Mat-2.108 Independent Research Project in Applied Mathematics

Potentially Optimal Portfolios in Capital Budgeting

January 30, 2003

Helsinki University of Technology
Department of Engineering Physics and Mathematics
Systems Analysis Laboratory
Pekka Mild
50124C
Contents

1 Introduction 1

2 Multi-Attribute Value 3
   2.1 Additive Value Representation ............................. 3
   2.2 Incomplete Weight Information ............................. 4
   2.3 Dominance and Potential Optimality ........................ 6

3 Multi-Attribute Capital Budgeting 8
   3.1 Portfolio of Multi-criteria Alternatives ..................... 9
   3.2 Optimal Portfolio Construction .............................. 11

4 Proposed Framework 12
   4.1 Geometric Grounds ........................................... 13
   4.2 The Algorithm ................................................. 14
       4.2.1 Notational Convention ................................ 15
       4.2.2 Progress of the Algorithm .............................. 16

5 Illustrative Numerical Example 20

6 Discussion 29
   6.1 Properties of the Algorithm ................................. 29
   6.2 Implications of the Assumptions ............................ 30
   6.3 Extensions and Further Research ............................ 32

7 References 34
1 Introduction

Capital budgeting (see, e.g., Luenberger, 1998) and multiple criteria decision making, MCDM (e.g., early classic of Keeney and Raiffa, 1976) are basic problems of operations research. Combining the two, i.e. selecting combinations of multiple attribute alternatives, has also gained attention in the literature. For example, Spronk and Hallerbach (1997) motivate a multi-attribute representation of financial securities, and Martikainen (2002) introduces a real-life case of constructing a multi-attribute project portfolio in a large Finnish corporation. With large and complex selection problems and possibly stochastic (fuzzy) parameters, finding a uniquely optimal solution can turn out impossible. Then, just approximate and directive (mathematical) elimination methods can offer substantial aid to decision making.

Consider a framework, in which basic alternatives’ (projects) performances are first assessed individually in multiple attributes, and the decision maker is then interested in the best feasible combination (portfolio) of the basic alternatives. The overall value is represented as a weighted sum of the attribute-specific performances. If the weight parameters are defined as intervals, different realizations of the weights indicate different optimal combinations. We are now interested in finding these potentially optimal portfolios.

In this paper, we introduce a search algorithm that efficiently detects potentially optimal portfolios in a multi-attribute capital budgeting problem with incompletely defined attribute weights. The incomplete weight information and its implications in single alternative selection problems are widely studied in the MCDM literature (see, e.g., Hazen, 1986; Weber, 1987; Salo and Hämäläinen, 1992). Our paper, however, combines the established concepts of capital budgeting and MCDM incomplete information. Since the problem is in fact a combinatorial problem, the number of actual decision alternatives, i.e. combinations, easily grows high. Thus, to first generate all possible combinations to create a set of fixed alternatives and then to rely on the established methods developed for fixed alternatives seems uninviting – there is room for more efficient methods that are able to distinguish the potential combinations straight from the set of basic alternatives.

In introducing the algorithm, the multi-attribute value construction is rather restrictive concerning the nature of the attributes. Thus, the method cannot be presented as a completely general MCDM tool. The issues of aggregating incommensurate attributes on portfolio level in the light of MCDM theory are briefly discussed in the end of the
paper (e.g. Martikainen, 2002, dismisses this consequential issue). However, the current assumptions of the algorithm are well met be a general Gray LP problem with a Gray price vector (see, e.g., Lin and Liu, 1999). Also, scenario analysis with loosely defined state occurrence probabilities suits well into the framework.

The paper is structured as follows. Section 2 considers additive multi-attribute value representation and introduces the concepts of dominance and potential optimality. Section 3 introduces the capital budgeting problem under known scores and weights. In Section 4 we extend the budgeting problem with incomplete weights and describe our algorithm. Section 5 provides a numerical example with some general notes on the characteristics of the framework. Section 6 discusses the properties of the algorithm and implications of our assumptions. Various branches of further research and methodological extensions are also proposed.
2 Multi-Attribute Value

2.1 Additive Value Representation

The term value refers to the strength of preference of an object. If a decision maker is facing a situation of choosing one of several alternatives, he/she chooses the one whose value is the highest. However, the alternatives often perform in various attributes. The attribute specific performances may contradict such that by solely focusing on a certain attribute, the value of a certain alternative would be the highest, but focusing on another attribute, another alternative would yield the highest value. In such cases, we need a framework for constructing an aggregate value, i.e. representation to describe the overall "goodness" of an alternative over all attributes. This aggregate value can then be used as the measure determining which alternative to choose. A very common aggregate value representation is additive value function, which is widely used in the theory of multiple criteria decision making (MCDM), for example (see, e.g., Keeney and Raiffa, 1976; Saaty, 1977; French 1988). Furthermore, objective functions of all linear programming problems are additive by definition.

Let an alternative be described by a vector $\mathbf{x}^j$, $j = 1, \ldots, m$. An alternative can be a multi-attribute project or a fixed basket of multiple attributes, for example. Let there be $n$ different attributes. All $m$ alternatives perform with regard to each $n$ attributes, and thus the alternative vector is $\mathbf{x}^j = (x_1^j, x_2^j, \ldots, x_n^j)$. In multi-attribute value theory, the attribute specific performances are typically converted into scores with a function $v_i(x_i^j)$, s.t. the notation $v_i(x_i^j)$ reads "the score of the j:th alternative on the i:th attribute". Generally, the mapping $v_i$ is optional, i.e. one can use identity mapping $v_i(x_i^j) = x_i^j$.

The additive value function is basically a weighted sum of the scores. The weights are scaling factors that can be used to adjust the construction of the aggregate value. Let $w_i$ denote the weight factor attached to the $i$:th attribute. Now, $w = (w_1, w_2, \ldots, w_n)$ denoting a weight vector, and $v(\mathbf{x}) = (v_1(x_1) v_2(x_2) \ldots v_n(x_n))$ collecting the attribute specific scores of an alternative. The overall additive value is

$$ V(\mathbf{x}, w) = \sum_{i=1}^{n} w_i v_i(x_i) = v(\mathbf{x}) w^T. \quad (1) $$
Without loss of generality, we may assume (or transform) that all the weights and scores are non-negative, i.e. $w_i \geq 0$, $v_i(x^j_i) \geq 0$, $\forall i, j$. In the basic context of MCDM, the weights reflect the subjective relative importance of the attributes, with respect to the available ranges of the attribute performances. In addition, the weights can be seen as more objective factors, e.g., unit prices (in general LP), discount factors or state occurrence probabilities (see, e.g., Weber, 1987; French, 1988. See also a real-life case of high-tech company valuation; Gustafsson et. al., 2001).

Simple form of the additive value function makes it flexible, and it is an inviting approach to many problems. An obvious restriction of direct additivity is that the attribute specific values, $w_i v$, need to be measured in commensurate units (one cannot sum up "apples and oranges", so to say). MCDM theory (see, e.g., Keeney and Raiffa, 1976; Saaty, 1977; French 1988; Clemen 1996) concentrates on value tradeoffs to elicit the preference weights and on construction of relative score functions to translate originally incommensurate attribute performances into additive units. Furthermore, certain independency assumptions are considered. Although the roots of our framework lie in MCDM, many of the very basic concepts of relative preference do not apply to the development of multi-attribute port-folios in a straightforward manner. However, throughout the paper the aggregate value of a multi-attribute alternative is represented with an additive value function of the form (1).

2.2 Incomplete Weight Information

In calculating the additive value function, it is assumed that the weights $w$ are known. To broaden the practical applicability of the multi-attribute value theory, the field of MCDM has concentrated on methods allowing more loosely defined parameters. The term "incomplete information" refers to parameters whose values are not known exactly but are defined as intervals, rank orders or bounded distributions. In MCDM, it is most often the preference information that is incomplete, and the weights are defined as uniform intervals. In linear programming, the same context is called Gray LP (see, e.g., Lin and Liu, 1999) with a gray price vector. Furthermore, Lin and Liu (1999), for example, associate of the context with fuzzy concepts and stochastic concepts.

With incomplete information the selection problem becomes somewhat different – instead of searching the absolutely best alternative by maximizing (1), we have to focus on estab-
lishing dominance relations, searching potentially optimal solutions or possibly deal with expected values (see, e.g., Hazen, 1986; Weber, 1987). Since part of the input information is defined only approximately, different realizations of the respective parameters yield different solutions. The basic approach is to deal with the incomplete information, instead of "completing" it (compare to whitenization in Gray LP or the use of expected values in many stochastic programming techniques). A drawback of the approach is, however, that the results may be inconclusive.

Methods for dealing with incomplete information have attained much attention in the MCDM literature (e.g., Hazen, 1986; Weber, 1987; Arbel, 1989; Salo and Hämäläinen, 1992; 1995). More recent papers of, e.g., Kim and Han (2000), Salo and Hämäläinen (2001) as well as Salo and Punkka (2002) provide a fresh overview. The methods do differ in their details, mostly in the elicitation of incomplete statements and in the solution techniques, but they all incorporate incomplete input information and produce dominance results between the alternatives by solving a series of (mostly linear) optimization problems. The papers discuss rationales for incorporating incomplete information as well as introduce the basic concepts of bounded feasible weight region and dominance structures. In these settings, the feasible weight region is defined through linear inequalities, and the value representation is linear as in (1).

Weber (1987) recognizes possible sources of incomplete input information. Such can be, for example, urgency of the decision situation, interpretation of intangible objectives or truly uncertain realizations of stochastic variables (e.g., unit prices). Furthermore, possibly divergent group statements can all be incorporated in an incomplete information framework. Weber (1987) also discusses issues concerning incomplete information on probability distributions and general utility functions from both descriptive and prescriptive points of view.

A general approach of handling incomplete information would be fuzzy set theory (see, e.g., financial applications of Ramaswamy, 1998; Mohamed and McCowan, 2001), and the inherent fuzziness could be incorporated in other parameters than the weights as well. Our model, however, approaches the problem of treating incomplete information from viewpoint of Hazen, Weber, Salo and Hämäläinen etc. Thus, the incomplete information is incorporated in the weights of an additive value function, and the aim is to produce possibly inconclusive dominance and potential optimality results without trying to extract the incompleteness from the model.
2.3 Dominance and Potential Optimality

Following Salo and Hämäläinen (1992), dominance structures for the alternatives can be established on the basis (1) the value intervals that the alternatives can assume, subject to the requirement that the weights belong to the feasible region, and (2) the minimization of value differences between pairs of alternatives, as computed from the pairwise bounds.

\[
\min_{w \in S} [V(x^k, w) - V(x^l, w)] = \min_{w \in S} \sum_{i=1}^{n} w_i [v_i(x^k_i) - v_i(x^l_i)]
\]  

(2)

Specifically, if the minimum in (2) is non-negative, with \( V(x^k, w) \) larger than \( V(x^l, w) \) for all \( w \in S \), the value of alternative \( x^k \) is greater than or equal to that of alternative \( x^l \), no matter how the feasible weights are chosen. In this case, alternative \( x^k \) dominates \( x^l \) in the sense of (pairwise) dominance criterion. Thus, a complete pairwise dominance structure of a set of \( m \) alternatives can be established by running total of \( \frac{m(m-1)}{2} \) small linear programs of form (2).

If an alternative is dominated, it cannot be optimal anywhere within the feasible region. A rational decision maker rules out dominated alternatives, since at least the respective pairwise dominant offers higher value. On the other hand, neither non-dominated alternatives are necessarily optimal with any realization of the weights. Potentially optimal alternatives are such that they are by definition optimal within a particular subset of the feasible weight region. Thus, potentially optimal alternatives are certainly non-dominated and, in addition, hold the possibility of offering the best performance. See Weber (1987), for example, for detailed introduction of these different dominance definitions and their relations. For potential optimality, a definition that is notionally consistent with our presentation can be found in, for example, the recent paper of Eum et.al. (2001). An alternative \( x^k \) is potentially optimal if and only if there exists a feasible weight vector \( w \in S \) such that

\[
\sum_{i=1}^{n} w_i v_i(x^k_i) = \max_{v_j} \left\{ \sum_{i=1}^{n} w_i v_i(x^l_i) \right\}
\]

(3)

Thus, checking whether an alternative is potentially optimal by the above definition, the decision maker has to determine the optimal alternative with all feasible realizations of the weight vectors.
The definitions (2) and (3) help to find the sets of non-dominated and potentially optimal alternatives – they only narrow the initial set of alternatives to provide potential candidates of choice for a rational decision maker. Further recommendations are typically drawn based on certain established decision rules (see, e.g., Salo and Punkka, 2002). Many decision rules are such they recommend robust solutions, i.e. solutions whose performance is satisfactory under all possible realizations of the incomplete information. The less there are alternatives to compare, the more carefully the decision maker can focus on them. Thus, the sole narrowing the set alternatives to the potential candidates is advantageous especially in large problems.
3 Multi-Attribute Capital Budgeting

Capital budgeting (see, e.g., Luenberger, 1998) is an allocation decision of allocating a fixed budget among a number of competing investment alternatives. Each alternative takes a certain cost and offers certain benefits. Both costs and benefits may be multi-dimensional. For the allocation problem to be relevant, the budgetary limit (resource constraints in general) should rule out the possibility of selecting all alternatives. Furthermore, we make an important assumption that each alternative has first gone through an individual appraisal that proves it desirable, i.e. each alternative offers positive net value. Thus, the decision maker would choose all alternatives if he/she had an unlimited budget.

Luenberger, 1998, draws an applicable distinction between capital budgeting and general portfolio problems, s.t. capital budgeting refers to allocating among fixed and discrete alternatives instead of choosing appropriate amounts of different continuous investment objects. In capital budgeting the instrumental variables are indicators of whether or not an alternative is chosen, and because of fixed costs of the alternatives, the budget may not be used up to its limit in the optimal solution. In general portfolio problems the budget to be invested is fixed and the instrumental variables are fractions of the budget invested in different alternatives. Considerations of attribute score aggregation are slightly different in the two types of problems. In its basic form, our method treats a capital budgeting problem.

There are several established techniques to solve the capital budgeting problem (see, e.g., Dye and Pennypacker, 1999 or Martino, 1995 for general RD project portfolio selection). The techniques include ranking and use of benefit-cost approximations or economic methods, for example. In this paper, we concentrate on a mathematical programming model. Mathematical programming models are advantageous in large problems with complex feasibility structures, such as multiple resources, strategic balance requirements or minimum performance levels (see, e.g., a comparative review by Archer and Ghasemzadeh, 1999; or a case example by Martikainen, 2002). If the alternatives are multi-attribute in benefits and/or costs, the complexity of the problem is likely to further increase.
3.1 Portfolio of Multi-Attribute Alternatives

Now, let us consider the capital budgeting problem with multi-attribute alternatives. Single alternative representation is similar to Section 2, but instead of choosing the one best alternative, the decision maker is searching for the best feasible combination of the alternatives. The objective is to maximize the overall portfolio value. Furthermore, portfolio approach can include other goals, such as trying to attain balanced performance by combining conflicting alternatives (see, e.g., Spronk and Hallerbach, 1997 with multi-attribute financial securities and multiple goal programming).

We will apply additive value function to represent the overall portfolio value. In our framework, the alternative appraisals are conducted independently, i.e. the scores and possible interactions are assessed alternative by alternative, s.t. an overall multi-attribute value (1) could first be calculated for each alternative individually. Then, combinations are constructed by studying aggregations of the alternative scores – the value of a portfolio is a direct function of the scores of the alternatives included. To introduce our algorithm, whose novelty is in treating incomplete weight information, we make fairly restrictive assumptions concerning value aggregation of multi-attribute alternatives. The properties and implications of the assumptions are discussed in Section 6.2.

We assume that the attribute specific scores are additive over the alternatives. For now, we consider alternatives whose benefits are independent, i.e. there are no synergism or cannibalization effects (see, e.g., Martino, 1995). Then, the attribute specific score of portfolio $\mathcal{P}$ is

$$v_i(\mathcal{P}) = \sum_{x^j \in \mathcal{P}} v_i(x^j_i).$$  \hspace{1cm} (4)

Considering a capital budgeting problem with positive net value alternatives, the assumption (4) implies that the attribute specific scores are expressed in absolute - instead of relative - terms, i.e. the score $v_i(x^j_i)$ of alternative $x^j$ is not affected by the set of alternatives available, and that the score functions are not bounded by an upper limit.
Let matrix $Q$ collect the scores of all the alternatives. Each row represents an alternative and each column represents an attribute.

$$Q = \begin{pmatrix}
    v_1(x_1) & v_2(x_1) & \cdots & v_n(x_1) \\
    v_1(x_2) & v_2(x_2) & \cdots & v_n(x_2) \\
    \vdots & \vdots & \ddots & \vdots \\
    v_1(x_m) & v_2(x_m) & \cdots & v_n(x_m)
\end{pmatrix}.$$ 

Let vector $p^k = (z_1^k, z_2^k, \ldots, z_m^k)$, $z_j^k \in \{0, 1\}$, $\forall j = 1, \ldots, m$ denote a portfolio, i.e. a combination of alternatives. In a capital budgeting problem, we can take the instrumental variables as binary indicators, i.e. $z_j^k = 1 \Rightarrow$ alternative $x_j$ is included in portfolio $p^k$, and $z_j^k = 0 \Rightarrow x_j$ excluded.

In our selection framework, portfolios $p^k$ are the effective decision alternatives, i.e. the selection is made between them. Altogether $m$ individual alternatives enable total of $2^m$ possible combinations $p^k$ — the feasibility conditions rule out part of them, but still the number of selection alternatives is easily very high. Now, $p^k$ is a binary vector and the matrix $Q$ contains the attribute specific scores of the basic alternatives. Then, let $P^k$ denote the (constant) score vector of a fixed portfolio $p^k$, i.e.

$$P^k = p^k Q = (\sum_{j \in p^k} v_1(x_1^j), \sum_{j \in p^k} v_2(x_2^j), \ldots, \sum_{j \in p^k} v_n(x_n^j)),$$ \hspace{1cm} (5)

where the notation $j \in p^k$ reads $z_j^k = 1$.

Once the portfolio score vector (5) is constructed, it can be seen as single multi-attribute alternative, and the scores are weighted and added up to obtain the overall portfolio value. Then, the overall value of portfolio $p^k$, with fixed weights $w$ is

$$V(p^k, w) = p^k Q w^T = P^k w^T.$$ \hspace{1cm} (6)

Thus, the portfolio is value a weighted linear combination of the portfolio scores, and is therefore linear with respect to the weights.
3.2 Optimal Portfolio Construction

Fixed portfolios act as individual alternatives. Then, an optimal portfolio could be found by constructing all feasible portfolios and comparing their values to each other. However, we are interested in finding the optimal portfolio without first constructing all portfolios as individual alternatives. If the weights $w$ are fixed and we can model all restrictions as linear constraints, we can construct a unique optimal portfolio as a solution to an LP problem of maximizing (6). Let us denote the optimal portfolio with particular weights $w$ by $p^*(w) = (z_1^* z_2^* \ldots z_m^*)$, i.e. $p^*(w)$ is a solution to the problem

$$
\max \quad V(p^k, w) \tag{7}
$$

subject to

$$
p^k C \leq B
$$

$$
z_i^k \in \{0, 1\} \quad \forall i = 1, \ldots, m,
$$

where $p^k C \leq B$ represents the set of problem specific constraint equations. Through the constraints the decision maker can model various types of restrictions, such as multiple resources, attribute balance requirements or mutually exclusive and mandatory alternatives. By adding auxiliary variables that are dependent on the instrumental indicators, we could add resource or benefit synergies to the model. From the viewpoint of our algorithm, modelling of the capital budgeting problem (7) is fairly flexible. The only definitive requirements are that (i) the objective has to be linear with respect to the weights, and (ii) the feasibility conditions have to be independent of the weights.
4 Proposed Framework

Consider a capital budgeting problem with multi-attribute alternatives and "incompletely defined weights." We assume that the basic alternatives are given and scored. Portfolio construction follows the representation of Section 3.2. Then, different weights may indicate different solutions to the budgeting problem (7), and a new problem is to determine, which portfolios of the basic alternatives can turn out optimal when the final weight vector can take any value within the feasible weight region. In other words, we are interested in finding the set of potentially optimal portfolios (see Section 2.3 for potential optimality).

The feasible weight region is defined through a set of linear inequalities. Without loss of generality, we may assume that each component of the weight region is non-negative. In fact, the only definitive requirement is that \( S_0 \) forms a polytope. By definition, polytope is a bounded polyhedron, i.e. an intersection of a finite number of closed halfspaces. Since halfspaces are always convex, \( S_0 \) is automatically convex as an intersection of convex sets (see, e.g., Bazaraa, 1993 for convexity and polyhedral sets). The form of the original feasible weight region to be split, denoted by \( S_0 \), is generally

\[
S_0 = \{ w = (w_1, w_2, \ldots, w_n) \mid w_i \geq 0, \, l_{ij} w_j \leq w_i \leq u_{ij} w_j, \, l_i \leq w_i \leq u_i, \, \sum_i w_i = s \}, \tag{8}
\]

where, \( l_{ij} \) and \( u_{ij} \) are relative lower and upper bounds, respectively, on comparison between attribute weights \( w_i \) and \( w_j \); \( l_i \) and \( u_i \) are absolute lower and upper bounds, respectively, for weight \( w_i \); and is \( s \) a fixed sum of the weights. By the definitive requirement of \( S_0 \) forming a polytope, the definition (8) does not have to include all the types of restrictions above and it may include more complex linear types as well.

Further to the assumptions considering the linear properties of portfolio value construction and the form of \( S_0 \), the algorithm requires that the weight region splits into finitely many subsets, i.e. there are finitely many potentially optimal portfolios. Thus, the number of alternatives must be finite, and the instrumental indicator variables, \( z_i \), must be discrete and take only finitely many values. In the context of capital budgeting we use binary \( z_i \)'s, and therefore it is sufficient to assume that there is a finite number of alternatives.
The algorithm is based on the linear properties of the selection problem. Before introducing the algorithm, let us summarize the key assumptions.

- The portfolio value is linear with respect to the weights.
- All scores and weights are non-negative
- The feasibility conditions of the portfolio construction are not affected by the weights
- The feasible weight region $S_0$ is a convex polytope
- The indicator variables $z_i$ the number of alternatives are discrete and finite.

4.1 Geometric Grounds

Consider two fixed portfolios, $p^1$ and $p^2$, and their respective score vectors $P^1$ and $P^2$. Since the portfolios are fixed, the score vectors are constant vectors of size $(1 \times n)$. Now, consider the variation of the respective portfolio values $V(p^k, w) = P^k w^\top$ when varying the weight within the region $S$. In this respect, the following Lemma 1 and Lemma 2 capture important aspects.

**Lemma 1** Assume that the values of two fixed portfolios, $p^1$ and $p^2$, are such that $V(p^1, w^1) > V(p^2, w^1)$ and $V(p^1, w^2) < V(p^2, w^2)$, $w^1, w^2 \in S$. Then, there exists a hyperplane $H = \{w \in S \mid V(p^1, w) - V(p^2, w) = 0\}$, that splits $S$ into two distinct halfspaces $H^+$ and $H^-$, s.t. $H^+ = \{w \in S \mid V(p^1, w) - V(p^2, w) > 0\}$ and $H^- = \{w \in S \mid V(p^1, w) - V(p^2, w) < 0\}$.

**Proof.** Take the difference $p^\# = p^1 - p^2$ and the respective score $P^\# = P^1 - P^2$. By the assumption, $V(p^\#, w^1) > 0$ and $V(p^\#, w^2) < 0$. As a linear function, $V(p^\#, w)$ is continuous. Therefore, by Bolzano’s theorem, there exists a $w \in S$ such that $V(p^\#, w) = P^\# w = 0$. Geometrically, the condition is fulfilled by vectors that are perpendicular to $P^\#$, i.e. vectors that compose the hyperplane stated. Then, based on linear algebra (see, e.g., Bazaraa, 1993), the region splits into $H^+$ and $H^-$ as proposed. □

Now, let two distinct weight vectors $w^1, w^2 \in S$ indicate different optimal portfolios, say $p^1 = p^*(w^1) \neq p^*(w^2) = p^2$. By Lemma 1 we know that the region can be divided into distinct halfspaces with inverse rank order for the two portfolios. Thus, portfolio $p^2$ is
certainly not potentially optimal in the halfspace \( \{ w \in S \mid V(p^1, w) - V(p^2, w) > 0 \} \), since at least portfolio \( p^1 \) offers higher value in the particular subset. Also, as a basic inequality the value order comparison \( V(p^{k_1}, w) > V(p^{k_2}, w) \) is transitive. Thus, if we have found a subset \( S_k \) indicating \( V(p^1, w) > V(p^2, w) \), and then within \( S_k \) we find \( S_{k+1} \subset S_k \) indicating \( V(p^3, w) > V(p^1, w) \), it immediately follows that \( V(p^3, w) > V(p^2, w) \), \( w \in S_{k+1} \).

On the other hand, if we find distinct weights that indicate the same optimal portfolio, then by Lemma 2 we can draw further conclusions.

**Lemma 2** Assume that there are weight vectors \( w^i \in S, \quad i = 1, 2, \ldots \) such that \( V(p^1, w^i) > V(p^2, w^i) \) \( \forall i \). Then, \( V(p^1, w) > V(p^2, w) \) in the convex hull spanned by the vectors \( w^i \).

**Proof.** If \( V(p^1, w^i) > V(p^2, w^i) \) \( \forall i \), then by Lemma 1, \( w^i \in H^+ \forall i \). As a halfspace, \( H^+ \) is convex. By definition, the weights in the convex hull of \( w^i \)'s are convex combinations of the \( w^i \)'s, and thus belong to \( H^+ \). Thus, by Lemma 1, the proposed condition holds. \( \Box \)

Now, recall the concept of optimality, i.e. \( p^*(w^i) \) is optimal \( \Leftrightarrow V(p^*, w^i) > V(p^k, w^i) \) for all feasible \( p^k \). Thus, if we find weight vectors \( w^i \) that all indicate the same optimal portfolio, say \( p^1 = p^*(w^i) \) \( \forall i \), then by Lemma 2, we can conclude that the respective convex hull also indicates this optimal portfolio. Arbel (1989) also proved similar result concerning rank orders, within the context of the AHP. A quote of Arbel’s theorem runs "If all vertices of the solution subset exhibit a certain rank order, then any interior will exhibit the same rank order too." This is basically a stronger form of our theorem, but Lemma 2 alone provides sufficient grounds for determining optimal portfolios.

### 4.2 The Algorithm

The above Lemmas provide tools to divide the initial \( S_0 \) into subsets \( S_k \) each indicating a unique optimal portfolio. Together these portfolios constitute the set of potentially optimal portfolios. By assuming that there is a finite number of potentially optimal portfolios, the initial \( S_0 \) is divided into finitely many subsets \( S_k \).

All along, the weight subsets take a form of a polytope. Then, with an all-linear capital budgeting problem, the computation is basically linear algebra and linear programming.
The computational efficiency of the algorithm follows from the facts that the maximization (7) takes place only in the extreme points of the subsets and that the extreme points of consequent subsets are efficiently generated by utilizing the existing ones. By Lemma 2, the algorithm is able to draw conclusions concerning entire subsets without examining the interior points of the respective area. Thus, a complex problem with lots of alternatives and incompletely defined parameters can be solved with few computational rounds.

4.2.1 Notational Convention

A polytope is uniquely determined by its extreme points. A polytope equals the convex hull spanned by its extreme points. Let $\text{ext}(S)$ denote the set of the extreme points of a convex polytope $S$, i.e.

$$\text{ext}(S) = \{w \in S \mid \exists w', w'' \in S \text{ s.t. } w' \neq w'' \land w = \lambda w' + (1-\lambda) w'' \text{ for some } \lambda \in (0, 1)\}$$

Because the number of constraints on $S_0$ is finite, the number of elements in $\text{ext}(S_0)$ is finite. The algorithm adds linear constraints to define new descendant sets, but by the assumption of finity, all of the subsets created are nevertheless defined by finitely many constraints and thus $\text{ext}(S_k)$ remains finite for all $k$. We assume that the extreme points of the initial $S_0$ can be found by an LP-package, for example.

Adding linear constraints to sets that are recursive descendants of $S_0$ implies that new subsets are created by cutting a polytope with a hyperplane. Thus, all the descendants are polytopes as well. When generating the extreme points of the descendant sets (i.e. identifying the sets), the extreme points of the parent set can utilized. This procedure reduces the computational effort of our algorithm. To get a grip on the procedure, we need a definition of adjacent extreme points. The first one below is a geometric definition (see, e.g., Murty, 2002).

Two extreme points $w^1$, $w^2$ of a convex polyhedron $S \subset \mathbb{R}^n$ are said to be adjacent iff every point $w$ on the line segment joining them satisfies:

$$w^3, w^4 \in S, \ w = \alpha w^3 + (1 - \alpha)w^4 \text{ for some } 0 < \alpha < 1 \Rightarrow w^3, w^4 \text{ are also on the line segment joining } w^1, w^2.$$ 

Another commonly used definition is based on the basic solutions of an LP problem over the respective polytope (see e.g. Reveliotis, 2002).
Two basic feasible solutions of an LP with \(m\) linear constraints in standard form are said to be adjacent, if their bases have \(m - 1\) variables in common.

Following the above definitions, let \(\text{adj}(S)\) denote the set of adjacent extreme points of region \(S\), i.e.

\[
\text{adj}(S) = \{(w^i, w^j) \mid w^i, w^j \in \text{ext}(S), \ w^i \neq w^j ; \ w^i, w^j \text{ adjacent}\}
\]

By the above references, for example, an \textit{edge} of a polytope refers to a line segment connecting two distinct adjacent extreme points. Let us define an edge, \(e(w^i, w^j)\),

\[
e(w^i, w^j) = \{w \in S \mid w = \alpha w^i + (1 - \alpha) w^j, \ \alpha \in [0, 1], \ w^i, w^j \in \text{adj}(S)\}
\]

With these notational conventions we may now introduce the algorithm dividing the feasible weight region between respective optimal portfolios.

### 4.2.2 Progress of the Algorithm

In each iteration, the algorithm treats a subset \(S_k \subseteq S_0\). Let set \(F\) serve as a register for generated \(S_k\):s that have not yet been treated, i.e. \textit{pending-fathomed} subsets. The algorithm removes a subset \(S_k\) from the register \(F\) and either splits it in two and replaces \(S_k\) in \(F\) with the resulting descendants or finds \(S_k\) \textit{fathomed} and adds it to set \(L\), the register of fathomed regions.

The choice of procedure on a particular \(S_k\) is determined by the number of different optimal portfolios found in the extreme points of \(S_k\):

\[\text{i.} \text{ If the extreme points of } S_k \text{ indicate two or more different optimal portfolios, then } S_k \text{ is divided in two, } S_{k+1} \text{ and } S_{k+2}, \text{ by Lemma } 1.\]

\[\text{ii.} \text{ If the extreme points of } S_k \text{ all indicate the same optimal portfolio, then by Lemma } 2, \text{ the interior of } S_k \text{ is concluded to indicate the particular portfolio.}\]

If there are more than two different optimal portfolios in \text{i}, selecting the two portfolios to determine the split is a matter of free choice. In \text{ii}, it is only the interior of \(S_k\) that
strictly indicates the particular portfolio *uniquely* optimal. For $S_k$, $k \geq 1$ all of its edges intersecting the initial $S_0$ are defined by equality of two portfolio values. Thus, if fathomed subsets $S_l$ and $S_l$ have a joint edge, then the edge indicates equal value to the portfolios that are strictly optimal within the sets.

Once a subset is fathomed, we set a label that links the subset and the respective optimal portfolio together. Now, let $S_k(p^{opt_k})$ denote that a particular portfolio $p^{opt_k}$ is optimal within the entire $S_k$, i.e.,

$$S_k(p^{opt_k}) = \{ w \in S \mid V(p^{opt_k}, w) \geq V(p^l, w) \text{ for all feasible } p^l \}.$$  

Note, that practically $S_k(p^{opt_k})$ indicates the subset $S_k$, with an extension of linking the optimal portfolio to it. In the main algorithm, set $L$ is then gradually incremented to contain the potentially optimal portfolios and their respective regions, i.e. $L = \bigcup_k S_k(p^{opt_k})$.

The algorithm terminates, when $S = \emptyset$ and consequently $L = S_0$, i.e. there are no pending-fathomed subsets to treat but the union of treated subsets equals the original $S_0$.

In addition, there is a subalgorithm which generates the extreme points of the descendants and updates the adjacency record. It utilizes the fact that all new sets generated by the main algorithm are subsets of earlier ones. Existing adjacent extreme points of parent $S_k$ that belong to its descendant are adjacent extremes of the particular descendant. New extreme points for the descendant arise in the intersection of the edges of parent $S_k$ and the cutting hyperplane. The subalgorithm determines the adjacency relations of the new extreme points based on previous adjacency information. Thus, information concerning the extreme points is updated recursively instead of generating it "from scratch" for each new subset (i.e. solely by the set of constraint equations).
Main algorithm

Step 0. Set $F = \{S_0\}$ and $L = \emptyset$. Furthermore, find sets $\text{ext}(S_0)$ and $\text{adj}(S_0)$.

Step 1. Take $S_k \in F$, such that $k \leq l \quad \forall S_l \in F$.

Step 2. Calculate $p^*(w^i) \quad \forall w^i \in \text{ext}(S_k)$.

Step 3. If $p^*(w^i) \neq p^*(w^j)$ for some $w^i, w^j \in \text{ext}(S_k)$,

Then define a hyperplane $H = \{w \in W \mid p^*(w^i) - p^*(w^j) = 0\}$ and halfspaces $H^+ = \{w \in W \mid p^*(w^i) - p^*(w^j) \geq 0\}$; $H^- = \{w \in W \mid p^*(w^i) - p^*(w^j) \leq 0\}$.

Next, define sets $S_{2k+1} = \{S_k\} \cap \{H^+\}$ and $S_{2k+2} = \{S_k\} \cap \{H^-\}$.

Also, set $F \leftarrow (F\backslash \{S_k\}) \cup \{S_{2k+1}\} \cup \{S_{2k+2}\}$.

Furthermore, update $\text{ext}(S)$ and $\text{adj}(S)$ with the subalgorithm for both $S_{2k+1}$ and $S_{2k+2}$.

On the other hand, if $p^*(w^i) = p^*(w^j)$ \forall $w^i, w^j \in \text{ext}(S_k)$, set $L \leftarrow \{L\} \cup \{S_k(p^{\text{pre}})\}$ and remove $S_k$ from $F$, i.e. $F \leftarrow (F\backslash \{S_k\})$.

Step 4. If $F = \emptyset$, terminate. Otherwise, go to Step 1.

Subalgorithm

**Step I.** Set $\text{ext}(S_{2k+1}) = \{w \in \text{ext}(S_k) \mid w \in H^+\} \cup \{w \in H \cap e(w^i, w^j) \mid w^i \in \text{ext}(S_k) \cap H^+ \land w^j \in \text{ext}(S_k) \cap H^-\}$.

**Step II.** Set $\text{adj}^i(S_{2k+1}) = \{(w^i, w^j) \mid (w^i, w^j) \in \text{adj}(S_k) \land w^i \in H^+\}$ and $\text{adj}^m(S_{2k+1}) = \{(w^i, w^k) \mid w^k \in H \cap e(w^i, w^j); (w^i, w^j) \in \text{adj}(S_k), w^i \in H^+ \land w^j \in H^-\}$

**Step III.** Set $\nu$-definitions for abbreviations,

$\nu_1 : w^{k'} = H \cap e(w^i, w^j'), \text{s.t. } w^i \in \text{ext}(S_k) \cap H^+ \land w^j \in \text{ext}(S_k) \cap H^-$

$\nu_2 : w^{k''} = H \cap e(w^{k'}, w^{j''}), \text{s.t. } w^{k'} \in \text{ext}(S_k) \cap H^+ \land w^{j''} \in \text{ext}(S_k) \cap H^-$

$\nu_3 : w^{k'} \neq w^{k''}$

$\nu_4 : w^i, w^{j''} \in \text{adj}(S_k) \cap H^+$

$\nu_5 : w^i, w^{j''} \in \text{adj}(S_k) \cap H^-$

Set $\text{adj}^m(S_{2k+1}) = \{(w^{k'}, w^{k''}) \mid \nu_1, \nu_2, \nu_3, \nu_4, \nu_5\}$

**Step IV.** Set $\text{adj}(S_{2k+1}) = \text{adj}^i(S_{2k+1}) \cup \text{adj}^m(S_{2k+1}) \cup \text{adj}^m(S_{2k+1})$

The above representation creates sets $\text{ext}(S_{2k+1})$ and $\text{adj}(S_{2k+1})$. To create the respective sets for $S_{2k+2}$, replace $S_{2k+1}$ with $S_{2k+2}$. In addition, swap $H^+$ with $H^-$ and vice versa.
The initial feasible weight region $S_0$ is divided into a hierarchical tree consisting of subsets of $S_0$. Note that the tree can be constructed either by "breadth first" or "depth first", depending on Step 1. The form of the algorithm above is "breadth first", since Step 1 chooses $S_k$ with the smallest index number $k$ to be further examined. "Depth first" principle could be applied by choosing $S_k$ with the highest index. Either way, the number of rounds required remains the same. The numerical example in Section 5 provides insight into how the algorithm proceeds.

Convergence of the algorithm is assured, because of the assumption that $S_0$ is divided into a finite number of subsets. In the algorithm, Step 3 manipulates the set $F$. In Step 3, a single element of $F$, a subset of the original $S_0$ effectively, is taken into consideration. In each round of the algorithm, such subset $S_k \subseteq F$ is either split in two or removed (i.e. transferred to increment the set $L$). Removal of a subset clearly reduces the size of the remaining $F$, in a sense that $(F\setminus\{S_k\}) \subseteq F$. If $S_k$ is split, the aggregate size of the descendants equals the size of the region just split, i.e. $\{S_{2k+1}\} \cup \{S_{2k+2}\} = \{S_k\}$, and it holds that $F = (F\setminus\{S\}) \cup \{S_{2k+1}\} \cup \{S_{2k+2}\}$. Thus, the union of the elements of $F$ remains unchanged if current $S_k$ is divided, and is strict subset of the preceding $F$ if $S_k$ is removed.

By the assumption of finite number of portfolios, each subset is finally divided into two $S_k$:s for which $p^*(w^i) = p^*(w^j) \ \forall w^i, w^j \in \text{ext}(S_k)$, that are then removed. Thus, the set $F$ is not enlarged in any phase, and one by one, each element is removed. Therefore, it holds that finally $F = \emptyset$, and the algorithm terminates.
5 Illustrative Numerical Example

The example represents a very simple case of selecting three-criteria portfolio from a set of 5 independent basic alternatives (projects) that consume one scarce resource. In this example the weights represent traditional preference weights, i.e. they sum to one. The criteria specific outcomes of the basic alternatives are fully hypothetical figures representing general units of component value, then scaled by the weight factors to aggregate the total value.

The main idea of this example is to graphically illustrate the progress of the algorithm. The size of the problem would easily allow us to solve it by other means as well, but on the other hand, the limited dimensionality enables the graphical step by step illustration. There are three criteria dimensions, and the summation requirement further drops the dimensionality by one. Thus, the feasible weight region to be split is a plane.

The size and simplicity of the problem do not highlighting the efficiency of the method in solving complex problems. With all the intermediate phases written down, the algorithm may seem inefficient in such a simple case. However, the actual computation is straightforward and sets fairly light requirements for the computational power running it. Much of the favorable properties of method lie within the treatment of interdependent alternatives and additional restrictions. Such features increase the complexity of the problem significantly, but considering the power of current LP-packages, they do not appreciably affect the efficiency of our method. Furthermore, the splitting process of the incompletely determined weight region illustrated in this example is basically not affected by the complexity of the portfolio construction LP-problem.

The following pages present the progress of the algorithm. At each phase, the contents of the data stacks are written down and the new cutting plane along with the resulting subsets are shown graphically. The tree-form division graphs next to the figures illustrate the branching process in set notation.

In this example, the solution portfolios differ from each other only by one project, and one of the projects is not included in any of the potentially optimal portfolios. This is partly a consequence of the small size of the problem, but also a fairly common situation with larger problems. With additivity and non-negative \( \nu_i \)'s, determining between such portfolios becomes a "traditional" pairwise comparison between the alternatives that distinguish the
portfolios. However, reducing the problem into such comparisons is difficult without our method. It is also noteworthy, that dominated alternatives cannot be excluded from the list of alternatives, since the problem constraints often lead to situations where alternatives that are dominated in isolation are still selected in the optimal portfolio.

The alternative projects for the example portfolio construction are presented in the table below. There is a budgetary limit of 10 units, which basically allows choosing 2-3 of the possible five projects.

<table>
<thead>
<tr>
<th></th>
<th>Alt. 1</th>
<th>Alt. 2</th>
<th>Alt. 3</th>
<th>Alt. 4</th>
<th>Alt. 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Crit. 1</td>
<td>2</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>Crit. 2</td>
<td>3</td>
<td>3</td>
<td>7</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>Crit. 3</td>
<td>7</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>8</td>
</tr>
<tr>
<td>Cost</td>
<td>4</td>
<td>5</td>
<td>4</td>
<td>1</td>
<td>6</td>
</tr>
</tbody>
</table>

The numerical elements in the table directly represent the scores \( v_i(x^j_i), \ i = 1, 2, 3, \ j = 1, \ldots, 5 \). Thus, the aggregate value to be maximized is obtained by multiplying the figures of the table above by the weights \( w_i, \ i = 1, 2, 3 \) and then adding these attribute-specific values together.

The original feasible weight region \( S_0 \) is defined by the following restrictions:

\[
\begin{align*}
    w_1 & \geq \frac{2}{3} w_2 \\
    w_1 & \leq 2 w_2 \\
    w_1 & \geq 0.85 w_3 \\
    w_2 & \leq 3 w_3 \\
    w_i & \geq 0, \ i = 1, 2, 3 \\
    \sum_{i=1}^{3} w_i & = 1.
\end{align*}
\]

Note, that by solely focusing on the scores of the alternatives, the alternative \( a_4 \) is dominated by all the others and alternative \( a_5 \) dominates all the others. However, these two are the least and the most expensive alternatives, respectively. Some of the solutions will include the dominated \( a_4 \), but none includes the dominant \( a_5 \).
In the beginning, the register for (sub)sets to be treated is $F = \{S_0\}$, and the register for fathomed (sub)sets is $L = \emptyset$. Now, treat the sole $S_0$.

Extreme points of $S_0$ and their adjacency relations are:

$$\text{ext}(S_0) = \begin{pmatrix} w^1 \\ w^2 \\ w^3 \\ w^4 \end{pmatrix} = \begin{pmatrix} (34/91 , 17/91 , 40/91) \\ (34/125 , 51/125 , 40/125) \\ (1/3 , 1/2 , 1/6) \\ (6/10 , 3/10 , 1/10) \end{pmatrix}$$

$$\text{adj}(S_0) = \{(w^1 , w^2) , (w^2 , w^3) , (w^3 , w^4) , (w^4 , w^1)\}$$

The optimal portfolios in each of extreme points, constructed by the LP-solution with fixed weights,

$$p^*(w^1) = (1, 1, 0, 1, 0)$$

$$p^*(w^2) = (1, 0, 1, 1, 0)$$

$$p^*(w^3) = (0, 1, 1, 1, 0)$$

$$p^*(w^4) = (0, 1, 1, 1, 0).$$
All of the extreme points of $S_0$ do not indicate the same optimal portfolio. Thus, a cutting plane is required. We *choose* the plane separating portfolios $p^*(w^1)$ and $p^*(w^2)$.

$$p^*(w^1) \neq p^*(w^2) \implies H = \{(w_1, w_2, w_3) \mid 4w_1 - 4w_2 + w_3 = 0\}$$

The plane and the resulting subsets are shown in Figure 2.

![Figure 2: The first split and the extreme points of $S_1$.](image)

Now, $F = \{S_1, S_2\}$ and still $L = \emptyset$. Select treating $S_1$.

Extreme points and their adjacencies:

$$\text{ext}(S_1) = \begin{pmatrix} w^1 \\ w^5 \\ w^6 \\ w^4 \end{pmatrix} = \begin{pmatrix} 34/91, 17/91, 40/91 \\ 17/59, 22/59, 20/59 \\ 11/27, 12/27, 4/27 \\ 6/10, 3/10, 1/10 \end{pmatrix}$$

$$\text{adj}(S_1) = \{(w^1, w^5), (w^5, w^6), (w^6, w^4), (w^4, w^1)\}$$
And the optimal portfolios in the extreme points of $S_1$:

$p^*(w^1) = (1, 1, 0, 1, 0)$
$p^*(w^5) = (1, 1, 0, 1, 0)$
$p^*(w^6) = (0, 1, 1, 1, 0)$
$p^*(w^4) = (0, 1, 1, 1, 0)$

The extreme points indicate two different optimal portfolios. We take a cutting plane to
separate these, say $p^*(w^1)$ and $p^*(w^4)$.

$p^*(w^1) \neq p^*(w^4) \Rightarrow H = \{(w_1, w_2, w_3) \mid w_1 - 4w_2 + 4w_3 = 0\}$

This split results in the situation of Figure 3. Note, that the new cut only affects within
the particular subset under treatment.

Figure 3: Further splitting $S_1$ and the extreme points of $S_3$.

Now we have three pending-fathomed subsets to be treated, $F = \{S_3, S_4, S_2\}$. Note that
the union $\cup\{S_3, S_4, S_2\}$, i.e. the "size" of $F$ still equals the original $S_0$. Thus, we have
not fathomed any subsets, and $L = \emptyset$. By applying the depth first principle, we select
treating the subset $S_3$ next.
The extreme points of $S_3$ that are already shown in Figure 3 are

$$\text{ext}(S_3) = \begin{pmatrix} w^1 \\ w^5 \\ w^8 \\ w^7 \end{pmatrix} = \begin{pmatrix} (34/91 , 17/91 , 40/91) \\ (34/125 , 51/125 , 40/125) \\ (4/13 , 5/13 , 4/13) \\ (4/7 , 2/7 , 1/7) \end{pmatrix}$$

$adj(S_3) = \{(w^1, w^5) , (w^5, w^8), (w^8, w^7), (w^7, w^1)\}$

And the optimal portfolios:

$p^*(w^1) = (1, 1, 0, 1, 0)$
$p^*(w^5) = (1, 1, 0, 1, 0)$
$p^*(w^8) = (1, 1, 0, 1, 0)$
$p^*(w^7) = (1, 1, 0, 1, 0)$

Now, all of the extreme points do indicate the same optimal portfolio $(1, 1, 0, 1, 0)$. Thus, the entire subset $S_3$ is fathomed to indicate this portfolio. A label set $S_k(p^{**})$ is created and attached to the register of treated subsets, $L$.

$p^*(w^i) = p^*(w^j) = (1, 1, 0, 1, 0) \; \forall w^i, w^j \in \text{ext}(S_3) \Rightarrow S_3((1, 1, 0, 1, 0))$

The bracketed $\{1,2,4\}$ in the branch-down graph of Figure 3 indicates that the respective subset $S_3$ is fathomed to indicate a portfolio of alternatives 1, 2 and 4.

Now, both registers are "open", $F = \{S_1, S_2\}$ and $L = S_3((1, 1, 0, 1, 0))$. The "size" of $F$ no longer equals $S_0$. Again, treating the top of the stack, i.e. $S_4$.

$$\text{ext}(S_4) = \begin{pmatrix} w^4 \\ w^6 \\ w^8 \\ w^7 \end{pmatrix} = \begin{pmatrix} (6/10 , 3/10 , 1/10) \\ (11/27 , 12/27 , 4/27) \\ (4/13 , 5/13 , 4/13) \\ (4/7 , 2/7 , 1/7) \end{pmatrix}$$

$adj(S_4) = \{(w^4, w^6) , (w^6, w^8) , (w^8, w^7), (w^7, w^4)\}$

$p^*(w^4) = (0, 1, 1, 1, 0)$
$p^*(w^6) = (0, 1, 1, 1, 0)$
$p^*(w^8) = (0, 1, 1, 1, 0)$
$p^*(w^7) = (0, 1, 1, 1, 0)$
Figure 4: The extreme points of \( S_4 \). There are no new subsets, since both \( S_3 \) and \( S_4 \) were fathomed and thus no new cutting planes were created.

Similarly to \( S_3 \), the subset \( S_4 \) is fathomed, and register \( L \) is supplemented.

\[
p^*(w^i) = p^*(w^j) = (0, 1, 1, 1, 0) \quad \forall w^i, w^j \in \text{ext}(S_4) \quad \Rightarrow S_4((0, 1, 1, 1, 0))
\]

Since there was no new split as \( S_3 \) was fathomed in the previous phase, Figure 4 only exhibits the extreme points of \( S_4 \). The branch-down graph again exhibits the conclusion of \( S_4 \).

After concluding the descendants of \( S_1 \), we have \( F = \{ S_2 \} \) and \( L = S_3((1, 1, 0, 1, 0)) \cup S_4((0, 1, 1, 1, 0)) \). Thus, we move to the other branch of the first split, i.e. treat \( S_2 \).

Figure 5 illustrates the extreme points of \( S_2 \).

\[
\text{ext}(S_2) = \begin{pmatrix} w^5 \\ w^2 \\ w^3 \\ w^5 \end{pmatrix} = \begin{pmatrix} (17/59 , 22/59 , 20/59) \\ (34/125 , 51/125 , 40/125) \\ (1/3 , 1/2 , 1/6) \\ (11/27 , 12/27 , 4/27) \end{pmatrix}
\]
Figure 5: The extreme points of $S_2$

$$\text{adj}(S_2) = \{(w^5, w^2), (w^2, w^3), (w^3, w^6), (w^6, w^5)\}$$

$$p^*(w^5) = (1, 0, 1, 1, 0)$$
$$p^*(w^2) = (1, 0, 1, 1, 0)$$
$$p^*(w^3) = (0, 1, 1, 1, 0)$$
$$p^*(w^6) = (0, 1, 1, 1, 0)$$

There are different optimal portfolios and thus a cutting plane is required.

$$p^*(w^5) \neq p^*(w^6) \Rightarrow H = \{(w_1, w_2w_3) \mid w_1 - w_3 = 0\}$$

The resulting subsets are shown in Figure 6. Again, the cut has no effect outsize the bounds of $S_2$, i.e. the subset $S_3$ remains fathomed in spite of the new plane cutting through it.
Figure 6: The split on $S_2$ and the extreme points of the descendants $S_5$ and $S_6$

Now, $F = \{S_5, S_6\}$ and $L = S_3((1, 1, 0, 1, 0)) \cup S_4((0, 1, 1, 1, 0))$. For short, we consider treating both $S_5$ and $S_6$.

It turns out that:

\[
p^*(w^t) = (1, 0, 1, 1, 0) \quad \forall w^t \in \text{ext}(S_5) \Rightarrow S_5((1, 0, 1, 1, 0))
\]

\[
p^*(w^t) = (0, 1, 1, 1, 0) \quad \forall w^t \in \text{ext}(S_6) \Rightarrow S_6((0, 1, 1, 1, 0))
\]

That is, both $S_5$ and $S_6$ become fathomed. Since there are no new cutting planes and there are no unexplored branches left in search graph, $F = \emptyset$ and $L = S_3((1, 1, 0, 1, 0)) \cup S_4((0, 1, 1, 1, 0)) \cup S_5((1, 0, 1, 1, 0)) \cup S_6((0, 1, 1, 1, 0)) = S_0$. Thus, the entire original $S_0$ is split and treated, and the algorithm terminates.

The resulting potentially optimal portfolios are:

- $(1, 1, 0, 1, 0)$, i.e. projects 1, 2 and 4, on subset $S_3$
- $(0, 1, 1, 1, 0)$, i.e. projects 2, 3 and 4, on subsets $S_4$ and $S_5$
- $(1, 0, 1, 1, 0)$, i.e. projects 1, 3 and 4, on subset $S_6$. 

28
6 Discussion

6.1 Properties of the Algorithm

Our capital budgeting framework extends "conventional" capital budgeting by allowing incompletely defined parameters. We have evolved the framework from the concept incomplete weights in the field of MCDM. The algorithm presented in this paper efficiently detects potentially optimal solutions from a large set of multi-attribute alternatives under complex feasibility restrictions. Furthermore, it can be extended to a more advanced algorithm that detects all non-dominated portfolios. Formulation of the sole capital budgeting problem is but an ordinary LP problem – the novelty of the framework is in the incomplete information. The algorithm itself is a combinator: basic alternatives and their relations are first assessed individually, and the algorithm seeks the potentially optimal combinations, with respect to the possible realizations of the incompletely defined parameters.

The strengths of our framework are that (i) potentially optimal portfolios are found without studying each feasible weight and/or combination, (ii) the weight region is accurately divided into subsets that each indicate a unique optimal portfolio and (iii) full dominance structure can be established more efficiently than calculating pairwise comparisons (However, the full dominance structure is only in the scope of the advanced algorithm). More efficient search method is advantageous especially in combinatorial problems, since the number of possible combinations often gets high (2\(^m\) combinations from \(m\) basic alternatives with binary indicators). Furthermore, if all basic alternatives offer non-negative net value, only the largest feasible combinations are interesting, since subsets of larger combinations are trivially dominated by the respective larger combination.

Reduction of calculation appears such that the algorithm does not run inconclusive iterations: each portfolio that is proposed during the iteration is potentially optimal – we do not have to construct all feasible combinations for comparison. Similar advantage could be attained by simulating the weight realizations and finding the optimal combination in each simulation point, but the linear properties of the problem allow an explicit and accurate division of the original feasible region. Furthermore, the algorithm proceeds top down on the value scale, i.e. if there is a single portfolio that offers the highest value in the entire feasible weight region then the algorithm stops in one iteration. In general, once
a subset is concluded to indicate a particular portfolio, it is found fathomed and cut off from further consideration. Thus, the search is highly focused, opposed to a "mechanical" search that always calculates through all possibilities.

However, this first version of the algorithm detects only the potentially optimal portfolios. Considering decision recommendations, all non-dominated portfolios should be exposed to further analysis. Especially with decision rules that emphasize robust solutions, the best feasible portfolio may be missed by analyzing the potentially optimal portfolios only. Furthermore, our algorithm does not measure the size of the of the resulting subsets in any sense. Attaching the respective region (defined by the set of extreme points) to the portfolios is yet more than solely indicating the portfolios, but as such our algorithm does not calculate, e.g., probability measures of any kind to determine mutual preferences between the potentially optimal portfolios detected.

6.2 Implications of the Assumptions

In our framework, the basic alternatives are first assessed individually and combinations are constructed using these numerical assessments. Thus, the actual decision alternatives are not explicitly appraised against each other. The assumption of portfolio score additivity (4) makes the portfolio value construction rather restrictive and thus limits nature of attributes applicable in this framework. Since we can use identity mapping as the score function, i.e. \( v_i(x^i_j) = x^i_j \), the assumption basically requires the alternatives’ attribute specific performances to be reasonably additive. Absolute scores are applicable, e.g., monetary (weights representing scenario probabilities, discount factors or spatial scaling factors) or direct quantity (weights representing unit prices).

If the attributes are subjective of nature and/or incommensurate as such, and the decision maker has used relative attribute value functions (Keeney and Raiffa, 1976; French, 1988; Clemen, 1996, for example) as the scores in the initial assessment, the assumption (4) is undermined. Value functions are either constructed through inter-attribute value tradeoffs to directly indicate component value or by first cardinally ranking the alternatives attribute by attribute and then considering the relative importance of the attributes. Either way, the set of alternatives available and the implied ranges of attribute specific performances are determinant in the multi-attribute value theory. In attribute-specific relative score construction, the score is typically normalized into \( v_i(x^i_j) \in [0, 1] \) such that
the best plausible $x_i^j$ takes value of one and the value of zero is attached to the worst plausible performance. Combinations of the alternatives are basically new alternatives that then invalidate the prior relative scores by altering the ranges of assessment. Direct summation of normalized scores would be unjustified, since the bounds of normalization would easily be broken. For example, how to interpret a score $0.8 + 0.7 = 1.5$ on a scale of $[0,1]$. Thus, relative (normalized) scores seem to lose their usual interpretation in the aggregation.

Using relative weights to indicate the attributes’ mutual strength of preference also requires re-evaluation in a portfolio setting. In the light of MCDM theory, relative weights that the decision maker would use in calculating the overall value of the basic alternatives in the appraisal phase are not applicable as the portfolio weights. By the theory, value tradeoffs determining the weights are again attached to the ranges of attribute performances and thus become invalid as the ranges broaden with the new alternatives. Thus, in the assumption (6), the weights would have to be absolute (such as scenario probabilities), or the decision maker would have to set new, theoretically coherent preference weights on portfolio level with respect to the attribute performance ranges that the feasible portfolios can assume.

Yet another fundamental question is whether the attribute specific utilities (portfolio score functions) are increasing in the first place. If individual attribute scores are subjective grades, the assumption of "more the merrier" is unjustified. In such settings, claiming that two bad ones (say, grade 5 on a scale of 0 to 10) together would be better than a single good one (grade 8 for example) seems intuitively invalid. Especially with such subjective grades, using average score as the portfolio score would seem inviting. In a general portfolio problem use of weighted average score is directly arguable (see, e.g., Hallerbach et al., 2002), but with the fixed alternatives in capital budgeting it will require careful further consideration.

All the questions concerning value aggregation are related to the theory MCDM and its ability to build preference relations under incommensurate attributes. With its current assumptions, our framework is fairly restrictive concerning the nature of the attributes, and it cannot be seen as a true multiple criteria decision making tool. However, it does hold potential in many applications. Perhaps most promising would seem scenario analyses, since the difficulties in assessing subjective probabilities would benefit from the use of incomplete information and the concept of expected value fits well in our aggregation.
framework.

In fact, the assumptions of our algorithm are perhaps best met by a general discrete linear programming problem in which the multi-attribute alternative dependency can be broken. Consider linear pricing scheme (e.g., Luenberger, 1998) or multiple choice knapsack problem (e.g., Pisinger, 1995), for example. In linear pricing, take $i$ different securities, and let $w_i$ represents the price of security $i$ and $y_i$ the number of securities $i$ selected in the portfolio. The fixed baskets of security quantities are broken, but the resulting portfolio price (value) is again of additive form, i.e. $V = \sum w_i y_i$. Thus, the proposed framework applies as long as the other assumptions are met. Since $y_i$’s are independent quantities, binary indicator variables $z_j$ are not needed. For the convergence of the algorithm, the possible quantities $y_i$ must be discrete and finite for all $i$ to ensure there is only a finite number of possible security quantity combinations. Emerging from the idea of preference weights, discrete multi-objective optimization with the weighting method (e.g., Taha, 1997) would be another field of application for the relaxed form of the algorithm.

6.3 Extensions and Further Research

The algorithm presented in this paper is still fairly limited and leaves plenty of room for extensions. The algorithm itself (under the current assumptions) can be taken further. On the other hand, a thorough study of the value aggregation issues should lead to procedures allowing incommensurate attributes as well. Attaining the level of attribute flexibility that the established MCDM methods offer would broaden the applicability of our framework significantly.

Considering the capacity of the algorithm, the first step in further research would be to detect all non-dominated alternatives. It seems, that with a particular elimination procedure the algorithm of this paper can be used recursively to find all non-dominated portfolios. Also, it would be advantageous to be able to mutually rank the highlighted portfolios, e.g. in terms of expected value or probability of being optimal. In this respect, laying a joint probability distribution of the realizations over the original feasible weight region could benefit the analysis considerably. Especially with the weights representing unit prices or discount factors, using a probability distribution would allow robustness and risk analyses in the spirit of Value at Risk measure (see, e.g., Jorion, 2001).
Use of originally continuous decision variables should also be studied. Unrestricted allocation would be called for especially in financial market applications. On the other hand, with truly multi-criteria alternatives in particular, it would add the flexibility of the framework to allow incompletely defined alternative performances. Allowing a hierarchical attribute tree for one should be rather straightforward, recognizing the established works in the field (e.g., Salo and Hämäläinen, 1995).

Towards more flexible value aggregation, using fuzzy utility or averaging would seem promising in describing portfolio preferences (see Ramaswamy, 1998, for fuzzy utility with interest rate scenario application and Hallerbach et.al., 2002, for averaging with a security portfolio problem). With incommensurate, yet quantitative attributes that imply increasing utility functions we could first one by one maximize the attribute specific portfolio scores with respect to the resource restrictions, and then aim at maximizing the weighted additive fuzzy utility of the portfolio. In this approach, the attribute specific fuzzy utilities would present the level of attainment with respect to the maximal feasible performance, and the weights would reflect the relative importance of the attributes, for example. Thus, we could limit the portfolio values in the range of $[0,1]$ and bring the portfolio framework closer to the established value concepts of MCDM. A detailed theoretical consideration and exact representation of the above briefly described approach take high priority in the future. Furthermore, dealing with fully subjective grades should be considered.

The framework would also call for a case application. The framework allows various kinds of problem-specific modelling possibilities, including interdependent basic alternatives, and only through a real-life case they would become authentically evident. Also, creating a software tool to run the framework would create additional value, especially considering real-life cases. All in all, the framework presented in this paper opens various branches of research topics for the future.
7 References


