Approximative Algorithms in Robust Portfolio Modeling

Jussi Kangaspunta, 60689D
31.01.2007
Contents

1 Introduction 1

2 Robust Portfolio Modeling 3
   2.1 Additive Value ......................................... 3
   2.2 Resource Constraints and Feasible Portfolios ............. 4
   2.3 Incomplete Information .................................. 4
   2.4 Non-dominated Portfolios ................................ 5
   2.5 Core Indices ............................................ 6

3 Approximative Computation of Non-dominated Portfolios 7
   3.1 Potentially Optimal Portfolios ............................. 7
   3.2 Calculation of Non-dominated Portfolios Regarding Tchebycheff Metric ........................................ 9
   3.3 Problems with Dominated Solutions ......................... 13
   3.4 Behaviour of Core Indices During Computations .......... 15

4 Computational Experiments 15
   4.1 Experimental Design ....................................... 15
   4.2 Results .................................................. 17

5 Software Demonstration 25

6 Conclusions 27
1 Introduction

In multiple criteria project portfolio optimization, a combination (i.e. a portfolio) of projects is chosen from a large set of proposals by taking many objectives and Decision Maker’s (DM) preference statements simultaneously into account. Due to difficulties in obtaining reliable estimates on the projects’ future outcomes and the elicitation of preferences, project portfolio decisions are often made under incomplete information. Such multiple criteria problem arise, e.g., when dealing with R&D portfolios where aspects such as several project versions and interdependencies need to be taken into account simultaneously (Stummer and Heidenberger, 2003).

Liesiö et al. (2006) develop Robust Portfolio Modeling (RPM) framework which extends principles of preference programming (see Salo and Hämäläinen, 2004) methods into multiple criteria portfolio problems under incomplete information. RPM is a decision support framework that produces a set of non-dominated portfolios for attractive narrowing the scope of portfolios rather than suggesting a single optimal portfolio. The approach provides a systematic and transparent way to support project portfolio selection especially when the number of proposals is high.

Yet the number of non-dominated portfolios can be very high when there are, hundreds of project proposals, for instance. Thus, approximative methods are useful because it is not possible to solve large problems with exact algorithms on a personal computer in reasonable time.

The main objective of the study is to implement and compare two different approximative algorithms for solving non-dominated portfolios with in-
complete information. Algorithms are implemented with Java\textsuperscript{TM} and Mixed Integer Linear Programming (MILP) solver lp_solve.

The remainder of this study is organized as follows. Section 2 summarizes the RPM framework. Section 3 focuses on the approximative computation of non-dominated portfolios with two different algorithms. The performance and differences between algorithms are analyzed in Section 4. Section 5 describes the implemented software and conclusions are made in Section 6.
2 Robust Portfolio Modeling

2.1 Additive Value

In RPM, a set of \( m \) project proposals that are to be evaluated against \( n \) criteria is denoted with \( X = \{x^1, \ldots, x^m\} \). The performance (score) of project \( x^j \) against criteria \( i \) is \( v^j_i \) for all \( i = 1, \ldots, n \) and \( j = 1, \ldots, m \). Thus, the vector \( v^j = [v^j_1, \ldots, v^j_n] \) contains the scores of project \( x^j \). These vectors form the rows of the score matrix \( v \in \mathbb{R}^{m \times n} \) such that \([v]_{ji} = v^j_i\). In Multiattribute Value Theory (MAVT) the overall value of project \( x^j \) is represented with the weighted average of its scores (see e.g. Keeney and Raiffa, 1976)

\[
V(x^j) = \sum_{i=1}^{n} w_i v^j_i,
\]  

(1)

where the weight \( w_i \) measures the relative importance of the \( i \)th criterion. A project is preferred to another if it has the higher overall value of the two. The weights (i.e DM’s preferences) of all \( n \) criteria \( w = (w_1, \ldots, w_n)^T \) are scaled so that

\[
w \in S^0_w = \left\{ w \in \mathbb{R}^n \mid w_i \geq 0, \sum_{i=1}^{n} w_i = 1 \right\}.
\]  

(2)

A project portfolio \( p \subseteq X \) is a subset of all available projects. Thus, the set of all possible portfolios is given by the power set \( P = 2^X \). The overall value of a portfolio is the sum of all its constituent projects’ overall values (see Golabi et al. 1981, Golabi 1987)

\[
V(p, w, v) = \sum_{x^j \in p} V(x^j) = \sum_{x^j \in p} \sum_{i=1}^{n} w_i v^j_i = z(p)^T vw,
\]  

(3)

where \( z(p) = [z_1(p), \ldots, z_m(p)]^T \) is a bijection \( z : P \rightarrow \{0,1\}^m \) such that \( z_j(p) = 1 \) if \( x^j \in p \) and \( z_j(p) = 0 \) if \( x^j \not\in p \). For instance, with 8 projects, portfolio \( p = \{x^1, x^2, x^6, x^8\} \) corresponds to \( z(p) = [1,1,0,0,0,1,0,1]^T \).
2.2 Resource Constraints and Feasible Portfolios

Typically only a subset of all available projects can be funded. For instance, there may be restricted resources such as a limited budget. These constraints are modeled through a set of linear inequalities. Let \( q \) be the number of the constraints, \( A \in \mathbb{R}^{q \times m} \) and \( B \in \mathbb{R}^q \) be a constraint matrix and a constraint vector respectively. The set of feasible portfolios \( P_F \subseteq P \) is

\[
P_F := \{ p \in P \mid Az(p) \leq B \},
\]

(4)

where \( \leq \) holds componentwise.

If there is complete information on the scores and weights available, the most preferred portfolio is the feasible portfolio that maximizes the overall value (3). This optimal portfolio can be obtained as the solution of the integer linear programming problem (ILP)

\[
\max_{z(p)} \{ z(p)^T vw \mid Az(p) \leq B, z_j(p) \in \{0, 1\} \}.
\]

(5)

2.3 Incomplete Information

A key feature of RPM is that DM can provide incomplete information about criterion weights and criterion-specific scores. Instead of exactly known parameters, the analysis is based on the sets of feasible parameters that are consistent with DM’s preferences.

Incomplete weight information is modeled as a set of feasible weights \( S_w \subseteq S^0_w \), where \( S^0_w \) is given by (2). Here we make the assumption about \( S_w \) that its convex hull \( \text{conv}(S_w) \) is a polyhedron (Liesiö et al., 2007). Thus, we denote

\[
\{ w^1, \ldots, w^t \} := \text{ext} (\text{conv}(S_w))
\]

(6)

\[
W_{ext} := [w^1, \ldots, w^t] \in \mathbb{R}^{n \times t},
\]

(7)
where \( t \) is the number of extreme points of the polyhedron of feasible weights.

Incomplete score information is modeled with value intervals \([v^i_j, \bar{v}^i_j]\), which are assumed to contain the ‘true’ scores \( v^i_j \) such that \( v^i_j \in [v^i_j, \bar{v}^i_j] \) for all \( j = 1, \ldots, m \) and \( i = 1, \ldots, n \). In matrix form the set of feasible scores \( S_v \) is the set of matrices

\[
S_v = \{ v \in \mathbb{R}^{m \times n} \mid \underline{v} \leq v \leq \bar{v} \}.
\] (8)

A non-empty information set \( S \) is the Cartesian product of feasible sets of weights and scores, i.e. \( S := S_w \times S_v \). Thus \( s = (w, v) \in S \) is equivalent to \( w \in S_w \) and \( v \in S_v \). For each portfolio \( p \in P \), the selection of the information set \( S \) defines an interval of the overall portfolio value such that

\[
V(p, w, v) \in \left[ \min_{w \in S_w} V(p, w, \underline{v}), \max_{w \in S_w} V(p, w, \bar{v}) \right].
\] (9)

### 2.4 Non-dominated Portfolios

Because of incomplete information the overall values of portfolios are no longer crisp numbers. Thus, there is not just one optimal portfolio and a dominance relation is used to compare portfolios. Portfolio \( p \) is preferred to portfolio \( p' \) if \( p \) dominates \( p' \).

**Definition 1.** Let \( p, p' \in P \). Portfolio \( p \) dominates \( p' \) with regard to the information set \( S \), denoted by \( p \succ_S p' \), if

i) \( V(p, w, v) \geq V(p', w, v) \) for all \( (w, v) \in S \)

ii) \( V(p, w, v) > V(p', w, v) \) for some \( (w, v) \in S \)

In view of the information set \( S \), dominated portfolios cannot be optimal and is therefore discarded. Thus, further analyses can be focused on non-dominated portfolios, denoted by \( P_N \)

\[
P_N := \{ p \in P_F \mid p' \not\succ_S p \ \forall p' \in P_F \}
\] (10)
2.5 Core Indices

One of the defining features of RPM is the characterization of projects that should be selected or rejected. It may be possible to make definitive recommendations about individual projects although the number of non-dominated portfolios is high. Characterization is made by computing core index $CI(x^j, S)$ for every project $x^j, j = 1, \ldots, m$.

$$CI(x^j, S) = \frac{| \{ p \in P_N(S) \mid x^j \in p \} |}{| P_N(S) |}$$  \hspace{1em} (11)

where $| \cdot |$ denotes the number of portfolios in the respective set. That is, for every project core index is the share of non-dominated portfolios that include the project.

We classify projects accordingly into three groups

Core projects : $X_{C}(S) = \{ x^j \in X \mid CI(x^j, S) = 1 \}$

Borderline projects : $X_{B}(S) = \{ x^j \in X \mid 0 < CI(x^j, S) < 1 \}$

Exterior projects : $X_{E}(S) = \{ x^j \in X \mid CI(x^j, S) = 0 \}$

The idea of classifying projects with core indices is that even if additional information is given (i.e., narrower score intervals and/or additional constraints for the feasible weights) core and exterior projects retain their status. Providing narrower score intervals for core or exterior projects has no impact on the set of non-dominated portfolios. Thus additional score information does not reduce the set of non-dominated portfolios unless it relates to borderline projects.
3 Approximative Computation of Non-dominated Portfolios

Under incomplete information there is not unique optimal portfolio, therefore we use the dominance structure to reduce the number of competing portfolios. Non-dominated portfolios are important in decision support because if a portfolio is dominated, it cannot be optimal in view of the information set $S$; we can therefore discard dominated portfolios from further analysis. When the number of project proposals increases, the number of all possible project portfolios is huge. For instance in case with 20 projects we have $2^{20} = 1\,048\,576$ and with 30 projects we have $2^{30} = 1\,099\,511\,627\,776$ possible portfolios. Thus, it is natural to search non-dominated portfolios using approximative methods.

3.1 Potentially Optimal Portfolios

The first approach is to search non-dominated portfolios among portfolios that maximize the overall value (3) with some information $(w, v)$ in the information set $S$. The set of optimal portfolios with information $(w, v) \in S$ is

$$P_O(w, v) = \arg \max_{p \in P_F} V(p, w, v) \quad (12)$$

$P_O(w, v)$ is a set because some portfolios could have the same overall value, which is the maximum with information $(w, v) \in S$. We also define a set of potentially optimal portfolios $P_{PO} \subset P_F$ in the whole information set $S$ as the union of optimal portfolios

$$P_{PO} = \bigcup_{(w, v) \in S} P_O(w, v). \quad (13)$$
It should be noted that a potentially optimal portfolio is not necessarily non-dominated \((p \in P_{PO} \Rightarrow p \in P_N)\), and neither is a non-dominated portfolio always potentially optimal \((p \in P_N \Rightarrow p \in P_{PO})\). These situations are illustrated in Figures 1a and 1b where we have two criteria and weights \(w_1, w_2\) such that \(w_2 = 1 - w_1\). Figure 1a represents the situation where \(p^1, p^2\) and \(p^3\) are all potentially optimal but only \(p^3\) is non-dominated. In Figure 1b portfolios \(p^1, p^2\) and \(p^3\) are all non-dominated portfolios but only \(p^1\) and \(p^2\) are potentially optimal.

![Figure 1: (a) \(p^1, p^2, p^3 \in P_{PO}\) but only \(p^3 \in P_N\) (b) \(p^1, p^2, p^3 \in P_N\) but only \(p^1, p^2 \in P_{PO}\)](image)

Still, it is useful to seek non-dominated portfolios by means of potentially optimal portfolios because with every information \((w, v)\) there exists at least one optimal portfolio which is also non-dominated.

**Theorem 1.** \(P_N(S) \cap P_O(w, v) \neq \emptyset \forall (w, v) \in S\)

**Proof.** Let \((w_0, v_0) \in S\), \(p \in P_O(w_0, v_0)\) and assume \(p \notin P_N\). Then \(\exists p' \in P_N\) s.t. \(p' \succ p \Rightarrow V(p', w, v) \geq V(p, w, v) \forall (w, v) \in S\). Thus \(p' \in P_O(w_0, v_0)\) and so \(P_N \cap P_O \neq \emptyset\) \(\square\)
Later on, we refer to the approximative algorithm for finding non-dominated portfolios through computation of potentially optimal portfolios \( P_{PO} \) as Algorithm 1.

**Algorithm 1**

1. \( \hat{P}_{PO} \leftarrow \{\emptyset\} \)

2. Generate random feasible scores \( v^j_i \in \left[v^j_i, \overline{v}^j_i\right] \) such that
   \[ v^j_i \leftarrow v^j_i + r^j_i \left( \overline{v}^j_i - v^j_i \right), \text{ where } r^j_i \sim UNI([0, 1]). \]

3. Generate random feasible weight vector \( w \) as a convex combination of the extreme points \( w^1, \ldots, w^t \) of \( S_w \) by setting \( w = \sum_{j=1}^{t} \alpha_j w^j \), where \( \alpha_j = \frac{r^j}{\sum_{j=1}^{t} r^j} \) and \( r_j \sim EXP(1) \) (These exponentially distributed \( r_j \) result in weight vectors uniformly distributed over \( S_w \) when \( S_w = S^0_w \)).

4. Solve zero-one linear programming (ZOLP) problem
   \[ \max_{z(p)} \left\{ z(p)^Tvw \mid Az(p) \leq B, z_j \in \{0, 1\} \right\} \]
   with random generated scores and weight vector \((w, v) \in S\)

5. \( \hat{P}_{PO} \leftarrow \{p\} \cup \hat{P}_{PO}, \) where \( p \) is the optimal solution to the problem in the previous step.

6. Repeat (2-5) until enough potentially optimal portfolios have been found.

**3.2 Calculation of Non-dominated Portfolios Regarding Tchebycheff Metric**

As we can see from Figure 1b, all non-dominated portfolios cannot be found (except in some special cases) with the Algorithm 1. Another approximative
algorithm for computing non-dominated portfolios is based on the Tchebycheff metric $L_\infty$ and series of integer linear problems (Miettinen, 1999). Tchebycheff metric in $\mathbb{R}^t$ is

$$\|x\| = \max_{i=1,\ldots,t} |\lambda_i x_i|$$

$$\lambda \in \Lambda = \left\{ \lambda \in \mathbb{R}^t_+ \mid \sum_{i=1}^t \lambda_i = 1 \right\}.$$  \hspace{1cm} (14)

Let $V' \in \mathbb{R}^t$ be an ideal point which is a solution to problems

$$V'_i = \max_{p \in P_F} V (p, w^i, v^i) \quad \forall i \in \{1, \ldots, t\}.$$  \hspace{1cm} (15)

An utopian vector denoted by $V^*$ is defined so that its components are greater than ideal point $V'$'s. The selection of the utopian vector $V^*$ is arbitrary as long as each of its components is greater than the maximal portfolio value at the corresponding space extreme point.

We denote the set of feasible portfolios that minimize the distance from the utopian vector with regard to the Tchebycheff metric by $P_T$.

$$P_T (\lambda, v) = \arg \min_{p \in P_F} \max_k \lambda_k (V^*_k - V (p, w^k, v)) ,$$  \hspace{1cm} (16)

where $V^* = [V^*_1, \ldots, V^*_t]$, $\lambda \in \Lambda$ and $v \in S_v$.

The procedure of finding non-dominated portfolios with the Tchebycheff algorithm with two criteria is illustrated in Figure 2a where we minimize the Tchebycheff distance of feasible portfolios from the ideal point $V'$. In Figures 2a and 2b, axes show the portfolio overall value at the extreme points of $S_w$. Algorithm 1 will not find all non-dominated portfolios. This same situation is presented in Figure 2b where we can see that the Tchebycheff algorithm is able to find portfolio $p'$ which corresponds to portfolio $p^3$ in Figure 1b.
Figure 2: (a) Tchebycheff algorithm finds every non-dominated portfolios (black and white points) but Algorithm 1 finds only white points. (b) A situation where Algorithm 1 cannot find portfolio $p'$, but Tchebycheff algorithm can.

**Theorem 2.** Given an information set $S = S_w \times S_v$ the corresponding set of non-dominated portfolios $P_N(S)$ and an utopian vector $V^* = [V_1^*, \ldots, V_t^*]$ the following properties hold:

i. $P_N(S) \cap P_T(\lambda, v) \neq \emptyset$ for any $\lambda \in \Lambda$ and $v \in S_v$.

ii. If $p \in P_N(S)$, then exists $\lambda \in \Lambda$ and $v \in S_v$ such that $p \in P_T(\lambda, v)$.

**Proof.**

i. Set $\lambda \in \Lambda$ and $v \in S_v$. Let $p \in P_T(\lambda, v)$ and assume that $p \notin P_N(S)$. Then exists $p' \in P_N$ such that $p' \succ p$

\[ V(p', w, v) \geq V(p, w, v) \quad \forall (w, v) \in S \]

\[ \iff V_k^* - V(p', w^k, v) \leq V_k^* - V(p, w^k, v) \quad \forall k = 1, \ldots, t \]

\[ \iff \lambda_k (V_k^* - V(p', w^k, v)) \leq \lambda_k (V_k^* - V(p, w^k, v)) \quad \forall k = 1, \ldots, t \]

\[ \iff \max_k \lambda_k (V_k^* - V(p', w^k, v)) \leq \max_k \lambda_k (V_k^* - V(p, w^k, v)) \]

Thus $p' \in P_T(\lambda, v)$ which implies $P_N(S) \cap P_T(\lambda, v) \neq \emptyset$.

ii. Contrary to the claim, assume that there exists $p \in P_N(S)$ such that $p \notin P_T(\lambda, v) \quad \forall \lambda \in \Lambda$ and $v \in S_v$. Set $\hat{v}$ such that $\hat{v} = \begin{cases} \overline{v^i}, & \text{when } x^i \in p \\ v^i, & \text{when } x^i \notin p \end{cases}$ and $\hat{\lambda}_k = \frac{\beta}{V_k^* - V(p, w^k, \hat{v})}$, where $\beta$ is a scaling factor such that $\sum_{k=1}^t \hat{\lambda}_k = 1$. 


Because $p \notin P_T\left(\hat{\lambda}, \hat{v}\right)$, there exists $p' \in P_F$ such that

$$\max_k \hat{\lambda}_k (V^*_k - V(p', w^k, \hat{v})) < \max_k \hat{\lambda}_k (V^*_k - V(p, w^k, \hat{v}))$$

$$= \max_k \beta (V^*_k - V(p, w^k, \hat{v})) = \beta.$$

Thus $\hat{\lambda}_k (V^*_k - V(p', w^k, \hat{v})) < \beta$ for all $k = 1, \ldots, t$, which implies

$$\frac{\beta}{V^*_k - V(p, w^k, \hat{v})} (V^*_k - V(p', w^k, \hat{v})) < \beta \quad \forall \ k = 1, \ldots, t$$

$$V(p', w^k, \hat{v}) > V(p, w^k, \hat{v}) \quad \forall \ k = 1, \ldots, t$$

$$V(p' \setminus p, w^k, \hat{v}) > V(p \setminus p', w^k, \hat{v}) \quad \forall \ k = 1, \ldots, t$$

Thus, $P_F \ni p' \succ p \in P_N$ which is a contradiction.

Later on, we refer to the approximative algorithm for finding non-dominated portfolios using the Tchebycheff metric as Algorithm 2.

**Algorithm 2**

1. Solve utopian vector $V^*$ or at least find an upper bound for it

2. $\hat{P}_T \leftarrow \{\emptyset\}$

3. Choose randomly $\lambda$ such that $\lambda_i \sim EXP(1)$

4. Choose randomly a subset $X^u \subseteq X$ and set $v^j = \overline{v}^j$ if $x^j \in X^u$ and $v^j = \overline{v}^j$ if $x^j \notin X^u$
5. Solve the integer linear programming (ILP) problem

\[ \min_{z(p)} \Delta + (\lambda_i w^i v)z(p) \geq \lambda_i V^*_i \quad \forall i = 1, \ldots, t. \]
\[ Az(p) \leq B \]
\[ z_j(p) \in \{0, 1\} \quad \forall j = 1, \ldots, m. \]

\[ \hat{P}_T \leftarrow \{p\} \cup \hat{P}_T, \text{ where } p \text{ is the optimal solution to the problem.} \]

6. Repeat (3-5) until enough non-dominated portfolios have been found.

### 3.3 Problems with Dominated Solutions

As mentioned in section 3.1, a potentially optimal portfolio is not necessarily non-dominated. This is illustrated in Figures 3a and 3b where there are two criteria and three portfolios \( p_1, p_2 \) and \( p_3 \). Overall values of portfolios are in the Figure 3a and in Figure 3b axes correspond to portfolio overall value in the extreme points of \( S_w \). In Figure 3b black points represents dominated portfolios. Portfolio \( p_1 \) is the only non-dominated portfolio and it dominates portfolios \( p_2 \) and \( p_3 \) (i.e. \( p_1 \geq p_2 \) and \( p_1 \geq p_3 \)). Still portfolio \( p_2 \) maximizes overall value \( V(p, w, v) \) when weights are chosen such that \( w = w^2 \) and \( p_3 \) maximizes overall value \( V(p, w, v) \) when \( w = w^1 \). Thus, all three portfolios \( p_1, p_2 \) and \( p_3 \) could be found by Algorithm 1. Because of linearity of the overall values this situation only arise when weights are chosen in the extreme points.

Although Theorem 2 states that for every non-dominated portfolio \( p \), there exists \( \lambda \) and \( v \) such that \( p \in P_T(\lambda, v) \), it does not imply that \( \lambda \) and \( v \) are unique. Thus, there are situations when Algorithm 2 may find dominated portfolios as illustrated in Figure 4 where black points represents two alter-
native optima, which minimize the distance from ideal point when \( \lambda_1 = 0 \), but only one is non-dominated.

Figure 3: (a) \( p^1, p^2 \) and \( p^3 \) could be found by Algorithm 1 but only \( p^1 \) is non-dominated. (b) Amount of profits from criterion 1 and 2 of portfolios \( p^1, p^2 \) and \( p^3 \) where black points are dominated portfolios.

Figure 4: Black points are alternative optima for Algorithm 2. However only one of them is non-dominated.

Although both algorithms can give dominated portfolios it is possible to discard them with a pairwise dominance checks between found portfolios.
3.4 Behaviour of Core Indices During Computations

If we consider core indices after the first approximation round, we have only core and exterior projects because we have just one portfolio and every project either is included in or excluded from that portfolio. When the number of approximation rounds increases we (generally) have also borderline projects that are included in some but not in all portfolios. However we might still have core and exterior projects but not as many as after the first round. Thus after additional approximation rounds the number of core and exterior projects decreases and simultaneously the number of borderline projects increases.

4 Computational Experiments

4.1 Experimental Design

Computational experiments on the algorithms were carried out using 2 different problem types

- Problem type 1:
  50 projects, 3 criteria and 1 resource constraints.

- Problem type 2:
  150 projects, 6 criteria and 3 resource constraints.

8 different problem instances were generated from both types and problems are solved with both approximative algorithms discussed earlier. Thus, total number of 16 multiple criteria project portfolio optimization problems were generated and solved.
Each instance included score matrices $\mathbf{v}, \mathbf{\overline{v}}$, constraint matrix $A$ and vector $B$. No weight information was given, i.e. $S_w = S^0_w$. Upper bounds for scores ($\mathbf{\overline{v}}$) were uniformly distributed integers in the interval $[0, 100]$ and lower bounds were set $v_i^j = (1 - r_i^j) \mathbf{\overline{v}}_i^j$, where $r_i^j \sim UNI([0, 0.1])$. Constraint matrix $A$ coefficients $a_i^j$ were uniformly distributed integers on the interval $[0, 1000]$ for all $l = 1, \ldots, q$, where $q$ is the number of the constraints and $b_l$ was set to 50\% of the sum $\sum_{j=1}^{m} a_i^j$.

Problems were entered as parameters to the Java\textsuperscript{TM} program and they were solved with both algorithms using lp\_solve software with AMD Duron 1.6Ghz personal computer as follows.

1. Initiate problem.
   (Approximation round begins)

2. Solve problem with Algorithm 1.


4. Save results of both algorithms.

5. Calculate key ratios e.g. core indices, number of non-dominated portfolios and number of common non-dominated portfolios found by algorithms as a function of approximation rounds.

6. Repeat (2-4) 1000 times.
4.2 Results

Both algorithms search non-dominated portfolios randomly and it is obvious that the probability of finding a particular portfolio in the first round is the same as in the last round. Thus, it is valuable to compare algorithms at the final stage of computations. Sometimes it is also interesting to compare algorithms time-wise in view of approximative computations but algorithms turned out not to be significant different in computation time.

The number of portfolios found by both algorithms in the smaller and larger problem instances are presented in Figure 5 and 6, respectively, as a function of approximation rounds (=solved linear programming problems). Also number of common portfolios found by algorithms are presented in the figures. As we can see from the figures, the Algorithm 1 finds more portfolios at the first approximation rounds than Algorithm 2. However the speed of finding new portfolios by Algorithm 1 seems to decrease as a function of approximation rounds more rapidly than in the case of Algorithm 2. Thus, after enough approximations rounds the number of portfolios found by Algorithm 2 becomes greater than Algorithm 1. The number of common portfolios found seems to increase almost linearly and approaches the number of portfolios found by Algorithm 1. The number of core indices during the computations as a function of approximation rounds are in the Figures 7 and 8. Core indices in the Figures 7 and 8 are only printed of the first four problem instances of 50 and 150 projects problems.

The number of portfolios found by algorithms and core indices after 1000 approximation rounds are listed in Table 1 and Table 2. Clearly the number of portfolios found by algorithm 2 is larger than number of portfolios found by Algorithm 1 in all the problem instances. Sets of core, borderline and exterior projects are quite similar with both algorithms at the final stage.
The average computation times for solving one zero-one linear programming problem in Algorithm 1 were 37.05 milliseconds in case of 50 projects and 1048.83 milliseconds in case of 150 projects. With Algorithm 2 the average computation times for solving one integer linear programming problem were 54.06 milliseconds in case of 50 projects and 1553.67 milliseconds in case of 150 projects. Computation times are very dependent on linear programming solver and computer performance.
Figure 5: Eight problem instances with 50 projects, 3 criteria and 1 constraint.
Figure 6: Eight problem instances with 150 projects, 6 criteria and 3 constraints.
Figure 7: Number of core, borderline and exterior projects of the problem with 50 projects, 3 criteria and 1 constraint.
Figure 8: Number of core, borderline and exterior projects of the problem with 150 projects, 6 criteria and 3 constraints.
Table 1: Results of the problem with 50 projects, 3 criteria and 1 constraint.

<table>
<thead>
<tr>
<th>Instance</th>
<th>Non-dominated portfolios</th>
<th>Algorithm 1</th>
<th>Algorithm 2</th>
<th>Common</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Instance 1</td>
<td>Instance 2</td>
<td>Instance 3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>89</td>
<td>141</td>
<td>44</td>
</tr>
<tr>
<td>X_C</td>
<td></td>
<td>9</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>X_B</td>
<td></td>
<td>38</td>
<td>39</td>
<td></td>
</tr>
<tr>
<td>X_E</td>
<td></td>
<td>3</td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>


Table 2: Results of the problem with 150 projects, 6 criteria and 3 constraint.

<table>
<thead>
<tr>
<th>Instance</th>
<th>Non-dominated portfolios</th>
<th>Algorithm 1</th>
<th>Algorithm 2</th>
<th>Common</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>179</td>
<td>294</td>
<td>49</td>
</tr>
<tr>
<td></td>
<td>(</td>
<td>X_C</td>
<td>)</td>
<td>57</td>
</tr>
<tr>
<td></td>
<td>(</td>
<td>X_B</td>
<td>)</td>
<td>78</td>
</tr>
<tr>
<td></td>
<td>(</td>
<td>X_E</td>
<td>)</td>
<td>15</td>
</tr>
<tr>
<td>Instance 2</td>
<td>Non-dominated portfolios</td>
<td>252</td>
<td>285</td>
<td>48</td>
</tr>
<tr>
<td></td>
<td>(</td>
<td>X_C</td>
<td>)</td>
<td>48</td>
</tr>
<tr>
<td></td>
<td>(</td>
<td>X_B</td>
<td>)</td>
<td>89</td>
</tr>
<tr>
<td></td>
<td>(</td>
<td>X_E</td>
<td>)</td>
<td>13</td>
</tr>
<tr>
<td>Instance 3</td>
<td>Non-dominated portfolios</td>
<td>219</td>
<td>285</td>
<td>50</td>
</tr>
<tr>
<td></td>
<td>(</td>
<td>X_C</td>
<td>)</td>
<td>48</td>
</tr>
<tr>
<td></td>
<td>(</td>
<td>X_B</td>
<td>)</td>
<td>86</td>
</tr>
<tr>
<td></td>
<td>(</td>
<td>X_E</td>
<td>)</td>
<td>16</td>
</tr>
<tr>
<td>Instance 4</td>
<td>Non-dominated portfolios</td>
<td>196</td>
<td>272</td>
<td>47</td>
</tr>
<tr>
<td></td>
<td>(</td>
<td>X_C</td>
<td>)</td>
<td>55</td>
</tr>
<tr>
<td></td>
<td>(</td>
<td>X_B</td>
<td>)</td>
<td>79</td>
</tr>
<tr>
<td></td>
<td>(</td>
<td>X_E</td>
<td>)</td>
<td>16</td>
</tr>
<tr>
<td>Instance 5</td>
<td>Non-dominated portfolios</td>
<td>213</td>
<td>303</td>
<td>52</td>
</tr>
<tr>
<td></td>
<td>(</td>
<td>X_C</td>
<td>)</td>
<td>53</td>
</tr>
<tr>
<td></td>
<td>(</td>
<td>X_B</td>
<td>)</td>
<td>86</td>
</tr>
<tr>
<td></td>
<td>(</td>
<td>X_E</td>
<td>)</td>
<td>11</td>
</tr>
<tr>
<td>Instance 6</td>
<td>Non-dominated portfolios</td>
<td>205</td>
<td>296</td>
<td>40</td>
</tr>
<tr>
<td></td>
<td>(</td>
<td>X_C</td>
<td>)</td>
<td>53</td>
</tr>
<tr>
<td></td>
<td>(</td>
<td>X_B</td>
<td>)</td>
<td>82</td>
</tr>
<tr>
<td></td>
<td>(</td>
<td>X_E</td>
<td>)</td>
<td>15</td>
</tr>
<tr>
<td>Instance 7</td>
<td>Non-dominated portfolios</td>
<td>216</td>
<td>298</td>
<td>49</td>
</tr>
<tr>
<td></td>
<td>(</td>
<td>X_C</td>
<td>)</td>
<td>54</td>
</tr>
<tr>
<td></td>
<td>(</td>
<td>X_B</td>
<td>)</td>
<td>81</td>
</tr>
<tr>
<td></td>
<td>(</td>
<td>X_E</td>
<td>)</td>
<td>15</td>
</tr>
<tr>
<td>Instance 8</td>
<td>Non-dominated portfolios</td>
<td>237</td>
<td>315</td>
<td>47</td>
</tr>
<tr>
<td></td>
<td>(</td>
<td>X_C</td>
<td>)</td>
<td>43</td>
</tr>
<tr>
<td></td>
<td>(</td>
<td>X_B</td>
<td>)</td>
<td>91</td>
</tr>
<tr>
<td></td>
<td>(</td>
<td>X_E</td>
<td>)</td>
<td>16</td>
</tr>
</tbody>
</table>
5 Software Demonstration

These algorithms were implemented as a part of the decision support software RPM-Decisions. Graphical User Interface (GUI) and computational framework are implemented with Java\textsuperscript{TM} and Mixed Integer Linear Programming solver lp\_solve. The screen shots are from the RPM-Decisions at the problem input and computation phases. At the phase represented in Figure 9 the criteria-specific scores and feasibility constraints are entered to the program. Also the names of projects, criteria and constraints are entered at this phase.

![Figure 9: Screen shot of RPM-Decisions software's input sheet.](image)

During the computations the number of core, borderline and exterior projects are plotted to the graph so it is easy to follow the progress of computation. The user can also choose whether we use both algorithms or just a one. The computation phase is illustrated in Figure 10. We can choose
whether to use both algorithms or just one for computations and we can also change the score and weight distributions of algorithms during computations. The distribution of core indices is drawn on the chart in function of approximation rounds and the drawing frequency can be changed with a slide bar on left.

Figure 10: Screen shot of RPM-Decisions software’s computation phase.
6 Conclusions

In this paper, two different algorithms were compared as a function of solved integer linear problems (=approximation rounds) and the main goal was to discover the advantages of the approximation algorithms when solving large problems that includes many project proposals. Approximation methods turned out to be useful for finding non-dominated project portfolios in the set of projects proposals. It should be remembered that the represented algorithms are not exact algorithms.

The algorithm that searches portfolios using Tchebycheff metric finds more non-dominated portfolios than the algorithm that finds potentially optimal solutions. However, core indices based on algorithms are quite similar and thus both algorithms are good choices for searching approximative solutions and non-dominated portfolios.

With both algorithms the computations were quite fast even in the case of 150 projects and Algorithm 1 was slightly faster than Algorithm 2. The most time consuming in the algorithms was solving the mixed integer linear programming problem. Computation time could be reduced with optimization of linear programming solvers. Also the use of Java™ and lp.solve solver was an easy and practical way to make user friendly interface but it might not be the best solution regarding computational time.

It would be interesting to focus future research on approximative or exact algorithms for solving only projects core indices. Also it is appropriate to develop faster exact algorithm for solving all non-dominated project portfolios when there are hundreds of project proposals.
References


