Acquisition Prioritization: A Multicriteria Approach Based on a Case Study

Horst W. Hamacher*, Stefan Ruzika*, Akin Tanatmis*

November 6, 2006

Fachbereich Mathematik, Technische Universität Kaiserslautern
Postfach 3049, Kaiserslautern, Germany
{hamacher,ruzika,tanatmis}@mathematik.uni-kl.de
corresponding author: Stefan Ruzika

Abstract. Selection of new projects is one of the major decision making activities in any company. Given a set of potential projects to invest, a subset which matches the company's strategy and internal resources best has to be selected. In this paper, we propose a multicriteria model for portfolio selection of projects, where we take into consideration that each of the potential projects has several - usually conflicting - values. We propose a method for computing a small set of efficient (Pareto-optimal) project portfolios which serves as a representation of all efficient portfolios. This method is realized in the software tool ProSel (project selection) which additionally assists the decision maker in choosing the final project portfolio. Our approach was tested in a case study for KEIPER, an international company which develops and manufactures vehicle seating systems.

Key words: Project prioritization - Multicriteria optimization - Representation - Decision support - Project selection

*The research of all authors was partially supported by Deutsche Forschungsgemeinschaft (DFG) grant HA 1737/7 "Algorithmik großer und komplexer Netzwerke", by New Zealand's Julius von Haast award, and by the Rheinland-Pfalz cluster of excellence "Dependable adaptive systems and mathematical modeling".

1 Introduction

Global companies usually have to select from a large set of new projects according to various evaluation criteria. This problem is widely known as project portfolio selection (PPS). Financial objectives like costs, profit, return on investment, etc. are part of this evaluation. Another group of criteria includes among others market shares, customer satisfaction, or developing strategic partnerships. Current trends like the international expansion of business activities, increasing number and variations of products, global competition, cost consciousness etc. have made the project prioritization and resource allocation more difficult and more crucial than ever.

Related problems appear in the literature among others under the terms "R&D project selection", "Project portfolio selection", "Project prioritization", or "Portfolio management for new products". The review in [3] summarizes the recent approaches for managing the portfolio of new products in 7 categories namely financial or economic models, scoring models and checklists, probabilistic financial models, behavioral approaches, mathematical optimization procedures, decision support systems, and mapping approaches. Despite theoretical advances in PPS, new approaches are only slowly deployed in practice (cf. [3], [12]) According to [12] many organizations utilize a variant of the following five steps for project prioritization and resource allocation.

1. Listing of potential projects.
2. Calculation of the benefit of each of the potential projects.
3. Ranking of the potential projects from the most beneficial to the least beneficial.
4. Estimation of costs to the potential project.
5. Choosing the most beneficial projects top-down until the total cost exceeds the budget.

The major drawback in this approach is that a ranking of the potential projects is in general not possible, since various conflicting objectives need to be considered. We therefore propose a multicriteria model and split the "benefit" into two apparently natural components: financial (e.g. sales, assets, etc.) and non-financial benefits, also called political benefits in this article (e.g. development of new markets or job creation). Political and financial benefits are typically non-commensurable and in conflict, but nevertheless they are to be maximized in the sense of Pareto optimality subject to budget restrictions. We thus are interested in project portfolios, for which no other portfolio has a better evaluation with respect to both political and financial objectives. The specific model we are using for tackling portfolio selection is the bicriteria binary multidimensional knapsack problem. We refer to [7] and [4] or [11] for in-depth treatments on the knapsack and the general multicriteria optimization problem, respectively.

Multicriteria optimization is used in many applications as the solution approach (see, e.g., [6], [8], or [15]). The primary goal in multicriteria optimization is to seek efficient (Pareto-optimal) solutions and/or nondominated points. Since it is usually not advisable (or even possible) to compute all of them we restrict ourselves to find a representative set of these solutions which serves as a preselection of alternatives for the decision maker. For a detailed discussion of various methods for computing representations we refer to [13]. Among the representing points, the decision maker then chooses a finally preferred solution.

In order to measure the quality of the representation we apply the box algorithm [5]. We implement this algorithm as the core of our decision support system (DSS) ProSel for project selection. ProSel combines the tasks of collecting and processing data, computing a high quality representation, and presenting alternatives with appropriate tools.

Our bicriteria approach and the DSS ProSel were originally designed for the acquisition prioritization in KEIPER. Acquisition prioritization is the term used in KEIPER to describe the activities of selecting a portfolio of new projects. KEIPER is a financially and legally independent company of the internationally active Keiper Recaro Group which includes also RECARO and RECARO Aircraft Seating. Business activities of KEIPER are concentrated on the development and manufacturing of metal components and structures for automobile seats as well as the development of complete seats. Following the globalization of the automobile industry KEIPER works with well-known system suppliers and automobile manufacturers all over the world. Currently the company operates at 23 (13 production facilities) locations in 11 countries.

The rest of this paper is organized as follows. In Section 2 the acquisition prioritization process in KEIPER is briefly explained using company specific terminology. Moreover the mathematical formulation is introduced. Section 3 includes the solution methodology with the used notation and a summary of the box algorithm. Section 4 contains information about the implementation and the numerical experiments. Conclusions and contribution of ProSel to the company are discussed in Section 5.

2 Acquisition prioritization: Problem description and mathematical modeling

Acquisition prioritization consists in selecting a best mix from a list of new projects in accordance with the company's strategy and internal resources. New projects are evaluated with respect to costs and benefits. The total cost of a project portfolio is subject to some budget. The benefit of new projects is measured and evaluated by several different criteria. From the set of new projects, a subset has to be found which complies with the budget constraints and which is most beneficial for the company. A detailed information of acquisition prioritization in KEIPER can be found in [9] and [14].

The time period between the acquisition of a new project and the delivery of the last product to the customer is referred to as the *project running time*. Each new project has a specific running time. Projects are associated with one or more KEIPER products (a complete seat structure or a seat component) and generate acquisition, investment, and development costs in various completion stages of their running times. Acquisition costs comprise among others the costs of submitting an order or building a concept. Tooling costs and costs resulting from the procurement of test equipment and raw materials are subsumed under the category investment costs. Development costs include costs arising from activities like construction, prototyping, testing, and developing. Production costs occur during the series production and involved in the model in the process of evaluating the financial value of a project. Acquisition costs, investment costs, and development costs are subject to acquisition budget, investment budget, and development budget, respectively. Any selected portfolio of projects has to comply with the budget constraints that the overall costs must not exceed the available budgets.

In this paper we emphasize the importance of the second issue that should be addressed in the process of acquisition prioritization - the fulfillment of the targets which are derived from the company's strategical goals. This issue is realized as follows. For each criterion the decision maker assigns integer values to the new projects. This value reflects the relative importance of the project compared to others under this particular criterion. Then, we group the evaluation criteria under financial and political criteria. Financial criteria are the indicators of the rentability of a project which can be expressed in monetary terms. Political criteria like market share, customer satisfaction, etc. are usually more difficult to measure. Obviously, the process of generating financial and political performance measures is company specific and due to confidentiality reasons we cannot disclose the specific approach used in our KEIPER case study.

In order to generate an appropriate operations research model we denote with $P = \{1, \ldots, n\}$ the set of new projects and with T the *planning horizon*. The Planning horizon T is the total number of the time periods that the decision maker takes into consideration when evaluating the new projects.

The binary decision variables $x_i \in \{0, 1\}$, $i = 1, \ldots, n$, are used to model if project i is chosen ($x_i = 1$) or not ($x_i = 0$). Then the acquisition prioritization problem can be formulated as the following discrete bicriteria optimization problem.

$$\max \sum_{i \in P} f_i x_i \tag{2.1}$$

$$\max \sum_{i \in P} p_i x_i \tag{2.2}$$

$$\text{s.t.} \quad \sum_{i \in P} AC_{it} x_i \leq AB_t \qquad \forall t = 1, \ldots, T \tag{2.3}$$

$$\sum_{i \in P} IC_{it} x_i \leq IB_t \qquad \forall t = 1, \ldots, T \tag{2.4}$$

$$\sum_{i \in P} DC_{it} x_i \leq DB_t \qquad \forall t = 1, \ldots, T \tag{2.5}$$

$$x_i = x_j \qquad \forall i, j \in D_l \quad \forall l = 1, \ldots, k \tag{2.6}$$

$$x_i \in \{0, 1\} \qquad \forall i = 1, \ldots, n \tag{2.7}$$

Here, AC_{it}, IC_{it}, and DC_{it} denote the acquisition, investment, and development costs of project i in period t. The corresponding budgets are denoted by AB_t, IB_t, and DB_t, respectively. Constraints 2.3, 2.4, and 2.5 thus express the budget limitations. Several new projects may belong to a particular subset $D_l \subseteq \{1, \ldots, n\}$ where $l = 1, \ldots, k$. In this case either all or none of the projects in this subset are chosen. Constraints 2.6 realize these dependencies. Two linear objective functions corresponding to the financial and political benefits, respectively, are to be optimized. Each of the two objective functions is a weighted sum aggregation of the criteria used by the company to evaluate the financial and political benefit of new projects. That means $f_i = \sum_{j \in I_f} \lambda_j f_{ij}$ and $p_i = \sum_{j \in I_p} \lambda_j p_{ij}$, where I_f and I_p denote the index sets of financial and political criteria, respectively, and f_{ij} and p_{ij} denote the values assigned by the company to project i for financial and political criteria j, respectively.

3 Solution methodology

The operations research formulation of acquisition prioritization is a special case of the more general discrete bicriteria optimization problem

$$\begin{array}{ll} \max & f(x) = \begin{pmatrix} f_1(x) \\ f_2(x) \end{pmatrix} \\ \text{s.t.} & \quad x \in X \end{array} \tag{$DBOP$}$$

where X is the discrete *feasible set* and $f : X \to \mathbb{Z}^2$ is a vector-valued objective function. We denote by $Y := f(X)$ the *set of attainable outcomes*.

The meaning of maximizing the vector-valued objective function needs to be specified since there is no canonical ordering defined in $\mathbb{Z}^2$. We use the optimality concept based on the componentwise order, also known as Pareto optimality, with the following notation. Let $y^1, y^2 \in \mathbb{Z}^2$. Then

$$\begin{aligned} y^1 \geq y^2 &:\Leftrightarrow y_i^1 \geq y_i^2 \quad \forall i = 1, 2 \quad \textit{and} \quad y^1 \neq y^2 \\ y^1 > y^2 &:\Leftrightarrow y_i^1 > y_i^2 \quad \forall i = 1, 2 \end{aligned}$$

A decision vector $x^1 \in X$ is *(weakly) efficient* if there does not exist another decision vector x^2 such that $f(x^2) \geq f(x^1)$ $(f(x^2) > f(x^1))$. An objective vector $y = f(x) \in \mathbb{Z}^2$ is *(weakly) nondominated* if x is (weakly) efficient. The *efficient set* X_E and the *weakly efficient set* X_{wE} are defined as

$$\begin{aligned} X_E &:= \{x^1 \in X : \text{there exists no } x^2 \in X : f(x^2) \geq f(x^1)\} \\ X_{wE} &:= \{x^1 \in X : \text{there exists no } x^2 \in X : f(x^2) > f(x^1)\}. \end{aligned}$$

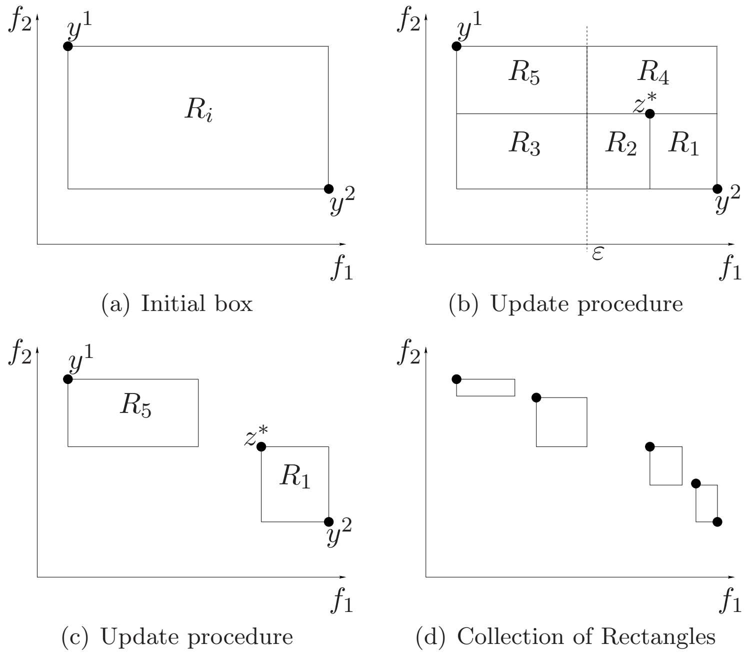
The images $Y_N := f(X_E)$ and $Y_{wN} := f(X_{wE})$ of these sets are the *nondominated set* and the *weakly nondominated set*, respectively. Let $y_q^I := \max\{f_q(x) : x \in X\}$ and let $y_q^N := \min\{f_q(x) : x \in X_E\}$ for $q = 1, 2$. The *ideal point* is $y^I := (y_1^I, y_2^I)^T$ and the *nadir point* is $y^N := (y_1^N, y_2^N)^T$.

In our model for project portfolio selection each efficient solution corresponds to an optimal project portfolio in the Pareto sense. This means given an efficient project portfolio, there does not exist any other portfolio which has a better evaluation with respect to political and financial objectives. More precisely, when comparing two efficient portfolios of new projects, the Pareto optimality concept implies that the project portfolio being better with respect to the financial criterion must be worse in the political criterion. This tradeoff characterizes the nondominated set.

The decision maker is not interested in any dominated project portfolio, since there exists another portfolio which is better with respect to both groups of criteria. On the other hand, the computation of all nondominated points is, in general, unrealistic for several reasons. The number of nondominated points is usually very large. Hence its computation is not at all possible, or - even if it is available - it is not helpful for the decision maker due to an abundance of information which cannot be put to use in a reasonable way.

We therefore apply in our approach the *box method* introduced in [5]. This method avoids information overkill and can be controlled in such a way that it computes the complete set of nondominated points (if this set is small) or selects a *representative system* of the nondominated set with certain quality guarantees. This representation is understood as a quality-proved substitute of the complete nondominated set. Loosely speaking these quality features guarantee that

- each alternative project portfolio is optimal and cannot be improved in both criteria,
- portfolios yielding similar financial and political results are not generated since they do not correspond to reasonable alternatives, and
- no reasonably different alternative project portfolio is missed in the representation.



(a) Initial box (b) Update procedure

(c) Update procedure (d) Collection of Rectangles

Figure 1: Stages of the box algorithm

In order to apply the box algorithm the following two mathematical programs are solved:

$$\text{lex } \max_{x \in X} \begin{pmatrix} f_1(x) \\ f_2(x) \end{pmatrix} \quad \text{and} \quad \text{lex } \max_{x \in X} \begin{pmatrix} f_2(x) \\ f_1(x) \end{pmatrix}$$

In the first program we determine the point with maximal f_2-value among all points which are optimal with respect to f_1. In the second program the roles of f_1 and f_2 are changed. These two lexicographic optimal solutions determine the ideal and the nadir point and a rectangle - the *starting box*, denoted by $R(y^1, y^2)$ with $y^1, y^2 \in \mathbb{Z}^2$ (see Figure 1(a)). The starting box obviously contains the complete nondominated set.

During the algorithm, rectangular parts of this starting box are iteratively discarded since they do not contain any nondominated point. These rectangles are defined by points which are found by solving the following lexicographic variant P_ε of the well-known ε-constraint method (see [2] and [5])

$$\begin{aligned} \text{lex max} \quad & \begin{pmatrix} f_2(x) \\ f_1(x) \end{pmatrix} \\ \text{s.t.} \quad & f_1(x) \geq \varepsilon \\ & x \in X \end{aligned} \tag{P_ε}$$

with adequate values for ε. Optimal solutions of P_ε are efficient. In each stage of the algorithm, a collection of rectangles or boxes containing the nondominated set is maintained. The upper left corner point of each of the rectangles is a representing point and the collection of these points builds the representing system. The box algorithm terminates if a given accuracy $\Delta > 0$ is achieved. We measure this accuracy by the area of the largest of the remaining boxes, i.e., if $a(R(y^1, y^2)) := (y_1^2 - y_1^1) \cdot (y_2^1 - y_2^2) \leq \Delta$ for all rectangles $R(y^1, y^2)$. Alternatively, the algorithm might stop after a given number of representing points has been found.

In the following, the refinement of a rectangle is described in more detail. Consider a box with area $a(R(y^1, y^2)) > \Delta$. Following the general idea of the box algorithm outlined above the representation has to be locally updated in $R(y^1, y^2)$. Consider P_ε with $\varepsilon := \left\lceil \frac{y_1^1 + y_1^2}{2} \right\rceil$. Let $x^* \in X$ be optimal for P_ε and let $z^* := f(x^*) := (f_1(x^*), f_2(x^*))$. Using the point z^* and ε, we divide $R(y^1, y^2)$ into five rectangles as visualized in Figure 1(b).

The following results can be established:

- The point z^* is nondominated.
- R_2, R_3, and R_4 can be eliminated, since $(R_2 \cup R_3) \cap Y_N \subseteq \{z^*\}$, and $R_4 \cap Y_N \subseteq \{z^*\}$.
- R_1 and R_5 contain all nondominated points in $R(y^1, y^2)$, i.e., $Y_N \cap R(y^1, y^2) \subseteq R_1 \cup R_5$ (see Figure 1(c)).
- The update has locally improved the representation by a factor of 2, i.e., $a(R_1) + a(R_5) \leq \frac{1}{2} a(R(y^1, y^2))$.

In each iteration of the algorithm, the box having the largest area is refined as described. Thus, the representation is updated where it is needed most. The algorithm is depicted in Figure 1. The rectangles shown in Figure 1(d) are used to measure and control the quality of the representation. For a more detailed exposition of the box algorithm and related theoretical results, we refer to [5].

4 ProSel: A software tool for project selection

For the implementation in our case study at KEIPER we developed an integrated decision support system called ProSel which allows

- collection and preparation of internal and external data,
- realization of the box algorithm, and
- presentation of optimal project portfolios and decision support.

The logical structure of ProSel is explained in more detail in the following and illustrated in Figure 2.

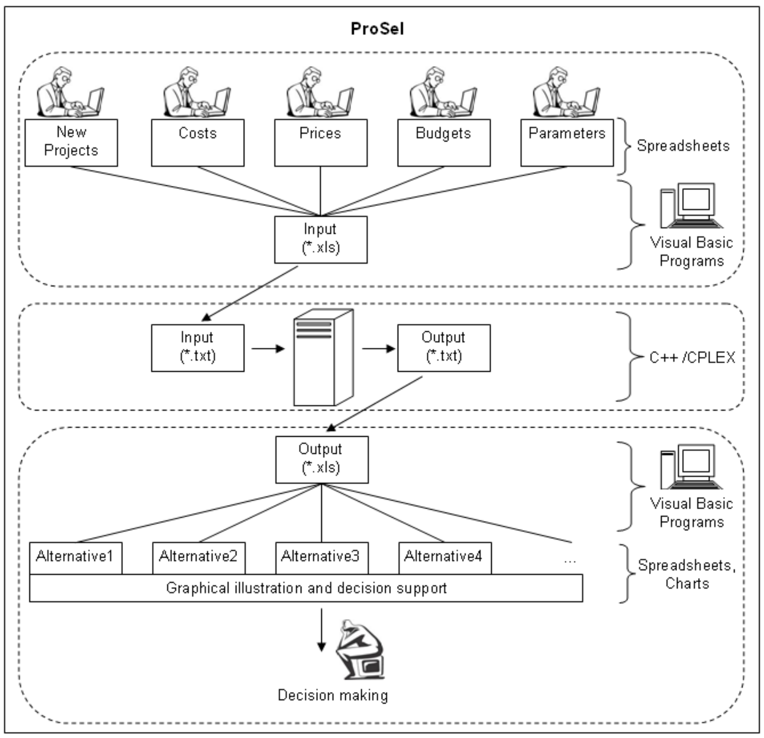


Figure 2: ProSel process flow in the case study KEIPER

The input data for ProSel - external and internal information as well as the parameters specified by the decision maker - are stored in various spreadsheets.

Visual Basic programs facilitate the entry of data and automate the tasks of combining information, performing initial computations, and generating the input file for the box algorithm.

The core of ProSel is the realization of the box algorithm. It is implemented as a C++ program. The main algorithm is written in C++ while all occurring subproblems are solved using ILOG CPLEX [1]. An all-purpose solver like CPLEX is preferred to a specialized solution algorithm for reasons of robustness and flexibility. The C++ program reads the data of the operations research model from the input file. After calculating the lexicographic maxima, the program evaluates the area of the initial box. The update procedure is executed by solving P_ε problems until the stopping criterion is fulfilled and the representation with the correct quality is found. Objective function values and solutions are written in an output file.

Using the output file, the objective function values of the representing points are scaled and stored in a database featured with different search and viewing options. Each representing point is associated with an efficient project portfolio and thus corresponds to an acquisition alternative. Different alternatives can be compared with each other with respect to financial and political values in a tradeoff chart. In Figure 3 an example of a representation is illustrated in such a tradeoff chart. For a closer investigation several tools as shown in Figure 4 provide comprehensive information about specific alternatives for the decision maker. The operations of transferring data, searching the representing set, selecting different alternatives, and creating charts are realized by Visual Basic programs. Spreadsheets are used for data storage.

ProSel has been tested with KEIPER data and performed well. For obvious confidentiality reasons we cannot report on these results, but describe instead in the following on test data. These data were designed based on our practical experience to provide worst-case benchmarks with respect to computation times resulting from real-world. We considered the average number of representing points and the average CPU times as measures of performance. We used the model formulation given in Section 2 with the assumption that there exists only independent projects. It should be pointed out that inclusion of dependent projects reduces the number of variables and facilitates the problem. Consequently, the bicriteria binary multidimensional knapsack problem we have tested can be viewed as a worst-case scenario with respect to dependency. Besides dropping dependency constraints, we employed additional means to generate more complex test data

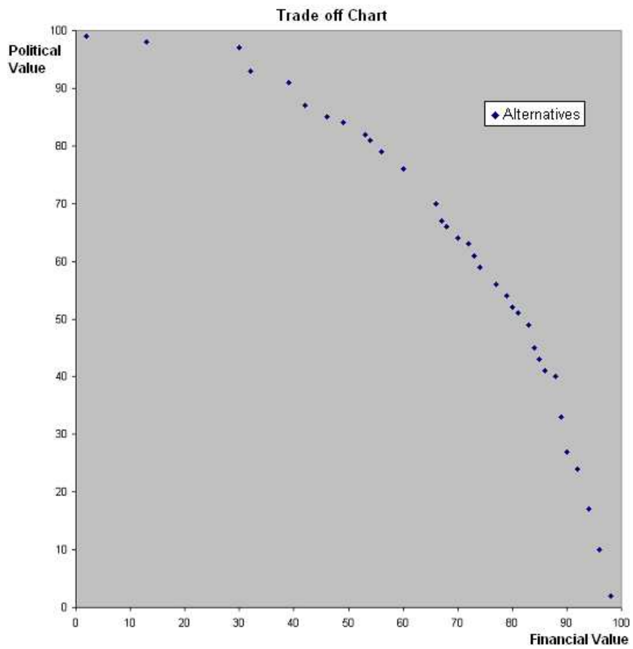


Figure 3: Tradeoff chart

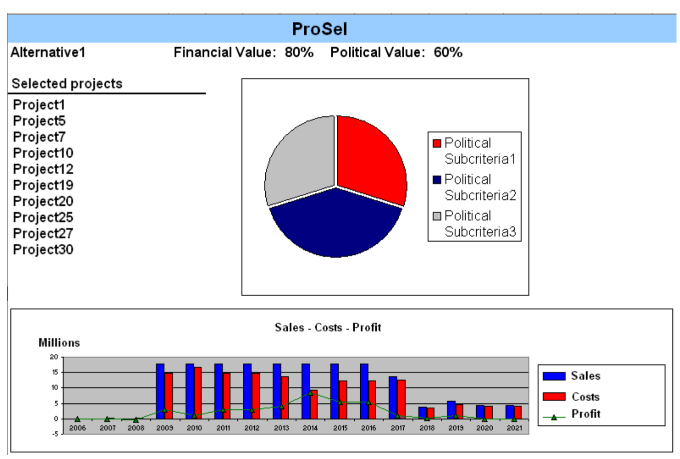


Figure 4: Presentation of an alternative

than the expected real-world data. The generated cost matrix is dense since it has all positive entries. In reality, this matrix contains many zeros since, for instance, acquisition costs do only occur at the beginning of the projects and not in later phases such as production. The visual basic function `rnd()` which generates uniformly distributed random numbers in the range $[0, 100]$ is used for generating objective function and constraint coefficients. We determined the right hand sides of the constraints by calculating the sum of the constraint coefficients for each constraint and then multiplying these values by a constant. This constant is chosen as 0.5 since the resulting instances can be expected to be particularly difficult to solve (cf. [10]). The accuracy Δ is chosen as 0.1% of the area of the initial rectangle. Computations with the C++ program were executed on a work station equipped with a Dual Intel Xeon 3.20GHz running under Linux Kernel 2.6.5 SMP.

The purpose of our numerical study is to evaluate the performance of our program under conditions which are worse than those expected from real data. Furthermore, we report the size of the representation under varying data. According to historical experience in KEIPER, the number of variables averages 50, the planning horizon does not exceed 16 years, the basic time period is chosen to be one year, and the number of constraints is consequently around 48. In the following, we either vary the number of variables or the number of constraints while keeping the other value fixed. For each setup, we generated 100 instances. The numbers we report are averages which explains non-integral numbers of representing points.

We started our analysis by setting the number of constraints to 48 and varying the number of variables between 25 and 75 with steps of 5. The average number of representing points and the average CPU times are given in Table 1 and plotted in Figures 5 and 6.

The average number of representing points roughly doubles as the number of variables triples from 25 to 75. Increasing the number of new projects leads to a modest increase in the number of alternative project portfolios. In contrast, the CPU times increase rapidly which is due to the numerical difficulty of the underlying knapsack problem. Note, however, that a CPU time of less than 10 minutes for a large problem is still acceptable since the problem does not have to be solved online. Nevertheless, having an acceptable average CPU time is significant since the ProSel might later allow to simulate and analyze various scenarios.

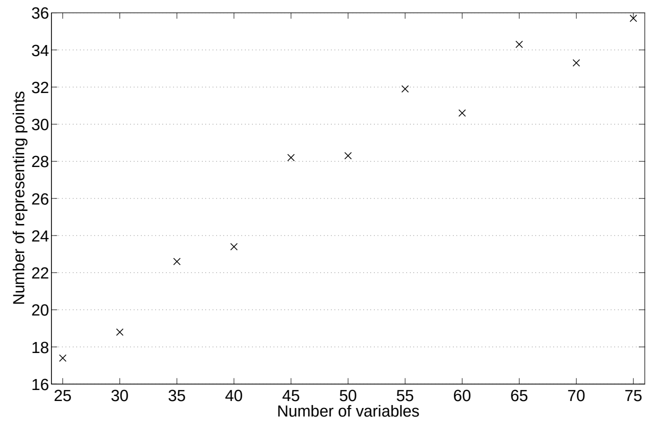


Figure 5: Average number of representing points

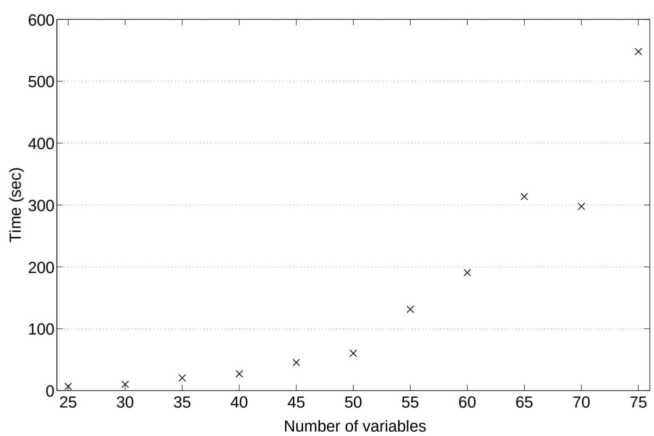


Figure 6: Average CPU time

Variables	Constraints	Average CPU time	Average number of representing points
25	48	6.8	17.4
30	48	10.2	18.8
35	48	20.5	22.6
40	48	27.1	23.4
45	48	45.4	28.2
50	48	60.4	28.3
55	48	131.4	31.9
60	48	190.8	30.6
65	48	313.6	34.3
70	48	297.8	33.3
75	48	547.9	35.7

Table 1: Varying the number of variables while keeping the number of constraints fixed.

In the second part of our testing, we fix the number of variables to 50 and vary the number of constraints from 24, 48, 96 to 192. Recall that on a yearly planning basis we get 48 constraints for a planning horizon of 16 years since we have 3 different constraints per year. Changing the number of constraints to 24, 96, and 192 allows the simulation of a biennial, a biannual, and a quarterly planning basis, respectively. The results can be viewed in Table 2 and Figures 7 and 8.

Variables	Constraints	Average number of representing points	Average CPU time
50	24	28.6	30.6
50	48	28.9	77.2
50	96	29.2	164.5
50	192	29.5	> 511.4

Table 2: Varying the number of constraints while keeping the number of variables fixed.

Interestingly, increasing the number of constraints has only a very small effect on the average number of representing points. However, the average CPU time increases tremendously with the number of constraints. In a few instances with 192 constraints the CPU time exceeded an hour. These instances were not considered in the average CPU time calculation of the

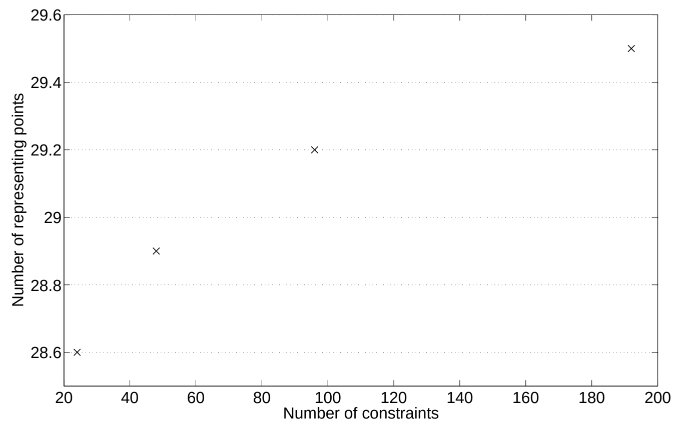


Figure 7: Average number of representing points

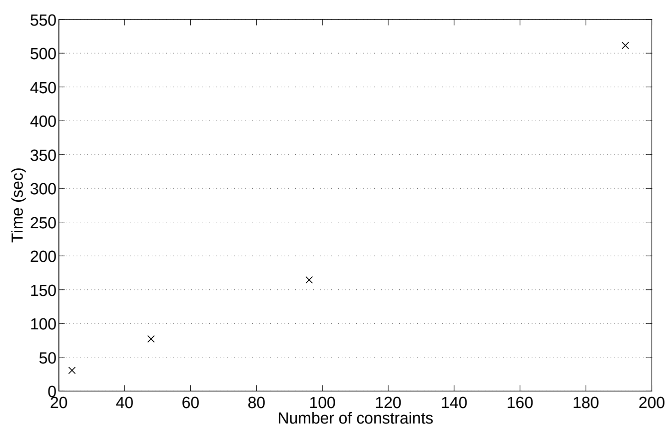


Figure 8: Average CPU time

setup with 192 constraints.

5 Conclusion

Portfolio management of new projects is a strategically important process in any large company since the decisions will effect all subsequent activities like procurement, development, and production. ProSel supports this process with a systematical and flexible approach. The data preparation process which ends with the creation of the input file is standardized and automated. On the other hand flexibility is preserved since it is possible to change parameters like weights of the criteria in order to simulate different scenarios. The solution method finds a limited amount of Pareto-optimal project portfolios in a representative system assuring the quality aspects explained in [5]. The decision maker can search in the resulting data base of representative solutions on-line to identify his ultimate acquisition decision.

ProSel has been tested in the KEIPER case study - nevertheless, the general approach is applicable to any project portfolio selection problem in which the evaluation criteria can be aggregated into two conflicting objectives - and in randomly generated benchmark problems. It has proven its usefulness in simulating, comparing and analyzing different scenarios with visualization and graphics. In this way decision makers can strengthen their decision arguments with quantitative data.

References

[1] *ILOG CPLEX 9.0 User's Manual*, 2003.

[2] V. Chankong and Y. Haimes. *Multiobjective Decision Making Theory and Methodology.* Elsevier Science, New York, 1983.

[3] R.G. Cooper, S.J. Edgett, and E.J. Kleinschmidt. *Portfolio Management for New Products 2^{nd} Edition.* Perseus Book, 2001. Chapter 2.

[4] M. Ehrgott. *Multicriteria Optimization.* Springer-Verlag, Berlin, second edition, 2005.

[5] H.W. Hamacher, C.R. Pedersen, and S. Ruzika. Finding Representative Systems for Discrete Bicriteria Optimization Problems. *Operations Research Letters*, 2006. In press, available online.

[6] C. Hillermeier and Jahn J. Multiobjective optimization: Survey of methods and industrial applications. *Surveys on Mathematics for Industry*, 11:1–42, 2005.

[7] H. Kellerer, U. Pferschy, and D. Pisinger. *Knapsack Problems*. Springer, Berlin, Heidelberg, 2004.

[8] M. Köksalan and S. Zionts, editors. *Multiple Criteria Decision Making in the New Millenium*. Springer Verlag, 2001.

[9] T. Klein. Bewertung von Marktchancen unter Beachtung der Unternehmensstrategie der KEIPER GmbH & Co. KG. Master's thesis, Wirtschaftsingenieurwesen, Fachhochschule Kaiserslautern, Germany, 2005.

[10] S. Martello and P. Toth. Algorithms for knapsack problems. In S. Martello, G. Laporte, M. Minoux, and C. Ribeiro, editors, *Surveys in Combinatorial Optimization*, volume 31 of Annals of Discrete Mathematics, pages 213–257. North Holland, 1987.

[11] K. M. Miettinen. *Nonlinear Multiobjective Optimization*. Kluwer Academic Publishers, Boston, London, Dordrecht, 1999.

[12] L.D. Phillips and C.A. Bana e Costa. Transparent prioritisation, budgeting and resource allocation with multi-criteria decision analysis and decision conferencing. *Annals of Operations Research*, 2006. Forthcoming (Preprint: Working Paper LSEOR 05-75, 2005, London School of Economics).

[13] S. Ruzika and M.M. Wiecek. A survey of approximation methods in multiobjective programming. *Journal of Optimization Theory and Applications*, 126(3), 2005.

[14] A. Tanatmis. Multicriteria Optimization Approach for KEIPER Acquisition Prioritization Problem. Master's thesis, Fachbereich Mathematik, Technische Universität Kaiserslautern, Germany, 2006.

[15] D.J. White. A bibliography on the application of mathematical programming multiple-objective methods. *Journal of the Operational Research Society*, 41(8):669–691, 1990.