-dominated solutions is generated. The obtained solutions may include both extreme efficient points and interior efficient points. Algorithm 1 describes the procedure to obtain a set $\mathbb{X}$ of solutions X that are non-dominated in $f_1 f_2$.

The symbols used in Algorithm 1 can be found in Table 2. The decision maker needs to set parameter K, which denotes an upper bound to the number of obtained solutions. If a high value for K is used, a large number of non-dominated solutions might be found, which will require a large number of iterations and therefore much computational effort. If, however, a low value for parameter K is used interesting solutions might be missed. As there is a finite number of feasible solutions, there is a certain value of K for which all non-dominated solutions will be found.

Table 2: Symbols used in Algorithm 1

K	upper bound to the number of obtained solutions
φ_2	lower bound objective value f_2
F_1	pareto objective value f_1
F_2	pareto objective value f_2
ϵ	minimum gap in objective f_2 between two consecutive solutions
Y	solution that is temporarily stored

3.2 A weighting method procedure

This section presents a procedure based on the weighting method to find non-dominated solutions X in $f_1 f_2$ for problems with a large but finite number of feasible solutions. The objective of the weighting method is to simultaneously minimize both f_1 and f_2, that is

$$\min\{f_1(X) + w f_2(X)\}, \tag{8}$$

where weight w denotes the number of units of f_1 a decision maker is willing to trade in to reduce the level of f_2 with one unit. Algorithm 2 describes a procedure to gather a set $\mathbb{X}$ of solutions X that is non-dominated in $f_1 f_2$. In Algorithm 2 we initially minimize objective $f_1(X)$ and $f_2(X)$ separately, and store the obtained solutions as X_1 and X_2. The following step is to determine the value of trade-off w for which X_1 and X_2 have the same objective function value according to

$$w = \frac{f_1(X_2) - f_1(X_1)}{f_2(X_1) - f_2(X_2)}. \tag{9}$$

The solution we obtain by using weight w in Equation 8 will be either a new one, or it will be a solution we have found before. If we find either X_1 or X_2 we can conclude that solutions X_1 and X_2 are optimal with respect to weight w, and we stop the search for alternative solutions. If a new solution is found, we can conclude that solutions X_1 and X_2 are not optimal for weight w. In that case we store the new solution as X_3, and restart the procedure to assess whether there are values of w corresponding to other extreme efficient points of the Pareto front other than X_1, X_2, and X_3. The procedure in

Algorithm 2 (in: problem data. out: Set $\mathbb{X}$ of extreme efficient solutions

Determine $X_1 = \text{argmin}\{f_1(X)\}$ and $X_2 = \text{argmin}\{f_2(X)\}$
$\mathbb{X} = \{X_1, X_2\}$ *store resulting solutions*
Set $k = 2$, $t = 3$, $w_t = \frac{f_1(X_2) - f_1(X_1)}{f_2(X_1) - f_2(X_2)}$
while $k < t$ *iteration < nr scheduled iterations*
    $k = k + 1$ *increment current iteration counter*
    $X_k = \text{argmin}\{f_1(X) + w_k f_2(X)\}$
    if $X_k \notin \mathbb{X}$ *if new solution is found*
        $X_{LB} = \underset{X \in \mathbb{X}}{\text{argmin}}(f_1(X_k) - f_1(X))$
        $w_{t+1} = \frac{f_1(X_k) - f_1(X_{LB})}{f_2(X_{LB}) - f_2(X_k)}$ *new weight w below w_k*
        $X_{UB} = \underset{X \in \mathbb{X}}{\text{argmin}}(f_1(X) - f_1(X_k))$
        $w_{t+2} = \frac{f_1(X_{UB}) - f_1(X_k)}{f_2(X_k) - f_2(X_{UB})}$ *new weight w above w_k*
        $t = t + 2$ *increment scheduled iteration counter*
        $\mathbb{X} = \mathbb{X} \cup X_k$ *store new solution*

Algorithm 2 will find all extreme efficient solutions for positive values of w (i.e. $0 \geq w \geq \infty$). The symbols adopted in Algorithm 2 can be found in Table 3.

If a decision maker is able to indicate a weight range where an optimal trade-off between objective f_1 and f_2 is expected she may determine X_1 and X_2 using Equation 8 with a lower and an upper bound for w instead. This may reduce the required computational effort.

Table 3: Symbols used in Algorithm 2

Indices	
k	current iteration counter
t	total iteration counter
Data	
w	maximum allowed increase in objective f_1 to reduce objective f_2 with one unit

3.3 Comparison with other algorithms

Recently many specific algorithms have been developed and investigated for multi-objective integer programming. In general, the newly proposed schemes are more and more sophisticated elaborations of the classical approaches in an attempt to capture all non-dominated solutions also for the case of more than two objectives. Recently Lokman and Köksalan [2013] extended the earlier work of Sylva and Crema [2004], where every found efficient point generates more constraints and binary variables extending the IP problem to be solved at each iteration considerably. Lokman and Köksalan [2013] propose a shortcut for that guaranteeing to reach all non-dominated points, whereas the presented Algorithm 1 simply generates a subset in the classical ϵ-constraint sense where solutions differ at least the preset accuracy ϵ in one of the objectives. Needless to say that this target requires far less steps from an optimization problem that does not increase in size.

Mavrotas and Florios [2013] and [Zhang and Reimann, 2014] elaborate further on the augmented ϵ-contraint method that is designed to deal with many objectives in an integer programming environment. Essential is the idea that the problem is converted in a problem with integer coefficients, such that the step sizes can be taken as one. It uses the characteristic that in fact the number of values for each objective is finite as long as the integer programming problem has a finite number of solutions. It requires weighting slacks and determination of step sizes for each objective. The approach is far more elaborated than a straightforward constraint method for two objectives.

Ozlen et al. [2014] present a new version of an earlier algorithm that is based on the constraint method combined with lexicographical optimization

of many objectives. The method aims at generating Pareto points for a problem where all objecive vector coefficients have integer value. This means that the step size ϵ is implicitly taken as one. The problem we would like to solve does not fall necessarily in this class of problems. Another difference with their algorithm is that Algorithm 1 never has to solve an infeasible subproblem.

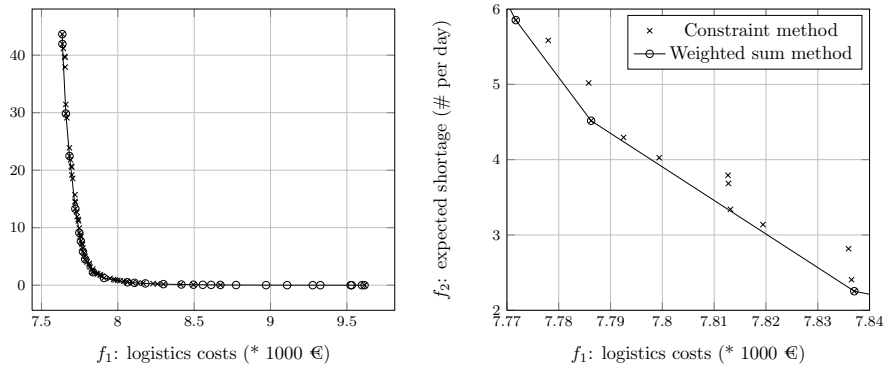
None of the mentioned modern algorithms aim at generating all extreme non-dominated points as Algorithm 2 does. The problem we aim to generate the efficient plans for does not necessarily contain integer valued coefficients. For the described logistic problem this would be a hard assumption given that thousands of quality data are used. Algorithm 1 does not take implicit step sizes of 1 and does not necessarily aim at generating all efficient solutions. It aims at supporting decisions by providing a trade-off curve of the two objectives.

4 Computational illustration

In this section, we illustrate the behaviour of both procedures presented in Section 3 using the livestock allocation problem presented in Section 2. The parameter settings used in this allocation problem can be found in Table 4. The solutions generated by both the constraint method procedure (Algorithm 1) and the weighting method procedure (Algorithm 2) in objective f_1 and f_2 are depicted in Figure 1. The complete solution sets are presented in Subfigure 1(a), whereas Subfigure 1(b) presents a small subset of obtained solutions. The data presented in Figure 1 are obtained for demand $h_1 = 250$,

Table 4: Parameter settings used in computational illustration

Parameter	Value	Definition
I	72	number of farmers supplying livestock
J	5	number of slaughterhouses receiving livestock
L	45	number of observations for each farmer
N	5000	number of samples in optimization
d_{ij}	0.133 to 4.33	livestock transportation costs in € per animal



(a) Complete set of solutions (b) Subset of solutions

Figure 1: Sets of non-dominated solutions obtained using both procedures

whereas a value of $K = 1000$ is used in the constraint method.

We now vary the demand parameter h_j in order to observe the effect on the set of non-dominated solutions. For clarity of presentation, we only present the solution set obtained using the weighting method procedure in Figure 2. A similar set of solutions can be obtained using the contraint method procedure as well. To provide insight in the performance and computational efficiency of both procedures, performance data for both procedures is provided in Table 5. In this table we present performance data of solutions obtained using a.) the constraint method procedure with $K = 10$, b.) the constraint method procedure with $K = 1000$, and c.) the weighting method

procedure. The results were obtained using an Intel i7 860 CPU with 2.80 GHz and 4 GB of RAM.

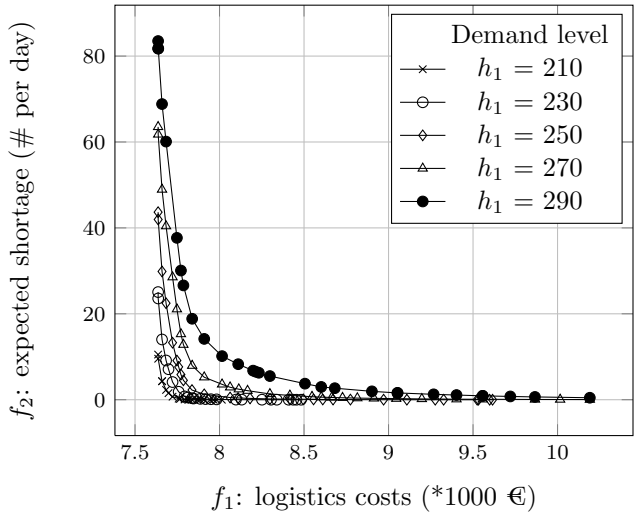


Figure 2: Pareto fronts obtained using the weighting method procedure for variations of h_1

Table 5: Optimization details for both procedures

	Constraint method procedure				**Weighting method procedure**	
	$K = 10$		$K = 1000$			
h_1	# of points	Total time [seconds]	# of points	Total time [seconds]	# of points	Total time [seconds]
210	7	5.21	41	32.07	20	14.58
230	7	6.52	57	62.25	24	30.23
250	9	10.71	69	113.19	29	126.50
270	10	11.96	87	192.26	29	67.08
290	9	14.86	112	348.57	29	67.66

In Figure 1 we observe that, following Romero and Rehman [2003], the weight method procedure finds all extreme efficient solutions of the Pareto front, whereas the constraint method procedure may find both extreme efficient and interior efficient solutions. Figure 2 can be used by decision makers that want to assess i.) what impact variations in demand level h_j have on both objectives, ii.) the trade-off between logistics costs and expected shortage for each fixed demand level. From Figure 1 and Table 5 we learn that

i.) selecting a low K will improve computational efficiency, but may lead to missing interesting points, particularly in the lower f_2 region, and ii.) selecting a high K will reduce the likelihood of missing non-dominated solutions (ultimately to the point where all non-dominated solutions will be found), but will increase required computational effort. Furthermore we observe from data in Table 5 that finding solutions becomes increasingly difficult for higher levels of demand h_j.

5 Discussion and conclusion

This paper develops two practical procedures derived from concepts of multi-criteria decision making to obtain sets of non-dominated solutions problems that have i.) two conflicting objective functions, and ii.) a large but finite number of feasible solutions. The performance and efficiency of both procedures is elaborated for a practical decision making problem where livestock batches need to be allocated to slaughterhouses. In this allocation problem both logistics costs and deviations from demanded livestock quality features are to be minimized. The livestock quality that will be supplied is modelled using a scenario-based approach using historical data on livestock quality delivered by individual farmers. The modelling results indicate that the presented procedures effectively i.) find sets of non-dominated solutions in both objective f_1 and f_2, ii.) provides insight in the trade-off between objective f_1 and f_2.

Acknowledgement

This paper has been supported by the Spanish state (project TIN2012-37483-C03-01) and Junta de Andalucía (P11-TIC-7176), in part financed by the European Regional Development Fund (ERDF).

References

J. Birge and F. Louveaux. *Introduction to Stochastic Programming*. Springer Series in Operations Research. Springer-Verlag, New York, 1997.

L. Chalmet, L. Lemonidis, and D. Elzinga. An algorithm for the bi-criterion integer programming problem. *European Journal of Operational Research*, 25(2):292 – 300, 1986.

J. Cohon, R. Church, and D. Sheer. Generating multiobjective trade-offs: An algorithm for bicriterion problems. *Water Resources Research*, 15(5): 1001–1010, 1979.

A. Geoffrion. Solving bicriterion mathematical programs. *Operations Research*, 15(1):39–54, 1967.

K. Grunert. How changes in consumer behaviour and retailing affect competence requirements for food producers and processors. *Economia Agraria y Recursos Naturales*, 6(11):3–22, 2006.

P. Hansen. *Multiple Criteria Decision Making Theory and Application*, chapter Bicriterion path problems. Springer Verlag, Berlin, 1979.

M. Lofti, M. Rabbani, and S. Ghaderi. A weighted goal programming approach for replenishment planning and space allocation in a supermarket. *Journal of the Operational Research Society*, 62(6):1128–1137, 2011.

B. Lokman and M. Köksalan. Finding all nondominated points of multi-objective integer programs. *Journal of Global Optimization*, 57(2):347–365, 2013. doi: 10.1007/s10898-012-9955-7. URL `http://dx.doi.org/10.1007/s10898-012-9955-7`.

S. Marglin. *Public Investment Criteria*. MIT Press, 1967.

G. Mavrotas and K. Florios. An improved version of the augmented ϵ-constraint method (augmecon2) for finding the exact pareto set in multi-objective integer programming problems. *Applied Mathematics and Computation*, 219(18):9652 – 9669, 2013.

M. Ozlen, B. Burton, and C. MacRae. Multi-objective integer programming: An improved recursive algorithm. *Journal of Optimization Theory and Applications*, 160(2):470–482, 2014. doi: 10.1007/s10957-013-0364-y.

C. Perez, R. de Castro, and M. Font i Furnols. The pork industry: a supply chain perspective. *British Food Journal*, 111(3):257–274, 2009.

W. Rijpkema, E. Hendrix, R. Rossi, and J. van der Vorst. Application of stochastic programming to reduce uncertainty in quality-based supply planning of slaughterhouses. *Annals of Operations Research*, to appear, 2013. doi: 10.1007/s10479-013-1460-y.

C. Romero and T. Rehman. *Multiple Criteria Analysis for Agricultural Decisions*. Developments in Agricultural Economics. Elsevier Science, 2003.

J. Sylva and A. Crema. A method for finding the set of non-dominated vectors for multiple objective integer linear programs. *European Journal of Operational Research*, 158(1):46 – 55, 2004.

S. F. Tóth, M. E. McDill, and S. Rebain. Finding the efficient frontier of a bi-criteria, spatially explicit, harvest scheduling problem. *Forest Science*, 52(1):93–107, 2006.

L. Zadeh. Optimality and non-scalar-valued performance criteria. *IEEE Transactions on Automatic Control*, 8(1):59–60, 1963.

W. Zhang and M. Reimann. A simple augmented ϵ-constraint method for multi-objective mathematical integer programming problems. *European Journal of Operational Research*, 234(1):15 – 24, 2014.