TECHNISCHE UNIVERSITÄT MÜNCHEN

Lehrstuhl für Operations Management

Considering Uncertainty in Project Management and Scheduling

Dipl.-Kfm. Univ. Thomas Richard Alexander Fliedner

Vollständiger Abdruck der von der Fakultät für Wirtschaftswissenschaften der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Wirtschaftswissenschaften (Dr. rer. pol.)

genehmigten Dissertation.

Vorsitzender:		Univ.-Prof. Dr. Stefan Minner
Prüfer der Dissertation:	1.	Univ.-Prof. Dr. Rainer Kolisch
	2.	Univ.-Prof. Dr. Walter J. Gutjahr Universität Wien, Österreich

Die Dissertation wurde am 31. März 2015 bei der Technischen Universität München eingereicht und durch die Fakultät für Wirtschaftswissenschaften am 15. Juni 2015 angenommen.

Acknowledgments

I wish to express my gratitude to my advisor, Prof. Dr. Rainer Kolisch, who encouraged me to conduct research in the area of project management and scheduling and guided me throughout the process of writing my dissertation. As my superior, Prof. Dr. Kolisch has provided me with the opportunity to gain invaluable experience while working at the Department of Operations Management of TUM School of Management. I wish to thank my colleagues and peers at TUM School of Management for the inspiring work environment in the past four years. I also would like to thank Prof. Dr. Walter Gutjahr for examining my dissertation as well as Prof. Dr. Stefan Minner for chairing the examination board. Finally, I am grateful to my parents, Monika and Bernhard Fliedner, as well as my partner, Eva Engelhardt, for their limitless support.

Contents

List of Figures

List of Tables

1. Introduction

[...] curator instructus esse debet, nec suae tantum stationis architectis uti, sed plurium advocare non minus fidem quam subtilitatem, ut aestimet quae repraesentanda, quae differenda sint, et rursus quae per redemptores effici debeant, quae per domesticos artifices.

[...] He [the commissioner] ought not only to consult the engineers in his own office, but also to call upon the reliable judgment and expertise of numerous others, that he may in the end determine which tasks are to be undertaken without delay and which are to be postponed, and, again, which are to be carried out by independent contractors and which by workmen of the domestic staff.

(Sextus Julius Frontinus, De aquaeductu urbis Romae)

This description of the responsibilities of the *curator aquarum*, the commissioner of the aqueducts supplying water to the imperial City of Rome, given by Sextus Julius Frontinus demonstrates that issues of planning and executing projects were as pervasive centuries ago as they are in modern society (Frontinus and Rodgers, 2004; Walker and Dart, 2011). Frequently, prominent projects are seen to miss time and cost targets. Among other factors, failure to consider uncertainty in selecting and scheduling projects can be blamed.

This dissertation covers three distinct topics within the domain of project management and project scheduling tied together by a real-life project management problem of selecting initiatives to improve the supply chain function of an international semiconductor manufacturer presented in Chapter 2.

Reyck et al. (2005) emphasize the importance of properly assessing project and portfolio risks in project portfolio management. Ward and Chapman (2003) criticize common risk management practice for emphasizing the downside potential of uncertainty and neglecting unforeseen positive effects. Based on Fliedner and Liesiö (2015), Chapter 3 addresses this issue by extending an established approach for project portfolio selection under uncertainty, Robust Portfolio Modeling (RPM) (Liesiö et al., 2007, 2008), to provide less conservative portfolio recommendations. The chapter develops a formal decision making framework, describes means of providing decision support based on this model, and gives an illustrative example adapted from the case study in Chapter 2.

If operations research is to successfully help people in solving problems, the problem owners should not be neglected (Hämäläinen et al., 2013). Chapter 2 serves as an overall positive example of the practical application of normative decision support models and operations research techniques. Yet, company decision makers have emphasized that they also take into account tacit knowledge and managerial experience in decision making not covered by the formal process. Salo et al. (2011) emphasize the need for research on the occurrence, the impact, and the avoidance of decision biases in portfolio decision making settings, while Kavadias (2014) underlines the potential of experimental research for the domain of project management in general. Chapter 4 based on Fliedner et al. (2014) addresses human behavior in project portfolio selection. Based on the knapsack problem, a generic and controllable problem setting, an experimental framework is designed where subjects may dynamically select and deselect from a list of items to build their desired portfolio. The framework makes it possible to study both subjects' decision quality as well as their selection process.

Hans et al. (2007) survey different perspectives on managing project-driven organizations and propose a positioning framework to aid in selecting better planning approaches. The issue of project portfolio selection, covered by Chapter 2, Chapter 3, and Chapter 4, is regarded as a tactical task that directly leads to underlying operational issues of project scheduling and resource allocation. During evaluation of the decision support system for project portfolio selection reported on in Chapter 2, company decision makers have expressed the wish to receive additional information on ongoing projects in order to aid project tracking and the process of allocating resources. Based on Fliedner and Kolisch (2015), Chapter 5 considers a dynamic project resource allocation and scheduling setting where projects arrive stochastically over time. The chapter proposes novel problem instances to study the performance and stability of approaches for project scheduling under uncertainty first in a static and later in a dynamic setting.

2. A Case Study on the Selection of Supply Chain Improvement Projects in the Semiconductor Industry

Project portfolio selection, choosing a subset from a discrete set of project proposals subject to various constraints, poses significant challenges to businesses and public organizations alike (Kleinmuntz, 2007; Salo et al., 2011). Organizations typically must choose from multiple proposals as they lack the funds, personnel, or time to pursue all of them. Projects may impact an organization in multiple dimensions and may affect multiple stakeholders with different preferences. Usually, significant initial investments are necessary with the anticipation of future benefits. Hereby, project resource requirements, project benefits, as well as aspects of operational project execution are subject to considerable uncertainty (Martinsuo et al., 2014). Issues in project portfolio selection arise on a strategic level in corporate research and development planning (Heidenberger and Stummer, 1999; Dickinson et al., 2001; Stummer and Heidenberger, 2003) or public policy development (Henriksen and Traynor, 1999; Liesiö et al., 2007). On an operational level, a common problem setting is the selection of process improvement initiatives (Santhanam and Kyparisis, 1996; Grushka-Cockayne et al., 2008; Gurgur and Morley, 2008). This chapter considers the latter domain.

We report on a practical decision problem of selecting projects to improve the supply chain function of an international semiconductor manufacturer. While the company has established a clear process for project evaluation and prioritizing overseen by a project portfolio committee, company decision makers have expressed difficulties in assessing project proposals evaluated in terms of multiple criteria and subject to interdependencies. When considered in isolation, a project proposal with high cost and low value may be rejected by traditional evaluation techniques, e.g., the value-for-money principle (Keisler, 2004; Phillips and Bana e Costa, 2007; Lourenço et al., 2012). Nonetheless, the project may be valuable to the portfolio as a whole as it enables the execution of other valuable projects or is rendered profitable due to synergistic effects in combination with other projects. Reyck et al. (2005) emphasize adequate consideration of project interdependencies, incorporation of selection constraints, as well as proper alignment of the portfolio to company strategy as key elements of project portfolio management and drivers of project success. Cooper et al. (2001) empirically study the popularity and effectiveness of different approaches to project portfolio selection. While most companies only employ financial metrics for project selection alone, this practice does not relate to the selection of projects perceived as most valuable by decision makers. Top performing companies additionally rely on non-financial metrics, which result in better alignment of the project portfolio with business strategy. In order for the semiconductor manufacturer to become a champion in project portfolio management, an academic project was initiated. Jointly with the supply chain management department of the company, an interactive decision support system has been developed to provide guidance in the decision making process. The system was designed to provide decision makers with portfolio recommendations that are aligned with the supply chain strategy and provide highest possible financial contribution.

Research in the area of "Portfolio Decision Analysis" (Salo et al., 2011) has brought forth a wide range of quantitative approaches to provide guidance for project selection problem settings. Reviews are given by e.g., Hall and Nauda (1990), Heidenberger and Stummer (1999), and Kolisch et al. (2008). Santhanam and Kyparisis (1996) propose an integer programming model for information system project selection taking into account logical interdependencies as well as synergies between projects. Dickinson et al. (2001) provide decision support for the selection and timing of technology projects at Boeing. A non-linear integer programming model is used to maximize the net present value of the portfolio subject to a budget constraint as well as aspiration levels on the strategic alignment of projects in the portfolio. Logical interdependencies between projects are modeled by a dependency matrix, which is also used to transfer value between interdependent project proposals. Stummer and Heidenberger (2003) develop a multi-objective integer programming model for the selection of research and development projects. All Pareto optimal project portfolios are determined through complete enumeration, which the authors regard as computationally tractable for problem settings with up to 30 projects. The model takes into account logical interdependencies and synergies between projects as well as aspiration levels for all evaluation criteria. Grushka-Cockayne et al. (2008) provide a decision making framework for the selection of operational improvement projects and their execution mode at the European air traffic management organization. The authors consider a linear-additive portfolio value model taking into account project scores evaluated by multiple stakeholders according to multiple criteria. An integer programming model is developed that maximizes overall portfolio value while taking into account multiple budget restrictions and aspiration levels for all score criteria. The model is extended to additionally account for non-linear project synergies by considering decisions on project clusters instead of single projects. Gurgur and Morley (2008) support the selection of maintenance and infrastructure projects at the Lockheed Martin Space Systems Company. A multi-criteria linear-additive portfolio value model is developed, where criteria weights are determined by swing-weighting. The authors propose an integer programming model that maximizes aggregate project utility. Uncertainty with regards to project cost and duration are taken into account by chance-constraints limiting the probability of time and cost overruns.

Aspects of these approaches for project portfolio selection have been considered in the development of the decision support system for the selection of supply chain improvement projects. Based on a multi-attribute portfolio value model (Golabi et al., 1981), we have developed an integer programming model taking into account logical interdependencies as well as synergies. Strategic alignment of the portfolio recommendation is ensured by allowing decision makers to express aspiration levels for evaluation criteria. We illustrate the proposed integer programming model using computational results obtained during a past iteration of the project selection process. Furthermore, we report on the introduction of the decision support system and initial feedback by decision makers.

The chapter is structured as follows. Section 2.1 proposes a formal decision making framework and integer programming model for project portfolio selection, which has been embedded in a decision support system introduced in Section 2.2. We demonstrate the proposed model in a case study in Section 2.3 before discussing the reception of the decision support system in Section 2.4.

2.1. Decision Making Framework

Initially, project owners develop business cases for their project proposals. Using a standardized template, they specify qualitative objectives, the scope, as well as the deliverables of their proposal. Formally, projects $j = 1, \ldots, m$ are evaluated in terms of execution cost giving a cost vector $c \in \mathbb{R}_+^m$ and multiple score criteria $i = 1, \ldots, n$ giving a score matrix $v \in \mathbb{R}_+^{m \times n}$. The additive overall value of project j is $\sum_{i=1}^n w_i v_{ji}$, where the vector of criterion weights $w \in \mathbb{R}_+^n$ relates a unit increase in the criterion-specific score to an increase in the overall project value. Specifically, the potential of supply chain improvement through projects is evaluated in terms of their financial and strategic value. Projects can create financial value by reducing cost within the company supply chain or by enabling additional revenue for the company. Strategically, projects can improve the lead time of customer orders, termed "Speed", or the accuracy of forecasted customer demands, called "Forecast Accuracy" (F/A). While financial value is measured in monetary units, the Speed criterion is measured in days and Forecast Accuracy in percentage points. Weights to translate strategic scores to monetary units have been determined by a previous internal study on supply chain strategy.

Project portfolios are subject to multiple sources of uncertainty stemming from the company environment, project interdependencies, or the projects themselves (Martinsuo et al., 2014). Different approaches of tackling uncertainty in project selection problems have been proposed: The specification of distributions for uncertain data (Gurgur and Morley, 2008), the definition of discrete scenarios of parameter realizations (Liesiö and Salo, 2012), the interval representation of input data (Düzgün and Thiele, 2010), as well as the representation of uncertain information by fuzzy numbers (Tavana and Sodenkamp, 2009). The process established at the semiconductor manufacturer requires project owners to specify a risk parameter with which nominal project scores are multiplied to obtain risk-adjusted scores v. This approach is in line with common practices of project risk management (Cooper et al., 2001; Phillips and Bana e Costa, 2007), yet follows the paradigm of interpreting uncertainty primarily as a threat to portfolio success (Ward and Chapman, 2003).

Evaluated project proposals are discussed, prescreened, and prioritized during monthly core team meetings by representatives from different departments. At this stage interdependencies between project proposals are identified. Firstly, projects can be logically connected, i.e., the execution of one project is dependent on the execution of one or more other projects. As an example, a project to improve company revenue by implementing decision support for supplier negotiations may be dependent on the execution of an independent project to introduce a database on previous procurement contracts. Secondly, synergies may arise from the execution of two or more related projects. Greater cost efficiency of joint project execution creates additional financial value, whereas greater effectiveness in supporting the supply chain strategy creates additional value in terms of Speed or Forecast Accuracy. During quarterly steering committee meetings, prescreened project proposals are presented and a portfolio is selected for execution subject to budget restrictions.

In order to aid decision makers during steering committee meetings, portfolio recommendations are developed by solving an integer programming problem for project portfolio selection. A project portfolio is modeled as a binary vector $x \in \{0,1\}^m$ indicating the selection ($x_j = 1$) of project $j = 1, \ldots, m$. The overall portfolio value is given by the sum of the additive overall values of selected projects (Golabi et al., 1981). Synergies $s = 1, \ldots, o$ account for non-additive beneficial effects $v^S \in \mathbb{R}_+^{o \times n}$ resulting from the joint execution of related projects (Santhanam and Kyparisis, 1996). They are modeled as a binary vector $y \in \{0,1\}^o$ indicating the application ($y_s = 1$) of synergy $s = 1, \ldots, o$ if all required projects $j \in \mathcal{J}_s$ are selected.

The integer programming problem solved in the decision support system is given by

$$\text{maximize} \sum_{i=1}^{n}\sum_{j=1}^{m} v_{ji} w_i x_j + \sum_{i=1}^{n}\sum_{s=1}^{o} v_{si}^S w_i y_s - \sum_{j=1}^{m} c_j x_j \tag{2.1}$$

subject to the constraints

$$\sum_{j=1}^{m} v_{ji} x_j + \sum_{s=1}^{o} v_{si}^S y_s \geq v_i^{\text{asp}} \qquad \forall i = 1, \ldots, n \tag{2.2}$$

$$\sum_{j=1}^{m} c_j x_j \leq B \tag{2.3}$$

$$x_{j'} \leq x_j \qquad \forall (j, j') \in \mathcal{E} \tag{2.4}$$

$$\sum_{j \in \mathcal{J}_s} x_j \geq |\mathcal{J}_s| y_s \qquad \forall s = 1, \ldots, o \tag{2.5}$$

$$x_j, y_s \in \{0, 1\} \qquad \forall j = 1, \ldots, m \quad s = 1, \ldots, o. \tag{2.6}$$

By company policy, projects may only receive funding if they are profitable, i.e., their financial value is higher than their cost, or if they are essential in order to achieve decision makers' strategic goals for the company supply chain. To this end, the integer programming problem is solved to recommend project portfolios that provide the highest possible value contribution (2.1), i.e., the sum of the portfolio value and the aggregate value of synergies reduced by the aggregate cost of selected projects. At the same time, aspiration level constraints (2.2) ensure minimum achievement levels $v^{\text{asp}} \in \mathbb{R}_+^n$ for each score criterion (Stummer and Heidenberger, 2003; Kleinmuntz, 2007), particularly reflecting strategic goals in terms of Speed and Forecast Accuracy.

Project selection is further restricted by an overall budget constraint (2.3) as well as logical interdependencies (2.4). The budget constraint specifies that the aggregate cost of all executed projects may not exceed a given budget level $B \in \mathbb{R}_+$. In the set of logical interdependencies $\mathcal{E}$, a tuple $(j, j') \in \mathcal{E}$ indicates that project j' can only be selected if another project j required for its execution is selected as well (Santhanam and Kyparisis, 1996). Finally, Constraints (2.5) ensure that synergy values are only taken into account if the required projects are selected.

2.2. Decision Support System

The integer programming model has been embedded in a decision support system to aid the selection process during steering committee meetings. The decision support system user interface consists of an interactive browser client integrated in the internal network of the company. The back-end system builds and solves the integer programming problem and manages the presentation of project and portfolio information in the interface.

Figure 2.1 provides an illustration of the user interface. At the bottom left of the interface, a list of initially all project proposals is displayed (Section "Project List"). The list contains the names of the projects as well as their financial value. By clicking on any project a window with detailed information appears. In the top right part of the interface, slide bars allow decision makers to adjust aspiration levels as a percentage of maximum attainable criteria scores $v^{\max} \in \mathbb{R}^n_+$. When the decision support system is started, maximum scores for each criterion $i = 1, \ldots, n$ are determined by solving integer programming problem

$$v_i^{\max} = \text{maximize} \sum_{j=1}^{m} v_{ji} x_j + \sum_{s=1}^{o} v_{si}^S y_s \tag{2.7}$$

subject to constraints (2.3) - (2.6) for each score criterion. Absolute aspiration levels are given below the slide bars and are automatically adjusted whenever the slide bars are changed.

By clicking the "Solve" button, decision makers can invoke the decision support system to solve integer programming problem (2.1) - (2.6) taking into account the chosen aspiration levels v^{asp}. The resulting portfolio recommendation, i.e., a list of all selected projects, is displayed at the bottom right of the interface (Section "Portfolio List"). In the center of the interface, a bubble chart (Cooper et al., 2001) of the recommended portfolio is displayed. All selected projects are plotted in the space of their financial and strategic impact, i.e., the sum of projects' monetized Speed and Forecast Accuracy scores. Independently, logical interdependencies and synergy relationships of pairs of projects can be visualized in a dependency matrix (Dickinson et al., 2001; Killen and Kjaer, 2012). The top left "Portfolio" Section contains information on the budget limit as well as aggregate information on the recommended project portfolio. Portfolio cost, aggregate values for each score criterion, as well as the overall portfolio value are presented. The section furthermore presents the objective function value of the integer programming problem (2.1) - (2.6), the portfolio contribution.

Decision makers can manually select projects by dragging them from the "Project List" to the "Portfolio List". Projects may be excluded from the portfolio by dragging them to the edge of the decision support system screen. Whenever a project is manually selected or excluded, the integer programming problem (2.1) - (2.6) is automatically solved again taking into account decision makers' manual adjustments. If the integer programming problem is infeasible, a notification is displayed prompting decision makers to adjust their choices.

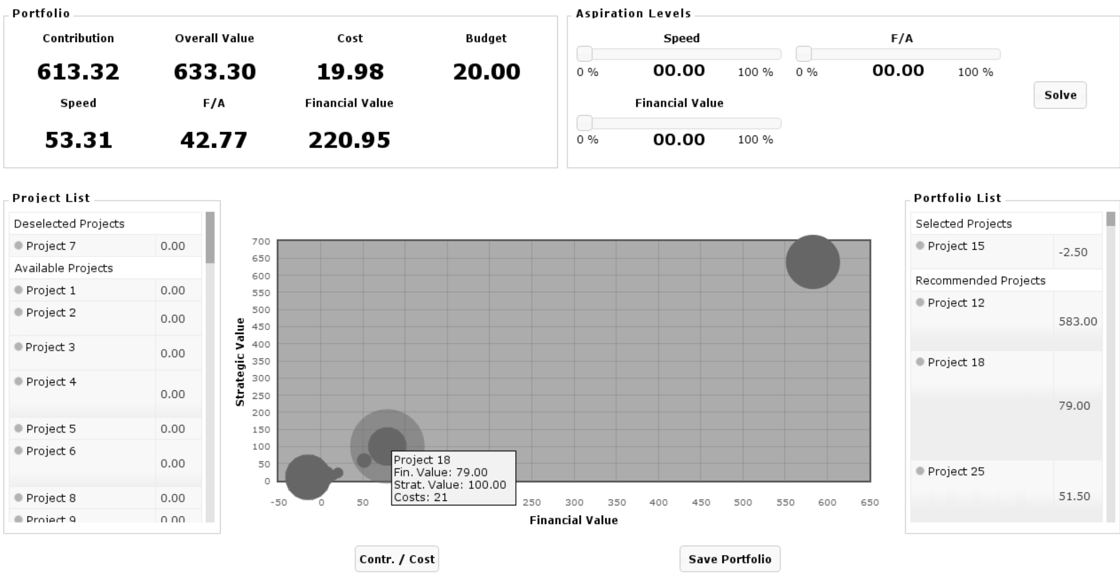


Figure 2.1.: Illustration of the decision support system interface

2.3. Case Study

We report computational results obtained during a past iteration of the project selection process. Table 2.1 summarizes all relevant project information. 58 projects (named P1 to P58) are evaluated in terms of $n = 3$ criteria, Speed ($i = 1$), Forecast Accuracy ($i = 2$), and financial value ($i = 3$). While financial value is directly measured in monetary units ($w_3 = 1$), improving Speed by one unit is worth $w_1 = 2.6$ monetary units, and one unit improvement in Forecast Accuracy is worth $w_2 = 6.4$ monetary units. 13 projects can only be selected if other projects are selected as well. Overall, 26 logical interdependencies between projects exist. Synergies are summarized in Table 2.2. For example, the joint selection of projects P9 and P27 creates additional value $v^S_{1,3} = 5.46$, amounting to 19% of the overall value of the two projects (synergy S1).

We report optimal solutions to integer programming problem (2.1) - (2.6) for different budget levels B and no restrictions on aspiration levels ($v_1^{\text{asp}} = v_2^{\text{asp}} = v_3^{\text{asp}} = 0$). Overall, the integer programming problem contains 61 decision variables and 33 constraints. Optimal portfolios for all budget levels $B = 5, 10, 15, \ldots, 125$ were determined in less than five seconds (IBM ILOG CPLEX 12.6, 3.2 GHz dual-core processor, 8 GB memory). For most budget levels, unique optimal solutions exist. Only for budget levels $B = 25$ and $B = 60$, ten distinct optimal portfolios exist.

j	v_{j1}	v_{j2}	v_{j3}	c_j	Required Projects
	$w_1 = 2.6$	$w_2 = 6.4$	$w_3 = 1$		
P1	0.0	0.0	0.4	0.44	
P2	0.0	35.57	0.0	1.24	
P3	0.0	0.0	29.38	1.2	
P4	0.0	0.0	28.32	0.52	
P5	0.0	0.0	29.68	16.2	
P6	0.0	0.0	112.42	51.82	P7, P8, P9, P10, P11, P28
P7	0.0	0.67	0.0	0.62	
P8	2.0	0.0	0.0	0.22	
P9	0.0	0.0	1.02	0.26	
P10	0.0	7.2	0.0	1.4	
P11	0.0	0.67	0.0	0.96	
P12	0.0	0.0	48.96	4.98	P27, P33, P34
P13	0.0	0.0	8.63	6.84	
P14	0.0	0.0	0.33	0.61	
P15	0.0	0.0	0.35	0.62	
P16	0.0	0.0	7.32	0.22	P37
P17	0.0	0.0	0.45	0.62	
P18	0.0	0.0	21.95	1.0	
P19	5.99	0.0	0.0	0.34	
P20	13.64	0.0	0.0	0.78	
P21	1.66	0.0	0.0	0.22	
P22	4.09	0.0	0.0	0.88	P8
P23	3.29	0.0	0.0	0.58	P19, P20, P21, P22, P24, P25, P26
P24	1.64	0.0	0.0	0.34	P8
P25	13.64	0.0	0.0	1.26	
P26	5.46	0.0	0.0	1.34	P27
P27	0.0	0.0	27.29	2.48	P28
P28	0.0	0.0	27.29	1.7	P29
P29	0.0	0.0	17.56	2.3	
P30	0.0	0.0	2.73	0.38	
P31	0.0	0.71	2.11	1.18	P34
P32	0.0	0.0	29.33	2.14	
P33	0.0	0.0	18.19	12.86	
P34	0.0	0.0	1.23	0.64	P37
P35	0.0	0.0	0.14	0.74	P31
P36	0.0	0.0	0.67	0.22	
P37	6.55	0.0	1.64	0.34	
P38	0.0	0.0	0.8	0.22	
P39	0.0	0.0	1.06	0.22	
P40	0.0	0.0	1.72	0.22	
P41	0.0	0.0	1.66	0.22	P37
P42	0.0	0.0	8.01	0.22	
P43	0.0	0.0	2.77	0.22	
P44	0.0	0.0	3.33	1.34	
P45	0.0	0.0	0.67	0.22	
P46	0.0	0.0	6.65	1.6	
P47	0.0	0.0	0.33	0.12	
P48	0.0	0.0	1.0	0.1	
P49	0.0	0.0	0.33	0.08	
P50	0.0	0.0	0.33	0.16	
P51	0.0	0.0	0.4	0.1	
P52	0.0	0.0	0.4	0.05	
P53	0.0	0.0	0.4	0.1	
P54	0.0	0.0	0.4	0.1	
P55	0.0	0.0	0.4	0.2	
P56	0.0	0.0	0.4	0.38	
P57	0.0	0.0	0.4	0.1	
P58	0.0	0.0	0.4	0.1	

Table 2.1.: Project scores, cost, and logical interdependencies

s	v^S_{s1}	v^S_{s2}	v^S_{s3}	Required Projects
	$w_1 = 2.6$	$w_2 = 6.4$	$w_3 = 1$	
S1	0.0	0.0	5.46	P9, P27
S2	4.09	0.0	8.19	P20, P27
S3	0.0	0.0	0.74	P11, P31

Table 2.2.: Synergy scores

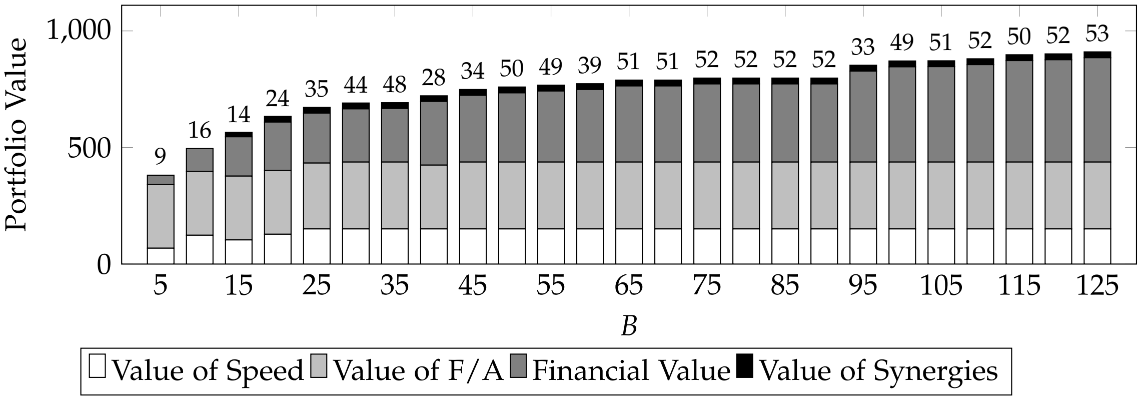


Figure 2.2.: Value composition of optimal portfolios for given budget levels

Figure 2.2 reports the value composition of optimal portfolios. Portfolio value with regards to each score criterion $i = 1, \ldots, n$ is given by $\sum_{j=1}^{m} v_{ji} w_i x_j$. Portfolio value due to synergies is calculated as $\sum_{i=1}^{n} \sum_{s=1}^{o} v^S_{si} w_i y_s$. The number of selected projects $\sum_{j=1}^{m} x_j$ is shown above the bar charts. For low budget levels $B = 5, \ldots, 25$, we find that portfolios are dominated by projects providing strategic impact in terms of Speed and particularly Forecast Accuracy. Scores in strategic criteria initially rise quickly and reach their peak at budget level $B = 30$ whereafter additional portfolio value is primarily driven by financial value. Value from synergies is utilized from budget level $B = 15$ onward. Value contribution from Speed is reduced when increasing the budget from $B = 10$ to $B = 15$ in order to accommodate projects that provide higher financial value.

Figure 2.3 reports the relative frequency of each project and each synergy being selected in an optimal portfolio for varying budget levels. While relative frequencies are either zero or one when there is only one unique optimal solution, fractional values are possible for B=25 and B=60 where multiple optimal solutions exist. High relative frequencies of selection across varying budget levels give an indication of the value of a project to decision makers. A similar approach termed “core index” is employed by Liesiö et al. (2007). We find that projects P2, P4, P10, P19, P20, P37, and P42 are selected

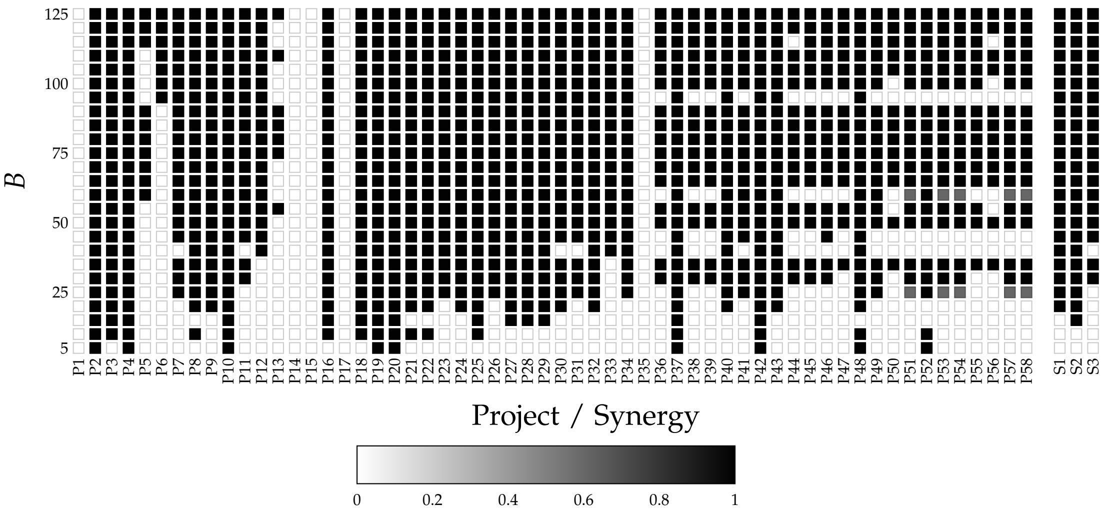


Figure 2.3.: Relative frequency of project and synergy selection for given budget levels

for all considered budget levels. P2 and P10 are valuable with regards to the Forecast Accuracy criterion, and P10 additionally enables project P6. P4 and P42 provide high financial value relative to their cost. P19 and P20 are valuable with regards to the Speed criterion relative to their cost, and P20 is additionally required for P23 as well as synergy S2. Furthermore, P37 is required for execution of P16 and P34. Invariant to the available budget, projects P1, P14, P15, P17, and P35 are never selected. Their overall value is lower than their cost and they do not contribute to the portfolio indirectly through logical interdependencies or synergies. As no aspiration levels warrant the selection of these projects for strategic purposes, these non-profitable projects are not considered in portfolio recommendations.

The decision support system allows decision makers to manually select or exclude specific projects. When a project included in the optimal portfolio for a given budget level is manually excluded, the highest obtainable portfolio contribution (2.1) taking into account the exclusion will be no higher than the contribution when not forcing this project out of the portfolio. The difference in contribution gives the opportunity cost of manually excluding the project (Ghasemzadeh et al., 1999). Likewise, opportunity cost or infeasibility of integer programming problem (2.1) - (2.6) arises when manually selecting a project not included in the optimal portfolio. In order to shed light on the effect of manual adjustments on the portfolio contribution, Figure 2.4 presents the percentage loss in portfolio contribution when requiring the selection ($x_j = 1$) or exclusion ($x_j = 0$) of each project $j = 1, \ldots, m$ for budget levels $B = 5, 10, \ldots, 125$. For each project, opportunity cost due to manual selection are indicated by an upward-facing arrow, while opportunity cost due to manual exclusion are represented by a downward-facing

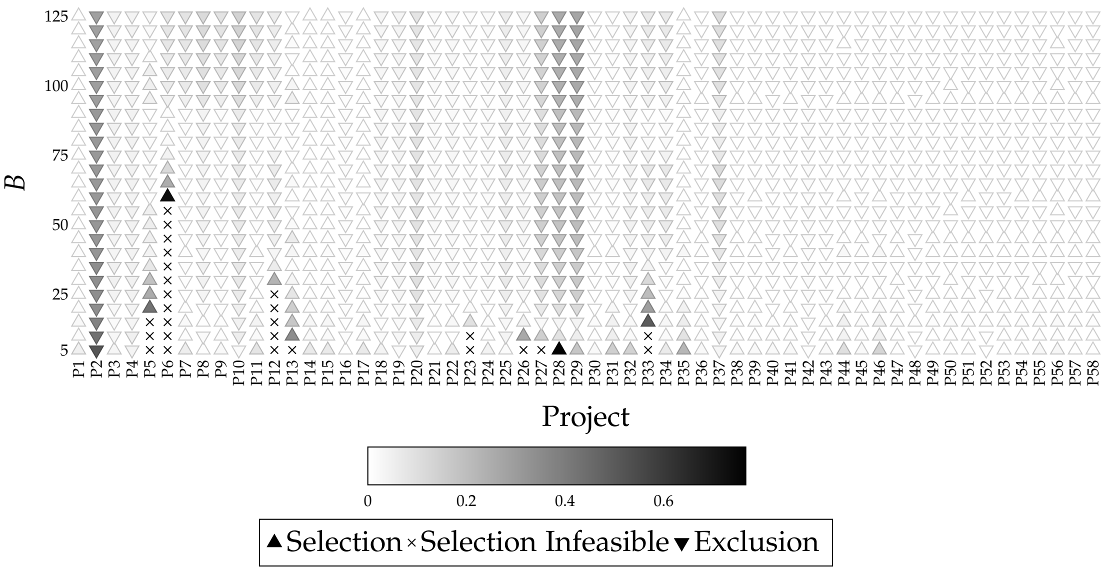


Figure 2.4.: Percentage loss in portfolio contribution when requiring the selection or exclusion of a project for given budget levels

arrow. If the manual selection of a project causes the integer programming problem to become infeasible, a cross is given in the chart. Across all budget levels, opportunity cost are particularly high when excluding project P2, valuable with regards to the Forecast Accuracy criterion. Due to high project cost, selecting projects P5, P6, and P12 cause infeasibility for low budget levels and opportunity cost for higher budget levels. While for low budget levels $B \leq 10$ opportunity cost are incurred when projects P27, P28, and P29 are selected, their exclusion is penalized for higher budget levels. This is due to the fact that projects P28 and P29 are required for the selection of project 27, which in turn shares synergies with projects P9 and P20. Selecting projects P27, P28, or P29 for low budget levels is not cost-efficient as these projects rank among the 15% most expensive projects. On the other hand, project interdependencies prove valuable when sufficient budget is available to select all interdependent projects and utilize synergies.

Finally, we study the effect of aspiration levels on portfolio performance in terms of portfolio contribution (2.1). Figure 2.5 reports the loss in portfolio contribution when aspiration levels are imposed compared to the case without aspiration levels. For a fixed budget level $B = 20$ and a fixed aspiration level of financial value $v_3^{\text{asp}} = 220$, which equals 95% of $v_3^{\max}$, we vary the aspiration levels of strategic score criteria v_1^{asp} and v_2^{asp}. For a wide range of aspiration levels, decision makers only experience negligible losses in portfolio contribution. Setting aspirations too high causes the problem setting to quickly become infeasible. A trade-off between Speed and Forecast Accuracy arises only at the edge of the space of feasible aspiration levels. Significant losses occur for extreme decision maker preferences, e.g., requiring v_1^{asp} to be greater than 85% of v_1^{max} and setting $v_2^{asp} = 0$. The limited tradeoff between Speed and Forecast Accuracy is explained by optimal portfolios accommodating as much strategic value as possible. Projects providing strategic value are preferable as they have a significantly higher (monetized) value-to-cost ratio, 26 for Speed and 46 for Forecast Accuracy on average, than projects providing financial value, which have a ratio of 8 (See e.g., Lourenço et al. 2012 for a discussion of the value-to-cost ratio an an evaluation metric). As portfolios already accommodate high levels of strategic value, aspiration levels are easily achieved without sacrificing much portfolio contribution. These results also hold for higher budget levels and different aspiration levels for financial value.

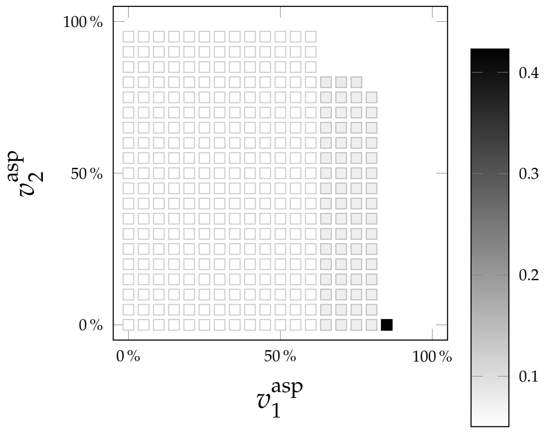


Figure 2.5.: Percentage loss in portfolio contribution for given aspiration levels v_1^{asp} and v_2^{asp} as percentages of the highest achievable criterion score

2.4. Observations and Insights

Stating that decision support systems frequently bear little relevance to management and fail to accommodate decision makers' working style, Marx et al. (2011) perform an empirical study to derive six principles for designing successful decision support systems. They emphasize system ease-of-use, system flexibility and adaptability, as well as a comprehensive and well manageable information model. A trade-off arises between the comprehensiveness of information and support system usability. Similar aspects were emphasized by decision makers within the semiconductor manufacturer during development of the decision support system. To aid usability of the decision support system, relevant project and portfolio information should be displayed in a clear fashion and the system should employ visualizations and wording aligned to the established project selection process. Decision support system interactivity was of prime importance to the decision makers. Displayed information and portfolio recommendations should be easily adjustable during meetings and should support live adjustments to aspiration levels and manual selection or exclusion of projects. Furthermore, to ensure information consistency and minimize system maintenance, the decision support system, which requires information on projects, project interdependencies, as well as the project portfolio environment, should seamlessly integrate with existing data sources.

When being presented with the final decision support system design, company decision makers expressed great satisfaction and were eager to employ it in future decision making sessions. After the first supported session, an initial survey was conducted with three key stakeholders of the decision support system. Decision makers regarded the decision support system as an important tool for the decision making process and expressed confidence in the system improving the quality of the project portfolio. They emphasized a need for proper system documentation and user training in order to get accustomed to the system interface as well as the unfamiliar paradigm of normative operations research models underlying the system.

In terms of future improvements to the decision support system, decision makers proposed including additional information regarding both new project proposals as well as previously selected projects. Firstly, project specific key performance indicators as well as qualitative decision criteria are used by decision makers to make portfolio choices and are currently not accommodated by the decision support system and the established portfolio selection process. Secondly, the decision support system shall also be employed to track ongoing projects and potentially adjust the previously established portfolio by terminating ongoing projects.

On a cautionary note, decision makers have also emphasized discrepancies between the formal project selection process supported by the decision support system and actual project selection behavior. While project proposals are evaluated by standardized, quantitative criteria, which build the foundation of the recommendations by the decision support system, decision makers also take into account tacit knowledge and managerial experience not covered by the formal process. This aspect of project portfolio selection has been discussed by Loch (2000), based on empirical findings obtained at a European technology manufacturer. He concludes that "it is too simplistic to hope for an application of general 'best practice' [. . .] processes". Loch suggests that organizations should develop a mixture of formal and informal processes, which together meet the strategic needs of the organization needs. To this end, this academic project and case study has not developed a universal "holy grail" solution to project portfolio selection, but one building block for successful project portfolio management within the supply chain function of the semiconductor manufacturer.

3. Adjustable Robustness for Multiobjective Project Portfolio Selection

Socio-economic decision problems often involve decision makers facing alternative courses of action, which if chosen consume resources and have multi-dimensional consequences. Research in the area of "Portfolio Decision Analysis" strives to bring greater rationality and transparency to such resource allocation decisions (Kleinmuntz, 2007; Salo et al., 2011). Particularly, linear-additive portfolio selection models have frequently found high impact applications (Ewing et al., 2006; Grushka-Cockayne et al., 2008). Due to the strategic nature of typical portfolio decisions (Salo et al., 2011; Lourenço et al., 2012), consequences, selection constraints, and decision maker preferences are subject to significant uncertainty. In uncertain environments any optimal portfolio determined for some point estimate parameter values might prove unsatisfactory for other possible realizations of uncertain parameter values.

Robust Portfolio Modeling (RPM) (Liesiö et al., 2007, 2008) is a Portfolio Decision Analysis approach designed for decision environments in the context of project portfolio selection where project consequences given as scores in multiple criteria and decision maker preferences are subject to considerable uncertainty. RPM identifies non-dominated portfolios, i.e., portfolios feasible in the sense that they satisfy relevant budgetary and other constraints and for which no other feasible portfolio yields greater value for all possible realizations of the uncertain parameters. Decision makers are provided with a set of non-dominated portfolios to choose from and receive decisional guidance on which projects' uncertainty has the greatest impact on the decision problem. By demanding dominance for all possible realizations of uncertain parameters, decision recommendations provided by RPM are by design explicitly conservative. RPM has found practical application to develop a research agenda for the forestry sector (Könnölä et al., 2007) and to select road maintenance projects (Liesiö et al., 2007).

In this chapter we combine RPM with the robust optimization concept by Bertsimas and Sim (2004). The number of project scores that are assumed to deviate from their most likely value when determining dominance relations between portfolios are limited.

By adjusting this deviation limit, decision makers are able to choose a desired level of conservatism. When no limits are imposed, our approach coincides with standard RPM, while strict limits cause our approach to omit deviations and compare portfolios based on their most likely scores. Decision makers may choose a deviation limit based on a metric for the share of possible realizations considered in determining non-dominated portfolios. Our approach is easily extended to consider adjustable robustness not only with regards to project and portfolio scores but also with regards to constraint satisfaction.

After presenting the theoretical background in the following section, we introduce the proposed decision making framework in Section 3.2. Section 3.3 describes how RPM with adjustable robustness can be applied to support project portfolio selection and how decision makers can determine desired levels of robustness. Section 3.5 extends the framework to account for interdependencies among project scores and for modeling uncertain coefficients in portfolio feasibility constraints. Sections 3.4 and 3.6 apply the developed framework to supply chain management in the semiconductor industry.

3.1. Earlier Approaches to Robust Project Portfolio Selection

Robust modeling approaches aid decision makers in accounting for ambiguity and stochastic uncertainty of parameters to decision problems when detailed information on the nature of uncertainty, e.g., probability distributions, is not readily available (Bertsimas et al., 2011). Both theoretical and practical contributions to the field are covered by the surveys of Roy (2010), Bertsimas et al. (2011), and Gabrel et al. (2014). Bertsimas et al. (2011) particularly emphasize the issue of conservatism when dealing with robust modeling techniques. Kouvelis and Yu (1997) seek to find solutions that minimize the worst case performance within a set of all possible realizations of uncertain parameters. Such a realization where "everything is assumed to go wrong" is bound to be very unlikely in practice.

The robust modeling approach by Bertsimas and Sim (2003, 2004) explicitly addresses the problem of conservatism. The authors model uncertainty in linear and integer programming problems by allowing uncertain coefficients of constraints or the objective function to realize within an interval, symmetrically extending around the most likely value of the coefficient. By limiting the number of coefficients that may deviate from their most likely value, the authors allow decision makers to choose their desired level of robustness. Bertsimas and Sim investigate the trade-off between the probability of constraint violations and the most likely objective function value, termed the price of robustness. The authors furthermore consider "correlated" constraint coefficients, whose deviation from their most likely values is determined by the linear combination of a common set of factors subject to uncertainty. Perfectly correlated constraint coefficients depend on one common factor only, while uncorrelated coefficients are each determined by independent factors. In this modeling approach parameter Γ controls the number of factors permitted to take on their worst-case value rather than the constraint coefficients themselves.

Several studies apply the robust modeling technique by Bertsimas and Sim (2003, 2004) to portfolio selection problems. Kachani and Langella (2005) develop integer programming models for multi-period capital rationing and budgeting problems that are robust with regards to uncertain net present values of projects. Robustness with regards to objective function and constraint coefficients is considered. Düzgün and Thiele (2010) propose an integer programming model for the selection of research projects. Uncertain project cash flows are allowed to fall into one of several value intervals. The number of parameters that may fall into each interval is chosen by the decision makers. The authors propose a robust heuristic procedure where projects are selected in order of decreasing cash-flows or cash-flow to cost ratios. Düzgün and Thiele (2012) compare the robust optimization approach to a stochastic programming model. Gregory et al. (2011) review robust optimization methods in financial portfolio selection and evaluate linear programming models for the robust selection of financial assets. The authors discuss model formulations that support uncorrelated as well as correlated uncertain returns. Different ways of deriving uncertainty sets from real life data, portfolio diversity, and tradeoffs between robustness and portfolio performance are evaluated using data on equity returns from the London stock exchange. Hassanzadeh et al. (2014) consider project portfolio selection under multiple objectives and subject to constraints. Feasible portfolios Pareto optimal with regards to all objectives are generated by solving an augmented weighted Tchebycheff program (Steuer, 1986). The authors propose a robust mixed integer programming model that protects against uncertainty in objective function coefficients, i.e., project scores, as well as coefficients of selection constraints. The authors do not assume a priori information on decision maker preferences with regards to objectives but implicitly elicit these in an interactive procedure. A set of candidate portfolios is generated by solving the mixed integer programming model several times while randomly weighting different objective functions. Based on the decision makers' selection from this set a new set of candidates is generated with random weights being more focused based on the decision makers' choice. The procedure ends when decision makers are content with their portfolio choice or a fixed number of iterations have been performed.

Liesiö et al. (2007) develop Robust Portfolio Modeling to address project portfolio selection under uncertainty. Portfolio values are given by a linear-additive model (Golabi et al., 1981; Liesiö and Punkka, 2014) where projects are evaluated in terms of multiple criteria. Preference Programming techniques (Salo and Hämäläinen, 1995) account for uncertain project scores, bounded by value intervals, as well as uncertain criteria weights, bounded by decision makers' preference statements. Parameters of selection constraints, e.g., budgetary restrictions or logical selection constraints, are assumed to be known with certainty. It is unlikely that there exists a portfolio that provides the highest overall value for all considered score realizations and criterion weights. RPM provides decision makers with a set of non-dominated portfolios, which each are not strictly inferior for all possible realizations to any other portfolio. The authors develop a dynamic programming algorithm to compute the set of non-dominated portfolios and propose an iterative process to reduce this set by acquiring additional preference and score information. They employ a metric termed core index to determine projects whose score information impacts dominances between portfolios. Liesiö et al. (2008) extend RPM to account for project interdependencies and uncertainty with regards to selection constraints. They consider interval-valued coefficients of a constraint that limits aggregate portfolio cost to a (variable) budget. Budgetary restrictions are required to hold for all possible cost realizations. The authors ensure portfolio cost-efficiency by including project cost as a weighted score criterion.

In this chapter we relax the strictly conservative robustness paradigm of Liesiö et al. (2007, 2008) by employing the adjustable robustness model by Bertsimas and Sim (2003, 2004). Decision makers control the level of conservatism with regards to uncertain project scores and coefficients of selection constraints by choosing a suitable parameter Γ. While the related project portfolio selection approach by Hassanzadeh et al. (2014) implicitly elicits decision makers' preferences through an interactive procedure, our approach, based on Robust Portfolio Modeling, separates preference elicitation from portfolio selection. Uncertain preference information, which may be elicited by different methods from preference programming (Salo and Hämäläinen, 1995), is taken into account when determining the complete set of non-dominated portfolios. The interactive RPM procedure furthermore provides decision makers with guidance in refining uncertain project scores.

3.2. Decision Making Framework

Project proposals $j = 1, \ldots, m$ are evaluated with regards to criteria $i = 1, \ldots, n$ giving a score matrix $v \in \mathbb{R}^{m \times n}$, where v_{ji} indicates the matrix element in the jth row and ith column. The additive overall value of project j is $\sum_{i=1}^{n} w_i v_{ji}$, where the vector of criterion weights $w \in \mathbb{R}^n$ is scaled so that

$$w \in S_w^\infty = \left\{ w \in \mathbb{R}^n \mid w_i \geq 0, \sum_{i=1}^{n} w_i = 1 \right\}. \tag{3.1}$$

The weight w_i thus relates a unit increase in the criterion-specific score to an increase in the overall value. S_w^∞ is termed the information set of criteria weights.

A project portfolio is a subset of the available proposals and is modeled as a binary vector $z \in \{0,1\}^m$, such that $z_j = 1$ if and only if project j is included in portfolio z. We assume a linear-additive portfolio model (Golabi et al., 1981; Liesiö and Punkka, 2014), where the overall portfolio value is given by the sum of its constituent projects' overall value

$$V(v,w,z) = \sum_{i=1}^{n}\sum_{j=1}^{m} v_{ji} w_i z_j = z^T v w. \tag{3.2}$$

Projects are selected subject to linear constraints $l = 1, \ldots, q$, whose coefficients form matrix $A \in \mathbb{R}^{q\times m}$ and whose respective limits are given by vector $B \in \mathbb{R}^q$. Constraints typically considered in portfolio selection problems are budgetary restrictions, which limit the execution of costly projects, or logical constraints, which impose restrictions on the combined execution of multiple projects (Ghasemzadeh et al., 1999; Stummer and Heidenberger, 2003). The set of project portfolios feasible in terms of all considered constraints is given by

$$Z_F = \left\{ z \in \{0,1\}^m \mid Az \leq B \right\}, \tag{3.3}$$

where $\leq$ holds componentwise.

If complete information on the weights and scores is available, the most preferred feasible portfolio maximizes the overall value (3.2). This optimal portfolio can be obtained as the solution to the integer programming problem

$$\max_{z\in Z_F} V(v,w,z) = \max_{z\in\{0,1\}^m} \left\{ z^T v w \mid Az \leq B \right\}. \tag{3.4}$$

3.2.1. Incomplete Information

Our model uses the Preference Programming approach to capture incomplete information on the importance of criteria (Weber, 1987; Salo and Hämäläinen, 1992; Salo and Punkka, 2005; Danielson et al., 2007; Sarabando and Dias, 2010). Preference statements are modeled as a set of feasible weights $S_w \subseteq S_w^\infty$ satisfying linear constraints. For instance, ranking three criteria in terms of their importance could result in the weight set $\{w \in S_w^\infty \mid w_2 \geq w_3 \geq w_1\}$. Thus, the set of feasible weights $S_w = S_w^\infty$ corresponds to lack of any preference information, while a single weight vector $S_w = \{w\}$ corresponds to complete information.

Uncertainty with regards to project scores is modeled by allowing each score v_{ji} to deviate from the most likely score $\hat{v}_{ji}$ in both directions by at most $\vec{v}_{ji}$, which results in a range of feasible scores $[\hat{v}_{ji} - \vec{v}_{ji}, \hat{v}_{ji} + \vec{v}_{ji}]$. The information set of all possible project score (matrices) is thus

$$S_v^\infty = \left\{ v \in \mathbb{R}^{m \times n} \mid v_{ji} = \hat{v}_{ji} + \vec{v}_{ji} y_{ji}, y \in [-1,1]^{m \times n} \right\}. \tag{3.5}$$

Comparing different portfolios based on their overall values over the full range of S_v^∞ (Liesiö et al., 2007, 2008) leads to strictly conservative judgments. Extreme realizations may be very unlikely as they require multiple scores to take on values at the border of their ranges but are equally considered in comparing alternatives.

Following the approach of budgeted robustness by Bertsimas and Sim (2003) and Bertsimas and Sim (2004), we base portfolio decision making on a subset of the polytope S_v^∞, in which deviations are limited by a parameter Γ.

Definition 3.2.1 *An adjustable uncertainty set for project scores is given by*

$$S_v^\Gamma = \left\{ v \in \mathbb{R}^{m \times n} \mid v_{ji} = \hat{v}_{ji} + \vec{v}_{ji} y_{ji}, y \in [-1,1]^{m \times n}, \sum_{i=1}^{n} \sum_{j=1}^{m} |y_{ji}| \leq \Gamma \right\}$$

with deviation limit $\Gamma \in [0, mn]$.

The number of scores that may deviate from their most likely values is limited by Γ. For $\Gamma = mn$ the largest possible score information set $S_v^\Gamma = S_v^\infty$ is considered, while $\Gamma = 0$ indicates that no deviation of scores from their most likely values is taken into account.

3.2.2. Dominance Relations

When considering uncertain preferences and scores, the overall value $V(v, w, z)$ of a portfolio is uncertain as well. It is typically not possible to determine one portfolio z optimal for all combinations of w and v. However, it is often possible to use dominance relationships between portfolios to rule out strictly inferior solutions.

Definition 3.2.2 *Portfolio z dominates z' with regard to the information set $S = (S_v^\Gamma \times S_w)$, denoted $z \succ_S z'$, if and only if $V(v, w, z) \geq V(v, w, z')$ for all $(v, w) \in S$ and $V(v, w, z) > V(v, w, z')$ for some $(v, w) \in S$.*

Theorem 3.2.1 shows that determining whether z dominates z' or not can be established by solving a continuous knapsack problem (solvable in linear time, see e.g., Martello and Toth 1990) in each extreme point of the convex hull of S_w, denoted $\text{ext}(\text{conv}(S_w))$. If a project is included in both portfolios z and z', it has an equal contribution to the overall values of both portfolios for any realization of scores. Therefore, it is not relevant for establishing dominance. All proofs are included in the appendix.

Theorem 3.2.1 *Let $z, z' \in \{0,1\}^m$ and information set $S = (S_v^\Gamma \times S_w)$ with parameter Γ. Then*

$$z \succ_S z' \Leftrightarrow \begin{cases} V(\hat{v}, w, z) \geq V(\hat{v}, w, z') + \beta(\vec{v}, w, z, z', \Gamma) & \forall w \in ext(conv(S_w)) \\ V(\hat{v}, w, z) > V(\hat{v}, w, z') & \exists w \in ext(conv(S_w)) \end{cases},$$

where

$$\beta(\vec{v}, w, z, z', \Gamma) = \max_{y \in [0,1]^{m \times n}} \left\{ \sum_{i=1}^{n} \sum_{j \in J(z,z')} w_i \vec{v}_{ji} y_{ji} | \sum_{i=1}^{n} \sum_{j=1}^{m} y_{ji} \leq \Gamma \right\}$$

and $J(z, z') = \left\{ j \in \{1, ..., m\} \ | (z_j = 1, z'_j = 0) \vee (z_j = 0, z'_j = 1) \right\}$.

The introduced notion of dominance has all analytical properties to establish theoretically sound preference orders between project portfolios as stated by the following lemma.

Lemma 3.2.1 $\succ$ *is (i) asymmetric, (ii) irreflexive, and (iii) transitive.*

As Γ increases the dominance relations are established more conservatively. For $\Gamma = mn$ dominance exists only if a portfolio has a higher overall value for any scores in their intervals. This case coincides with the standard RPM case. At the other extreme, $\Gamma = 0$ implies that dominance exists whenever a portfolio has a higher overall value for most likely scores than another portfolio for all feasible weights.

Corollary 3.2.1 *Let $z, z' \in \{0,1\}^m$.*

(i) $z \succ_{(S_v^{mn} \times S_w)} z'$ *if and only if* $V(v, w, z) \gneq V(v, w, z')$ *for all* $(v, w) \in (S_v^\infty \times S_w)$,

(ii) $z \succ_{(S_v^0 \times S_w)} z'$ *if and only if* $V(\hat{v}, w, z) \gneq V(\hat{v}, w, z')$ *for all* $w \in S_w$,

where $\gneq$ *denotes that the inequality is strict for some values of v and w.*

If decision makers were to choose a dominated portfolio, another feasible portfolio would provide equal or higher value for all allowed realizations of weights and project scores. Decision makers should thus choose a portfolio from the set of non-dominated portfolios.

Definition 3.2.3 *For information set* $S = (S_v^\Gamma \times S_w)$ *with parameter* Γ *the set of non-dominated portfolios is*

$$Z_N(S) = \left\{z \in Z_F \mid \nexists z' \in Z_F \text{ s.t. } z' \succ_S z\right\}.$$

Reducing the deviation limit $\Gamma' \leq \Gamma$ causes fewer score deviations to be considered and reduces the score information set.

Theorem 3.2.2 *Let information set* $(S_v^\Gamma \times S_w)$ *and deviation limits* $\Gamma' \leq \Gamma$. *Then*

$$Z_N(S_v^{\Gamma'} \times S_w) \subseteq Z_N(S_v^\Gamma \times S_w).$$

The set of non-dominated portfolios can be identified by first applying the dynamic programming algorithm of Liesiö et al. (2008) to determine the set $Z_N(S_v^\infty \times S_w)$ and then using Theorem 3.2.1 to discard dominated portfolios to obtain $Z_N(S_v^\Gamma \times S_w)$, i.e.,

$$Z_N(S_v^\Gamma \times S_w) = \left\{z \in Z_N(S_v^\infty \times S_w) \mid \nexists z' \in Z_N(S_v^\infty \times S_w) \text{ s.t. } z' \succ_{(S_v^\Gamma \times S_w)} z\right\}. \quad (3.6)$$

This is since the set $Z_N(S_v^\Gamma \times S_w)$ is a subset of $Z_N(S_v^\infty \times S_w)$ for any desired robustness level Γ by Theorem 3.2.2. Another possible approach is to employ the dominance check of Theorem 3.2.1 within the dynamic programming algorithm (for details see Appendix A). Since dominance relations are less conservative, the computation can be faster, but use of this approach requires that decision makers have fixed their desired (maximum) deviation limit Γ a priori. If on the other hand the decision makers wish to choose a deviation limit $\Gamma \in [0, mn]$ ex post or examine how the set of non-dominated portfolios depends on the value of parameter Γ, the former approach is preferable.

3.3. Interactive Decision Support

The developed framework serves a basis for an interactive procedure that aids decision makers in choosing a project portfolio. Based on initial specifications of uncertain criteria weights and project scores $(S_v^\Gamma \times S_w)$, a set of non-dominated portfolios is determined for decision makers to choose from. If the set of candidate portfolios is too large to make proper portfolio choices, more conclusive decision recommendations can be obtained by considering an information subset $(S_{v'}^\Gamma \times S_{w'}) \subseteq (S_v^\Gamma \times S_w)$.

A subset of uncertain criteria weights $S_{w'} \subseteq S_w$ is obtained by imposing additional or tighter preference statements (Salo and Hämäläinen, 1995). The information set of project scores is reduced by re-evaluating most likely scores $\hat{v}'_{ji}$ and possible score deviations $\vec{v}'_{ji}$. From Liesiö et al. (2008) it follows that if scores are re-evaluated so that $S^{\infty}_{v'}$ is a subset of S^{∞}_{v} then this also holds for the sets of non-dominated portfolios resulting from these information sets. By Theorem 3.2.2 the set of non-dominated portfolios for $S^{\Gamma}_{v'}$ with arbitrary deviation limit Γ is a subset of non-dominated portfolio for S^{∞}_{v}. Finally, if most likely scores remain unchanged and only deviation limits are reduced the resulting sets of feasible scores and non-dominated portfolios are subsets of the original sets. These results are formalized by the following Theorem.

Theorem 3.3.1 *Let information sets $S = (S^{\infty}_{v} \times S_w)$, $S' = (S^{\infty}_{v'} \times S_{w'})$ with $S' \subseteq S$, $int(S) \cap S' \neq \emptyset$ and an arbitrary deviation limit Γ.*

(i) If $\hat{v}'_{ji} - \vec{v}'_{ji} \geq \hat{v}_{ji} - \vec{v}_{ji}$ and $\hat{v}'_{ji} + \vec{v}'_{ji} \leq \hat{v}_{ji} + \vec{v}_{ji}$, then $S^{\Gamma}_{v'} \subseteq S^{\infty}_{v}$ and

$$Z_N(S^{\Gamma}_{v'} \times S_{w'}) \subseteq Z_N(S^{\infty}_{v} \times S_w).$$

(ii) If $\hat{v}'_{ji} = \hat{v}_{ji}$ and $\vec{v}'_{ji} \leq \vec{v}_{ji}$, then $S^{\Gamma}_{v'} \subseteq S^{\Gamma}_{v}$ and

$$Z_N(S^{\Gamma}_{v'} \times S_{w'}) \subseteq Z_N(S^{\Gamma}_{v} \times S_w).$$

We use core indexes (Liesiö et al., 2007) to identify those projects for which more accurate information can reduce the set of non-dominated portfolios. The core index of a projects measures the share of non-dominated portfolios that include the project.

Definition 3.3.1 *For information set $S = (S^{\Gamma}_{v} \times S_w)$ core indexes $C(S) \in [0,1]^m$ are given by*

$$C_j(S) = \frac{\sum_{z \in Z_N(S)} z_j}{|Z_N(S)|} \quad j = 1, \ldots, m.$$

Projects with a core index of zero, included in none of the non-dominated portfolios, as well as projects with a core index of one, included in all non-dominated portfolios, do not aid in reducing the set of non-dominated portfolios. When reducing the information set of scores for any project j with core index $C_j(S^{\Gamma}_{v} \times S_w) \in]0,1[$, additional dominance relations may arise causing portfolios to fall out of the set of non-dominated portfolios. The following lemma formalizes this result.

Lemma 3.3.1 *Let information sets $S = (S^{\Gamma}_{v} \times S_w)$ and $S' = (S^{\Gamma}_{v'} \times S_w)$, where $S^{\Gamma}_{v'} \subset S^{\Gamma}_{v}$ such that $\hat{v}'_{ji} = \hat{v}_{ji}$ and $\vec{v}'_{ji} = \vec{v}_{ji}$ for $i = 1, \ldots, n$ and all j with $C_j(S) \in]0,1[$. Then*

$$Z_N(S') = Z_N(S)$$

By Theorem 3.2.2 the set of non-dominated portfolios can also be reduced by lowering deviation limit Γ causing dominance relations to be determined less conservatively. A tradeoff arises between precise coverage of the information set and the size of the set of non-dominated portfolios. In order to aid decision makers in choosing a suitable level of robustness Γ, we provide information on the fraction of the information set S_v^∞ taken into account when determining dominance relations between project portfolios for different deviation limits. To this end, we consider uncertain scores as random variables $\tilde{v}$. Each $\tilde{v}_{ji}$ is given by $\tilde{v}_{ji} = \hat{v}_{ji} + \eta_{ji}\vec{v}_{ji}$, with η_{ji} being identically, independently, and symmetrically distributed with support $[-1, 1]$. The probability that random scores realize within S_v^Γ is given by

$$\mathbb{P}(\tilde{v} \in S_v^\Gamma) = \mathbb{P}(\sum_{i=1}^{n}\sum_{j=1}^{m} |\eta_{ji}| \leq \Gamma) = \int_0^\Gamma (f_{|\eta_{11}|} * \ldots * f_{|\eta_{mn}|}), \tag{3.7}$$

where $*$ indicates convolutions. In cases where no additional distributional information is available, it is adequate to assume η to be uniformly distributed in the range $[-1, 1]$, i.e., $\eta_{ji} \sim \mathcal{U}(-1, 1)$ (Shakhsi-Niaei et al., 2011). The compound random variable $\sum_{i=1}^{n}\sum_{j=1}^{m} |\eta_{ji}|$ then follows an Irwin-Hall probability distribution (Irwin, 1927; Hall, 1927), and a precise probability of $\tilde{v}$ falling in the considered information set is given by

$$\mathbb{P}(\tilde{v} \in S_v^\Gamma) = \mathbb{P}(\sum_{i=1}^{n}\sum_{j=1}^{m} |\eta_{ji}| \leq \Gamma) = \frac{1}{(mn)!}\sum_{g=0}^{\lfloor\Gamma\rfloor}(-1)^g \binom{mn}{g}(\Gamma - g)^{mn}. \tag{3.8}$$

Figure 3.1 illustrates the probability $\mathbb{P}(\tilde{v} \in S_v^\Gamma)$ for η being distributed uniformly. As the exact calculation of quantity (3.8) is not computationally tractable when high numbers of parameters are considered, we approximate probabilities by Monte Carlo simulation. $1,000,000$ realizations of nm independently distributed uniform random variables with support $[-1, 1]$ have been considered. The same approach is used to obtain a good approximation of $\mathbb{P}(\tilde{v} \in S_v^\Gamma)$ for triangularly distributed $\eta_{ji} \sim \mathrm{Tr}(-1, 0, 1)$, where no general formula in line with (3.8) can be given.

For arbitrary, symmetric distributions of η, $\mathbb{P}(\tilde{v} \in S_v^\Gamma)$ can be bound using Markov's inequality

$$\mathbb{P}(\tilde{v} \in S_v^\Gamma) \geq 1 - \frac{mnE[|\eta|]}{\Gamma}. \tag{3.9}$$

The level of $E[|\eta|]$ depends on the assumed distribution of the random variable η and controls the tightness of the approximate probability bound. If the decision makers are able to specify a best, worst, and most realistic value, scores can be assumed as triangularly or beta-distributed in the domain of these three values. In case η is triangularly distributed in $[-1, 1]$, $E[|\eta|] = \frac{1}{3}$.Figure 3.1 also illustrates the lower bound (3.9) to $\mathbb{P}(\tilde{v} \in S_v^\Gamma)$ for different levels of Γ and mn.

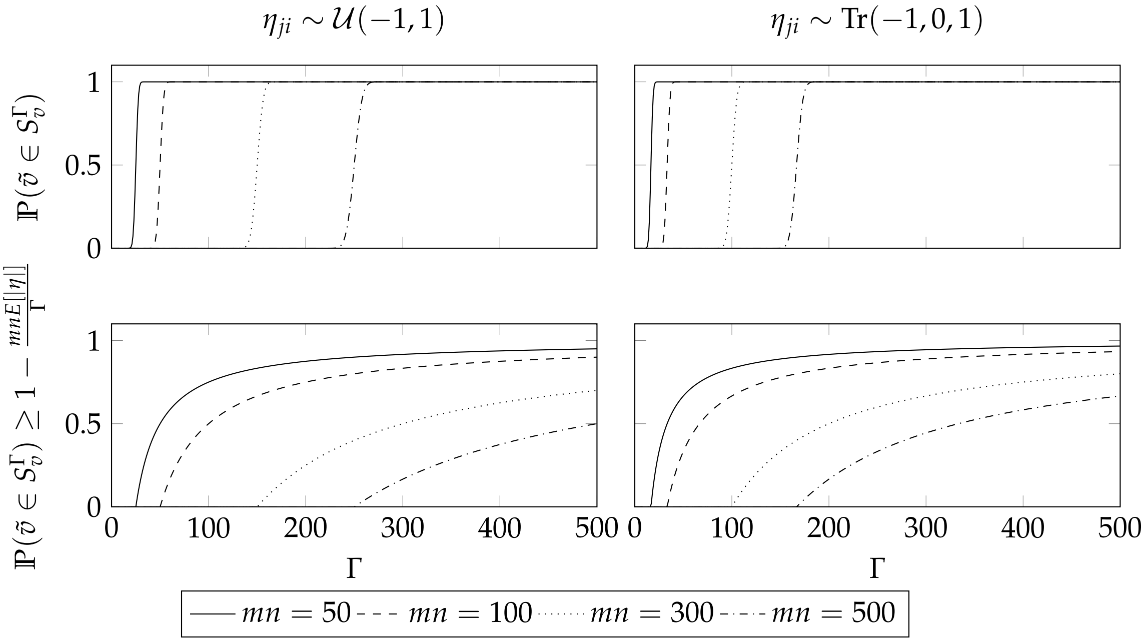


Figure 3.1.: Monte Carlo simulation and Markov bound for $\mathbb{P}(\tilde{v} \in S_v^\Gamma)$ for uniformly and triangularly distributed η, given levels of Γ, and different settings of mn

3.4. Application to the Selection of Supply Chain Improvement Projects

We illustrate the developed framework with an example based on a real life project portfolio selection problem at an international semiconductor manufacturer, where initiatives are chosen to improve supply chain performance (Kolisch et al., 2012). While the project and portfolio value model are derived from practice, project scores and decision making constraints have been adapted to illustrate the concept of RPM with adjustable robustness.

3.4.1. Project Value Model

Table 3.1 summarizes all relevant project information used in the case study. The 58 project proposals (named $P1$ to $P58$) are evaluated in terms of their impact on supply chain competitiveness, which is measured with $n = 3$ criteria. Projects may impact the speed of order processing, termed "Speed" ($i = 1$), as well as the accuracy of forecasting future customer demands, termed "Forecast Accuracy" ($i = 2$). These strategic contributions to critical success factors of the semiconductor supply chain are measured in "natural units", i.e., days ($i = 1$) and percentage points ($i = 2$). The direct financial impact of a project ($i = 3$), capturing anticipated increases in turnover or cost reductions, is measured in monetary units. The score information set S_v^∞ is given by most likely project scores $\hat{v}$ as well as possible score deviations $\vec{v}$. The criteria are aggregated by converting the strategic criteria ($i = 1, 2$) to monetary units using monetization rates, which capture decision makers' incomplete preferences. These rates are modeled by the set of feasible weights

$$S_w = \left\{ w \in [0,1]^3 \mid \sum_{i=1}^{3} w_i = 1,\ 2.6 \leq \frac{w_1}{w_3} \leq 3.0,\ 3.0 \leq \frac{w_2}{w_3} \leq 6.4 \right\}.$$

3.4.2. Portfolio Value and Constraints

Portfolio value is additive across projects up to synergies, which account for beneficial effects resulting from the joint execution of related projects (Santhanam and Kyparisis, 1996). When choosing both projects $P9$ and $P27$, their overall value increases by 20% as synergy $S1$, when choosing projects $P20$ and $P27$ synergy $S2$ amounts to 30%, and for projects $P11$ and $P31$ synergy $S3$ is 25%. Synergies are modeled through additional, artificial projects $S1$, $S2$, and $S3$, whose values amount to the proposed synergetic effects. Logical constraints ensure that these artificial projects can be selected only if the projects producing the synergy effect are also selected (Liesiö et al., 2008). Furthermore, there are 13 projects whose selection requires that some other specific projects have been included in the portfolio (cf. last column in Table 3.1), which results in 13 additional logical constraints. Finally, the aggregate cost of executed projects may not exceed a given budget level of 50 monetary units ($\sum_{j=1}^{m} c_j x_j \leq 50$).

3.4.3. Results

The problem contains $m = 61$ projects (58 real projects and 3 synergies), which are evaluated in terms of $n = 3$ criteria. 62 of the 58×3 criterion scores are non-zero and can deviate from their most likely scores. The deviation limit therefore may take on values $\Gamma \in [0, 62]$. The set of non-dominated portfolios $Z_N(S_v^\infty \times S_w)$ was computed using the dynamic programming algorithm by Liesiö et al. (2008) and then sets $Z_N(S_v^\Gamma \times S_w)$ for different levels of Γ were obtained through pairwise comparisons (3.6). The calculation of $Z_N(S_v^\infty \times S_w)$ took 54 minutes, while the pairwise comparisons to derive $Z_N(S_v^\Gamma \times S_w)$ for all levels of $\Gamma \in [0, 62]$ were performed in 27 seconds (2.5 GHz dual–core processor, 4 GB memory).

j	$\hat{v}$			$\vec{v}$			c_j	Required Projects
	$\hat{v}_{j1}$	$\hat{v}_{j2}$	$\hat{v}_{j3}$	$\vec{v}_{j1}$	$\vec{v}_{j2}$	$\vec{v}_{j3}$		
P1	0.0	0.0	0.4	0.0	0.0	0.04	0.44	
P2	0.0	35.57	0.0	0.0	0.0	0.0	1.24	
P3	0.0	0.0	48.96	0.0	0.0	48.96	1.2	
P4	0.0	0.0	56.64	0.0	0.0	56.64	0.52	
P5	0.0	0.0	29.38	0.0	0.0	29.38	16.2	
P6	0.0	0.0	112.33	0.0	0.0	74.89	51.82	P7, P8, P9, P10, P11, P28
P7	0.0	0.33	0.0	0.0	0.33	0.0	0.62	
P8	2.0	0.0	0.0	0.22	0.0	0.0	0.22	
P9	0.0	0.0	1.17	0.0	0.0	0.29	0.26	
P10	0.0	7.2	0.0	0.0	4.8	0.0	1.4	
P11	0.0	0.67	0.0	0.0	0.0	0.0	0.96	
P12	0.0	0.0	48.96	0.0	0.0	48.96	4.98	P27, P33, P34
P13	0.0	0.0	8.63	0.0	0.0	8.63	6.84	
P14	0.0	0.0	0.33	0.0	0.0	0.11	0.61	
P15	0.0	0.0	0.35	0.0	0.0	0.09	0.62	
P16	0.0	0.0	7.32	0.0	0.0	7.32	0.22	P37
P17	0.0	0.0	0.91	0.0	0.0	0.91	0.62	
P18	0.0	0.0	21.95	0.0	0.0	7.32	1.0	
P19	5.99	0.0	0.0	0.67	0.0	0.0	0.34	
P20	13.64	0.0	0.0	4.55	0.0	0.0	0.78	
P21	1.66	0.0	0.0	0.55	0.0	0.0	0.22	
P22	4.09	0.0	0.0	1.36	0.0	0.0	0.88	P8
P23	3.29	0.0	0.0	1.1	0.0	0.0	0.58	P19, P20, P21, P22, P24, P25, P26
P24	1.64	0.0	0.0	0.18	0.0	0.0	0.34	P8
P25	13.64	0.0	0.0	4.55	0.0	0.0	1.26	
P26	5.46	0.0	0.0	3.64	0.0	0.0	1.34	P27
P27	0.0	0.0	27.29	0.0	0.0	9.1	2.48	P28
P28	0.0	0.0	27.29	0.0	0.0	9.1	1.7	P29
P29	0.0	0.0	17.56	0.0	0.0	11.71	2.3	
P30	0.0	0.0	2.73	0.0	0.0	0.91	0.38	
P31	0.0	0.67	2.08	0.0	0.22	0.69	1.18	P34
P32	0.0	0.0	29.28	0.0	0.0	7.32	2.14	
P33	0.0	0.0	18.19	0.0	0.0	18.19	12.86	
P34	0.0	0.0	1.23	0.0	0.0	0.82	0.64	P37
P35	0.0	0.0	0.14	0.0	0.0	0.05	0.74	P31
P36	0.0	0.0	0.67	0.0	0.0	0.22	0.22	
P37	6.55	0.0	1.64	0.73	0.0	0.18	0.34	
P38	0.0	0.0	0.8	0.0	0.0	0.09	0.22	
P39	0.0	0.0	1.06	0.0	0.0	0.27	0.22	
P40	0.0	0.0	1.8	0.0	0.0	0.6	0.22	
P41	0.0	0.0	1.66	0.0	0.0	0.55	0.22	P37
P42	0.0	0.0	8.08	0.0	0.0	2.69	0.22	
P43	0.0	0.0	2.84	0.0	0.0	0.95	0.22	
P44	0.0	0.0	3.33	0.0	0.0	1.11	1.34	
P45	0.0	0.0	0.67	0.0	0.0	0.22	0.22	
P46	0.0	0.0	6.65	0.0	0.0	2.22	1.6	
P47	0.0	0.0	0.33	0.0	0.0	0.11	0.12	
P48	0.0	0.0	1.0	0.0	0.0	0.33	0.1	
P49	0.0	0.0	0.33	0.0	0.0	0.11	0.08	
P50	0.0	0.0	0.33	0.0	0.0	0.11	0.16	
P51	0.0	0.0	0.4	0.0	0.0	0.13	0.1	
P52	0.0	0.0	0.4	0.0	0.0	0.13	0.05	
P53	0.0	0.0	0.4	0.0	0.0	0.13	0.1	
P54	0.0	0.0	0.4	0.0	0.0	0.13	0.1	
P55	0.0	0.0	0.4	0.0	0.0	0.13	0.2	
P56	0.0	0.0	0.4	0.0	0.0	0.13	0.38	
P57	0.0	0.0	0.4	0.0	0.0	0.13	0.1	
P58	0.0	0.0	0.4	0.0	0.0	0.13	0.1	
S1	0.0	0.0	5.46	0.0	0.0	1.82	0.0	P9, P27
S2	4.09	0.0	8.19	1.36	0.0	2.73	0.0	P20, P27
S3	0.0	0.0	0.55	0.0	0.0	0.19	0.0	P11, P31

Table 3.1.: Project and synergy scores, project cost, and logical constraints

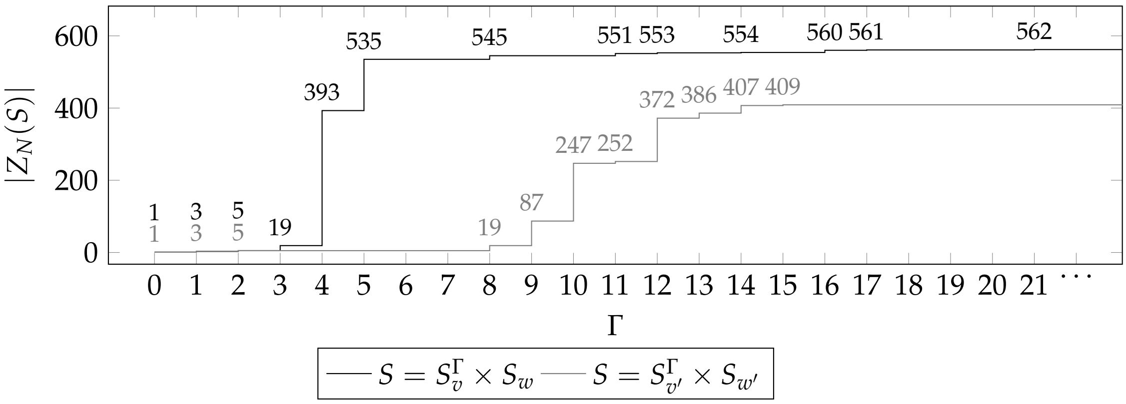


Figure 3.2.: Number of non-dominated portfolios for given levels of Γ

Figure 3.2 reports the number of non-dominated portfolios for different levels of Γ. When comparing portfolios based on most likely scores alone ($\Gamma = 0$), one optimal portfolio exists. With increasing levels of robustness, additional portfolios become non-dominated up to a maximum of $|Z_N(S_w \times S_v^\Gamma)| = 562$ for $\Gamma \geq 21$.

Portfolios non-dominated for $\Gamma \geq 21$ are composed of 34 to 55 projects (41 on average). Pairs of portfolios differ by between 2 and 23 projects (7 on average). When comparing portfolios in terms of dominance, deviations of up to Γ project scores whose most likely realization is larger than zero are taken into account. If Γ is increased beyond the number of project scores with most likely realizations greater than zero, dominance relationships between portfolios do no longer change. Therefore, the set of non-dominated portfolios obtains its maximum size for roughly two thirds of the value range of Γ.

The left side of Figure 3.3 presents the sets of uncertain scores for all non-dominated portfolios $Z_N(S_v^\infty \times S_w)$. Most likely overall portfolio values, indicated by crosses, are 72 for Speed, 45 to 47 for Forecast Accuracy, and 454 to 459 for Financial Impact. All portfolios have at least worst-case overall values of 62 in terms of Speed, 42 in terms of Forecast Accuracy, and 306 in terms of Financial Impact.

Applying probability bounds discussed in Section 3.3, Monte Carlo simulation gives the probability of covering a particular realization of uniformly distributed project scores $\tilde{v}$ for $\Gamma = 21$ as $\mathbb{P}(\tilde{v} \in S_v^\Gamma) \approx 4 \cdot 10^{-6}$ and for triangularly distributed project scores as $\mathbb{P}(\tilde{v} \in S_v^\Gamma) \approx 0.58$. Decision makers wishing to reduce the set of non-dominated portfolios may either reduce robustness, i.e., the probability of taking into account a particular realization of $\tilde{v}$, or they may try to provide revised project scores $S_{v'}^\infty$ and preferences $S_{w'}$ so that $S_{v'}^\infty \times S_{w'} \subset S_v^\infty \times S_w$.

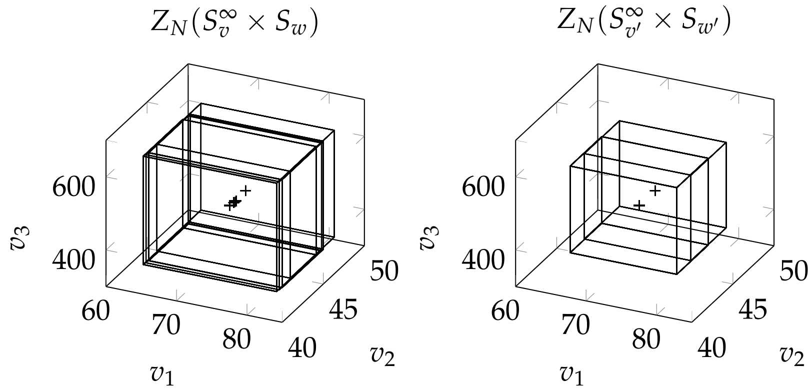


Figure 3.3.: Score information set of non-dominated portfolios

$S_{v'}^\infty$ and $S_{w'}$ are obtained by reducing all score deviations by 20%, leaving most likely scores unchanged, and adjusting preferences so that both Speed and Forecast Accuracy are regarded 3.0 times as important as financial impact. By Theorem 3.3.1 (ii), $Z_N(S_{v'}^\Gamma \times S_{w'})$ can be derived from $Z_N(S_v^\Gamma \times S_w)$ through pairwise comparisons. Figure 3.2 reports the number of non-dominated portfolios and the average number of projects included in each non-dominated portfolio. Starting with one non-dominated portfolio for $\Gamma = 0$, the set grows to 409 portfolios for $\Gamma \geq 15$. The right side of Figure 3.3 indicates that non-dominated portfolios have at least a worst-case overall value of 63 in terms of Speed, 43 in terms of Forecast Accuracy, and 339 in terms of financial impact.

Figure 3.4 illustrates project and synergy core indexes for both the original and reduced information set for different levels of Γ. The figure shows that by reducing Γ or transitioning from information set $S_v^\infty \times S_w$ to $S_{v'}^\infty \times S_{w'}$ all core projects ($C_j(S_w \times S_v^\Gamma) = 1$) and exterior projects ($C_j(S_w \times S_v^\Gamma) = 0$) are maintained. As for $\Gamma = 0$ only one non-dominated portfolio remains, all borderline projects gradually either become core or exterior projects. For the original information set, 29 projects have a core index of one for all levels of Γ. When transitioning from information set $S_v^\infty \times S_w$ to $S_{v'}^\infty \times S_{w'}$, 6 additional projects become core projects for all levels of Γ. Projects P5, P6, and P35 are exterior projects for all levels of Γ. Score information regarding projects that are core or exterior projects invariant to Γ, i.e., are core or exterior projects with regards to $S_w \times S_v^\infty$, does not influence the set of non-dominated portfolios. Therefore, decision makers should focus on the remaining 24 borderline projects if they wish to further refine the information set $S_{v''}^\infty \times S_{w''} \subset S_{v'}^\infty \times S_{w'}$.

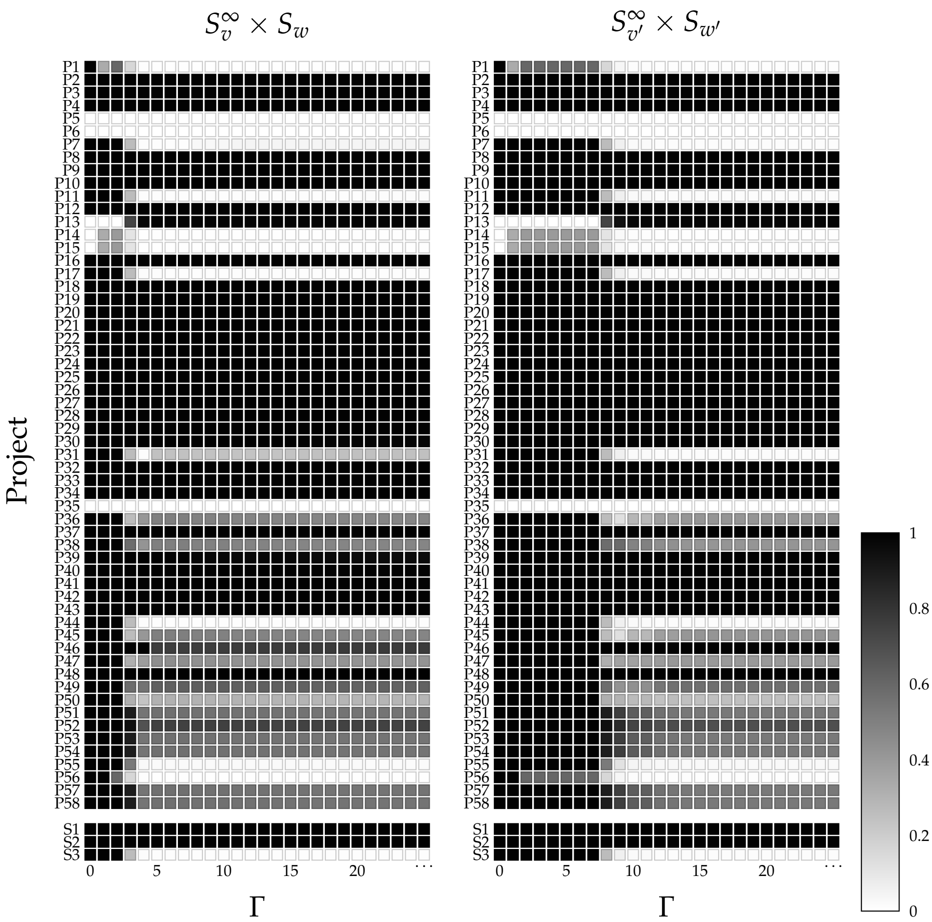


Figure 3.4.: Project core indexes for given levels of Γ

3.5. Extensions to the Decision Making Framework

We present two extensions of the basic decision making framework to accommodate two aspects common to practical decision making environments, interdependencies of uncertain project scores, and uncertainty with regards to constraint coefficients.

3.5.1. Modeling Interdependence of Uncertain Project Scores

The decision making framework from Section 3.2 assumes that project scores take on any value within the information set independent of one another. Frequently, project scores cannot be regarded as completely independent; for instance, project proposals within the same organization share technological or organizational properties that can cause their scores to manifest jointly within their specified value ranges (Loch and Kavadias, 2002; Gregory et al., 2011; Hall, 2012). Such interdependencies can be modeled through constraints on the score information set.

Definition 3.5.1 *An adjustable uncertainty set for project scores subject to interdependence is given by*

$$I_v^\Gamma = \left\{ v \in S_v^\infty \mid v_{ji} = \hat{v}_{ji} + \vec{v}_{ji} y_{ji}, A^I Y \leq B^I, y \in [-1,1]^{m \times n}, \sum_{i=1}^{n} \sum_{j=1}^{m} |y_{ji}| \leq \Gamma \right\},$$

where $Y \in [-1,1]^{mn}$ is a vector containing all entries y_{ji} of Matrix y, $A^I \in \mathbb{R}^{r \times mn}$, and $B^I = (b_1^I, \ldots, b_r^I)^T \in \mathbb{R}^r$.

The matrix of constraint coefficients A^I facilitates co- and diametric movement of scores across different criteria as well as different projects. For example, scores in different criteria for one project can be modeled to deviate jointly as their realizations are both tied to the development of the project. Perfect correlation of two scores ji and $j'i'$ is modeled by two constraints $[A^I]_k = [a_{11}^I, \ldots, a_{mn}^I]$ and $[A^I]_{k'} = [a_{11}^{I\,\prime}, \ldots, a_{mn}^{I\,\prime}]$, where $a_{ji}^I = a_{j'i'}^{I\,\prime} = 1$, $a_{j'i'}^I = a_{ji}^{I\,\prime} = -1$, $b_k^I = b_{k'}^I = 0$. Furthermore, it is possible to model joint deviation of the scores of different projects, which makes sense if these projects are interdependent in their execution. We only require that constraints $A^I Y \leq B^I$ induce a valid polytope within the space $[-1,1]^{mn}$ such that I_v^Γ is a non-empty set.

The conditions for portfolio dominance from Theorem 3.2.1 can be adjusted to consider score uncertainty sets with interdependence by substituting

$$\beta^I(\vec{v}, w, z, z', \Gamma) = \max_{y \in [-1,1]^{m \times n}} \left\{ \sum_{i=1}^{n} \sum_{j=1}^{m} w_i \vec{v}_{ji} y_{ji} (z_j{}' - z_j) | A^I Y \leq B^I, \sum_{i=1}^{n} \sum_{j=1}^{m} |y_{ji}| \leq \Gamma \right\} \tag{3.10}$$

for $\beta(\vec{v}, w, z, z', \Gamma)$. The substitution does not affect the validity of Theorem 3.2.1 or the properties given in Lemma 3.2.1. Interdependency constraints only limit the solution polytope of optimization problem $\beta^I(\vec{v}, w, z, z', \Gamma)$ compared to $\beta(\vec{v}, w, z, z', \Gamma)$ while all properties of continuity and convexity are maintained.

Constraining the realizations of decision variables y in I_v^Γ rule out the most extreme realizations within the score information set S_v^Γ. Therefore, the score information set considering interdependence is a subset of the information set without interdependence $I_v^\Gamma \subseteq S_v^\Gamma$ and the set of non-dominated portfolios $Z_N(I_v^\Gamma \times S_w)$ is a subset of $Z_N(S_v^\Gamma \times S_w)$.

Lemma 3.5.1 *Let preference information set S_w, score information set S_v^Γ, and score information set I_v^Γ, limited by constraints $A^I Y \leq B^I$ such that $I_v^\Gamma \neq \{\emptyset\}$. Then*

$$Z_N(I_v^\Gamma \times S_w) \subseteq Z_N(S_v^\Gamma \times S_w).$$

3.5.2. Modeling Uncertain Constraint Coefficients

Not only decision maker preferences and project scores may be subject to uncertainty but also decision making constraints. Previous project portfolio selection approaches have considered uncertainty with regards to project cost (Gutjahr and Froeschl, 2013; Liesiö et al., 2008), available budget (Li, 2009), and the achievement of aspiration levels for overall portfolio value with regards to different score criteria (Hall et al., 2009).

Uncertain coefficients in constraints, which determine the set of feasible portfolios Z_F, are modeled in line with Bertsimas and Sim (2003) and Bertsimas and Sim (2004). Constraint coefficients are given by a matrix $A \in \mathbb{R}^{q \times m}$, where for each coefficient a_{lj} the most likely realization is $\hat{a}_{lj}$ and realizations may deviate from this value by at most $\vec{a}_{lj}$. The number of coefficients that may deviate from their most likely value is limited by parameter Γ^A.

Definition 3.5.2 *The set of feasible portfolios assuming uncertain constraint coefficients is given by*

$$Z_F^{\Gamma^A} = \left\{ z \in \{0,1\}^m \mid Az \leq B \quad \forall A \in S_A^{\Gamma^A} \right\},$$

where

$$S_A^{\Gamma^A} = \left\{ A \in \mathbb{R}^{q \times m} \mid a_{lj} = \hat{a}_{lj} + \vec{a}_{lj} y_{lj}, y \in [-1,1]^{q \times m}, \sum_{j=1}^{m} |y_{lj}| \leq \Gamma^A \quad l = 1, \ldots, q \right\}$$

is the information set of uncertain constraint coefficients for deviation limit $\Gamma^A \in [0, m]$.

The set of feasible portfolios Z_F^Γ when considering coefficient uncertainty set $S_A^{\Gamma^A}$ can be directly derived from Theorem 1 of Bertsimas and Sim (2003).

Lemma 3.5.2 *For a given constraint coefficient information set $S_A^{\Gamma^A}$ parameterized with Γ^A the set of constraint-feasible portfolios is given by*

$$Z_F^{\Gamma^A} = \left\{ z \in \{0,1\}^m \mid \hat{A}z + \beta(\vec{A}, z, \Gamma^A) \leq B \right\},$$

where $\beta(\vec{A}, z, \Gamma^A) = [\beta_1(\vec{A}, z, \Gamma^A), \ldots, \beta_q(\vec{A}, z, \Gamma^A)]^T$ *is given by*

$$\beta_l(\vec{A}, z, \Gamma^A) = \max_{y \in [0,1]^q} \left\{ \sum_{j \in J(z)} \vec{a}_{lj} y_j | \sum_{j=1}^{m} y_j \leq \Gamma^A \right\},$$

with $J(z) = \{j \in \{1, ..., m\} \ | z_j = 1\}$.

In extension of Definition 3.2.3, the set of non-dominated portfolios assuming uncertain constraint coefficients is given by

$$Z_N^{\Gamma^A}(S) = \left\{ z \in Z_F^{\Gamma^A} \mid \nexists z' \in Z_F^{\Gamma^A} \text{ s.t. } z' \succ_S z \right\}. \quad (3.11)$$

The set $Z_N^{\Gamma^A}(S)$ can be determined by adjusting the dynamic programming algorithm given in Appendix A to check portfolio feasibility while taking into account constraint robustness. $\beta(\vec{A}, z, \Gamma^A)$ hereby constitutes q continuous knapsack problems, which are solvable in linear time (Martello and Toth, 1990). For an arbitrary level of constraint robustness Γ^A, some non-dominated portfolios $z \in Z_N(S)$ may be rendered infeasible ($z \notin Z_F^{\Gamma^A}$) and thus are not viable options for decision makers when taking into account constraint robustness ($z \notin Z_N^{\Gamma^A}(S)$). On the other hand, feasible portfolios $z' \in Z_F^{\Gamma^A}$ that are dominated by a portfolio $z \succ z', z \notin Z_F^{\Gamma^A}$ in turn may be rendered non-dominated when taking into account constraint robustness ($z' \in Z_N^{\Gamma^A}(S)$).

Interpreting the constraint matrix $\tilde{A}$ as independent random variables $\tilde{a}_{lj}$, symmetrically distributed with most likely value $\hat{a}_{lj}$, support $[\underline{a}_{lj}, \overline{a}_{lj}]$, and half-range $\vec{a}_{lj} = \overline{a}_{lj} - \hat{a}_{lj} = \hat{a}_{lj} - \underline{a}_{lj}$ Bertsimas and Sim (2003) and Bertsimas and Sim (2004) derive three bounds for the probability of a constraint being feasible in dependence of the deviation parameter. These probability bounds may aid the decision makers in choosing a desired level of Γ^A.

3.6. Supply Chain Improvement Example Revisited

In order to illustrate the developed extensions, we expand the model developed for selecting supply chain improvement projects. First, we assume that for each project proposal the three criteria scores deviate jointly within their relative value ranges. This relationship can be accounted for by constraints $y_{ji} - y_{ji'} \leq 0 \quad i, i' \in \{1, \ldots, n\}, i \neq i', j = 1, \ldots, m$. Furthermore, score deviations for synergy projects are assumed to be given by the average deviation of their enabling projects. This corresponds to constraints $\frac{1}{|\mathcal{E}_j|} \sum_{j' \in \mathcal{E}_j} y_{j'i} - y_{ji} \leq 0$ and $-\frac{1}{|\mathcal{E}_j|} \sum_{j' \in \mathcal{E}_j} y_{j'i} + y_{ji} \leq 0 \quad \forall j \in \mathcal{S}, i = 1, \ldots, n$. Set

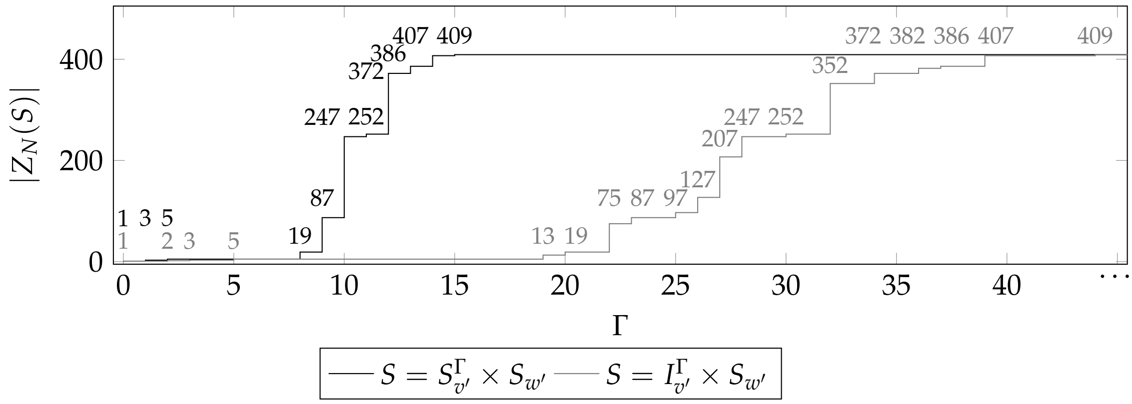


Figure 3.5.: Number of non-dominated portfolios for given levels of Γ

$\mathcal{S}$ denotes synergy projects (S1, S2, S3) and set $\mathcal{E}_j$ contains all projects required for the execution of project j.

Imposing these constraints $A^I Y \leq B^I$ on score information set $S^{\Gamma}_{v'}$ results in the score information set with interdependence constraints $I^{\Gamma}_{v'}$. By Lemma 3.5.1, we derive $Z_N(I^{\Gamma}_{v'} \times S_{w'})$ from $Z_N(S^{\Gamma}_{v'} \times S_{w'})$ through pairwise comparisons. As $\beta^I(\vec{v}, w, z, z', \Gamma)$ given by Equation (3.10) constitutes a non-trivial linear programming problem, the pairwise comparisons require considerably more computation time than when deriving $Z_N(S^{\Gamma}_{v} \times S_{w})$ and $Z_N(S^{\Gamma}_{v'} \times S_{w'})$. For all levels of Γ, $Z_N(I^{\Gamma}_{v'} \times S_{w'})$ was determined in 22 minutes.

Figure 3.5 compares the number of non-dominated portfolios $Z_N(I^{\Gamma}_{v'} \times S_{w'})$ to $Z_N(S^{\Gamma}_{v'} \times S_{w'})$ for different levels of Γ. Both sets of non-dominated portfolios converge to a set comprising 409 portfolios, with $Z_N(I^{\Gamma}_{v'} \times S_{w'})$ increasing in size more slowly and reaching its maximum size only at $\Gamma \geq 44$.

Second, we assume that the decision makers want to ensure minimum achievement levels $v^{asp} \in \mathbb{R}^n_+$ for each score criterion $i = 1, \ldots, n$ (Stummer and Heidenberger, 2003; Kleinmuntz, 2007). Aspiration levels are modeled by portfolio constraints $\sum_{j=1}^{m} -v_{ji} \geq -v_i^{\text{asp}} \quad i = 1, \ldots, n$, with project scores deviating from their most likely realization $\hat{v}_{ji}$ by at most $\vec{v}_{ji}$. Constraint feasibility is ensured for up to Γ^A project scores deviating from their most likely realization.

Recalling Figure 3.3, non-dominated portfolios in sets $Z_N(S^{\Gamma}_{v'} \times S_{w'})$ and $Z_N(I^{\Gamma}_{v'} \times S_{w'})$ obtain most likely scores of 72, 45 to 47, and 454 to 459 with regards to Speed, Forecast Accuracy, and financial impact respectively. Scores take on values of 62, 42, and 306 in the worst case. If decision makers require project portfolios to have a score of 60, 35, and 345 with regards to Speed, Forecast Accuracy, and financial impact, some non-dominated portfolios $z \in Z_N(I^{\Gamma}_{v'} \times S_{w'})$ are rendered infeasible when considering worst-case scores and constraint coefficients.

For increasing levels of constraint robustness Γ^A, we employ the dynamic programming algorithm to determine $Z_N^{\Gamma^A}(I^{\infty}_{v'} \times S_{w'})$ whenever a previously non-dominated portfolio is rendered infeasible. For arbitrary dominance robustness Γ, $Z_N^{\Gamma^A}(I^{\Gamma}_{v'} \times S_{w'})$ is derived through pairwise comparisons. In our computational setup the calculation of $Z_N^{\Gamma^A}(I^{\Gamma}_{v'} \times S_{w'})$ for $\Gamma \in [0, 62]$ and $\Gamma^A \in [0, 61]$ took roughly 9 hours overall.

Figure 3.6 reports the number of non-dominated portfolios $Z_N^{\Gamma^A}(I^{\Gamma}_{v'} \times S_{w'})$ for different levels of dominance robustness and robustness for aspiration level constraints, Γ and Γ^A. For levels of constraint robustness $\Gamma^A < 12$, aspiration level constraints do not impact feasibility of non-dominated portfolios. $Z_N^{\Gamma^A}(I^{\Gamma}_{v'} \times S_{w'})$ remain valid as presented in Figure 3.5 with a maximum size of 409. For higher levels of constraint robustness, the maximum size of the set of non-dominated portfolios changes in a non-monotonous fashion with increasing constraint robustness. For $\Gamma \geq 32$ the set of non-dominated portfolios remains unchanged with a maximum size of 27.

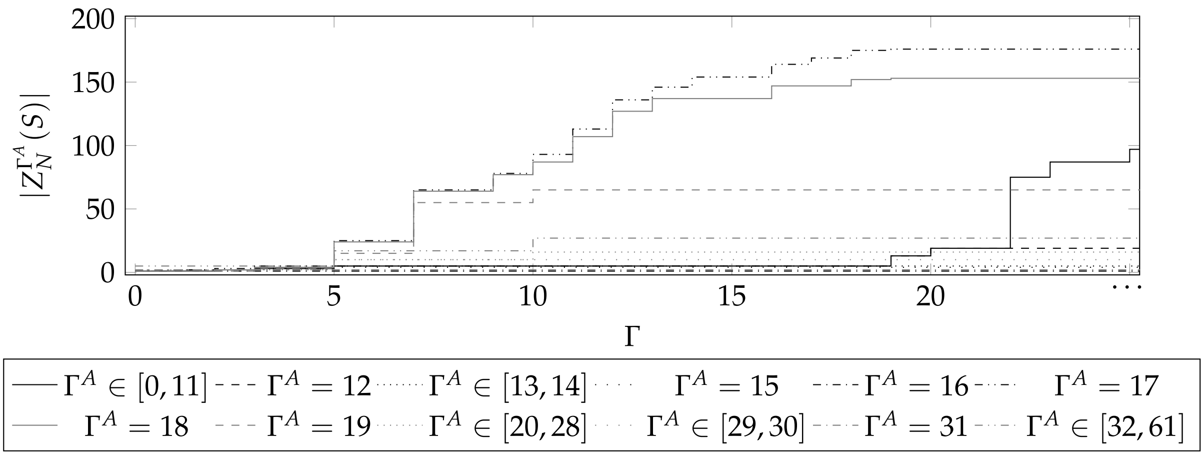


Figure 3.6.: Number of non-dominated portfolios for given levels of Γ and Γ^A

3.7. Conclusion and Outlook

In this chapter we have developed new methods for modeling uncertainty in multi-criteria project portfolio selection problems. Specifically, the developed methods allow the decision makers to control the level of conservatism employed in determining dominance relations among portfolios. Judgments can be based on most likely scores only or they can be made in line with Liesiö et al. (2007, 2008), where scores can assume any value within their intervals. We also showed how these methods can be used to model interdependencies among the possible realizations of projects' scores and capture uncertainties in coefficients of portfolio feasibility constraints. The developed methods can be used for interactive decision support. Decision makers can reduce the set of non-dominated portfolios by reducing potential score deviations through more precise preference statements about the importance of different criteria and by adjusting the level of conservatism.

Our work suggests ample opportunities for future research. For instance, Poss (2013, 2014) argues that the parameter measuring the level of conservatism should actually depend on the solution whose robustness is analyzed to ensure the same fixed probability of constraint feasibility or reaching aspiration levels for all solutions. The dominance concept proposed in this chapter could be extended to facilitate solution dependent conservatism when comparing project portfolios with varying numbers of unique projects. In our model project core indexes aid decision makers in revising uncertain project scores by identifying borderline projects, for which more precise score information can reduce the number of non-dominated portfolios. However, additional research is needed to develop methods for identifying those scores that are likely to have the greatest impact on the set of non-dominated portfolios. Finally, the developed models should be extensively evaluated in terms of their application in practical decision making settings in areas such as project portfolio selection (Heidenberger and Stummer, 1999), new product development (Loch and Kavadias, 2002), and supplier selection (Ho et al., 2010).

4. Human Behavior in Project Portfolio Selection: Insights from an Experimental Study

Selecting a subset from a discrete set of alternatives subject to various constraints is an ubiquitous problem in socio-economic decision making (Kleinmuntz, 2007). Research in the area of "Portfolio Decision Analysis" (Salo et al., 2011) has brought forth a wide range of quantitative approaches to provide guidance for such problems. Frequently, decision problems in supplier selection (Ho et al., 2010), new product development (Loch and Kavadias, 2002) and project portfolio selection (Heidenberger and Stummer, 1999) are considered. In contrast to numerous scientific publications on the subject, quantitative decision support approaches have only seen limited practical application (Booker and Bryson, 1985; Cooper et al., 2001; Loch, 2000). Unique decision making environments, difficulties in evaluating projects and decision maker preferences, as well as the strategic nature of decision problems cause practitioners to rely on management expertise rather than utilizing elaborate quantitative decision support approaches (Kester et al., 2009; Martinsuo, 2013). Thus, responsibility for portfolio decisions with grave impact often lies with human decision makers alone, who have been shown to behave irrationally in various decision environments (Bendoly et al., 2010). For this reason, Salo et al. (2011) emphasize the need for research on the occurrence, the impact, and the avoidance of decision biases in portfolio decision making settings. Kavadias (2014) underlines the potential of experimental research for the domain of project management in general.

In this chapter we address human behavior in project portfolio selection. Based on the knapsack problem, a generic and controllable problem setting, we design an experimental framework where subjects may dynamically select and deselect from a list of items to build their desired portfolio. Our framework allows us to study both subjects' decision quality as well as their selection process.

Our goal is to raise awareness of caveats of human decision making in project portfolio selection by investigating decision biases. It is well known that the application of decision heuristics in complex problem settings can result in systematic errors with serious implications (Gino and Pisano, 2008). We therefore strive to obtain greater understanding of decision maker heuristics in order to aid the development of debiasing strategies and effective decision support more compatible with human decision making (Gigerenzer and Selten, 2001).

To the best of our knowledge, no previous study addresses behavioral heuristics and biases in project portfolio selection. Fasolo et al. (2011) review experimental and empirical studies on behavioral issues in portfolio decision making. Gingrich and Soli (1984) investigate human decision making and suboptimization in a basic resource allocation problem, in which variable amounts of resources have to be assigned to a set of alternatives. In a similar setting, Busemeyer et al. (1986) study subjects' learning behavior as well as the effect of giving subjects feedback on their performance. Langholtz et al. (1993) examine a multi-period resource-allocation problem under certainty, risk, and uncertainty while Langholtz et al. (1994, 1995) study how subjects cope with possible resource breakdowns and abundance of resources. Langholtz et al. (1997) consider a three-dimensional resource allocation problem that is solvable by integer programming, for which Ball et al. (1998) examine decision making strategies using a verbal protocol analysis technique. Gonzalez et al. (2002) examine resource-allocation problems, where the goal is to achieve a fixed objective while minimizing resource consumption. Most recently, Gettinger et al. (2013) and Killen (2013) focus on the effect of different visualization techniques in support systems for portfolio decision making.

The remainder of this chapter is structured as follows. Section 4.1 introduces the knapsack problem setting as well as our hypotheses on human decision making in this environment. Based on an experimental framework introduced in Section 4.2, we set up two experimental studies and discuss the results in Section 4.3. We conclude this chapter with potential extensions and managerial implications in Section 4.4.

4.1. Decision Making Behavior in the Knapsack Problem

The knapsack problem (Martello and Toth, 1990)

$$\max_{x \in \{0,1\}^m} \left\{ v^T x | k^T x \leq c \right\} \tag{4.1}$$

considers a set of items $j = 1, \ldots, m$ with vector $v \in \mathbb{R}_+^m$ indicating value and vector $k \in \mathbb{R}_+^m$ indicating required resources of each item. Binary decision variables $x \in \{0, 1\}^m$ indicate the selection ($x_j = 1$) or exclusion ($x_j = 0$) of item $j = 1, \ldots, m$. The objective is to choose a subset of items of maximum sum of values, termed portfolio value, while the sum of required resources, termed portfolio resource requirement, must not exceed resource capacity $c \in \mathbb{R}$. While the knapsack problem is a NP-hard optimization problem, dynamic programming approaches exist to solve it in pseudo-polynomial time (Martello and Toth, 1990).

When facing difficult tasks, decision makers frequently reach the limits of their cognitive capacity (Loch and Wu, 2007) and do not solve problems to optimality. We expect decision makers to solve the knapsack problem suboptimally even in small problem instances with two-digit numbers of alternatives, which are common in real-life portfolio decision making (Golabi et al., 1981; Loch et al., 2001; Grushka-Cockayne et al., 2008; Gurgur and Morley, 2008). Furthermore, we expect that decision makers will not overcome suboptimization through learning by repetition.

Decision makers have been found to apply simple heuristics in many problem settings (Gino and Pisano, 2008; Gans and Croson, 2008; Bendoly et al., 2010), e.g., the secretary problem (Seale and Rapoport, 1997), the newsvendor problem (Schweitzer and Cachon, 2000), or revenue management (Bearden et al., 2008). In real-life project portfolio selection, decision makers typically prioritize projects based on their value and resource requirement (Koç et al., 2009; Salo et al., 2011). Advanced evaluation metrics consider the difference between value and resource requirements, such as the net present value (Heidenberger and Stummer, 1999), or the ratio of value and resource requirements, following the "value for money" principle (Keisler, 2004; Phillips and Bana e Costa, 2007; Lourenço et al., 2012). We conjecture that decision makers apply similar heuristics in the knapsack problem as well.

In iterative steps $s = 1, \dots, n$ a "constructive heuristic" for the knapsack problem selects item $h(\mathcal{A}_s)$, ranked highest according to an evaluation criterion, from the set of items $\mathcal{A}_s$ that have not been previously selected and the selection of which does not exceed the resource capacity. Items can be ranked by maximum value (MaxV), minimum resource requirement (MinK), maximum ratio of value to resource requirement (MaxR), or maximum difference between value and resource requirement (MaxD).

$$h^{\text{MaxV}}(\mathcal{A}_s) = \arg\max_{j \in \mathcal{A}_s} \{v_j\} \tag{4.2}$$

$$h^{\text{MinK}}(\mathcal{A}_s) = \arg\min_{j \in \mathcal{A}_s} \{k_j\} \tag{4.3}$$

$$h^{\text{MaxR}}(\mathcal{A}_s) = \arg\max_{j \in \mathcal{A}_s} \{v_j / k_j\} \tag{4.4}$$

$$h^{\text{MaxD}}(\mathcal{A}_s) = \arg\max_{j \in \mathcal{A}_s} \{v_j - k_j\}\,. \tag{4.5}$$

Iterations stop when the selection of any remaining unselected item would exceed the resource capacity. Constructive heuristics thus terminate with "complete portfolios", whose value cannot be increased by selecting any additional unselected item without violating the capacity constraint. We define the "construction phase" as a decision maker's selection process until the first complete portfolio is achieved. In contrast to the constructive heuristics, the human decision making process can involve deselection steps, through which decision makers might further adjust a complete portfolio built in the construction phase. We define this phase of decision making after establishing a first complete portfolio as the "improvement phase".

To investigate whether people rely on constructive heuristics in the knapsack problem, we formulate hypotheses for the MaxV, MinK, MaxR and MaxD heuristics.

> *H1: The selection process of decision makers during the construction phase is based on a) the MaxV heuristic, b) the MinK heuristic, c) the MaxR heuristic, or d) the MaxD heuristic.*

Constructive heuristics require sorting all available items, with a theoretical worst-case complexity of $O(m^2)$ (Knuth, 1968). Sorting items is less demanding to a decision maker's cognitive system than solving a knapsack problem to optimality through portfolio enumeration but becomes increasingly difficult with growing instance size. Ericcson et al. (1980) emphasize the limited capacity of human short-term memory placing constraints on the ability to process information for problem solving. While Miller (1994) claims that the limit on the capacity for processing information is about seven elements, Cowan (2001), reviewing a wide range of studies, proposes a limit of four. Although quantifying human mental capacity is matter of debate, there is no doubt that only a limited amount of information can be bound into one functional context (Jonides et al., 2008). For the knapsack problem, we assume that decision makers' ability to keep track of all available items is limited. We distinguish between ideal "global selection behavior", which considers all available items, and "localized selection behavior", which due to cognitive limitations considers only a subset of available items. For small problem sizes, this subset might be equal to the complete set of items but we expect that the impact of localized selection behavior increases with increasing problem size.

> *H2: Decision makers apply localized selection behavior.*

In order to investigate our research hypotheses, we develop an experimental framework to observe human decision making processes and selection performance when solving the knapsack problem in a laboratory setting.

4.2. Experimental Framework

During an experimental session, lasting 35 minutes, subjects are asked to solve a series of knapsack problems. For each problem subjects are given a list of items, their values and resource requirements, as well as the available capacity. Subjects may freely select and deselect items from the list of items. The current portfolio value and remaining resource capacity is communicated after each decision. Subjects are informed if they try to select an item whose resource requirement exceeds the remaining capacity and the attempted selection is denied. Subjects are provided with a calculator as well as pen and paper.

After being exposed to any knapsack problem for 1 minute, subjects are free to irrevocably proceed to the next problem and are required to do so at latest after 5 minutes. Preliminary studies have shown that 5 minutes is enough so that subjects do not perceive any time pressure for the kinds and sizes of problems we consider while ensuring that all subjects are presented with at least seven knapsack problems during a 35-minute session. The sequence in which knapsack problems are presented is prespecified and identical for all subjects.

Money is the only incentive offered ensuring that differences in selection performance result in clear differences in payout. At the end of the experiment, for each subject one solved knapsack problem is randomly drawn to determine the payout. The value of the last portfolio selected for this problem is converted to € using a problem-specific conversion factor communicated to subjects while they solve the knapsack problem. A subject's payout is obtained by reducing the converted portfolio value by a fixed charge of € 100. The conversion factors and fixed charge are chosen so that subjects can achieve a maximum payout of € 20 when solving the drawn knapsack problem to optimality. All subjects receive at least a show-up fee of € 3.

Before the experimental session begins, subjects are asked to read instructions, given in Appendix C. Subjects are furthermore presented with three "training" knapsack problems consisting of 25 items, which they are asked to solve within 5 minutes each. Training problems are not considered in the incentive scheme. At the end of the session, participants are asked to fill out a short questionnaire and are informed about their performance for each problem as well as the resulting payout.

We present subjects with challenging knapsack problems, neither overwhelming their mental capacity nor presenting trivial problems. Pisinger (2005) evaluates different solution procedures for knapsack problems of varying difficulty. In a computational study, the author considers between 50 and 10,000 items. Such dimensions are not adequate for investigating human decision making due to limited cognitive capacity as well as limited time in experimental settings. We examine considerably smaller problems consisting of between 5 and 25 items. Solution times of algorithms increase with growing number of available items, increasing number range of values and resource requirements, as well as with higher correlation of item values and resource requirements. Knapsack problems, whose values and resource requirements are independently drawn from the same range $[1, r]$, are termed "uncorrelated" problems. "Weakly correlated" problems are obtained by randomly drawing resource requirements from the range $[1, r]$ and sampling the value of each item from the reduced range $[\max(1, k_j - r/10), k_j + r/10]$. For "strongly correlated" problems the value is fixed to $v_j = k_j + r/10$. We opt to use weakly correlated problems in our experiments as Pisinger (2005) argues that they represent real-world knapsack problems in the most realistic way. All values and resource requirements fall into the range [1, 1,000] and values are higher than resource requirements for all items.

Smith-Miles and Lopes (2012) characterize knapsack problems by their "constraint slackness", the ratio of available budget to the sum of the resource requirements of all items. For slackness levels close to zero, only a few items may be selected without violating the budget restriction while for slackness levels close to one almost all items may be selected and the problem arises which items not to select. Such problems are assumed to be easier to solve while Chvátal (1980) proposes slackness ratios of around 0.5 for difficult problems. In line with Chvátal, we consider knapsack problems with slackness levels between 0.4 and 0.6. At these levels knapsack problems have the highest number of complete portfolios limiting subjects' possibilities to achieve an optimal solution by random selection.

In order to distinguish decision making behavior clearly, we ensure that each problem has a unique optimal solution and that the four constructive heuristics lead to different, unique, and non-optimal solutions. In order to ensure unique selection according to heuristics, no two items within a problem may have the same value, resource requirement, difference between value and resource requirement, or ratio of value divided by resource requirement. The order in which items are presented to subjects is random and identical for all subjects.

4.3. Experimental Studies

Based on the experimental framework introduced in Section 4.2, two experimental studies were performed at the laboratory "experimenTUM" of TUM School of Management. The experiments were programmed and conducted with the software z-tree (Fischbacher, 2007) and were administrated using the software ORSEE (Greiner, 2004).

Problem	1	2	3	4	5	6	7	8	9	...
m	5	10	15	25	15	25	15	25	25	...
Repeated Problem	-	-	-	-	3	4	3	4	-	...
Scaling Factor	-	-	-	-	$\frac{1}{3}$	$\frac{1}{2}$	$\frac{2}{3}$	$\frac{3}{2}$	-	...
Random Order	-	-	-	-	✓	✓	✓	✓	-	...

Table 4.1.: Specifications for the knapsack problems of Study 1

Study 1 provides general insights into the performance of decision makers, their learning behavior, as well as how the performance changes with the problem size. Furthermore, we derive initial results regarding the use of heuristics. Based on these findings we investigate adherence of selection steps to constructive heuristics as well as localized selection behavior in Study 2.

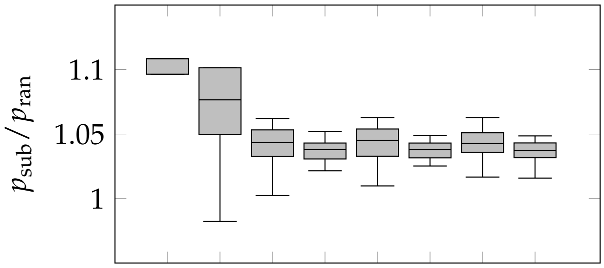
4.3.1. Experimental Protocol of Study 1

Table 4.1 provides the details of the knapsack problems used in Study 1. Subjects are presented with knapsack problems consisting of 5, 10, 15, and 25 items. To investigate learning behavior, problems 5 and 7 are repetitions of problem 3 with 15 items and problems 6 and 8 repeat problem 4 with 25 items. To prevent subjects from noticing repetitions, all item values, resource requirements, and capacities are multiplied by a "scaling factor" preserving integrity of all values. Furthermore, the order in which items are presented is randomly changed each time a problem is repeated. After Problem 8, subjects are presented with a series of unique problems consisting of 25 items until the session ends after 35 minutes.

The study has been conducted with 29 undergraduate business students in two separate experimental sessions. Including time to read instructions, time for three training problems, as well as time to fill out the questionnaire, each session took approximately 60 minutes. At the end of the sessions, subjects were paid in private earning on average €10.97 with a standard deviation of €3.28 including a show-up fee of €3.00.

4.3.2. Results of Study 1

Our analysis focuses on knapsack problems 1 to 8, which were solved by all subjects. Out of 232 "decision making processes", i.e., attempts to solve a knapsack problem, five are excluded from our analysis as subjects had advanced to the next problem without having selected a complete portfolio.



Problem	1	2	3	4	5	6	7	8
m	5	10	15	25	15	25	15	25
Repeated Problem	–	–	–	–	3	4	3	4
% Solved Optimally	59%	31%	7%	0%	4%	0%	4%	0%

Figure 4.1.: Box-and-whisker plots of $p_{\text{sub}}/p_{\text{ran}}$ for all subjects and problems 1 to 8

We measure the quality of subjects' final portfolio choice using the ratio of the obtained portfolio value p_{sub} to p_{ran}, the expected portfolio value if items are randomly selected until a complete portfolio is obtained. p_{ran} is determined by sampling 10,000 complete portfolios with Monte Carlo simulation. In a scheme similar to the constructive heuristics introduced in Section 4.1, unselected items whose selection does not exceed the resource capacity are chosen at random until a complete portfolio is obtained. Figure 4.1 reports the distribution of the ratio $p_{\text{sub}}/p_{\text{ran}}$ for all subjects. Outliers are omitted for reasons of clarity. Average ratios $p_{\text{sub}}/p_{\text{ran}}$ are higher than 1 showing that subjects' selection approach is better than random choice. However, in line with our expectation subjects solve knapsack problems suboptimally. Already for problem 1, consisting of only 5 items, 41% of subjects are unable to find the optimal solution. With increasing problem size, suboptimality becomes even more prominent. While few subjects succeed in finding the optimal solution for the problems with 15 items, problems consisting of 25 items are not solved to optimality by any subject.

Suboptimization is maintained on a similar level throughout the experiment. We assess learning behavior across all subjects through linear regression of $p_{\text{sub}}/p_{\text{ran}}$ and focus on problems 3, 5, and 7 ($m = 15$) as well as 4, 6, and 8 ($m = 25$) where subjects predominantly show suboptimal behavior. Slope parameter values and R^2-values close to zero, $b_1 = 0.00$ ($R^2 = 0.02$) for $m = 15$ and $b_1 = 0.00$ ($R^2 = 0.01$) for $m = 25$, demonstrate that in the short run subjects do not improve by repeatedly solving a knapsack problem.

In our experimental framework decision makers may select and deselect items at will giving them the opportunity to revise previously made decisions. In Figure 4.2 we analyze selection and deselection steps for the construction phase, i.e., steps leading to a subject's first complete portfolio, and the subsequent improvement phase, i.e., all

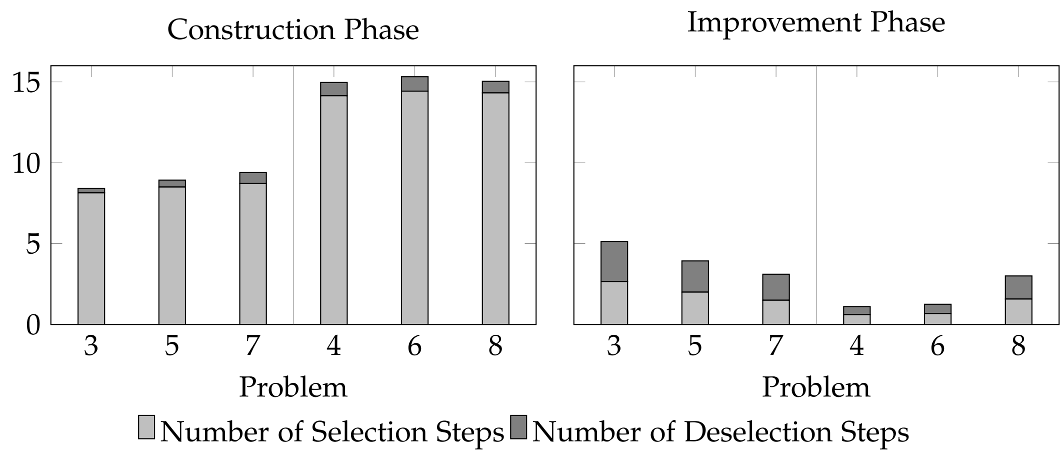


Figure 4.2.: Average number of selection and deselection steps in the construction and the improvement phase

steps undertaken after a complete portfolio has been obtained. For problems with $m = 15$ the construction phase has more than twice as many steps as the improvement phase. While during the construction phase more selection steps are performed than deselection steps, this relationship is almost equalized in the improvement phase. This implies that on average after a complete portfolio has been obtained one item is removed from the portfolio in order to add a new one. For larger problems with $m = 25$ the relationships between selection and deselection steps in the construction and improvement phase are similar.

Analyzing decision making patterns of selection and deselection steps, we find that decision makers frequently annul previous selection or deselection steps. In an "annulment pattern" two subsequent steps consider the same item either first selecting and immediately deselecting an item or deselecting and immediately reselecting an item. Both patterns can coincide if, for example, an item is selected, deselected, and immediately reselected. Table 4.2 gives the percentage of selection and deselection steps which are associated with annulment patterns for the construction and improvement phase. For all considered problems less than 18% of the selection steps and more than 76% of the deselection steps during the construction phase can be explained by annulment. This is in line with our expectation that decision makers follow a constructive heuristic during the construction phase, where deselection steps are only undertaken to correct erroneously selected items. In contrast, in the improvement phase between 45% and 77% of steps are associated with annulment patterns.

Constructive heuristics (4.2) - (4.5) iteratively select items ranked highest according to an evaluation criterion. Subjects conjectured to adhere to these heuristics frequently

Problem	m	Construction Phase		Improvement Phase	
		Selection	Deselection	Selection	Deselection
3		15.52	100.00	59.09	65.15
5	15	13.57	90.00	45.10	67.35
7		17.04	87.50	52.94	69.23
4		11.18	88.89	66.67	76.92
6	25	8.21	76.19	61.11	75.00
8		9.44	91.67	54.55	66.67

Table 4.2.: Percentage of steps associated with annulment patterns within the construction and the improvement phase

Heuristic	$m = 15$			$m = 25$		
	b_1	R^2	p	b_1	R^2	p
MaxV	0.02	0.06	$= 0.398$	0.01	0.02	$= 0.475$
MinK	-0.02	0.09	$= 0.283$	-0.01	0.06	$= 0.231$
MaxR	-0.06	0.72	< 0.001	-0.04	0.74	< 0.001
MaxD	-0.06	0.79	< 0.001	-0.04	0.79	< 0.001

Table 4.3.: Regression statistics for selection frequencies dependent on items' ranks according to the four evaluation criteria

select items with high rank according to the evaluation criteria as well. Figure 4.3 reports the relative frequency of items being included in subjects' complete portfolios depending on items' rank according to the evaluation criteria of the four heuristics. Subjects' first complete portfolios at the end of the construction phase as well as their final complete portfolios at the end of the improvement phase are considered. We compare subjects' behavior to random selection behavior. To this end, we determine the relative frequencies of items being included in all possible complete portfolios. We find that subjects choose highly ranked items and omit low-ranked items for the MaxR, and MaxD heuristics more often while for the MaxV and MinK heuristics no rank dependent differences between random selection and subjects' portfolio choices can be observed. A regression analysis provided in Table 4.3 verifies that subjects put stronger emphasis on items with high ratio of value to resource requirement and high difference between value and resource requirement in the construction phase. Regression line slopes b_1 smaller than zero are statistically significant for both criteria.

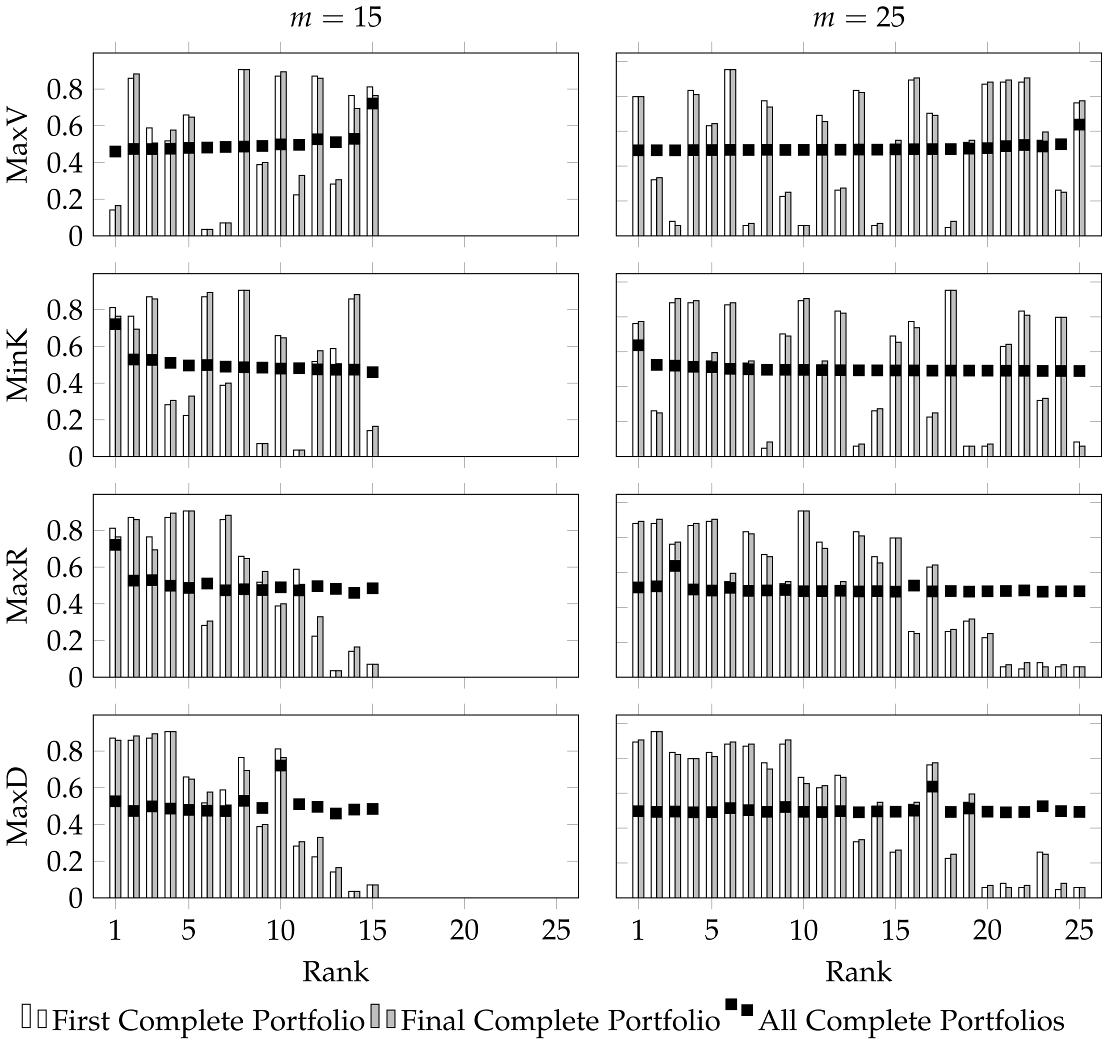


Figure 4.3.: Selection frequency of the ith highest ranked item for all complete portfolios as well as subjects' first and final complete portfolios

Problem	1	2	3	4	5	6	7	8	9	...
m	25	25	25	25	25	25	25	25	25	...
Repeated Problem	-	-	-	-	2	1	3	4	-	...
Scaling Factor	-	-	-	-	$\frac{1}{3}$	$\frac{1}{2}$	3	2	-	...
Reversed Order	-	-	-	-	✓	✓	✓	✓	-	...

Table 4.4.: Specifications for the knapsack problems of Study 2

4.3.3. Experimental Protocol of Study 2

While we focus on the composition of final complete portfolios in Study 1, we specifically investigate subjects' selection process and the use of heuristics in the construction phase in Study 2. Table 4.4 provides the details of the knapsack problems used in Study 2. In order to extend the number of selection steps in the construction phase, we consider a new set of knapsack problems consisting of 25 items. We investigate whether the sequence in which items are presented to subjects affects decision making by repeating the first four problems with reversed order in which items are presented. To prevent subjects from noticing the repetition, item values, resource requirements, and budgets are multiplied by a scaling factor preserving integrity of all values. After having finished problem 8, subjects are presented with a series of unrelated knapsack problems until the experiment ends after 35 minutes.

Study 2 was performed with 53 undergraduate business students in three separate sessions excluding participants of Study 1. Sessions lasted approximately 60 minutes and subjects earned €10.48 on average with a standard deviation of €3.39 including a show-up fee of €3.00.

4.3.4. Results of Study 2

Our analysis focuses on problems 1 to 8. Out of in total 424 attempts to solve these knapsack problems, 23 are excluded from the analysis because no complete portfolio was achieved during the construction phase. Study 1 has shown that most deselection steps in the construction phase can be explained by annulment patterns, which reflect reconsidered decisions not constructive decision making. In order to focus on systematic portfolio development, we exclude steps associated with items that are selected but are deselected later on. Out of 5,924 selection and deselection steps for all considered problems and subjects, we exclude 620 steps in the construction phase.

We investigate subjects' adherence to a constructive heuristic by measuring the relative frequency of a subject selecting items in line with the heuristic. For each step $s = 1, \ldots, n$ during the construction phase of solving a knapsack problem,

$$\alpha(s) = \begin{cases} 1, & \text{if} \quad j_s = h(\mathcal{A}_s) \\ 0 & \text{else} \end{cases} \tag{4.6}$$

indicates whether the item $j_s \in \{1, \ldots, m\}$ selected by a subject is chosen in line with heuristic $h(\cdot)$, given by (4.2) - (4.5). In each step heuristics consider all available items $\mathcal{A}_s$, i.e., items that have not been selected in steps $s = 1, \ldots, s-1$ and the selection of which does not cause the resource capacity to be exceeded. Absolute and relative heuristic adherence are given by

$$A_{\text{abs}} = \sum_{s=1}^{n} \alpha(s) \text{ and} \tag{4.7}$$

$$A_{\text{rel}} = \frac{A_{\text{abs}}}{n}. \tag{4.8}$$

If a subject selects all items exactly in line with a heuristic, then $A_{\text{rel}} = 1$. If a subject ignores items chosen by a heuristic in every single step, then $A_{\text{rel}} = 0$ holds. In Figure 4.4 we compare the distribution of A_{rel} for subjects' decision making with the distribution resulting from randomly selecting items. Distributions are aggregated for all considered problems and subjects. As in our first experiment, random selection behavior is approximated by Monte Carlo simulation with sample size 10,000. Subjects do not completely adhere to one single heuristic during the construction phase as A_{rel} values are strictly smaller than one. The degree of subjects' heuristic adherence for the MaxR as well as the MaxD heuristic is higher than for random selection behavior. No significant difference can be found for the MaxV and MinK heuristic.

Heuristic adherence, formalized by hypotheses *H1a* - *H1d*, has been statistically tested considering absolute adherence A_{abs}. We find no support for Hypothesis *H1a*, adherence to the MaxV heuristic, as there is no significant difference between A_{abs} for subjects' selections and random behavior across all problems (one-tailed Mann-Whitney, $p = 0.121$). We find weak support for adherence to the MinK heuristic, Hypothesis *H1b*. Although subjects' adherence is significantly higher than for random selection (one-tailed Mann-Whitney, $p < 0.001$), this difference is only significant for two out of eight problems under consideration, as presented in Table 4.5. Hypothesis *H1c*, adherence to the MaxR heuristic, is confirmed. Across all problems and in each problem separately, subjects' behavior is significantly more often in line with this heuristic than would be expected for random behavior (one-tailed Mann-Whitney, $p < 0.001$). We find support for adherence to the MaxD heuristic, Hypothesis *H1d*. Across all problems the absolute adherence is significantly higher than for random selection (one-tailed Mann-Whitney, $p < 0.001$). While the effect is only mildly significant for problem 3 (one-tailed Mann-Whitney, $p = 0.003$), there is strong significance for the other 7 problems (one-tailed Mann-Whitney, $p < 0.001$).

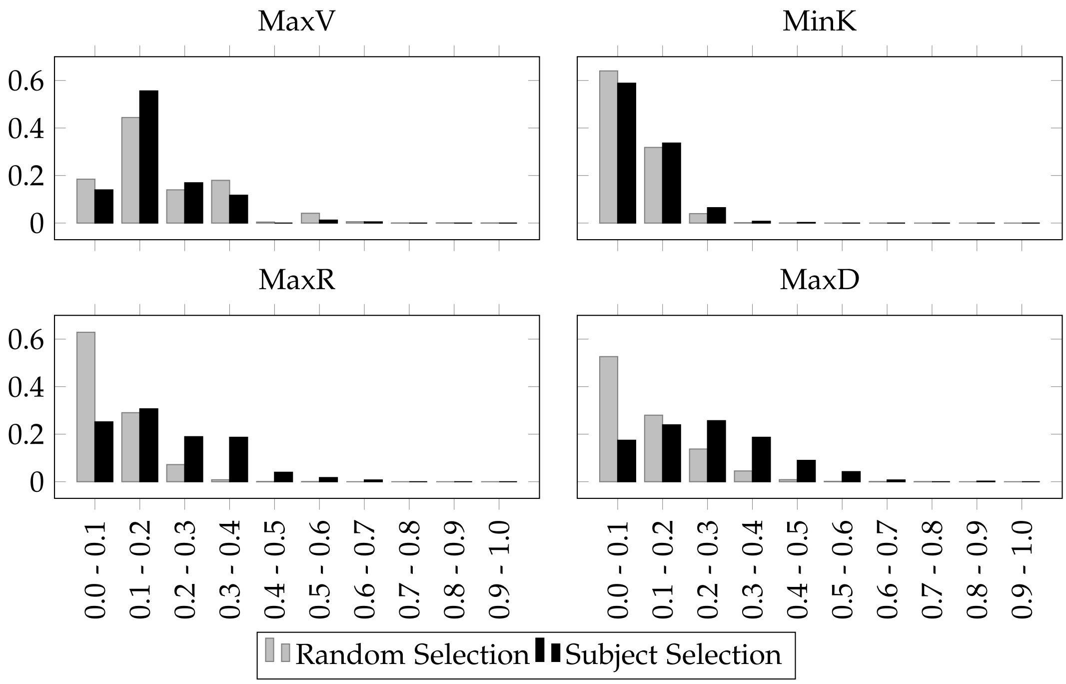


Figure 4.4.: Histogram of relative adherence to heuristic selection by subjects and in case of random selection

Problem							
1	2	3	4	5	6	7	8
$p < 0.001$	$p = 0.496$	$p = 0.197$	$p = 0.016$	$p = 0.464$	$p < 0.001$	$p = 0.081$	$p = 0.319$

Table 4.5.: One-tailed p-values of the Mann-Whitney test for the MinK heuristic

Overall, we find significant support that decision makers prefer to select items with the highest ratio or the highest difference. Nevertheless, selections are not consistently in line with the MaxR or the MaxD heuristic as across all problems the highest adherence of a subject to any heuristic is $A_{\text{rel}} = 0.41$. To obtain insights into these deviations, we investigate whether the order in which items are presented to subjects impacts their decision making.

Items are numbered in order in which they are presented to subjects. Item $j = 1$ is presented to subjects first, e.g., at the top of a list, while item $j = 25$ is presented last, e.g., at the bottom of a list. For two consecutive selection steps $s - 1$ and s, we define the "selection span" as the number of omitted available items listed between items j_{s-1} and j_s. Figure 4.5 presents the histogram of subjects' selection spans across all considered problems in comparison to the selection spans expected for random selection behavior.

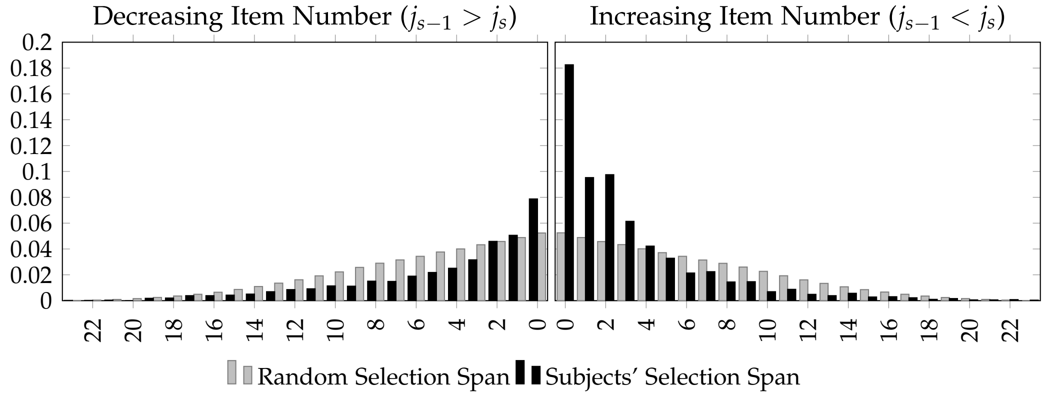


Figure 4.5.: Histogram of selection spans for consecutive selection steps

We consider the cases that the first selected item has a lower number than the second item ($j_{s-1} < j_s$), i.e., the decision maker has moved from top to bottom within the item list, or that the item number has decreased ($j_{s-1} > j_s$), i.e., the decision maker has moved from the bottom to the top. More than 50% of all selection spans are smaller than or equal to 3, almost twice as much as expected for random selection behavior (Binomial Test, $p < 0.001$). For each knapsack problem, subjects' selection span is significantly smaller than for random selection (one-tailed Mann-Whitney, $p < 0.001$) confirming Hypothesis *H2*. Subjects prefer to select items in close proximity to the previously selected item while large spans are underrepresented. Furthermore, Figure 4.5 indicates that subjects move along the item list from top to bottom more frequently than from the bottom to the top.

Figure 4.6 illustrates the distribution of j_s, the item number selected by subjects for the first ten selection steps ($s = 1, \ldots, 10$) over all problems. During the first five selection steps, j_s has a significantly positive trend (Jonckheere trend test, $p < 0.001$). This trend disappears for the following five selection steps (Jonckheere trend test, $p = 0.547$). Subjects are likely to have traversed the complete item list at that point. We conclude that subjects initially are biased toward items at the top of the item list and only later consider items from the bottom. To assess whether the order of items also influences the final portfolio, we compare portfolios resulting from the construction phase for problems 1 to 4 with the following identical problems with reversed order of items. There is no significant difference between the selected items for each pair of problems, i.e., problems 1 and 6 (two-tailed Mann-Whitney, $p = 0.889$), problems 2 and 5 (two-tailed Mann-Whitney, $p = 0.936$), problems 3 and 7 (two-tailed Mann-Whitney, $p = 0.271$), as well as problems 4 and 8 (two-tailed Mann-Whitney, $p = 0.725$).

We investigate whether subjects' selections are better explained by the MaxR or MaxD

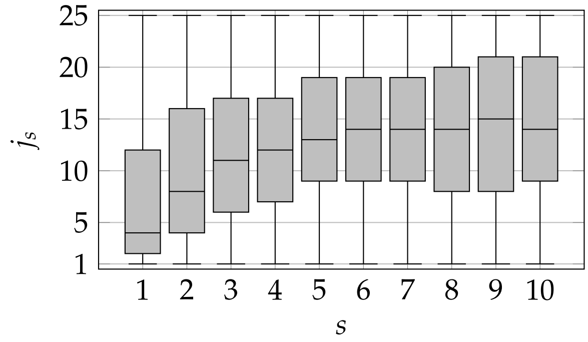


Figure 4.6.: Box-and-whisker plots representing the item numbers of the selected items in selection steps 1 to 10

heuristic when accounting for localized selection behavior. For each selection step

$$\alpha^{b,f}(s) = \begin{cases} 1, & \text{if} \quad j_s = h(\mathcal{A}_s^{b,f}(j_{s-1})) \\ 0 & \text{else} \end{cases} \tag{4.9}$$

indicates whether the item $j_s \in \{1, \dots, m\}$ selected by a subject is chosen in line with heuristic $h(\cdot)$, given by (4.2) - (4.5). Heuristics only consider available items $\mathcal{A}_s^{b,f}(j_{s-1}) \subseteq \mathcal{A}_s, s = 2, \dots, n$ in proximity of the previously selected item j_{s-1}. Items with an item number $j < j_{s-1}$ are considered if their selection span compared to j_{s-1} is no greater than b, while items with an item number $j > j_{s-1}$ are considered if their selection span is no greater than f. Relative heuristic adherence considering localized selection behavior is given by

$$A_{\text{rel}}^{b,f} = \frac{\sum_{s=1}^{n} \alpha_s^{b,f}}{n}. \tag{4.10}$$

Figure 4.7 reports average $A_{\text{rel}}^{b,f}$ values considering the MaxR and MaxD heuristic for different parameters b and f across all problems and subjects. The figure illustrates that subjects' selection process coincides more with a heuristic with small selection spans b and f. Average adherence to the MaxR and MaxD heuristics is 0.22 and 0.21 when not considering localized selection behavior ($b = f = \infty$). Limiting subjects' selection spans to $b = 2$ and $f = 3$ causes average adherence to the MaxR heuristic to increase to 0.32 while adherence to the MaxD heuristic rises to 0.30. An incomplete search pattern within a small range explains subjects' selection process better than global selection behavior.

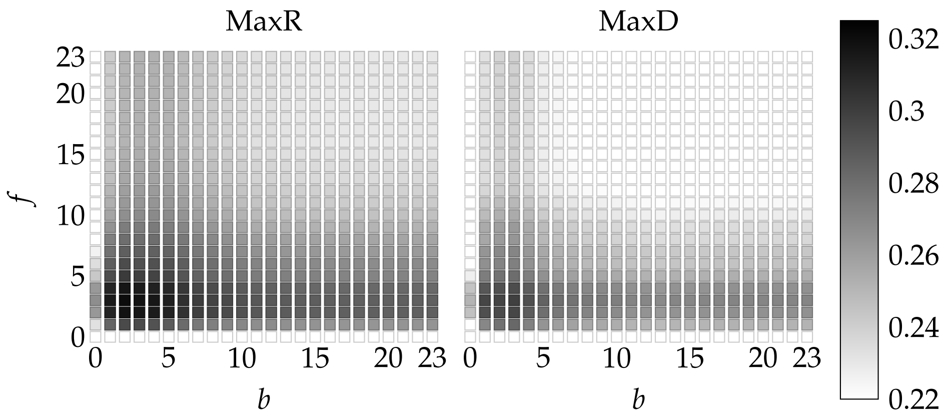


Figure 4.7.: Average $A_{\mathrm{rel}}^{b,f}$ values considering the MaxR and MaxD heuristic for given parameters b and f

4.4. Conclusion and Outlook

Managers are responsible for portfolio decisions in strategic environments, where the application of quantitative decision support is limited. Chapter 2 as well as Kolisch et al. (2012) report on an industrial application to provide decision aid for project portfolio selection within the supply chain function of an international semiconductor manufacturer. While decision recommendations are derived by solving an integer programming problem embedded in a decision support system for visualization and interactivity, final portfolio decisions are made by managers based on a listing of available projects. Understanding fallacies of human decision making in such settings enables organizations to design decision processes and support systems in order to counteract adverse effects of decision biases and heuristic decision making.

Study 1 emphasizes the limits of human decision making by showing that decision makers behave suboptimally in the abstract setting of our experiments even when accounting for learning behavior. We confirm the conjecture that human decision making first focuses on selecting alternatives to construct an initial portfolio during a construction phase, which serves as a baseline solution for further improvement. The improvement phase is dominated by annulment patterns and has limited impact on decision quality. Motivated by portfolio selection practice, we investigate subjects' adherence to simple constructive heuristics during the construction phase, which consider the value and resource requirement of alternatives. Study 1 and in more detail Study 2 provide evidence that subjects' behavior is partially explained by adherence to the MaxR heuristic, selecting items according to the maximum ratio of value divided by resource requirement, as well as the MaxD heuristic, considering the maximum difference between value and resource requirement. Study 2 demonstrates that subjects' selection processes are only partially in line with a construction heuristic due to limitations on the amount of information that decision makers are able to receive, process, or retain. Subjects compare and select items "locally" based on a limited subset of the set of presented items in close vicinity to previously selected items. They start at the top of the presented list of items and gradually move through it. Accounting for localized selection behavior increases the explanatory power of the MaxR and MaxD heuristics.

Our work suggests ample opportunities for future research. We investigate learning effects by considering three identical but rescaled and rearranged problem instances. We infer that the complexity of the problem hinders learning in the short run but make no projections regarding long-term learning behavior. Further investigations on learning effects as well as training would be worthwhile in order to overcome the difficulty that people will realize when the same instance has been rescaled and rearranged too often. As people are able to improve their memory span (Ericcson et al., 1980), decision quality can improve through training due to less localized selection behavior. Other factors besides localized selection behavior might prevent decision makers from strictly adhering to constructive heuristics. Subjects might try to apply a heuristic but fail at comparing and sorting items with regard to the evaluation criterion. Also, decision makers might apply more complex heuristics that combine evaluation criteria of the four basic heuristics treated in this study.

Operational research aims to help people in problem solving and in order to come up with better results the problem owners should not be neglected (Hämäläinen et al., 2013). Therefore, future research most importantly should provide guidance in predicting critical decision making environments and provide decision makers with debiasing methods to handle them appropriately. One promising research avenue would be to investigate whether knapsack problems where heuristics lead to suboptimal performance coincide with a worse performance of decision makers compared to instances where the heuristics lead to good or even optimal solutions. If this is the case, particular critical situations could be predicted in advance. The baseline knapsack problem can be extended in various ways taking into account e.g., uncertainty, multiple objectives, group decisions, project dependencies, and so on. Our experimental framework can be easily adapted to address these topics while experimental results can be compared to our findings as a baseline. We believe that human behavior in portfolio decision making, whose various facets are all of great practical relevance, is a promising field for research.

5. Performance and Robustness of Priority Policies for Static and Dynamic Project Scheduling under Uncertainty

Project managers are frequently confronted with scarce resources, e.g., limited budgets, limited availability of skilled workforce, or limited access to required tools and infrastructure. Scarce project resources are allocated subject to considerable uncertainty in terms of activity durations, varying resource requirements and supplies, or changes in the content or structure of the entire project. Furthermore, the majority of organizations do not execute projects in isolation but have to support several projects simultaneously (Payne, 1995; Lova et al., 2000).

In this chapter we consider a dynamic project scheduling problem where projects with stochastic activity durations arrive stochastically over time. Every arrival of a new project changes the scheduling environment and makes (partial) rescheduling advisable. Dynamic project scheduling was first addressed by Adler et al. (1995), who model the processing of R&D projects as a queuing network. The authors report on a simulation study that investigates the impact of different factors on the average project flow time such as problem parameters, the number and pooling of resources, as well as limitations on the arrival of new projects in the environment. Anavi-Isakow and Golany (2003) also analyze the performance of different policies that limit the number of projects in the environment. Choi et al. (2007) model the resource-constrained project scheduling problem (RCPSP) with stochastic activity durations and dynamic project arrival as a Markov decision process. A Q-learning-based approach is employed to heuristically determine policies for activity initiation, cancellation of ongoing projects, as well as resource reservation for future projects. Likewise, Melchiors and Kolisch (2010) employ a Markov decision process model, which is solved by value iteration, while Melchiors and Kolisch (2009) investigates different priority rules. In addition, dynamic scheduling of activities with uncertain processing times has been considered frequently within the domain of machine scheduling (Ouelhadj and Petrovic, 2009; Vredeveld, 2012).

Unlike previous approaches to dynamic scheduling, we propose to adapt solution procedures originally developed for the stochastic resource-constrained project scheduling problem (SRCPSP) (Möhring et al., 1984, 1985) to the dynamic setting of stochastically arriving projects. The SRCPSP is a direct adaption of the well-known deterministic resource-constrained project scheduling problem to the context of uncertainty regarding activity durations. A solution to the SRCPSP is not a concrete schedule, but a scheduling policy. A policy is applied dynamically during project execution to decide on activity starting times based on the realized durations of finished activities and duration estimates for pending activities. Möhring and Stork (2000) develop exact branch-and-bound procedures working with earliest start, linear preselective, and activity-based priority policies. Ballestín (2007) reports on a sampling heuristic and genetic algorithm, while Ballestín and Leus (2009) present a greedy randomized adaptive search procedure, both employing activity-based priority policies. Ashtiani et al. (2011) introduce the class of pre-processor policies, a combination of earliest start policies and resource-based priority policies. Deblaere et al. (2011) propose resource-based policies with release times as a new scheduling policy class, which are chosen by a simulation-based descent procedure. Recently, Fang et al. (2015) develop an estimation of distribution metaheuristic to determine resource-based priority policies for the SRCPSP.

In this chapter SRCPSP solution procedures are used to provide multi-project scheduling policies, which are executed until the arrival of a new project requires rescheduling. Frequent rescheduling renders dynamic project scheduling problems computationally challenging even by standards of the NP-hard RCPSP (Blazewicz et al., 1983). For this reason, the first half of this chapter is dedicated to the evaluation of different approaches for the SRCPSP in order to assess their suitability for the dynamic scheduling problem. Based on these results, we adapt specific approaches to the dynamic scheduling setting and compare their performance in a second computational study.

Section 5.1 formally introduces the dynamic project scheduling problem, for which we propose a solution approach in Sections 5.1.1 and 5.1.2. We report on two computational studies in Section 5.2. The first study considers the SRCPSP and investigates the performance and stability of two classes of scheduling policies. The second study investigates the performance of the dominant scheduling policy class for the dynamic scheduling problem.

5.1. Problem Setting

We consider a dynamic project scheduling problem where projects arrive stochastically over time according to a Poisson arrival process with arrival rate λ. Each arriving project consists of activities $i = 1, \ldots, n$ connected by precedence relations $(i, i') \in \mathcal{E}$, where $i, i' \in \{1, \ldots, n\}$. Each activity requires a non-negative amount of any resource $k = 1, \ldots, m$ during its execution giving a matrix of resource requirements $r \in \mathbb{N}^{n \times m}$. Resources are available in constant maximum amounts $R \in \mathbb{N}^m$ per period shared by all projects. When a project arrives in the environment, only the activities, their precedence relations, and resource requirements are known with certainty. Information on activity durations is given by distinct, mutually independent probability distributions $\mathbf{d} = (\mathbf{d}_1, \ldots, \mathbf{d}_n)^T$. The objective is to minimize the average expected flow time of projects, given by the difference between the completion time of a project and the time of its arrival.

A special case of this problem is a single project arriving at time $t = 0$. Then the problem boils down to the stochastic resource-constrained project scheduling problem (Möhring and Stork, 2000). For solving the SRCPSP, a scheduling policy π defines how to gradually build a project schedule during execution of the project while actual realizations of activity durations are revealed when an activity is completed. When considering activity durations as random variables $\mathbf{d}$, activity starting times, completion times, and any regular performance measure $\kappa(\pi, \mathbf{d})$ are random variables as well. The SRCPSP is typically solved by finding a scheduling policy π^* that minimizes the expected project makespan within a computationally tractable class of scheduling policies Π. Due to computational limitations (Hagstrom, 1988), solution approaches to the SRCPSP typically resort to simulation in order to approximate project makespan distributions and determine the expected value.

5.1.1. Proposed Solution Approach

We propose to employ scheduling policies generated by solution procedures for the SRCPSP to schedule projects in the dynamic project scheduling problem. We make use of the fact that between arrivals of two subsequent projects the dynamic environment is an extension of the SRCPSP setup considering multiple projects combined in one network. Whenever a new project arrives, solution procedures adapted from SRCPSP approaches are used to generate a scheduling policy for all unexecuted project activities in the environment. The policy is generated under the objective of minimizing the average expected flow time of all uncompleted projects and is employed to schedule activities for execution until the next project arrives in the environment.

In order to illustrate the proposed solution approach, we assume that identical projects adapted from an example by Igelmund and Radermacher (1983) arrive over time, each consisting of 5 activities $i = 1, ..., 5$, which are connected by precedence relations $\mathcal{E} = \{(1, 4), (3, 5)\}$. Activities require two resources with availabilities $R_1 = 1$ and $R_2 = 2$ in the amounts $r_{1,1} = r_{5,1} = r_{2,2} = r_{3,2} = r_{4,2} = r_{5,2} = 1$ and $r_{ik} = 0$ otherwise.

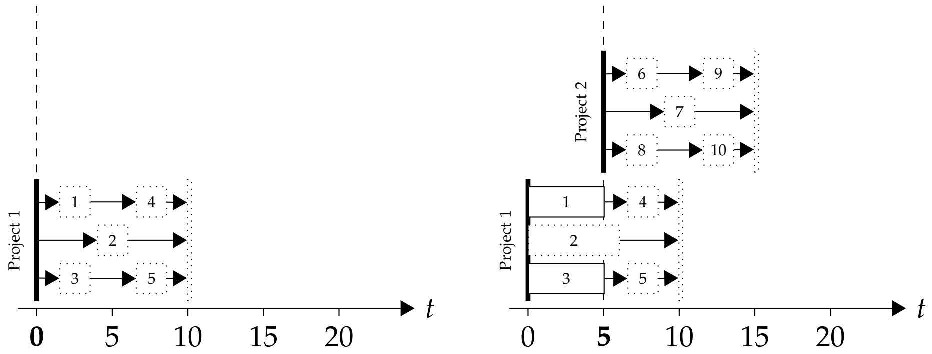


Figure 5.1.: Illustrative example of the development of a dynamic scheduling problem where two projects arrive over time

The left side of Figure 5.1 presents the dynamic scheduling problem where a first project arrives at time $t = 0$. Activities are given as rectangular nodes, whereas arcs indicate precedence relations. When a project arrives activity durations and thus also the project flow time are uncertain, indicated by dotted lines.

At time $t = 0$ the setting corresponds to the SRCPSP, which may be solved by determining a policy for activity scheduling. The right side of Figure 5.1 illustrates the setting if a policy starts activities 1, 2, and 3 at $t = 0$, and a second project, identical to the first one, arrives at $t = 5$. The previously determined scheduling policy does not provide any guidance in scheduling the second project. When determining a new policy for both the first and the second project, it must be taken into account that activities 1 and 2 of the first project have been completed by $t = 5$. The stochastic scheduling problem at $t = 5$ is solved to provide a scheduling policy for the unscheduled activities 4 and 5 of the first project as well as activities 6 to 10 of the second projects. When scheduling these activities, the resource demand of activity 2, which has been started at $t = 0$, but has not been completed by $t = 5$, must be taken into account. For exponentially distributed activity durations, the distribution of the remaining duration of activity 2 is the same as the initial distribution. This memorylessness does not hold for other distributions. In these cases the change in distribution must be accounted for, e.g., when performing simulations to determine expected flow time values in SRCPSP solution procedures.

Figure 5.2 presents activity starting and completion times if two different resource-based priority policies π_1 and π_2 are applied after $t = 5$ until both projects are completed. Policy π_1 is given by the priority list $5 \succ 4 \succ 6 \succ 7 \succ 8 \succ 9 \succ 10$, where $i \succ i'$ indicates that i has higher priority than i'. Policy π_2 corresponds to $6 \succ 7 \succ 8 \succ 9 \succ 10 \succ 5 \succ 4$. Figure 5.2 shows that the choice of scheduling policy strongly impacts resulting project flow time values. For π_1 the first project is completed after 10 time units and the second

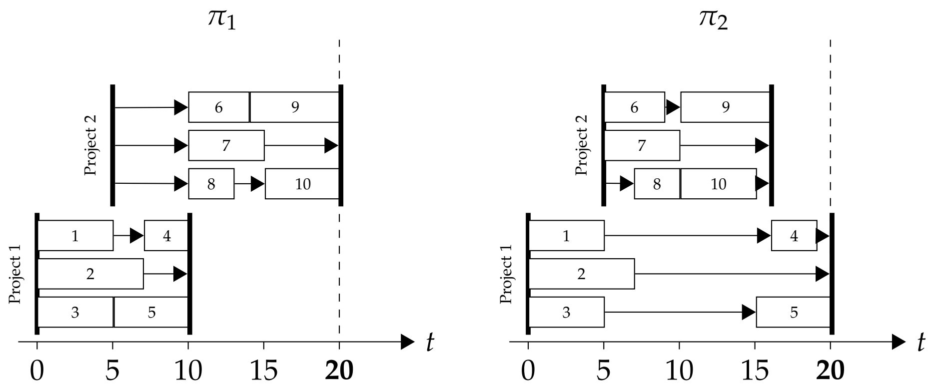


Figure 5.2.: Illustrative example of the scheduling decisions of two resource-based priority policies π_1 and π_2

project after 15, giving an average flow time of 12.5. For π_2 the last two activities of the first project are delayed giving a flow time of 20, while the second project takes 11 time units. In this case, the average flow time value is 15.5. The example shows that the choice of scheduling policy has significant impact on the project flow time.

One well established metaheuristic for the RCPSP is the priority list based genetic algorithm by Hartmann (1998) and Hartmann (2002). This procedure has been adapted to the SRCPSP setting by Ballestín (2007). The genetic algorithm generates populations of priority lists, which are interpreted as scheduling policies and whose expected project makespan is assessed by means of Monte Carlo simulation. This handling of stochastic activity durations corresponds to a sample average approximation (SAA) scheme (Kleywegt et al., 2002). Furthermore, Ballestín (2007) finds experimental evidence that for cases with small variance of activity durations the distributions can be adequately approximated by their expected values. In this case, the genetic algorithm by Hartmann (1998) is directly applied to generate priority lists, which can be interpreted and evaluated as scheduling policies for the SRCPSP. When computational effort is restricted, solution procedures using the latter approach are able to explore a larger part of the solution space than their counterparts using SAA, but for the price of underestimating stochastic deviations.

5.1.2. Proposed Scheduling Policy Classes

In literature, different classes of scheduling policies have been proposed, which are derived from the general class of set policies. At some point in time t, a set policy may start any subset of the set of activities, whose predecessors have already been completed and which obey the resource constraints. Decisions may only be based on the sets of activities started and completed up to t (Möhring et al., 1985). Earliest start policies are defined by an extension $\mathcal{E}' \supseteq \mathcal{E}$ of precedence relations for which there exist no minimal forbidden sets. Sets of activities are minimal forbidden if they form an antichain with respect to $\mathcal{E}$ and may not be executed simultaneously due to resource constraints (Möhring et al., 1984). For each minimal forbidden set, preselective policies define a preselected activity which may first be executed upon completion of any other activity from the forbidden set (Igelmund and Radermacher, 1983).

The number of minimal forbidden sets may be exponential in the number of project activities (Stork, 2001). The dynamic project scheduling problem builds on combinations of several single projects in one network. In the example illustrated by Figure 5.1, at $t = 0$ the first project has three minimal forbidden sets. In the combined network of both the first project (without completed activities 1 and 2) and the second project at $t = 5$, 29 minimal forbidden sets exist. This illustrates that scheduling policies based on minimal forbidden sets, i.e., preselective policies and earliest start policies, easily become computationally intractable for large multi-project networks.

Möhring and Stork (2000) describe two policy classes that do not rely on minimal forbidden sets and are characterized by a precedence-feasible priority ordering L of all project activities. For each decision point, resource-based priority policies (Π^{RB}) consider all unscheduled activities whose predecessors have been completed for scheduling in the order of the priority list. Activities are started under the condition that the resulting partial schedule is resource-feasible. If an activity cannot be started due to resource restrictions, the succeeding precedence-feasible activity in the priority list is considered. Activity-based priority policies (Π^{AB}) work under the additional side constraint that an activity may not be started earlier than any other activity with higher priority according to the priority list. Considering the example in Figure 5.1, any resource-based priority policy will greedily utilize all available resources and start activities 1, 2, and 3 at $t = 0$. On the other hand, activity-based priority policies might defer the execution of up to two of the activities leaving resources idle. For regular performance measures (Sprecher et al., 1995), Möhring and Stork (2000) and Stork (2001) show that analytically neither Π^{RB} nor Π^{AB} are dominant in terms of the optimum expected performance measure value $\rho = E[\kappa(\pi^*, \mathbf{d})]$ obtainable within each class. It is possible to construct instances where either class dominates the other one. However, the authors do not systematically study the performance of both policy classes for a range of different problem instances.

Möhring et al. (1984) give stability requirements for scheduling policy classes with regards to slight changes of input parameters, i.e., particularly the distribution of activity durations. For a stable policy class, the distribution of the performance measure $\kappa(\pi, \mathbf{d})$ associated with every policy $\pi \in \Pi$ and in particular the expected performance $E[\kappa(\pi, \mathbf{d})]$ will converge for every weakly convergent sequence of activity duration

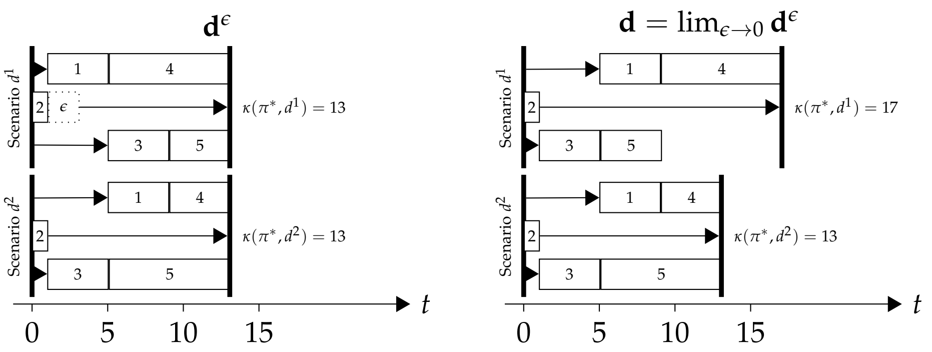


Figure 5.3.: Illustrative example of the instability of scheduling policy π^*

distributions $\mathbf{d}^\epsilon \to \mathbf{d}$ and every uniformly convergent sequence of project performance measures $\kappa^\epsilon \to \kappa$. The same holds true for the optimal expected performance measure value and the identity of being an optimal policy. This notion of stability ensures that when applying an approximation $(\kappa^\epsilon, \mathbf{d}^\epsilon)$ of input data $(\kappa, \mathbf{d})$ an established optimal policy will also be almost optimal for $(\kappa, \mathbf{d})$. As procedures for the SRCPSP rely on simulation to approximate the distributions of activity durations as well as policy performance measures, stability should ensure reliable performance of identified optimal policies as well as reliable optimum performance measure values.

To illustrate this concept of stability, we consider a single five-activity project arriving at $t = 0$. As before, precedence relations are given by $\mathcal{E} = \{(1,4),(3,5)\}$. Adopting an example by Möhring (2000), resource requirements and availabilities are set to prevent activities 1 and 3 from being executed in parallel ($R_1 = 1$, $r_{1,1} = r_{3,1} = 1$, and $r_{ik} = 0$ otherwise). Activity durations follow the multivariate, discrete (two-scenario) probability distribution

$$\mathbf{d}^\epsilon = \begin{cases} d^1 = (4, \quad 1+\epsilon, \quad 4, \quad 8, \quad 4) & \text{with probability } 0.5 \\ d^2 = (4, \quad 1, \quad\quad 4, \quad 4, \quad 8) & \text{with probability } 0.5 \end{cases} .$$

The overall optimal scheduling policy π^*, illustrated in Figure 5.3, starts activity 2 at $t = 0$ and waits until $t = 1$ to start either activity 1 or 3. If activity 2 is not completed by $t = 1$ (scenario d^1), activity 1 is started at $t = 1$ and activity 3 at $t = 5$. Otherwise (scenario d^2), activity 3 is started at $t = 1$ and activity 1 at $t = 5$. When applying this policy with $0 < \epsilon \leq 12$, the expected project duration is $E[\kappa(\pi^*, \mathbf{d}^\epsilon)] = 13$. Unstable behavior of π^* become apparent when considering $\mathbf{d} = \lim_{\epsilon\to 0} \mathbf{d}^\epsilon$. In this case the expected project duration discontinuously jumps to $E[\kappa(\pi^*, \mathbf{d})] = 15 \neq \lim_{\epsilon\to 0} E[\kappa(\pi^*, \mathbf{d}^\epsilon)] = 13$.

Möhring et al. (1984) characterize earliest start and preselective policies as finite, positively homogeneous, uniformly continuous, and monotonically increasing functions rendering these classes stable in terms of convergence in κ and $\mathbf{d}$. In general, set policies and particularly resource-based priority policies Π^{RB} are neither continuous, monotonically increasing, or convex and thus unstable and non-monotonous. Limiting instability and non-monotonicity, Möhring et al. (1985) argue that set policies show quasi-stable behavior, as within the space of activity durations they are piecewise composed of earliest start policies, which are stable. Instabilities and non-monotonous behavior appear only on the borders between these stable sets. When restricting activity durations to distributions with Lebesgue-density the combination of these borders has Lebesgue measure zero and set policies show both stability and monotonous behavior. Möhring (2000) therefore regards both set policies and priority policies as "robust" in the sense of convergence. Unstable and non-monotonous behavior in set policies may yet occur when considering discrete activity duration distributions, particularly for small sample sizes. Discrete distributions are common in procedures for the SRCPSP, which rely on simulation to approximate activity duration distributions.

5.2. Computational Studies

Past research on the SRCPSP has preferred activity- to resource-based priority policies based on analytical insights. No detailed experimental studies have been performed to compare the performance of both classes when scheduling practically interesting problem instances. Likewise, the impact of stability and monotonicity on solution procedures working with discrete approximations of activity durations has not been experimentally assessed. This leaves room for more thorough analyses of the relationship of activity- and resource-based priority policies. The first half of the computational study will consider both policy classes in the SRCPSP setting to choose one scheduling policy class for application to the dynamic project scheduling problem. Based on the promising results reported by Ballestín (2007) we adapt both the deterministic and SAA procedure to the dynamic project scheduling setting in the second half of our computational study. To give an indication of the value of the approaches, we compare them with random sampling procedures (Kolisch, 1996), likewise proposed by Ballestín (2007), as well as scheduling policies derived from common priority rules.

5.2.1. Computational Setup

We employ novel problem instances designed with both the SRCPSP and the dynamic project scheduling problem in mind. Project networks consist of 15 non-dummy activities and are created using the Progen/Max generator (Schwindt, 1998). The complexity of precedence relations is controlled by the order strength (OS) parameter. Lower OS is commonly associated with increasingly difficult problem instances due to higher numbers of precedence-feasible schedules (Herroelen and Reyck, 1999). We consider parameter levels of approximately 0.1, 0.5, and 0.9.

Expected durations $E[\mathbf{d}_i]$ of activities $i = 1, \ldots, n$ fall in the range from 1 to 10. They have been chosen to ensure a project makespan of between 10 and 20 when resource requirements are omitted and activity durations realize as their expected values. For stochastic activity durations, Jensen's inequality causes expected project makespan values to deviate from this value with increasing activity duration variance (Möhring, 2001), e.g., for exponential activity durations project flow times may vary between 11 and 32.

Every activity requires one or more types of resources where $r_{ik} \leq 10$ for all activities and resource types. The number of different resource types required by each activity is determined by the Resource Factor (RF) parameter (Kolisch et al., 1995). High parameter values indicate high resource interdependency of project activities and thus are associated with difficult scheduling environments. RF levels are set to 0.25, 0.5, 0.75, and 1.0. Ten project networks are generated for each combination of OS and RF levels giving 120 project networks overall.

For each project network, resource availabilities are determined by different parameters when dealing with either the SRCPSP or the dynamic project scheduling problem. When dealing with the SRCPSP in section 5.2.2, the parameter resource strength (RS) is used (Kolisch et al., 1995). For a resource strength level of zero, available resources only cover the highest demand for any single activity while for a level of one resource levels are set to the maximum aggregate resource requirement when scheduling all activities at their earliest start time assuming mean activity durations. Reyck and Herroelen (1996) indicate that the solution time required for problem instances with different resource strength values follows a bell-shaped curve with instances having low or high resource strength levels being significantly easier to solve than others. We set resource availabilities to give resource strength levels of approximately 0.1, 0.5, and 0.9, for all resource types. Resource requirements of activities are set so that integer resource availability levels exist that support each desired level of resource strength for each resource type.

When dealing with the dynamic project scheduling problem in section 5.2.3, we consider projects arriving according to a Poisson process with arrival rate λ, all sharing the same precedence relations, resource requirements, expected activity durations $E[\mathbf{d}_i]$, and duration distribution type. If the arrival rate of projects is lower than or equal to the rate of project completion, a stable environment is established and the average number of projects in the environment converges in the long run. If the project arrival rate is too high, the average number of projects in the environment increases toward infinity. In order to account for stability and different levels of resource scarcity we control the

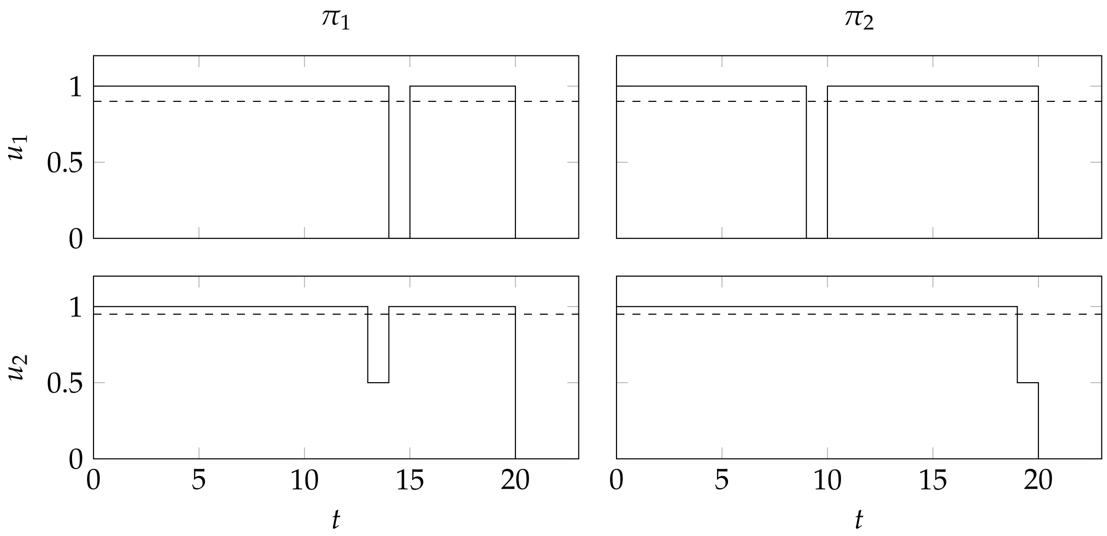


Figure 5.4.: Illustrative example of the relative resource consumption per time period for policies π_1 and π_2

ratio of expected per-period workload to per-period resource availability. As a metric, we consider the expected utilization of each resource type

$$u_k = \lambda \frac{\sum_{i=1}^{n} (E[\mathbf{d}_i] \cdot r_{ik})}{R_k}, \quad k = 1, \ldots, m. \tag{5.1}$$

Due to precedence constraints or interdependencies of activities requiring multiple resource types, available resources may remain idle. Resource utilization by Equation 5.1 therefore is only an approximation of the true utilization relating the rate of project arrival and project execution. As an illustration, we consider again the example from the previous section. For an assumed arrival rate of 0.05, corresponding to an expected inter-arrival-time of 20, we obtain utilization levels of $u_1 = 0.9$ and $u_2 = 0.95$. Figure 5.4 reports the per-period ratio of required to available resources for priority policies π_1 and π_2.

We control average utilization levels across all resource types $\overline{u} \in]0, 1.0]$ by fixing all resource availabilities to $R_k = \sum_{i=1}^{n} E[\mathbf{d}_i] \cdot r_{ik}$ for all resource types $k = 1, \ldots, m$ and adjusting λ to correspond to the desired utilization level.

In line with Ballestín and Leus (2009), we consider five different types of probability distributions for activity durations for each project network and level of resource strength or resource utilization: Two (continuous) uniform distributions (U1 and U2), two beta distributions (B1 and B2) and the exponential distribution (EXP). Durations of all project activities are assumed to follow the same type of distribution. Uniform distributions are defined with support $[E[\mathbf{d}_i] - \sqrt{E[\mathbf{d}_i]}, E[\mathbf{d}_i] + \sqrt{E[\mathbf{d}_i]}]$ for U1 and $[0, 2 \cdot E[\mathbf{d}_i]]$ for U2. The exponential distribution EXP has a mean value of $E[\mathbf{d}_i]$, whereas the beta distributions are defined with support $[E[\mathbf{d}_i]/2, 2 \cdot E[\mathbf{d}_i]]$ and a variance of $E[\mathbf{d}_i]/3$ for B1 and $E[\mathbf{d}_i]^2/3$ for B2. B1 is defined with $\alpha = (E[\mathbf{d}_i]/2) - (1/3)$ and B2 with $\alpha = 1/6$. $\beta = 2\alpha$ holds for both distribution classes. The distributions are designed so that U1 and B1 share the same low variance ($E[\mathbf{d}_i]/3$), U2 and B2 have the same medium variance ($E[\mathbf{d}_i]^2/3$), and EXP represents a case of high activity duration variance ($E[\mathbf{d}_i]^2$).

Figure 5.5 summarizes the generation process of problem instances as well as the problem parameters considered in our computational study on the SRCPSP and the dynamic project scheduling problem.

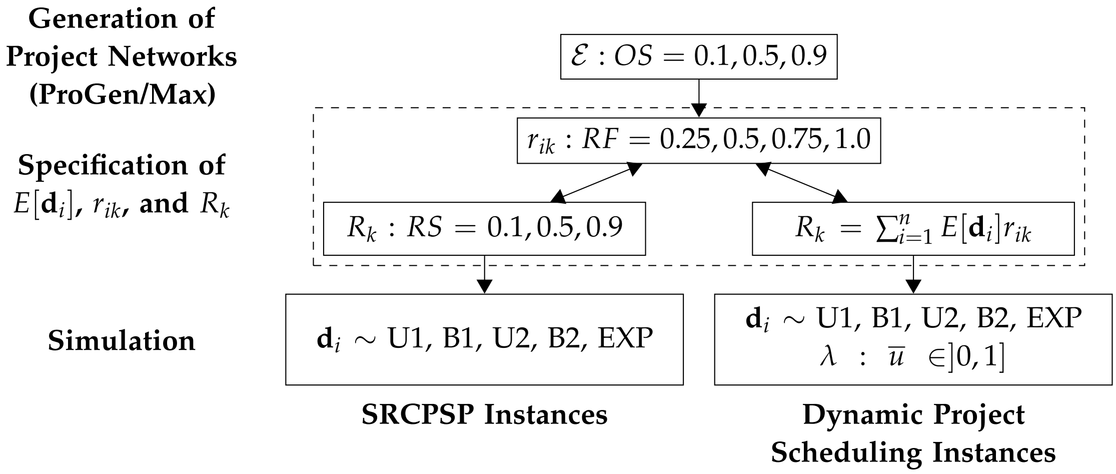


Figure 5.5.: Generation process and parameter combinations of problem instances for the SRCPSP and dynamic project scheduling problem

5.2.2. Performance and Stability of Priority Policy Classes for the SRCPSP

For the considered SRCPSP problem instances, we enumerate all possible precedence feasible priority lists. On average, 113,000 (minimum 1,356, maximum 630,630) priority lists are obtained per instance. Each priority list is evaluated as an activity- and resource-based priority policy. For each type of activity duration distribution, we determine the distribution of the project makespan obtained by each policy through descriptive sampling (Saliby, 1990) using 10,000 scenarios of activity durations. We

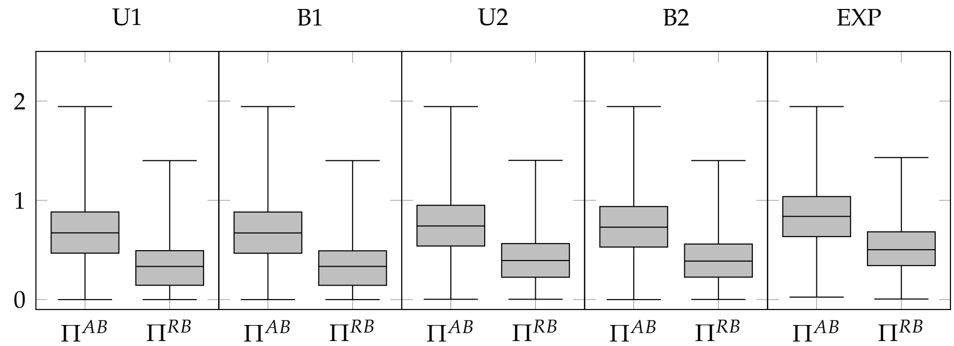


Figure 5.6.: Box-and-whisker plots of expected makespan values within policy classes Π^{AB} and Π^{RB} for different duration distribution types

compare makespan values across instances by reporting the percentage distance of the obtained makespan values to the critical path length of projects when assuming mean activity durations and omitting resource requirements.

In Figure 5.6 box-and-whisker plots illustrate distance values for all enumerated activity- and resource-based priority policies across all problem instances. Separate plots are given for different duration distribution types. Boxes cover the 25% to 75% percentile of evaluated policies, and whiskers extend to the lowest and highest value. For all duration distribution types, values are lower for resource- than for activity-based priority policies. We scrutinize this result by applying the Wilcoxon signed-rank test (Wilcoxon, 1945). We study the null hypothesis H_0 that there is no difference in expected makespan if each priority list is interpreted as an activity- or resource-based priority policy for each problem instance versus the alternative H_1 that the expected makespan of resource-based priority policies is lower than for activity-based priority policies. We can reject H_0 on a level of significance of $p < 0.001$. This holds for each type of activity duration distribution.

Ashtiani et al. (2011) also present computational results indicating significantly better expected makespan performance for resource- than for activity-based priority policies, especially for duration distribution types with medium to high variability. Based on these results, the authors opt to use resource-based priority policies in their proposed hybrid pre-processor scheduling policies and solution procedure. Similarly, Deblaere et al. (2011) argue that resource-based priority policies perform better than activity-based priority policies due to the additional precedence relations imposed on the project network by the latter. The presented performance gap partially explains the superior performance of the solution procedures by Ashtiani et al. (2011) and Deblaere et al. (2011), working with variants of resource-based priority policies, compared to procedures by Ballestín (2007) and Ballestín and Leus (2009), which employ activity-

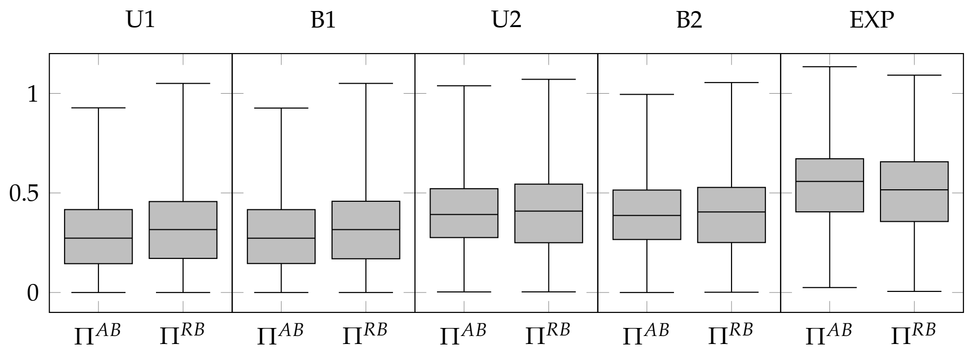


Figure 5.7.: Box-and-whisker plots of optimum expected makespan values within policy classes Π^{AB} and Π^{RB} for different duration distribution types

based priority policies.

Figure 5.7 compares activity- and resource-based priority policies in terms of the optimum expected makespan ρ^{AB} and ρ^{RB} obtainable within each class. Box-and-whisker plots illustrate ρ^{AB} and ρ^{RB} for each problem instance. For medium to high variation in activity durations (U2 and EXP), optimal resource-based priority policies significantly outperform optimal activity-based priority policies (one-tailed Wilcoxon signed-rank, $p < 0.001$). For cases of low variation (U1 and B1) this relationship is reversed (one-tailed Wilcoxon signed-rank, $p < 0.001$). For distribution type B2 the policy classes are statistically indistinguishable on a 99% level of significance (one-tailed Wilcoxon signed-rank, $p = 0.013$).

While resource-based priority policies mostly outperform activity-based priority policies in a direct comparison of policies for the same priority list, activity-based priority policies can outperform resource-based priority policies in terms of their obtainable optimum value when activity duration variance is low. In order to shed light on this discrepancy, Figure 5.8 reports the optimality gap of each activity- and resource-based priority policy for all problem instances. We find that resource-based priority policies have significantly lower optimality gaps than activity-based priority policies. Furthermore, within the class of resource-based priority policies on average 25% of all policies provide the optimum expected makespan, while only 0.03% of policies provide the optimum value within the class of activity-based priority policies.

In this study, resource-based priority policies show predominantly superior performance compared to activity-based priority policies, yet they are associated with unstable behavior from a theoretical perspective. We analyze stability in terms of the convergence of the expected makespan for any given policy. In order to justify the configuration of his SRCPSP procedure with respect to the number of scenarios, Stork (2001) shows for

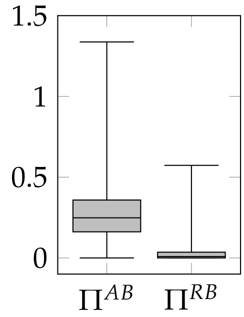


Figure 5.8.: Box-and-whisker plots of optimality gaps within policy classes Π^{AB} and Π^{RB}

different numbers of scenarios the average and maximum percentage deviation of the expected makespan from the makespan obtained when activity durations only take on their mean value. Leus and Herroelen (2004) and Ballestín and Leus (2009) extend this approach by considering the standard deviation of the percentage deviation of expected makespan values obtained through repeated simulation using the same number of scenarios, versus the "true" expected makespan value obtained by simulation using a large number of scenarios.

We adopt this approach and compare the standard deviation of the percentage deviation of approximated expected makespan values from their "true" values for multiple activity- and resource-based priority policies. For each problem instance, we select 100 policies from both policy classes, whose "true" expected makespan value obtained by simulation using 10,000 scenarios corresponds to the 1st, 2nd, ..., 100th percentile within the class. For each selected policy, we obtain 100 approximations of the expected makespan value by performing repeated simulations using either 3, 10, 100, or 1,000 scenarios of activity durations. Figure 5.9 reports the standard deviation of the deviation of expected makespan values for activity- and resource-based priority policies across problem instances for different duration distribution types and varying numbers of scenarios. Average deviation values are lower for activity- than for resource-based priority policies for all numbers of scenarios and distribution types. The difference in deviation is statistically significant for all duration distribution types and scenario numbers (one-tailed Wilcoxon signed-rank, $p < 0.001$). Deviations for both policy classes as well as differences in deviations are particularly pronounced when considering only 3 or 10 scenarios and exponentially distributed activity durations.

In terms of application, the stability of the performance measure associated with arbitrary policies is less relevant than stability with regards to the optimal policy and the optimum expected performance measure value. To this end, we analyze stability properties of the optimum expected makespan values obtainable within the classes of

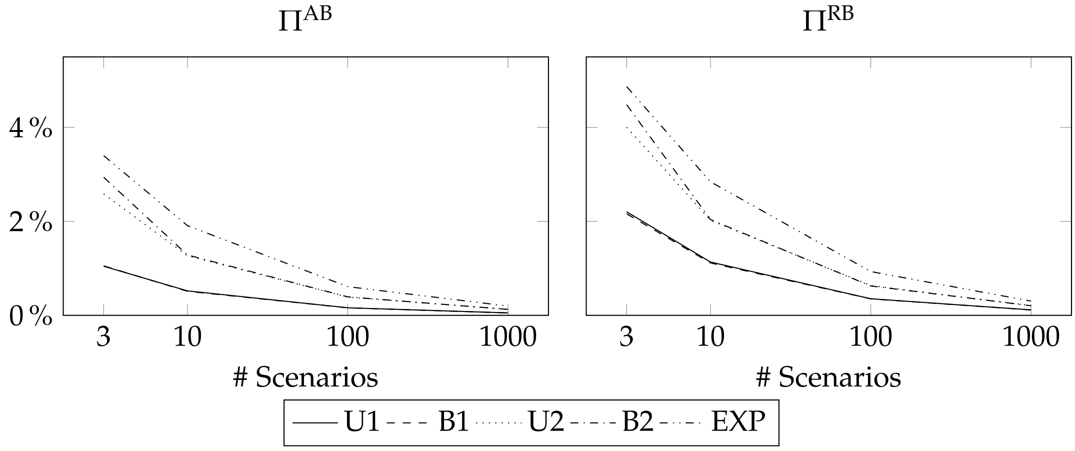


Figure 5.9.: Standard deviation of the deviation of expected makespan values within policy classes Π^{AB} and Π^{RB} for given numbers of scenarios and different duration distribution types

activity- and resource-based priority policies. We consider an experimental routine by Kleywegt et al. (2002), who study convergence and stability of optimum values in general sample average approximation problems. We evaluate the "true" expected makespan of all enumerated activity- and resource-based priority policies using 10,000 scenarios of activity durations and determine "true" optimum expected makespan values ρ^{AB} and ρ^{RB}. We then repeat the evaluation 100 times using only three scenarios, which are varied in each repetition. Hereby, 100 sets of approximately optimal policies are obtained. Figure 5.10 reports the relative frequency that approximately optimal activity- and resource-based priority policies are truly ϵ-optimal, i.e., their "true" value falls in the range $[\rho^{AB}, (1+\epsilon)\rho^{AB}]$ or $[\rho^{RB}, (1+\epsilon)\rho^{RB}]$. Relative frequencies are calculated across all problem instances. We find that approximately optimal activity-based priority policies are significantly further away from ρ^{AB} than approximately optimal resource-based priority policies from ρ^{RB} (one-tailed Wilcoxon signed-rank, $p < 0.001$). We conjecture that although the expected performance of arbitrary activity-based priority policies behaves stably, the smaller number of (near-)optimal policies within the solution space reported earlier in Figure 5.8 causes less stable behavior of optimal activity-based priority policies.

Summarizing the first study, we conclude that on average resource-based priority policies outperform activity-based priority policies. In cases of low variation in activity durations, there exist activity-based priority policies that provide a lower expected makespan value than the best obtainable resource-based priority policy. These "good" activity-based priority policies appear to be less frequent within their solution space than "good" resource-based priority policies. We confirm that activity-based priority

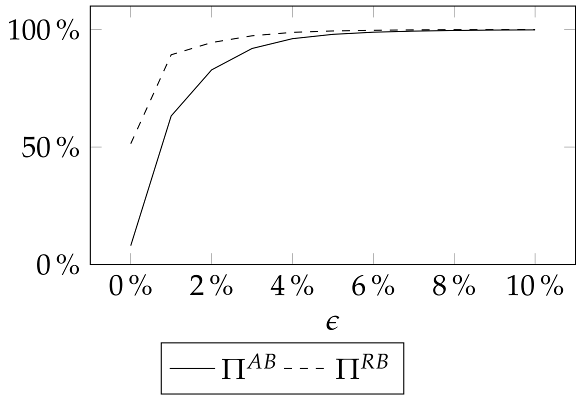


Figure 5.10.: Relative frequency of ϵ-optimality of approximately optimal policies within policy classes Π^{AB} and Π^{RB}

policies have better stability properties than resource-based priority policies when considering approximations of activity durations and makespan distributions. When considering approximations using 100 scenarios, deviations for both policy classes fall below 1% limiting the extent of unstable behavior of resource-based priority policies to approximations using only few scenarios. Unsatisfactory stability properties of resource-based priority policies can be mitigated by choosing higher numbers of scenarios when approximating activity duration and makespan distributions. Furthermore, optimal resource-based priority policies have been shown to behave more stably than activity-based priority policies even when only few scenarios are considered.

5.2.3. Comparison of Solution Procedures for the Dynamic Project Scheduling Problem

Due to the superior performance of resource-based priority policies, we focus on this policy class when applying SRCPSP solution procedures to the dynamic project scheduling problem. Starting with an empty environment, a SRCPSP solution procedure is used to derive a scheduling policy whenever a new project arrives. The policy is used for activity execution until the arrival of the next project. Then a new policy is determined. In order to aid comparisons of several solution procedures across different problem instances, we employ common random numbers as realized activity durations and within SAA procedures as activity duration scenarios. Whenever rescheduling is performed, new common activity duration scenarios are used in SAA procedures. Furthermore, we employ common random numbers as project arrival times.

To obtain initial results, we consider a very basic solution procedure, the first-come-

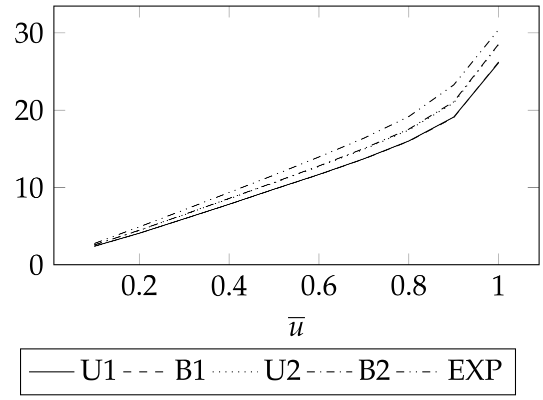


Figure 5.11.: Average number of projects within the scheduling environment for given levels of utilization and different duration distribution types

first-served (FCFS) priority rule. FCFS provides scheduling policies that prioritize activities in order of their (project) arrival and the order induced by project precedence constraints. Iterations of project arrivals and rescheduling are performed until a total of 200 projects have been completed. For different levels of resource utilization $\overline{u}$ and activity duration distribution types, Figure 5.11 reports the average number of projects that are executed in parallel within the environment. Values increase with increasing utilization and increase slightly with increasing variance of activity durations.

For different utilization levels $\overline{u}$, Figure 5.12 presents the development of the number of projects within the environment. Solid lines give average values, whereas dotted lines indicate dispersion of one standard deviation around the averages. Initially, the dynamic scheduling environment behaves unstably while the first projects arrive, and the number of projects executed in parallel grows. If sufficient resources are available to process arriving workload, the environment gradually becomes stable, and the average number of projects in the environment converges.

We ensure that solution procedures for dynamic project scheduling are evaluated in stable environments. To this end, we start evaluations only after a sufficient number of projects have arrived for the environment to achieve stability. We determine the required length of this "warm up period" by performing linear regression analyses on the number of projects in the environment dependent on the number of project arrivals. Until a threshold level of project arrivals in the environment has been reached, executed projects are excluded from the regression analysis. For different regression thresholds and different levels of utilization, Figure 5.13 reports the average slope of the linear regression lines. For $\overline{u} \leq 0.9$, slopes are positive when all arriving projects are considered and decrease to zero with increasing threshold, i.e., the environments become stable after sufficiently many projects have arrived. Stability is achieved after

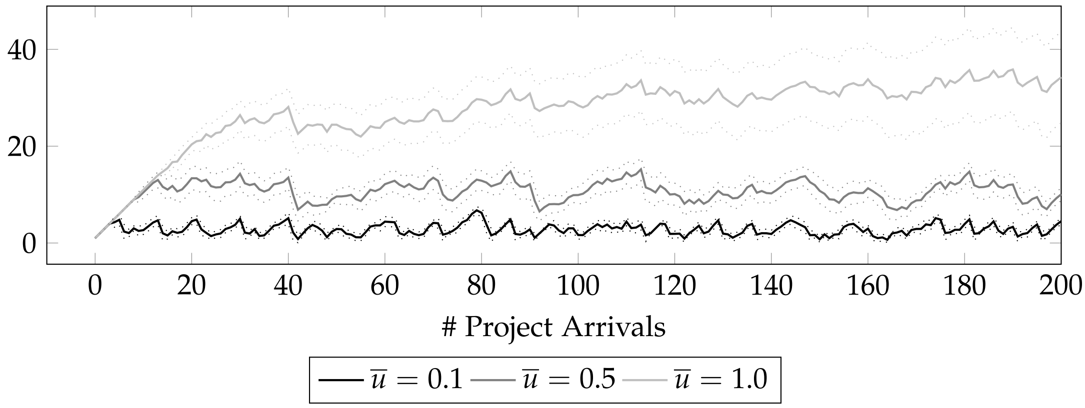


Figure 5.12.: Development of the number of projects within the environment for different levels of utilization

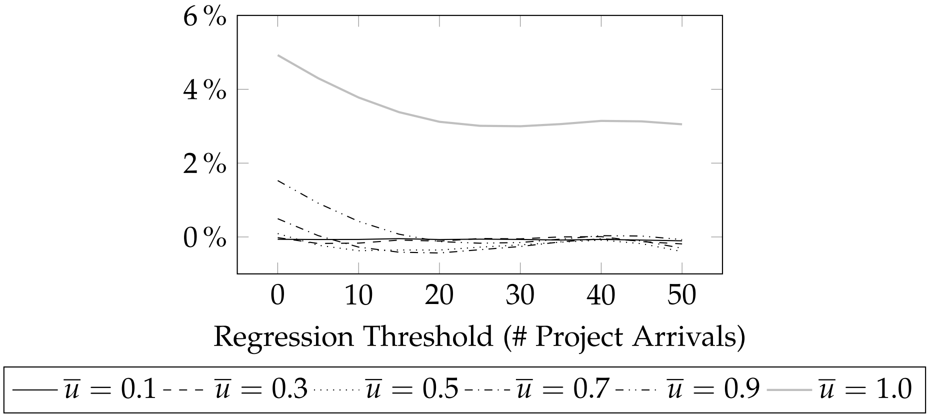


Figure 5.13.: Average linear regression line slopes obtained after warm up periods of given length for different levels of utilization

about 20 projects have arrived. For the other solution procedures covered by this computational study similar values are obtained. For $\overline{u} = 1.0$, slopes are positive even if a warm up period is accounted for. We conclude that this utilization level is too high to support stable environments, and therefore exclude $\overline{u} = 1.0$ when evaluating solution procedures.

We now investigate the random sampling procedures and genetic algorithms (GA) proposed by Ballestín (2007). These procedures take into account stochastic activity durations either by sample average approximation (SAA) (Kleywegt et al., 2002) or by considering expected activity durations (Mean). In sampling procedures, regret-based biased random sampling (RRS) based on the latest finish times of activities is used. This sampling scheme is also employed to create initial populations for the genetic algorithms. All algorithms are terminated after 5,000 schedules have been generated. This common approach in scheduling studies makes it possible to compare algorithms without having to consider the computer architecture used in the trials (Hartmann and Kolisch, 2000). Ballestín (2007) shows that the use of fewer scenarios in approximating activity duration distributions is beneficial when a limit is set on the number of generated schedules. The loss in precision regarding the project duration distribution is more than compensated by the benefit of exploring more solutions. The less variability in activity durations, the fewer scenarios should be used. This result is also supported by Ballestín and Leus (2009) as well as Ashtiani et al. (2011). In our computational study SAA using 3, 10, and 100 scenarios (denoted SAA-3, SAA-10, and SAA-100) will be compared while applying descriptive sampling (Saliby, 1990) as a variance reduction technique. In line with Hartmann (2002), the genetic algorithms apply a two-point crossover and mutation operator. Through pretests we have found a population size of 100 and a mutation probability of 0.1 to give the best results.

Table 5.1 reports average flow time values for different configurations of the random sampling procedure and genetic algorithm. In order to ensure stability, we employ a "warm up period" of 20 project arrivals, before evaluating project flow time values. On average, we exclude flow time values of four projects. Flow time values for different problem instances are compared by reporting the percentage distance of flow time values to the critical path length of a single project when assuming mean activity durations and omitting resource requirements. As seen in Figure 5.13, utilization level $\overline{u} = 1.0$ does not support stable environments. For $\overline{u} < 0.5$, flow time values do not differ significantly from the critical path length (0.33% on average). We therefore focus on utilization levels $\overline{u} = 0.5$, $\overline{u} = 0.7$, and $\overline{u} = 0.9$ in our evaluations. For all considered solution procedure configurations, flow time values increase with increasing resource utilization and variance of activity durations. We find that the genetic algorithm clearly outperforms the random sampling procedure for all utilization levels and activity duration distribution types (one-tailed Wilcoxon signed-rank, $p < 0.01$). For $\overline{u} = 0.5$ and exponentially distributed activity durations, random sampling procedure RRS-SAA-100 has slightly lower average flow time values than all genetic algorithms, but the difference is only significant when comparing to genetic algorithm GA-Mean (one-tailed Wilcoxon signed-rank, $p < 0.001$).

Ballestín (2007) claims for the SRCPSP that using fewer scenarios to evaluate activity-based priority policies during a solution procedure is beneficial when a limit on the number of generated schedules is imposed. The loss in precision regarding the performance measure value of individual policies is more than compensated by the benefit of exploring more solutions. The less variability in activity durations, the fewer scenarios should be used. For the dynamic project scheduling problem, we find weak evidence of this claim. GA-SAA-10 and GA-SAA-100 significantly outperform GA-Mean for most cases covered by Table 5.1 (one-tailed Wilcoxon signed-rank, $p < 0.01$).

			Distribution				
			U1	B1	U2	B2	EXP
$\overline{u} = 0.5$	RRS	Mean	5.26%	5.11%	15.12%	15.66%	27.03%
		SAA-3	5.24%	5.08%	15.12%	15.64%	27.00%
		SAA-10	5.22%	5.06%	15.10%	15.62%	26.98%
		SAA-100	5.22%	5.06%	15.09%	15.61%	**26.97%**
	GA	Mean	5.24%	5.09%	15.11%	15.64%	27.00%
		SAA-3	5.22%	5.07%	15.10%	15.62%	26.98%
		SAA-10	**5.21%**	5.05%	15.09%	15.61%	26.98%
		SAA-100	5.21%	**5.05%**	**15.08%**	**15.60%**	26.98%
$\overline{u} = 0.7$	RRS	Mean	6.86%	6.72%	16.74%	17.29%	28.43%
		SAA-3	6.80%	6.65%	16.62%	17.19%	28.30%
		SAA-10	6.73%	6.56%	16.57%	17.10%	28.17%
		SAA-100	6.72%	6.54%	16.57%	17.08%	28.18%
	GA	Mean	6.76%	6.58%	16.65%	17.17%	28.29%
		SAA-3	6.68%	6.51%	16.59%	17.08%	28.31%
		SAA-10	**6.64%**	**6.44%**	16.51%	**17.01%**	**28.15%**
		SAA-100	6.65%	6.45%	**16.50%**	17.01%	28.17%
$\overline{u} = 0.9$	RRS	Mean	15.48%	15.32%	27.20%	28.55%	42.37%
		SAA-3	15.12%	14.97%	26.78%	27.72%	41.62%
		SAA-10	14.93%	14.83%	26.41%	27.30%	41.18%
		SAA-100	15.17%	15.05%	26.62%	27.50%	41.07%
	GA	Mean	14.56%	14.42%	26.10%	27.49%	41.55%
		SAA-3	14.30%	14.08%	26.10%	26.87%	41.05%
		SAA-10	**14.06%**	**13.94%**	**25.60%**	**26.60%**	**40.67%**
		SAA-100	14.52%	14.38%	25.90%	27.02%	40.91%

Table 5.1.: Average flow time values obtained by solution procedures for given levels of utilization and duration distribution types

With regards to GA-SAA-100, significance is weak for $\overline{u} = 0.9$ and activity duration distributions U1 (one-tailed Wilcoxon signed-rank, $p = 0.012$) and B1 (one-tailed Wilcoxon signed-rank, $p = 0.026$). Comparing GA-SAA-10 and GA-SAA-100, no clear preference can be given for either configuration for $\overline{u} = 0.5$ and $\overline{u} = 0.7$. For utilization level $\overline{u} = 0.9$, GA-SAA-10 significantly outperforms GA-SAA-100 (one-tailed Wilcoxon signed-rank, $p < 0.01$), except for the case of exponentially distributed activity durations where significance is weak (one-tailed Wilcoxon signed-rank, $p = 0.04$). This partially supports the claim by Ballestín (2007) that when considering high levels of resource utilization fewer scenarios are beneficial for solution procedure performance. Yet, the bad performance of GA-Mean and the statistically ambiguous performance of GA-SAA-3 indicate that using too few solutions to properly take into account activity duration distributions is detrimental to flow time performance.

For special cases of stochastic and dynamic scheduling problems, priority policies derived from simple priority rules have been shown to lead to optimal performance. For example, Weiss and Pinedo (1980) show the problem of scheduling jobs with exponentially distributed activity durations on identical parallel machines to be solved optimally in terms of the expected makespan by scheduling jobs in decreasing order of their longest expected processing times. Chou et al. (2006) derive asymptotic optimality of scheduling based on weighted shortest expected processing times for a dynamic scheduling problem. We compare all implemented procedures to FCFS and other common priority rules for (multi-)project scheduling (Demeulemeester and Herroelen, 2002; Browning and Yassine, 2010). The longest/shortest expected processing time (LEPT/SEPT) rules schedule activities in order of non-increasing/non-decreasing expected duration, while the greatest rank positional weight (GRPW/GRPW2) rules schedule activities in order of non-increasing aggregate expected duration of the considered activity as well as all its immediate/transitive successors. The maximum immediate/transitive successors (MIS/MTS) priority rules schedule activities according to non-decreasing number of immediately/transitively succeeding activities in the project network, while the least non-related jobs (LNRJ) priority rule first selects activities with fewest activities that are not precedence related. The minimum earliest start time (MINEST), minimum latest start time (MINLST), minimum latest finish time (MINLFT), and minimum slack (MINSLK) rules rely on information from the earliest start schedule for all active projects, which considers expected activity durations, precedence relations, and project arrival times, but ignores resource constraints. Finally, the maximum/minimum total work content (MAXTWK/MINTWK) priority rules schedule activities according to a measure of work content. Work content is measured by the quotient of aggregate resource consumption of an activity, i.e., the product of per-period consumption and expected activity duration, and aggregate resource consumption of previously scheduled project activities.

Table 5.2 compares average flow time values of GA-SAA-10, GA-SAA-100, and priority rules for different utilization levels. We find that both genetic algorithms outperform scheduling policies induced by priority rules. For all utilization levels, we reject the hypothesis that there is no difference in performance between GA-SAA-10 and GA-SAA-100 compared to most priority rules (one-tailed Wilcoxon signed-rank, $p < 0.01$). Only for $\overline{u} = 0.5$ GA-SAA-10 and MINLFT (one-tailed Wilcoxon signed-rank, $p = 0.24$) as well as GA-SAA-10 and MINLST (one-tailed Wilcoxon signed-rank, $p = 0.30$) cannot be statistically distinguished. For $\overline{u} = 0.9$ LEPT, GRPW, GRPW2, MIS, MTS, and MINSLK underperform considerably compared to advanced solution procedures as well as the other priority rules.

Solution Procedure	$\overline{u}$ 0.5	0.7	0.9
GA-SAA-10	13.62%	**14.96%**	**24.16%**
GA-SAA-100	**13.62%**	14.97%	24.53%
FCFS	13.70%	15.36%	27.35%
LEPT	13.83%	16.65%	62.18%
SEPT	13.75%	15.56%	29.98%
GRPW	13.81%	16.56%	69.04%
GRPW2	13.73%	16.07%	57.93%
MIS	13.78%	16.05%	44.77%
MTS	13.72%	15.88%	45.64%
LNRJ	13.69%	15.44%	35.15%
MINEST	13.74%	15.58%	27.53%
MINLFT	13.63%	15.01%	24.71%
MINLST	13.63%	15.05%	25.23%
MINSLK	13.67%	15.49%	40.49%
MAXTWK	13.73%	15.40%	27.49%
MINTWK	13.71%	15.38%	27.62%

Table 5.2.: Average flow time values obtained by GA-SAA-10, GA-SAA-100, and priority rules for given levels of utilization

5.3. Conclusion and Outlook

In this chapter we have considered the dynamic project scheduling problem where projects are modeled as in the SRCPSP and multiple projects arrive stochastically over time. Established solution procedures originally developed for the static SRCPSP have been adapted and provide multi-project scheduling policies, which are executed until the arrival of a new project.

To investigate scheduling policy classes and solution procedure performance, we have employed a full factorial test design. Five distribution types for stochastic activity durations have been considered, while novel test instances with 15 activities have been defined by the established parameters order strength, resource factor, and resource strength as well as a resource utilization metric derived from queuing theory.

In the first computational study we have compared the performance and stability of activity- and resource-based priority policies. Overall, we have found significantly better performance of resource-based priority policies in terms of the expected project makespan. Optimal activity-based priority policies only outperform resource-based priority policies for cases of low variance of activity durations, while overall the number of nearly optimal policies is significantly lower. While activity-based priority policies show better stability of expected makespan values obtained by approximating activity durations through as few as three scenarios, the best resource-based priority policies determined through such approximations are closer to the "true optimal policy" than for activity-based priority policies.

Based on the results of the first computational study, we adapt SRCPSP solution procedures working with resource-based priority policies to the dynamic project scheduling problem. The random sampling procedure and genetic algorithm by Ballestín (2007) have been employed for policy generation. Our results show superior performance of the proposed scheduling procedures compared to common priority rules for all considered problem instances. Among the priority rules, the MINLFT and MINLST rule present the closest match to our procedures. The genetic algorithm outperforms the random sampling procedure in all cases. The approximation of stochastic activity durations by their respective mean values instead of sample average approximation, initially proposed by Ballestín (2007) for the static SRCPSP, proves only valuable to a limited extent when considering dynamic project scheduling environments. Only for high levels of resource utilization and when considering sample sizes greater than three, the loss in quality of approximation is outweighed by increased computational capacity to explore additional candidate policies.

In order to gain deeper insights into the nature of dynamic project scheduling, our investigations can be extended into different directions. First, additional solution procedures can be considered in our proposed experimental routine using the novel test instances. A comparison of the metaheuristic solution procedures considered in this study to e.g., Markov decision-based and approximate dynamic programming approaches (Melchiors, 2013) appears promising. While this chapter is based on purely reactive stochastic project scheduling in the sense that no baseline schedule is established prior to project execution, real life project planning and budgeting requires a certain level of robustness with regard to anticipated activity starting times (Herroelen and Leus, 2005). A decision maker thus would be interested in establishing scheduling policies with low deviation of activity starting times from predictable means, even for the price of longer project flow times. It appears promising to investigate the performance of our procedures for this alternative performance measure in order to quantify the trade-off between activity start time robustness and project flow time. Finally, the general setting of the dynamic multi-project scheduling problem under activity duration uncertainty could be adapted to fit different problem settings from practice.

6. Conclusion

Frequently, prominent projects are seen to miss time and cost targets. Among other factors, failure to consider uncertainty in selecting and scheduling projects can be blamed. Motivated by a real-life project management problem of selecting projects to improve the supply chain function of an international semiconductor manufacturer presented in Chapter 2, this dissertation has covered three distinct topics within the domain of project management and project scheduling.

The real-life case study illustrates that choosing the right subset from a set of candidate projects is a key driver of success and failure for organizations. Chapter 3 has considered the Robust Portfolio Modeling (RPM) approach to multi-objective project portfolio selection with uncertain project scores and decision maker preferences. By determining non-dominated portfolios for all possible realizations of uncertain parameters, decision recommendations produced by RPM may prove too conservative for real-life decision problems. In this chapter a methodology has been developed to reduce the set of possible realizations by limiting the number of project scores that may simultaneously deviate from their most likely value. By adjusting this limit, decision makers can choose desired levels of conservatism. The approach also allows to capture dependencies among project scores as well as uncertainty in portfolio constraints.

Chapter 4 has investigated how human decision makers behave in the context of project portfolio selection using an experimental study based on the knapsack problem. Within the proposed experimental framework, it is possible to study both subjects' decision quality as well as their decision making process. Decision makers select suboptimal portfolios across all knapsack problems considered. Presented results show that human decision making focuses on selecting items to construct an initial portfolio, which serves as a baseline solution for further improvement. The chapter investigates subjects' adherence to simple constructive heuristics motivated by portfolio selection practice. Decision making is partially explained by adherence to two heuristics, but problem complexity limits the application of such heuristics to a subset of items.

Finally, Chapter 5 has considered uncertainty in the operational domain of project scheduling. In the setting of dynamic project scheduling, projects arrive stochastically over time. Each arriving project is modeled as a stochastic resource-constrained project scheduling problem. Activity durations are stochastic, while all other parameters are deterministic. Activities are started based on a priority policy for all unscheduled activities to minimize the average expected flow time of the projects. The policy is updated at each project arrival. The chapter develops novel test instances to perform computational studies on the stochastic resource-constrained project scheduling problem as well as the dynamic project scheduling problem. Firstly, activity- and resource-based priority policies for the SRCPSP have been compared in terms of their expected makespan performance and stability properties, which are relevant when employing solution procedures based on simulation. Findings include that resource-based priority policies outperform activity-based priority policies while only having negligible drawbacks in terms of stability. Focusing on resource-based priority policies, the performance of state-of-the-art SRCPSP solution procedures applied to the dynamic project scheduling problem has been evaluated in comparison with different priority rules. The proposed genetic algorithm outperforms random sampling procedures as well as all considered priority rules.

A. Computation of Non-Dominated Portfolios

The algorithm developed here to compute the set of non-dominated portfolios is based on the algorithm of Liesiö et al. (2008), which determines non-dominated portfolio in the standard RPM-framework in which scores can take any value within an interval, i.e., $v \in S_v^\infty$. The core of the algorithm is the enumeration of the sets of portfolios $Z^0, Z^1, \ldots, Z^m$ using the iteration scheme

$$\begin{aligned} Z^0 &= \left\{[0,\ldots,0]^T\right\} \\ Z^s &= Z^{s-1} \cup \left\{z + e^s \mid z \in Z^{s-1}\right\},\ s = 1,...,m, \end{aligned} \tag{A.1}$$

where e^s is a unit vector with the element s equal to one and the rest of the elements equal to zero. At each iteration stage $s = 0, ..., m$, portfolios in set Z^s only contain projects from the set $\{1, ..., s\}$. Furthermore, at the final stage the set Z^m is equal to the set of all possible portfolios $z \in \{0,1\}^m$.

To avoid enumerating all possible portfolios, we deploy three techniques that at each iteration identify and discard portfolios that cannot become non-dominated if projects from the set $\{s+1, ..., m\}$ are added to them. Discarding portfolio z from the set Z^s implies that any portfolio that is obtained by adding some projects from the set $\{s+1, ..., m\}$ to portfolio z is not included in any of the pursuant sets $Z^{s+1}, \ldots, Z^m$.

The first of these techniques is based on the fact that a non-dominated portfolio must be feasible; Hence, an infeasible portfolio in Z^s which cannot become feasible by including some projects from the set $\{s+1, ..., m\}$ can be discarded. A sufficient condition for discarding $z \in Z^s$ is that the inequality

$$\sum_{j=1}^{s} z_j a_{lj} + \sum_{j=s+1}^{m} \min\left\{0, a_{lj}\right\} > b_l \tag{A.2}$$

holds for some $l = 1, \ldots, q$ (Liesiö et al., 2008, Lemma 3).

The second technique is based on pairwise comparison of portfolios in Z^s to identify if a portfolio is dominated by another portfolio that also has more slack in the feasibility constraints. Such as portfolio can then be safely discarded as stated by the following lemma.

Lemma A.0.1 *Let $z', z'' \in Z^s$ and information set $S = (S_v^\Gamma \times S_w)$. If $z' \succ_S z''$ and $Az' \leq Az''$ (where $\leq$ holds element-wise), then $[z''_1, \ldots, z''_s, z_{s+1}, \ldots, z_m]^T \notin Z_N(S)$ for any $z \in \{0,1\}^m$.*

The final technique requires that some feasible portfolios with a high value are available before the start of the iteration scheme (A.1). These reference portfolios then serve as a benchmark for how much added value including projects from the set $\{s+1, \ldots, m\}$ must produce for the resulting portfolio to be non-dominated. This condition is formalized in the following lemma.

Lemma A.0.2 *Let $z' \in Z_F$ and $z'' \in Z^s$. If*

$$V(\hat{v}, w, z') > V(\hat{v}, w, z'') + UB_k(w, z', z'') \; \forall \; w \in ext(conv(S_w)),$$

where $UB_s(w, z', z'')$ is an upper bound for the optimum of the mixed integer linear programming problem

$$\begin{aligned}
\max_{\substack{y \in [0,1]^{m \times n} \\ z \in \{0,1\}^m}} \quad & \sum_{i=1}^{n} \sum_{j=s+1}^{m} \hat{v}_{ji} w_i z_j + \sum_{i=1}^{n} \sum_{j=1}^{m} \vec{v}_{ji} w_i y_{ji} \\
& Az \leq B - Az'' \\
& \sum_{i=1}^{n} \sum_{j=1}^{m} y_{ji} \leq \Gamma \\
& y_{ji} = 0 \; \forall \; i \in \{1, \ldots, n\}, j \in \left\{ j \in \{1, ..., s\} \mid z'_j + z''_j \neq 1 \right\} \\
& y_{ji} \leq 2 - z_j - z'_j \; \forall \; i \in \{1, \ldots, n\}, j \in \{s+1, ..., m\} \\
& y_{ji} \leq z_j + z'_j \; \forall \; i \in \{1, \ldots, n\}, j \in \{s+1, ..., m\}
\end{aligned}$$

then $[z''_1, \ldots, z''_s, z_{s+1}, \ldots, z_m]^T \notin Z_N(S_v^\Gamma \times S_w)$ for any $z \in \{0,1\}^m$.

In principle, it is possible to solve the exact optimum of the mixed integer linear programming problem in Lemma A.0.2 instead of relying on an upper bound for the optimum. Although the exact solution would allow to discard more portfolios at each iteration stage, an approximate upper bound can be obtained with significantly less computational effort by solving either the linear programming relaxation or Lagrangian dual of the mixed integer linear programming problem (Bertsimas and Tsitsiklis, 1997).

The algorithm for solving the set of non-dominated portfolios $Z_N(S_v^\Gamma \times S_w)$ can be formalized as follows:

1 Solve $Z_R \subset Z_F$

2 $Z^0 \leftarrow \{[0,\ldots,0]^T\}$.

3 **for** $s = 1,\ldots,m$ **do**

4 $\quad Z^s \leftarrow Z^{s-1} \cup \{z + e^s \mid z \in Z^{s-1}\}$

5 $\quad Z^s \leftarrow \left\{z \in Z^s \mid \nexists l \in \{1,\ldots,q\} \text{ s.t. } \sum_{j=1}^{s} z_j a_{lj} + \sum_{j=s+1}^{m} \min\{0, a_{lj}\} > b_l\right\}$

6 $\quad Z^s \leftarrow$
$\quad \{z \in Z^s \mid \nexists z' \in Z_R \text{ s.t. } V(\hat{v},w,z') > V(\hat{v},w,z) + UB_s(w,z',z)\ \forall\, w \in \text{ext}(\text{conv}(S_w))\}$

7 $\quad Z^s \leftarrow \left\{z \in Z^s \mid \nexists z' \in Z^s \text{ s.t. } z' \succ_{S_v^\Gamma \times S_w} z \text{ and } Az' \leq Az\right\}$

8 **end**

9 $Z_N(S_v^\Gamma, S_w) \leftarrow \left\{z \in Z^m \mid \nexists z' \in Z^m \text{ s.t. } z' \succ_{S_v^\Gamma \times S_w} z\right\}$

The first step is to obtain a set of reference portfolio $Z_R \subset Z_F$ by generating some random weights and scores from sets S_w and S_v^Γ, respectively, and then solving the resulting the integer programming problems (3.4). The loop on lines 3–8 runs through the iteration scheme (A.1) and at each stage deploys the three techniques to discard portfolios, i.e., Equation (A.2), Lemmas A.0.1 and A.0.2. At the last iteration stage $s = m$, line 5 discards all infeasible portfolios from the set Z^m and line 9 carries out pairwise dominance checks to identify the non-dominated portfolios.

B. Proofs

Proof of Theorem 3.2.1

Part 1. The condition $V(v,w,z) \geq V(v,w,z')$ for all $(v,w) \in S = (S_v^\Gamma \times S_w)$ holds iff the optimum of $\min_{(v,w)\in S} V(v,w,z) - V(v,w,z')$ is non-negative. By denoting $J = \left\{j \in \{1,...,m\} \,|z_j = 1, z'_j = 0\right\}$ and $J' = \left\{j \in \{1,...,m\} \,|z_j = 0, z'_j = 1\right\}$, this problem is given by

$$\min_{(v,w)\in S} \left\{\sum_{i=1}^{n}\sum_{j=1}^{m} v_{ji}w_i z_j - \sum_{i=1}^{n}\sum_{j=1}^{m} v_{ji}w_i z'_j\right\} = \min_{(v,w)\in S} \left\{\sum_{i=1}^{n}\sum_{j\in J} v_{ji}w_i - \sum_{i=1}^{n}\sum_{j\in J'} v_{ji}w_i\right\}$$

$$= \min_{w\in S_w} \min_{\substack{y_{ji}\in[-1,1] \\ \sum|y_{ji}|\leq\Gamma}} \left\{\sum_{i=1}^{n}\sum_{j\in J} w_i(\hat{v}_{ji} + \vec{v}_{ji}y_{ji}) - \sum_{i=1}^{n}\sum_{j\in J'} w_i(\hat{v}_{ji} + \vec{v}_{ji}y_{ji})\right\}$$

$$= \min_{w\in S_w} \min_{\substack{y_{ji}\in[-1,1] \\ \sum|y_{ji}|\leq\Gamma}} \left\{\sum_{i=1}^{n}\sum_{j\in J} \hat{v}_{ji}w_i + \sum_{i=1}^{n}\sum_{j\in J} \vec{v}_{ji}w_i y_{ji} - \sum_{i=1}^{n}\sum_{j\in J'} \hat{v}_{ji}w_i - \sum_{i=1}^{n}\sum_{j\in J'} \vec{v}_{ji}w_i y_{ji}\right\}$$

$$= \min_{w\in S_w} \min_{\substack{y_{ji}\in[-1,1] \\ \sum|y_{ji}|\leq\Gamma}} \left\{\underbrace{\sum_{i=1}^{n}\sum_{j\in J} \hat{v}_{ji}w_i - \sum_{i=1}^{n}\sum_{j\in J'} \hat{v}_{ji}w_i}_{=V(\hat{v},w,z)-V(\hat{v},w,z')} + \sum_{i=1}^{n}\sum_{j\in J} \vec{v}_{ji}w_i y_{ji} - \sum_{i=1}^{n}\sum_{j\in J'} \vec{v}_{ji}w_i y_{ji}\right\}$$

$$= \min_{w\in S_w} \left\{V(\hat{v},w,z) - V(\hat{v},w,z') + \min_{\substack{y_{ji}\in[-1,1] \\ \sum|y_{ji}|\leq\Gamma}} \left\{\sum_{i=1}^{n}\sum_{j\in J} \vec{v}_{ji}w_i y_{ji} - \sum_{i=1}^{n}\sum_{j\in J'} \vec{v}_{ji}w_i y_{ji}\right\}\right\}$$

$$= \min_{w\in S_w} \left\{V(\hat{v},w,z) - V(\hat{v},w,z') - \underbrace{\max_{\substack{y_{ji}\in[-1,1] \\ \sum|y_{ji}|\leq\Gamma}} \left\{\sum_{i=1}^{n}\sum_{j\in J} \vec{v}_{ji}w_i(-y_{ji}) + \sum_{i=1}^{n}\sum_{j\in J'} \vec{v}_{ji}w_i y_{ji}\right\}}_{:=\beta(\vec{v},w,z,z',\Gamma)}\right\}.$$

By changing the decision variables $y_{ji}, j \in J$, to $-y_{ji}$,

$$\beta(\vec{v}, w, z, z', \Gamma) = \max_{\substack{y_{ji}\in[-1,1]\\ \sum|y_{ji}|\leq\Gamma}} \left\{ \sum_{i=1}^{n}\sum_{j\in J} \vec{v}_{ji}w_i y_{ji} + \sum_{i=1}^{n}\sum_{j\in J'} \vec{v}_{ji}w_i y_{ji} \right\} = \max_{\substack{y_{ji}\in[-1,1]\\ \sum|y_{ji}|\leq\Gamma}} \left\{ \sum_{i=1}^{n}\sum_{j\in J\cup J'} \vec{v}_{ji}w_i y_{ji} \right\}$$

since $J \cap J' = \emptyset$. The objective function coefficients $\vec{v}_{ji}w_i$ of this maximization problem are non-negative. Therefore, the optimum is always found at y such that $y_{ji} \geq 0$ for all $j \in J \cap J', i \in \{1, ..., n\}$. To show that the optimum of

$$\min_{w\in S_w} \left\{ V(\hat{v}, w, z) - V(\hat{v}, w, z') - \beta(\vec{v}, w, z, z', \Gamma) \right\}$$

$$= \min_{w\in S_w} \min_{\substack{y_{ji}\in[0,1]\\ \sum|y_{ji}|\leq\Gamma}} \left\{ \underbrace{V(\hat{v}, w, z) - V(\hat{v}, w, z') - \sum_{i=1}^{n}\sum_{j\in J\cup J'} \vec{v}_{ji}w_i y_{ji}}_{:=\alpha(w,y)} \right\}$$

is always found at some point in $W = \text{ext}(\text{conv}(S_w))$, take any $w' \in S_w \setminus W$ and let

$$y^* \in \arg\max_{y} \left\{ \alpha(w', y) \mid y \in [0,1]^{m\times n}, \sum |y_{ji}| \leq \Gamma \right\}.$$

Since $\alpha(w, y)$ is linear in w, there exists $w'' \in W$ such that $\alpha(w'', y^*) \leq \alpha(w', y^*)$. Hence,

$$\min_{\substack{y_{ji}\in[0,1]\\ \sum|y_{ji}|\leq\Gamma}} \alpha(w'', y) \leq \alpha(w'', y^*) \leq \alpha(w', y^*) = \min_{\substack{y_{ji}\in[0,1]\\ \sum|y_{ji}|\leq\Gamma}} \alpha(w', y).$$

We thus have established that $V(v, w, z) \geq V(v, w, z')$ for all $(v, w) \in S$ iff

$$\min_{w\in W} \left\{ V(\hat{v}, w, z) - V(\hat{v}, w, z') - \beta(\vec{v}, w, z, z', \Gamma) \right\} \geq 0. \tag{B.1}$$

Part 2. The condition $V(v, w, z) > V(v, w, z')$ for some $(v, w) \in S$ holds iff the optimum of

$$\max_{(v,w)\in S} \left\{ V(v, w, z) - V(v, w, z') \right\}$$

is strictly positive. Following the same steps as in Part 1, this problem is given by

$$= \max_{w\in S_w} \left\{ V(\hat{v}, w, z) - V(\hat{v}, w, z') + \max_{\substack{y\in[-1,1]^{m\times n}\\ \sum|y_{ji}|\leq\Gamma}} \left\{ \sum_{i=1}^{n}\sum_{j\in J} \vec{v}_{ji}w_i y_{ji} - \sum_{i=1}^{n}\sum_{j\in J'} \vec{v}_{ji}w_i y_{ji} \right\} \right\}$$

$$= \max_{w\in S_w} \left\{ V(\hat{v}, w, z) - V(\hat{v}, w, z') + \underbrace{\max_{\substack{y\in[-1,1]^{m\times n}\\ \sum|y_{ji}|\leq\Gamma}} \left\{ \sum_{i=1}^{n}\sum_{j\in J} \vec{v}_{ji}w_i (y_{ji}) + \sum_{i=1}^{n}\sum_{j\in J'} \vec{v}_{ji}w_i (-y_{ji}) \right\}}_{:=\beta'(\vec{v}, w, z, z', \Gamma)} \right\}.$$

By changing the decision variables $y_{ji}, j \in J'$, to $-y_{ji}$,

$$\beta'(\vec{v}, w, z, z', \Gamma) = \max_{\substack{y\in[-1,1]^{m\times n} \\ \sum|y_{ji}|\le\Gamma}} \left\{ \sum_{i=1}^{n}\sum_{j\in J} \vec{v}_{ji} w_i y_{ji} + \sum_{i=1}^{n}\sum_{j\in J'} \vec{v}_{ji} w_i y_{ji} \right\} = \max_{\substack{y\in[0,1]^{m\times n} \\ \sum|y_{ji}|\le\Gamma}} \left\{ \sum_{i=1}^{n}\sum_{j\in J\cup J'} \vec{v}_{ji} w_i y_{ji} \right\}$$

since $J \cap J' = \emptyset$ and the objective function coefficients $\vec{v}_{ji} w_i$ are non-negative. Therefore, the optimum is always found at y such that $y_{ji} \geq 0$ for all $i \in \{1, ..., n\}, j \in J \cap J'$ and $\beta'(\vec{v}, w, z, z', \Gamma) = \beta(\vec{v}, w, z, z', \Gamma)$. Thus, the condition $V(v, w, z) > V(v, w, z')$ for some $(v, w) \in S$ holds iff

$$\max_{w\in W} \left\{ V(\hat{v}, w, z) - V(\hat{v}, w, z') + \beta(\vec{v}, w, z, z', \Gamma) \right\} > 0. \tag{B.2}$$

Part 3. We have established

$$z \succ_S z' \Leftrightarrow \begin{cases} \min_{w\in W} \{V(\hat{v}, w, z) - V(\hat{v}, w, z') - \beta(\vec{v}, w, z, z', \Gamma)\} \geq 0 \\ \max_{w\in W} \{V(\hat{v}, w, z) - V(\hat{v}, w, z') + \beta(\vec{v}, w, z, z', \Gamma)\} > 0 \end{cases}. \tag{B.3}$$

Finally, we show that these conditions hold iff

$$\begin{matrix} \min_{w\in W} \{V(\hat{v}, w, z) - V(\hat{v}, w, z') - \beta(\vec{v}, w, z, z', \Gamma)\} \geq 0 \\ \max_{w\in W} \{V(\hat{v}, w, z) - V(\hat{v}, w, z')\} > 0 \end{matrix}. \tag{B.4}$$

(B.3)⇒(B.4): Assume (B.3) holds. First, if $\beta(\vec{v}, w, z, z', \Gamma) = 0$ for all $w \in W$ then

$$0 < \max_{w\in W} \left\{ V(\hat{v}, w, z) - V(\hat{v}, w, z') + \beta(\vec{v}, w, z, z', \Gamma) \right\} = \max_{w\in W} \left\{ V(\hat{v}, w, z) - V(\hat{v}, w, z') \right\}.$$

Second, if there exists $w' \in W$ such that $\beta(\vec{v}, w', z, z', \Gamma) > 0$ then

$$0 \leq V(\hat{v}, w', z) - V(\hat{v}, w', z') - \beta(\vec{v}, w', z, z', \Gamma) < V(\hat{v}, w', z) - V(\hat{v}, w', z')$$

$$V(\hat{v}, w', z) - V(\hat{v}, w', z') \leq \max_{w\in W} \left\{ V(\hat{v}, w, z) - V(\hat{v}, w, z') \right\}.$$

(B.4)⇒(B.3). Assume (B.4) holds. Then

$$0 < \max_{w\in W} \left\{ V(\hat{v}, w, z) - V(\hat{v}, w, z') \right\} \leq \max_{w\in W} \left\{ V(\hat{v}, w, z) - V(\hat{v}, w, z') + \beta(\vec{v}, w, z, z', \Gamma) \right\}$$

since $\beta(\vec{v}, w, z, z', \Gamma) \geq 0$ for any w. □

Proof of Lemma 3.2.1

Let $z, z', z'' \in \{0,1\}^m$, $S = (S_v^\Gamma \times S_w)$, and $W = \text{ext}(\text{conv}(S_w))$. Asymmetry, $z \not\succ z$, and irreflexivity, $z \succ z' \Rightarrow z' \not\succ z$, follow directly from

$$z \succ z' \Leftrightarrow \begin{cases} V(v,w,z) \geq V(v,w,z') & \forall (v,w) \in S \\ V(v,w,z) > V(v,w,z') & \exists (v,w) \in S \end{cases}.$$

Transitivity, $z \succ z' \wedge z' \succ z'' \Rightarrow z \succ z''$, can be shown by considering Theorem 3.2.1 for $z \succ_S z'$, $z' \succ_S z''$.

$$z \succ_S z' \Leftrightarrow \begin{cases} V(\hat{v},w,z) \geq V(\hat{v},w,z') + \beta(\vec{v},w,z,z',\Gamma) & \forall w \in W \\ V(\hat{v},w,z) > V(\hat{v},w,z') & \exists w \in W \end{cases}$$

$$z' \succ_S z'' \Leftrightarrow \begin{cases} V(\hat{v},w,z') \geq V(\hat{v},w,z'') + \beta(\vec{v},w,z',z'',\Gamma) & \forall w \in W \\ V(\hat{v},w,z') > V(\hat{v},w,z'') & \exists w \in W \end{cases}$$

$$\Leftrightarrow \begin{cases} V(\hat{v},w,z) \geq V(\hat{v},w,z'') + \beta(\vec{v},w,z,z',\Gamma) + \beta(\vec{v},w,z',z'',\Gamma) & \forall w \in W \\ V(\hat{v},w,z) > V(\hat{v},w,z'') & \exists w \in W \end{cases}$$

For $z \succ_S z''$ by Theorem 3.2.1 to hold, it suffices to show

$$\beta(\vec{v},w,z,z',\Gamma) + \beta(\vec{v},w,z',z'',\Gamma) \geq \beta(\vec{v},w,z,z'',\Gamma) \quad \forall w \in W.$$

$J(z,z'') \subseteq J(z,z') \cup J(z',z'')$ holds for the 0-1 knapsack problems presented by $\beta(\vec{v},w,z,z',\Gamma)$, $\beta(\vec{v},w,z',z'',\Gamma)$ and $\beta(\vec{v},w,z,z'',\Gamma)$, sharing the same budget limit Γ. Thus, the optimum value $\beta(\vec{v},w,z,z'',\Gamma)$ of decision set $J(z,z'')$ is obtainable within $J(z,z') \cup J(z',z'')$ by combining $\beta(\vec{v},w,z,z',\Gamma)$, and $\beta(\vec{v},w,z',z'',\Gamma)$. □

Proof of Corollary 3.2.1

By Definition 3.2.1, $S_v^\Gamma = \left\{ v \in \mathbb{R}^{m\times n} \mid v_{ji} = \hat{v}_{ji} + \vec{v}_{ji} y_{ji}, y \in [-1,1]^{m\times n}, \sum_{i=1}^n \sum_{j=1}^m |y_{ji}| \leq \Gamma \right\}$.
Part (i) For $\Gamma = mn$ the inequality $\sum_{i=1}^n \sum_{j=1}^m |y_{ji}| \leq mn$ always holds for $y \in [-1,1]^{m\times n}$ and may be omitted. The resulting expression

$$S_v^{mn} = \left\{ v \in \mathbb{R}^{m\times n} \mid v_{ji} = \hat{v}_{ji} + \vec{v}_{ji} y_{ji}, y \in [-1,1]^{m\times n} \right\}$$

is equivalent to S_v^∞ by Equation 3.5. Part (i) then follows from Definition 3.2.2.

Part (ii) For $\Gamma = 0$ the inequality $\sum_{i=1}^n \sum_{j=1}^m |y_{ji}| \leq 0$ only holds for $y_{ji} = 0$ with $i = 1, \ldots, n$ and $j = 1, \ldots, m$, reducing the information set of project scores to a point

$$S_v^0 = \left\{ v \in \mathbb{R}^{m\times n} \mid v_{ji} = \hat{v}_{ji} \right\} = \{\hat{v}\}.$$

Part (ii) then follows from Definition 3.2.2 for information set $S = (\{\hat{v}\} \times S_w)$. □

Proof of Theorem 3.2.2

Let information sets $S = (S_v^\Gamma \times S_w)$ and $S' = (S_v^{\Gamma'} \times S_w)$ as well as $W = \text{ext}(\text{conv}(S_w))$. Assume $Z_N(S') \not\subseteq Z_N(S)$ and $z' \in Z_N(S'), z' \notin Z_N(S)$. Then $\exists z \in Z_N(S)$ so that $z \succ_S z'$ and $z \not\succ_{S'} z'$.

$$\begin{array}{lll} z \succ_S z' & \Rightarrow & V(\hat{v},w,z) \geq V(\hat{v},w,z') + \beta(\vec{v},w,z,z',\Gamma) \quad \forall w \in W \\ z \not\succ_{S'} z' & \Rightarrow & V(\hat{v},w,z) < V(\hat{v},w,z') + \beta(\vec{v},w,z,z',\Gamma') \quad \exists w \in W \\ z \succ_S z' \wedge z \not\succ_{S'} z' & \Rightarrow & \beta(\vec{v},w,z,z',\Gamma') > \beta(\vec{v},w,z,z',\Gamma) \qquad \exists w \in W. \end{array}$$

For the last expression to hold, there must exist weights $w \in W$ such that

$$\max_{y \in [0,1]^{m\times n}} \left\{ \sum_{i=1}^{n} \sum_{j \in J(z,z')} \vec{v}_{ji} w_i y_{ji} | \sum_{i=1}^{n}\sum_{j=1}^{m} y_{ji} \leq \Gamma' \right\} > \max_{y \in [0,1]^{m\times n}} \left\{ \sum_{i=1}^{n} \sum_{j \in J(z,z')} \vec{v}_{ji} w_i y_{ji} | \sum_{i=1}^{n}\sum_{j=1}^{m} y_{ji} \leq \Gamma \right\}.$$

The maximization problem on the left-hand side is identical to the one on the right-hand side except for the tighter constraint $\Gamma' \leq \Gamma$. The inequality $\beta(\vec{v},w,z,z',\Gamma') > \beta(\vec{v},w,z,z',\Gamma)$ cannot be true. □

Proof of Theorem 3.3.1

Part (i) Applying Theorem (2) of Liesiö et al. (2007), $Z_N(S') \subseteq Z_N(S)$ holds. As by Theorem 3.2.2 $Z_N(S_{v'}^\Gamma \times S_{w'}) \subseteq Z_N(S')$ holds, $Z_N(S_{v'}^\Gamma \times S_{w'}) \subseteq Z_N(S)$ holds as well.

Part (ii) By Theorem 3.2.2 $Z_N(S_{v'}^{\Gamma'} \times S_{w'}) \subseteq Z_N(S_v^\Gamma \times S_w)$ follows from $Z_N(S_{v'}^\Gamma \times S_{w'}) \subseteq Z_N(S_v^\Gamma \times S_w)$. Applying Theorem (2) of Liesiö et al. (2007), $Z_N(S_{v'}^\Gamma \times S_{w'}) \subseteq Z_N(S_v^\Gamma \times S_w)$ holds if $S_{w'} \subseteq S_w$, $S_{v'}^\Gamma \subseteq S_{v'}^\Gamma$ and $\text{int}(S_v^\Gamma \times S_w) \cap S_{v'}^\Gamma \times S_{w'} \neq \emptyset$. Assume $S_{v'}^\Gamma \not\subseteq S_v^\Gamma$ and $\exists v^* \in S_{v'}^\Gamma$, $v^* \notin S_v^\Gamma$.

$$\nexists y \in [-1,1]^{m\times n}, \sum_{i=1}^{n}\sum_{j=1}^{m} |y_{ji}| \leq \Gamma \quad \hat{v}_{ji} + \vec{v}_{ji} y_{ji} = v^*_{ji} \qquad \forall i = 1,\ldots,n \quad j = 1,\ldots,m$$

$$\nexists y \in [-1,1]^{m\times n}, \sum_{i=1}^{n}\sum_{j=1}^{m} |y_{ji}| \leq \Gamma \quad \hat{v}_{ji} + \vec{v}_{ji} y_{ji} = \hat{v}'_{ji} + \vec{v}'_{ji} y^*_{ji} \qquad \forall i = 1,\ldots,n \quad j = 1,\ldots,m$$

With $\hat{v}'_{ji} = \hat{v}_{ji}$:

$$\nexists y \in [-1,1]^{m\times n}, \sum_{i=1}^{n}\sum_{j=1}^{m} |y_{ji}| \leq \Gamma \qquad \vec{v}_{ji} y_{ji} = \vec{v}'_{ji} y^*_{ji} \qquad \forall i = 1,\ldots,n \quad j = 1,\ldots,m$$

$$\nexists y \in [-1,1]^{m\times n}, \sum_{i=1}^{n}\sum_{j=1}^{m} |y_{ji}| \leq \Gamma \qquad y_{ji} = \frac{\vec{v}'_{ji}}{\vec{v}_{ji}} y^*_{ji} \qquad \forall i = 1,\ldots,n \quad j = 1,\ldots,m$$

$\frac{\vec{v}'}{\vec{v}} \in [0,1]^{m \times n}$ as $\vec{v}'_{ji} \leq \vec{v}_{ji} \forall i = 1, \ldots, n \quad j = 1, \ldots, m$. By definition, any y^* falls in the domain $y^* \in [-1,1]^{m \times n}$ and obeys $\sum_{i=1}^{n} \sum_{j=1}^{m} |y^*_{ji}| \leq \Gamma$. Both the domain and limiting condition also hold for any term $\frac{\vec{v}'}{\vec{v}} y^*$ with $\frac{\vec{v}'}{\vec{v}} \in [0,1]^{m \times n}$. There will always

$$\exists y \in [-1,1]^{m \times n}, \sum_{i=1}^{n} \sum_{j=1}^{m} |y_{ji}| \leq \Gamma \qquad y_{ji} = \frac{\vec{v}'_{ji}}{\vec{v}_{ji}} y^*_{ji} \qquad \forall i = 1, \ldots, n \quad j = 1, \ldots, m,$$

disproving the assumption. Thus $S^{\Gamma}_{v'} \subseteq S^{\Gamma}_{v}$, by Part (i) $S^{\Gamma'}_{v'} \subseteq S^{\Gamma}_{v}$, and by Theorem (2) of Liesiö et al. (2007) Part (ii) holds. □

Proof of Lemma 3.3.1

Let $z, z', z'' \in \{0,1\}^m$ and $W = \text{ext}(\text{conv}(S_w))$. From Theorem 3.3.1 Part (ii) it follows that $Z_N(S') \subseteq Z_N(S)$. Assume contrary to the claim that $\exists z' \in Z_N(S), z' \notin Z_N(S')$. Then $\exists z \in Z_N(S')$ so that

$$z \succ_{S'} z' \Leftrightarrow \begin{cases} \Delta(\hat{v}', w, z, z') \geq \beta(\vec{v}', w, z, z', \Gamma) & \forall w \in W \\ \Delta(\hat{v}', w, z, z') > 0 & \exists w \in W \end{cases},$$

where

$$\Delta(\hat{v}', w, z, z') = V(\hat{v}', w, z) - V(\hat{v}', w, z') = \sum_{i=1}^{n} \sum_{j \in J} \hat{v} w_i - \sum_{i=1}^{n} \sum_{j \in J'} \hat{v} w_i,$$

$J = \left\{ j \in \{1, ..., m\} \,|z_j = 1, z'_j = 0 \right\}$, $J' = \left\{ j \in \{1, ..., m\} \,|z_j = 0, z'_j = 1 \right\}$, and

$$\beta(\vec{v}', w, z, z', \Gamma) = \max_{y \in [0,1]^{m \times n}} \left\{ \sum_{i=1}^{n} \sum_{j \in J(z,z')} \vec{v}'_{ji} w_i y_{ji} | \sum_{i=1}^{n} \sum_{j=1}^{m} y_{ji} \leq \Gamma \right\}.$$

By definition, $J(z,z')$ in $\beta(\vec{v}', w, z, z', \Gamma)$ considers only projects j not included in both z and z', i.e., $C_j(S) \in]0,1[$, for which by assumption $\vec{v}'_{ji} = \vec{v}_{ji}$. This implies

$$\beta(\vec{v}', w, z, z', \Gamma) = \beta(\vec{v}, w, z, z', \Gamma).$$

$\Delta(\hat{v}', w, z, z')$ is likewise only determined by projects $j \in J \cup J' = J(z,z')$ so that

$$\Delta(\hat{v}', w, z, z') = \Delta(\hat{v}, w, z, z').$$

Therefore,

$$\begin{matrix} \Delta(\hat{v}, w, z, z') \geq \beta(\vec{v}, w, z, z', \Gamma) & \forall w \in W \\ \Delta(\hat{v}, w, z, z') > 0 & \exists w \in W \end{matrix} \Leftrightarrow z \succ_S z'$$

holds. $Z_N(S') \subseteq Z_N(S)$ implies $z \in Z_N(S)$ rendering $z' \in Z_N(S)$ invalid. □

Proof of Lemma 3.5.1

$\beta(\vec{v}, w, z, z', \Gamma)$ in Theorem 3.2.1 dominance is given by

$$\begin{aligned}\beta(\vec{v}, w, z, z', \Gamma) &= \max_{y \in [0,1]^{m \times n}} \left\{ \sum_{i=1}^{n} \sum_{j \in J(z,z')} w_i \vec{v}_{ji} y_{ji} | \sum_{i=1}^{n} \sum_{j=1}^{m} y_{ji} \leq \Gamma \right\} \\ &= \max_{y \in [-1,1]^{m \times n}} \left\{ \sum_{i=1}^{n} \sum_{j=1}^{m} w_i \vec{v}_{ji} y_{ji} ({z_j}' - z_j) | \sum_{i=1}^{n} \sum_{j=1}^{m} y_{ji} \leq \Gamma \right\}.\end{aligned}$$

$\beta^I(\vec{v}, w, z, z', \Gamma)$ only differs from this expression by additional constraints $A^I Y \leq B^I$. It follows that the objective value of maximization problem $\beta(\vec{v}, w, z, z', \Gamma)$ is no smaller than the objective value of $\beta^I(\vec{v}, w, z, z', \Gamma)$,

$$\beta(\vec{v}, w, z, z', \Gamma) \geq \beta^I(\vec{v}, w, z, z', \Gamma).$$

For arbitrary $z, z' \in \{0,1\}^m$

$$z \succ_{S_v^\Gamma \times S_w} z' \Rightarrow z \succ_{I_v^\Gamma \times S_w} z'$$

holds, which implies $Z_N(I_v^\Gamma \times S_w) \subseteq Z_N(S_v^\Gamma \times S_w)$. □

Proof of Lemma 3.5.2

The Lemma follows directly from Theorem 1 of Bertsimas and Sim (2003). □

Proof of Lemma A.0.1

Assume $z', z'' \in Z^s$ such that $z' \succ_S z''$ and $Az' \leq Az''$. Take any $z \in \{0,1\}^m$ and denote $\tilde{z}'' = [z_1'', \ldots, z_s'', z_{s+1}, \ldots, z_m]^T$. If $\tilde{z}'' \notin Z_F$ then $\tilde{z}'' \notin Z_N(S)$. In turn, if $\tilde{z}'' \in Z_F$ then

$$B \geq A\tilde{z}'' = Az'' + A \begin{bmatrix} 0 \\ \vdots \\ 0 \\ z_{s+1} \\ \vdots \\ z_m \end{bmatrix} \geq Az' + A \begin{bmatrix} 0 \\ \vdots \\ 0 \\ z_{s+1} \\ \vdots \\ z_m \end{bmatrix} = A \underbrace{\begin{bmatrix} z_1' \\ \vdots \\ z_s' \\ z_{s+1} \\ \vdots \\ z_m \end{bmatrix}}_{=\tilde{z}'},$$

which implies $\tilde{z}' \in Z_F$. Furthermore, for any $(v,w) \in S$ the value difference of portfolios $\tilde{z}'$ and $\tilde{z}''$ is equal to

$$V(v,w,\tilde{z}') - V(v,w,\tilde{z}'') = \tilde{z}'^T vw - \tilde{z}''^T vw = (z'^T + \begin{bmatrix} 0 \\ \vdots \\ 0 \\ z_{s+1} \\ \vdots \\ z_m \end{bmatrix})vw - (z''^T + \begin{bmatrix} 0 \\ \vdots \\ 0 \\ z_{s+1} \\ \vdots \\ z_m \end{bmatrix})vw$$

$$= z'^T vw - z''^T vw = V(v,w,z') - V(v,w,z''),$$

i.e., the value difference of portfolios z' and z''. Hence, $z' \succ_S z''$ implies $\tilde{z}' \succ_S \tilde{z}''$ and thus $\tilde{z}'' \notin Z_N(S)$. □

Proof of Lemma A.0.2

Let information set $S = (S_v^\Gamma \times S_w)$ and $W = \text{ext}(\text{conv}(S_w))$. Assume $z' \in Z_F$ and $z'' \in Z^s$ such that

$$V(\hat{v},w,z') > V(\hat{v},w,z'') + UB_s(w,z',z'') \ \forall\ w \in W.$$

For any $w \in W$

$$V(\hat{v},w,z') \quad > \quad \max_{\substack{y_{ji}\in[0,1] \\ z_j\in\{0,1\}}} \sum_{i=1}^{n}\sum_{j=1}^{s} \hat{v}_{ji} w_i z''_j + \sum_{i=1}^{n}\sum_{j=s+1}^{m} \hat{v}_{ji} w_i z_j + \sum_{i=1}^{n}\sum_{j=1}^{m} \vec{v}_{ji} w_i y_{ji} \tag{B.5}$$

$$Az'' + Az \leq B \tag{B.6}$$

$$\sum_{i=1}^{n}\sum_{j=1}^{m} y_{ji} \leq \Gamma \tag{B.7}$$

$$y_{ji} = 0 \ \forall\ i \in \{1,\ldots,n\}, j \in \left\{ j \in \{1,...,s\} \,|z'_j + z''_j \neq 1 \right\} \tag{B.8}$$

$$y_{ji} \leq 2 - z_j - z'_j \ \forall\ i \in \{1,\ldots,n\}, j \in \{s+1,...,m\} \tag{B.9}$$

$$y_{ji} \leq z_j + z'_j \ \forall\ i \in \{1,\ldots,n\}, j \in \{s+1,...,m\}. \tag{B.10}$$

By denoting $\tilde{Z} = \left\{ \tilde{z} \in \{0,1\}^m \,|\tilde{z}_1 = z''_1, \ldots, \tilde{z}_s = z''_s \right\}$, the mixed integer linear programming problem (B.5) - (B.10) can be written as

$$\max_{\substack{\tilde{z}\in\tilde{Z} \\ A\tilde{z}\leq B}} \left\{ V(\hat{v},w,\tilde{z}) + \max_{y_{ji}\in[0,1]} \left\{ \sum_{i=1}^{n}\sum_{j=1}^{m} \vec{v}_{ji} w_i y_{ji} \,\Big|\, \begin{matrix} \sum_{i=1}^{n}\sum_{j=1}^{m} y_{ji} \leq \Gamma \\ y_{ji} = 0 \text{ if } z'_j + \tilde{z}_j \neq 1 \end{matrix} \right\} \right\}$$

$$= \quad \max_{\tilde{z}\in\tilde{Z}\cap Z_F} \left\{ V(\hat{v},w,\tilde{z}) + \max_{y_{ji}\in[0,1]} \left\{ \sum_{i=1}^{n}\sum_{j\in J(z',\tilde{z})} \vec{v}_{ji} w_i y_{ji} \,\Big|\, \sum_{i=1}^{n}\sum_{j=1}^{m} y_{ji} \leq \Gamma \right\} \right\},$$

where $J(z',\tilde{z}) = \left\{ j \in \{1,\dots,m\} \,|\, (z'_j = 1, \tilde{z} = 0) \vee (z'_j = 0, \tilde{z} = 1) \right\}$. For any $w \in W$

$$V(\hat{v}, w, z') > \max_{\tilde{z} \in \tilde{Z} \cap Z_F} \left\{ V(\hat{v}, w, \tilde{z}) + \beta(\vec{v}, w, z', \tilde{z}) \right\},$$

which by Theorem 3.2.1 implies that $z' \succ_S \tilde{z}$ for any $\tilde{z} \in \tilde{Z} \cap Z_F$. Hence, $\tilde{Z} \cap Z_N(S) = \emptyset$.
□

C. Experimental Instructions

We give the English translation of the originally German subject instructions for the conducted experiments. The instructions for the first and second experiment are identical except for the stated number of items in the knapsack problems. An illustrative screenshot and its description included in the original instructions are omitted.

1. General information

You are about to participate in an experiment in decision making. In the course of the experiment, you can earn a considerable amount of money depending on how good your decisions are. In the experiment, all monetary amounts are specified in Experimental Currency Units (ECU), which are converted according a fixed exchange rate to € at the end of the experiment (see experimental payout). All your decisions and answers will be treated confidentially. Please read the following instructions carefully. Should you have any questions, please ask. During the experiment you have to switch off your cell phone, and communication with other participants is prohibited.

2. Experimental task and procedure

A set of items is given, and each item generates a value but requires a capacity. Your task is to select a subset of items given that a higher aggregate value results in a higher payout while the aggregate resource requirement must not exceed the available capacity.

The experiment consists of several independent rounds with different number of items (5, 10, 15, or 25), different item properties, and different capacities. Every round consists of a single screen displaying all items in a table containing information about the item properties (value and resource requirement) as illustrated in Figure C.1. For each item, you can decide to select it from the list and you are free to deselect already selected items at any time. Furthermore, the remaining capacity as well as the value of the portfolio is displayed on the screen. Please note that if the selected item results in an aggregate resource requirement exceeding the available capacity an error message will appear. Please take sufficient time to make your decisions , and once you have made your selection press the continue button to go to the next round. At most, you have 5

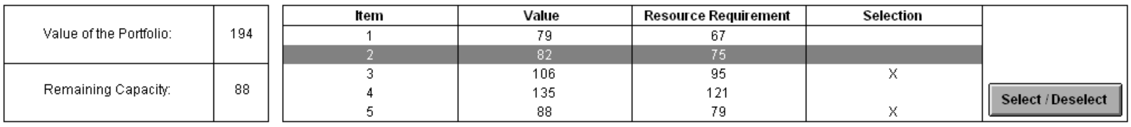


Figure C.1.: Illustration of the interface presented to subjects

minutes to complete each round, and when the time is over you will be automatically taken to the next screen. The experiment ends after 35 minutes.

Before the main rounds start, there are three training rounds and you have to fill out a short questionnaire at the end of the experiment. In total the experiment will take about 60 minutes.

3. Experimental payout

In each round the aggregate value of all selected items in ECU is converted by a linear exchange-rate to €. The exchange-rate is round-dependent and displayed on the screen. At the end of the experiment, one round out of all completed rounds is randomly chosen for payout. For the payout a fixed amount of €100 is subtracted from the aggregate € value in this round. In the unlikely case that the resulting payout is less than €3, you still receive a minimum of €3.

D. Abbreviations and Symbols

Abbreviations

ECU Experimental currency units
FCFS First-come-first-served priority rule
GA Genetic algorithm
GRPW Greatest rank positional weight priority rule
LEPT Longest expected processing time priority rule
LNRJ Least non-related jobs priority rule
MaxD Maximum difference between value and resource requirement heuristic
MaxR Maximum ratio of value divided by resource requirement heuristic
MAXTWK Maximum total work content priority rule
MaxV Maximum value heuristic
MINEST The minimum earliest start time priority rule
MinK Minimum resource requirement heuristic
MINLFT Minimum latest finish time priority rule
MINLST Minimum latest start time priority rule
MINSLK Minimum slack priority rule
MINTWK Minimum total work content priority rule
MIS Maximum immediate successors priority rule
MTS Maximum transitive successors priority rule
ORSEE Online recruitment system for economic experiments
OS Order strength metric
RCPSP Resource-constrained project scheduling problem
RF Resource factor metric
RPM Robust portfolio modeling
RRS Regret-based biased random sampling
RS Resource strength metric
SAA Sample average approximation
SEPT Longest expected processing time priority rule
SRCPSP Stochastic resource-constrained project scheduling problem

Symbols in Chapter 2

Indexes

$i = 1, \dots, n$ Evaluation criteria
$j = 1, \dots, m$ Projects
$s = 1, \dots, o$ Synergies

Sets

$\mathcal{E}$ Set of logical interdependencies (j, j'), where $j, j' \in \{1, \dots, m\}$
$\mathcal{J}_s$ Set of projects required for the activation of synergy $s = 1, \dots, o$

Parameters

$B \in \mathbb{R}_+$ Budget limit
$c \in \mathbb{R}_+^m$ Vector of cost of projects $j = 1, \dots, m$
$v \in \mathbb{R}_+^{m \times n}$ Matrix of risk-adjusted scores of projects $j = 1, \dots, m$ evaluated with regards to criteria $i = 1, \dots, n$
$v^{\mathbf{asp}}, v^{\mathbf{max}} \in \mathbb{R}_+^n$ Vector of aspiration levels and maximum achievable scores with regards to criteria $i = 1, \dots, n$
$v^S \in \mathbb{R}_+^{o \times n}$ Matrix of risk-adjusted scores of synergies $s = 1, \dots, o$ evaluated with regards to criteria $i = 1, \dots, n$
$w \in \mathbb{R}_+^n$ Vector of weights of score criteria $i = 1, \dots, n$

Variables

$x \in \{0, 1\}^m$ Vector of binary variables modeling a portfolio of projects $j = 1, \dots, m$
$y \in \{0, 1\}^o$ Vector of binary variables modeling the activation of synergies $s = 1, \dots, o$

Symbols in Chapter 3

Indexes

$i = 1, \ldots, n$ Evaluation criteria
$j = 1, \ldots, m$ Projects
$k = 1, \ldots, r$ Score interdependency constraints
$l = 1, \ldots, q$ Portfolio selection constraints

Sets

$\mathcal{E}_j$ Set of projects required for the execution of project $j = 1, \ldots, m$
$S \subseteq (S_w^\infty \times S_v^\infty)$ Information set of uncertain criteria weights and project scores
$\mathcal{S}$ Set of synergy projects
$S_A^{\Gamma^A}$ Information set of uncertain coefficients of portfolio selection constraints
$S_v^\infty \supseteq S_v^\Gamma, I_v^\Gamma$ Overall information set of uncertain project scores, adjustable information subset without interdependence, and adjustable information subset subject to interdependence
$S_w^\infty \supseteq S_w$ Overall information set and information subset of uncertain criteria weights
Z_F, Z_F^A Set of feasible portfolios and set of feasible portfolios for uncertain constraint coefficients
$Z_N(S), Z_N^{\Gamma^A}(S)$ Set of portfolios $z \in Z_F$ and set of portfolios $z \in Z_F^A$ non-dominated with regards to information set S

Parameters

$A, \hat{A}, \vec{A}, \tilde{A}, \underline{A}, \overline{A} \in \mathbb{R}^{q \times m}$ Matrix of nominal coefficients a_{lj}, most likely coefficients $\hat{a}_{lj}$, coefficient deviations $\vec{a}_{lj}$, uncertain coefficients $\tilde{a}_{lj}$, lower coefficient value bounds $\underline{a}_{lj}$, and upper coefficient value bounds $\overline{a}_{lj}$ of portfolio selection constraints $l = 1, \ldots, q$
$A^I \in \mathbb{R}^{r \times mn}$ Matrix of coefficients of score interdependency constraints $k = 1, \ldots, r$
$B \in \mathbb{R}^q$ Vector of limits to portfolio selection constraints $l = 1, \ldots, q$
$B^I \in \mathbb{R}^r$ Vector of limits to score interdependency constraints $k = 1, \ldots, r$
$c \in \mathbb{R}_+^m$ Vector of cost of projects $j = 1, \ldots, m$
$v, \hat{v}, \vec{v}, \tilde{v} \in \mathbb{R}^{m \times n}$ Matrices of nominal scores, most likely scores, score deviations, and uncertain scores of projects $j = 1, \ldots, m$ evaluated with regards to criteria $i = 1, \ldots, n$
$v^{\mathbf{asp}} \in \mathbb{R}_+^n$ Vector of aspiration level values with regards to score criteria $i = 1, \ldots, n$
$w \in \mathbb{R}^n$ Vector of weights of score criteria $i = 1, \ldots, n$
$\Gamma \in [0, mn]$ Score deviation limit
$\Gamma^A \in [0, m]$ Constraint coefficient deviation limit
η Matrix of identically, independently, and symmetrically distributed random variables with support $[-1, 1]$ modeling score deviations of projects $j = 1, \ldots, m$ evaluated with regards to criteria $i = 1, \ldots, n$

Variables

$y \in [-1,1]^{m \times n}, Y \in [-1,1]^{mn}$ Matrix and vector of continuous variables modeling value realizations of projects $j = 1, \ldots, m$ evaluated with regards to criteria $i = 1, \ldots, n$

$z \in \{0,1\}^m$ Vector of binary variables modeling a portfolio of projects $j = 1, \ldots, m$

Functions

$C(S)$ Core indexes of projects $j = 1, \ldots, m$ for information set S

$V(v, w, z)$ Linear-additive value of portfolio $z \in \{0,1\}^m$ with regards to weight vector $w \in \mathbb{R}^n$ and project score matrix $v \in \mathbb{R}^{m \times n}$

$J(z, z')$ Set of projects either included in portfolio $z \in \{0,1\}^m$ or $z' \in \{0,1\}^m$, but not in both

$\beta(\vec{v}, w, z, z', \Gamma), \beta^I(\vec{v}, w, z, z', \Gamma)$ Maximum deviation in value between portfolios $z \in \{0,1\}^m$ and $z' \in \{0,1\}^m$ when considering score deviations $\vec{v} \in \mathbb{R}^{m \times n}$, criteria weights $w \in \mathbb{R}^n$, and deviation limit $\Gamma \in [0, mn]$

$\beta(\vec{A}, z, \Gamma^A)$ Maximum deviation in constraint value for portfolio $z \in \{0,1\}^m$ when considering constraint coefficient deviations $\vec{A} \in \mathbb{R}^{q \times m}$ and deviation limit $\Gamma^A \in [0, m]$

Symbols in Chapter 4

Indexes

$j = 1, \dots, m$ Items

$s = 1, \dots, n$ Selection steps of a subject or constructive heuristic

Sets

$\mathcal{A}_s \supseteq \mathcal{A}_s^{b,f}(j_{s-1})$ Set of items $j = 1, \dots, m$ that have not been previously selected and the selection of which does not exceed the resource capacity in selection step $s = 1, \dots, n$ considering global selection behavior and considering localized selection behavior

Parameters

$b, f \in [2, n]$ Backward and forward search range

$b_1 \in \mathbb{R}$ Regression line slope

$c \in \mathbb{R}_+$ Resource capacity

$j_s \in \{1, \dots, m\}^n$ Vector of subject selection in steps $s = 1, \dots, n$

$k \in \mathbb{R}_+^m$ Vector of required resources of items $j = 1, \dots, m$

$p \in [0, 1]$ Observed level of significance

$p_{\mathbf{ran}} \in \mathbb{R}_+$ Expected portfolio value if items are randomly selected

$p_{\mathbf{sub}} \in \mathbb{R}_+$ Portfolio value obtained by a subject

$r \in \mathbb{N}$ Value range limit for instance generation

$R^2 \in [0, 1]$ Coefficient of determination

$v \in \mathbb{R}_+^m$ Vector of values of items $j = 1, \dots, m$

Variables

$x \in \{0, 1\}^m$ Vector of binary variables modeling a portfolio of projects $j = 1, \dots, m$

Functions

$A_{\mathbf{abs}}$ Absolute heuristic adherence

$A_{\mathbf{rel}}, A_{\mathbf{rel}}^{b,f}$ Relative heuristic adherence considering global selection behavior and considering localized selection behavior

$h(\mathcal{A}_s), h(\mathcal{A}_s^{b,f}(j_{s-1}))$ Highest ranked item according to some evaluation criterion from a set of items $\mathcal{A}_s$ or $\mathcal{A}_s^{b,f}(j_{s-1})$

$\alpha(s), \alpha^{b,f}(s)$ Indicator whether subject selection in step $s = 1, \dots, n$ is in line with a constructive heuristic for global selection behavior and for localized selection behavior

Symbols in Chapter 5

Indexes

$i = 1, \dots, n$ Project activities
$k = 1, \dots, m$ Resource types
$t \in \mathbb{R}_+$ Points in time

Sets

$(i, i') \in \mathcal{E}$ Set of precedence relations between project activities $i, i' \in \{1, \dots, n\}$

Parameters

$\mathbf{d} = (\mathbf{d}_1, \dots, \mathbf{d}_n)^T$ Duration distributions of activities $i = 1, \dots, n$
$d \in \mathbb{R}_+^n$ Vector of durations of activities $i = 1, \dots, n$
L Priority list
$p \in [0, 1]$ Observed level of significance
$r \in \mathbb{N}^{n \times m}$ Matrix of resource requirements of activities $i = 1, \dots, n$ with regards to resource types $k = 1, \dots, m$
$R \in \mathbb{N}^m$ Vector of per-period availabilities of resource types $k = 1, \dots, m$
$\lambda \in \mathbb{R}_+$ Project arrival rate
$\rho \in \mathbb{R}$ Optimum performance measure value associated with optimal policy π^*

Variables

$\pi^*, \pi \in \Pi$ Scheduling policies within class Π with optimal policy π^*

Functions

$u \in \mathbb{R}_+^m$ Utilization levels for resource types $k = 1, \dots, m$
$\overline{u} \in \mathbb{R}_+$ Average utilization level across all resource types
$\kappa(\pi, \mathbf{d})$ Performance measure distribution associated with policy $\pi \in \Pi$ and activity duration distributions $\mathbf{d}$

Bibliography

Adler, Paul S., Avi Mandelbaum, Viên Nguyen, Elizabeth Schwerer. 1995. From Project to Process Management: An Empirically-Based Framework for Analyzing Product Development Time. *Management Science* **41**(3) 458–484.

Anavi-Isakow, S., Boaz Golany. 2003. Managing Multi-Project Environments through Constant Work-in-Process. *International Journal of Project Management* **21**(1) 9–18.

Ashtiani, Behzad, Roel Leus, Mir-Bahador Aryanezhad. 2011. New Competitive Results for the Stochastic Resource-Constrained Project Scheduling Problem: Exploring the Benefits of Pre-Processing. *Journal of Scheduling* **14**(2) 157–171.

Ball, Christopher T., Harvey J. Langholtz, Jacqueline Auble, Barron Sopchak. 1998. Resource-Allocation Strategies: A Verbal Protocol Analysis. *Organizational Behavior and Human Decision Processes* **76**(1) 70–88.

Ballestín, Francisco. 2007. When It Is Worthwhile to Work with the Stochastic RCPSP? *Journal of Scheduling* **10**(3) 153–166.

Ballestín, Francisco, Roel Leus. 2009. Resource-Constrained Project Scheduling for Timely Project Completion with Stochastic Activity Durations. *Production and Operations Management* **18**(4) 459–474.

Bearden, J. Neil, Ryan O. Murphy, Amnon Rapoport. 2008. Decision Biases in Revenue Management: Some Behavioral Evidence. *Manufacturing & Service Operations Management* **10**(4) 625–636.

Bendoly, Elliot, Rachel Croson, Paulo Goncalves, Kenneth L. Schultz. 2010. Bodies of Knowledge for Research in Behavioral Operations. *Production and Operations Management* **19**(4) 434–452.

Bertsimas, Dimitris, David B. Brown, Constantine Caramanis. 2011. Theory and Applications of Robust Optimization. *SIAM Review* **53**(3) 464–501.

Bertsimas, Dimitris, Melvyn Sim. 2003. Robust Discrete Optimization and Network Flows. *Mathematical Programming* **98**(1-3) 49–71.

Bertsimas, Dimitris, Melvyn Sim. 2004. The Price of Robustness. *Operations Research* **52**(1) 35–53.

Bertsimas, Dimitris, John N. Tsitsiklis. 1997. *Introduction to Linear Optimization*. Athena Scientific Series in Optimization and Neural Computation, Athena Scientific, Belmont.

Blazewicz, Jacek, Jan K. Lenstra, Alexander H. G. Rinnooy Kan. 1983. Scheduling Subject to Resource Constraints: Classification and Complexity. *Discrete Applied Mathematics* **5**(1) 11–24.

Booker, Jane M., Maurice C. Bryson. 1985. Decision Analysis in Project Management: An Overview. *IEEE Transactions on Engineering Management* **EM-32**(1) 3–9.

Browning, Tyson R., Ali A. Yassine. 2010. Resource-Constrained Multi-Project Scheduling: Priority Rule Performance Revisited. *International Journal of Production Economics* **126**(2) 212–228.

Busemeyer, Jerome R., Kenneth N. Swenson, Alejandro Lazarte. 1986. An Adaptive Approach to Resource Allocation. *Organizational Behavior and Human Decision Processes* **38**(3) 318–341.

Choi, Jaein, Matthew J. Realff, Jay H. Lee. 2007. A Q-Learning-Based Method Applied to Stochastic Resource Constrained Project Scheduling with New Project Arrivals. *International Journal of Robust and Nonlinear Control* **17**(13) 1214–1231.

Chou, Mabel C., Hui Liu, Maurice Queyranne, David Simchi-Levi. 2006. On the Asymptotic Optimality of a Simple On-Line Algorithm for the Stochastic Single-Machine Weighted Completion Time Problem and Its Extensions. *Operations Research* **54**(3) 464–474.

Chvátal, Vasek. 1980. Hard Knapsack Problems. *Operations Research* **28**(6) 1402–1411.

Cooper, Robert G., Scott J. Edgett, Elko J. Kleinschmidt. 2001. Portfolio Management for New Product Development: Results of an Industry Practices Study. *R&D Management* **31**(4) 361–380.

Cowan, Nelson. 2001. The Magical Number 4 in Short-Term Memory: A Reconsideration of Mental Storage Capacity. *The Behavioral and Brain Sciences* **24**(1) 87–114.

Danielson, Mats, Love Ekenberg, Jim Idefeldt, Aron Larsson. 2007. Using a Software Tool for Public Decision Analysis: The Case of Nacka Municipality. *Decision Analysis* **4**(2) 76–90.

Deblaere, Filip, Erik L. Demeulemeester, Willy S. Herroelen. 2011. Proactive Policies for the Stochastic Resource-Constrained Project Scheduling Problem. *European Journal of Operational Research* **214**(2) 308–316.

Demeulemeester, Erik L., Willy S. Herroelen. 2002. *Project Scheduling: A Research Handbook*, *International Series in Operations Research & Management Science*, vol. 49. Kluwer Academic Publishers, Boston.

Dickinson, Michael W., Anna C. Thornton, Stephen Graves. 2001. Technology Portfolio Management: Optimizing Interdependent Projects over Multiple Time Periods. *IEEE Transactions on Engineering Management* **48**(4) 518–527.

Düzgün, Ruken, Aurélie Thiele. 2010. Robust Optimization with Multiple Ranges: Theory and Application to R&D Project Selection. URL `http://www.optimization-online.org/`.

Düzgün, Ruken, Aurélie Thiele. 2012. Multi-Range Robust Optimization vs Stochastic Programming in Prioritizing Project Selection. URL `http://www.optimization-online.org/`.

Ericcson, K. Anders, William G. Chase, Steve Faloon. 1980. Acquisition of a Memory Skill. *Science* **208**(4448) 1181–1182.

Ewing, Paul L., William Tarantino, Gregory S. Parnell. 2006. Use of Decision Analysis in the Army Base Realignment and Closure (BRAC) 2005 Military Value Analysis. *Decision Analysis* **3**(1) 33–49.

Fang, Chen, Rainer Kolisch, Ling Wang, Chundi Mu. 2015. An Estimation of Distribution Algorithm and New Computational Results for the Stochastic Resource-Constrained Project Scheduling Problem. *Flexible Services and Manufacturing Journal* 1–21.

Fasolo, Barbara, Alec Morton, Detlof von Winterfeldt. 2011. Behavioural Issues in Portfolio Decision Analysis. Ahti Salo, Jeffrey M. Keisler, Alec Morton, eds., *Portfolio Decision Analysis*, *International Series in Operations Research & Management Science*, vol. 162. Springer, New York, 149–165.

Fischbacher, Urs. 2007. Z-Tree: Zurich Toolbox for Ready-Made Economic Experiments. *Experimental Economics* **10**(2) 171–178.

Fliedner, Thomas, Rainer Kolisch. 2015. Performance and Robustness of Priority Policies for Static and Dynamic Project Scheduling under Uncertainty. URL `www.researchgate.com`.

Fliedner, Thomas, Juuso Liesiö. 2015. Adjustable Robustness for Multiobjective Project Portfolio Selection. URL `www.researchgate.com`.

Fliedner, Thomas, Sebastian Schiffels, Rainer Kolisch. 2014. Human Behavior in Project Portfolio Selection: Insights from an Experimental Study. URL `www.researchgate.com`.

Frontinus, Sextus Julius, Robert H. Rodgers. 2004. *De Aquaeductu Urbis Romae*, *Cambridge Classical Texts and Commentaries*, vol. 42. Cambridge University Press, Cambridge.

Gabrel, Virginie, Cécile Murat, Aurélie Thiele. 2014. Recent Advances in Robust Optimization: An Overview. *European Journal of Operational Research* **235**(3) 471–483.

Gans, Noah, Rachel Croson. 2008. Introduction to the Special Issue on Behavioral Operations. *Manufacturing & Service Operations Management* **10**(4) 563–565.

Gettinger, Johannes, Elmar Kiesling, Christian Stummer, Rudolf Vetschera. 2013. A Comparison of Representations for Discrete Multi-Criteria Decision Problems. *Decision Support Systems* **54**(2) 976–985.

Ghasemzadeh, Fereidoun, Norm P. Archer, Paul Iyogun. 1999. A Zero-One Model for Project Portfolio Selection and Scheduling. *Journal of the Operational Research Society* **50**(7) 745–755.

Gigerenzer, Gerd, Reinhard Selten. 2001. *Bounded Rationality: The Adaptive Toolbox*. MIT Press, Cambridge.

Gingrich, Gerry, Sigfrid D. Soli. 1984. Subjective Evaluation and Allocation of Resources in Routine Decision Making. *Organizational Behavior and Human Performance* **33**(2) 187–203.

Gino, Francesca, Gary Pisano. 2008. Toward a Theory of Behavioral Operations. *Manufacturing & Service Operations Management* **10**(4) 676–691.

Golabi, Kamal, Craig W. Kirkwood, Alan Sicherman. 1981. Selecting a Portfolio of Solar

Energy Projects Using Multiattribute Preference Theory. *Management Science* **27**(2) 174–189.

Gonzalez, Roxana M., Harvey J. Langholtz, Barron Sopchak. 2002. Minimizing Cost in Resource-Allocation Decisions. *Organizational Behavior and Human Decision Processes* **89**(2) 1032–1057.

Gregory, Christine, Ken Darby-Dowman, Gautam Mitra. 2011. Robust Optimization and Portfolio Selection: The Cost of Robustness. *European Journal of Operational Research* **212**(2) 417–428.

Greiner, Ben. 2004. An Online Recruitment System for Economic Experiments. Kurt. Kremer, Volker Macho, eds., *Forschung und wissenschaftliches Rechnen 2003*, *GWDG Bericht*, vol. 63. Gesellschaft für Wissenschaftliche Datenverarbeitung, Göttingen, 79–93.

Grushka-Cockayne, Yael, Bert de Reyck, Zeger Degraeve. 2008. An Integrated Decision-Making Approach for Improving European Air Traffic Management. *Management Science* **54**(8) 1395–1409.

Gurgur, Cigdem Z., Charles T. Morley. 2008. Lockheed Martin Space Systems Company Optimizes Infrastructure Project-Portfolio Selection. *Interfaces* **38**(4) 251–262.

Gutjahr, Walter J., Karl A. Froeschl. 2013. Project Portfolio Selection under Uncertainty with Outsourcing Opportunities. *Flexible Services and Manufacturing Journal* **25**(1-2) 255–281.

Hagstrom, Jane N. 1988. Computational Complexity of PERT Problems. *Networks* **18**(2) 139–147.

Hall, David L., Alexander Nauda. 1990. An Interactive Approach for Selecting IR&D Projects. *IEEE Transactions on Engineering Management* **37**(2) 126–133.

Hall, Nicholas G. 2012. Project Management: Recent Developments and Research Opportunities. *Journal of Systems Science and Systems Engineering* **21**(2) 129–143.

Hall, Nicholas G., Zhuoyu Long, Jin Qi, Melvyn Sim. 2009. A Satisficing Model for Project Selection. URL `http://www.optimization-online.org`.

Hall, Philip. 1927. The Distribution of Means for Samples of Size N Drawn from a Population in Which the Variate Takes Values between 0 and 1, All Such Values Being Equally Probable. *Biometrika* **19**(3/4) 240.

Hämäläinen, Raimo P., Jukka Luoma, Esa Saarinen. 2013. On the Importance of Behavioral Operational Research: The Case of Understanding and Communicating about Dynamic Systems. *European Journal of Operational Research* **228**(3) 623–634.

Hans, Erwin W., Willy S. Herroelen, Roel Leus, G. Wullink. 2007. A Hierarchical Approach to Multi-Project Planning under Uncertainty. *Omega* **35**(5) 563–577.

Hartmann, Sönke. 1998. A Competitive Genetic Algorithm for Resource-Constrained Project Scheduling. *Naval Research Logistics* **45**(7) 733–750.

Hartmann, Sönke. 2002. A Self-Adapting Genetic Algorithm for Project Scheduling under Resource Constraints. *Naval Research Logistics* **49**(5) 433–448.

Hartmann, Sönke, Rainer Kolisch. 2000. Experimental Evaluation of State-Of-The-Art Heuristics for the Resource-Constrained Project Scheduling Problem. *European Journal of Operational Research* **127**(2) 394–407.

Hassanzadeh, Farhad, Hamid R. Nemati, Minghe Sun. 2014. Robust Optimization for Interactive Multiobjective Programming with Imprecise Information Applied to R&D Project Portfolio Selection. *European Journal of Operational Research* **238**(1) 41–53.

Heidenberger, Kurt, Christian Stummer. 1999. Research and Development Project Selection and Resource Allocation: A Review of Quantitative Modelling Approaches. *International Journal of Management Reviews* **1**(2) 197–224.

Henriksen, Anne D., Ann J. Traynor. 1999. A Practical R&D Project-Selection Scoring Tool. *IEEE Transactions on Engineering Management* **46**(2) 158–170.

Herroelen, Willy S., Roel Leus. 2005. Project Scheduling under Uncertainty: Survey and Research Potentials. *European Journal of Operational Research* **165**(2) 289–306.

Herroelen, Willy S., Bert de Reyck. 1999. Phase Transitions in Project Scheduling. *Journal of the Operational Research Society* **50**(2) 148–156.

Ho, William, Xiaowei Xu, Prasanta K. Dey. 2010. Multi-Criteria Decision Making Approaches for Supplier Evaluation and Selection: A Literature Review. *European Journal of Operational Research* **202**(1) 16–24.

Igelmund, G., Franz Josef Radermacher. 1983. Preselective Strategies for the Optimization of Stochastic Project Networks under Resource Constraints. *Networks* **13**(1) 1–28.

Irwin, Joseph Oscar. 1927. On the Frequency Distribution of the Means of Samples from a Population Having Any Law of Frequency with Finite Moments, with Special Reference to Pearson's Type II. *Biometrika* **19**(3/4) 225.

Jonides, John, Richard L. Lewis, Derek Evan Nee, Cindy A. Lustig, Marc G. Berman, Katherine Sledge Moore. 2008. The Mind and Brain of Short-Term Memory. *Annual Review of Psychology* **59** 193–224.

Kachani, Soulaymane, Jerome Langella. 2005. A Robust Optimization Approach to Capital Rationing and Capital Budgeting. *The Engineering Economist* **50**(3) 195–229.

Kavadias, Stylianos. 2014. 10-Year Anniversary of the New Product Development, R&D, and Project Management Department in Production and Operations Management -Progress, Thoughts, and Perspectives. *Production and Operations Management* **23**(8) 1259–1264.

Keisler, Jeffrey M. 2004. Value of Information in Portfolio Decision Analysis. *Decision Analysis* **1**(3) 177–189.

Kester, Linda, Erik Jan Hultink, Kristina Lauche. 2009. Portfolio Decision-Making Genres: A Case Study. *Journal of Engineering and Technology Management* **26**(4) 327–341.

Killen, Catherine P. 2013. Evaluation of Project Interdependency Visualizations through Decision Scenario Experimentation. *International Journal of Project Management* **31**(6) 804–816.

Killen, Catherine P., Cai Kjaer. 2012. Understanding Project Interdependencies: The Role of Visual Representation, Culture and Process. *International Journal of Project Management* **30**(5) 554–566.

Kleinmuntz, Don N. 2007. Resource Allocation Decisions. Ward Edwards, Ralph F. Miles, Detlof von Winterfeldt, eds., *Advances in Decision Analysis*. Cambridge University Press, Cambridge.

Kleywegt, Anton J., Alexander Shapiro, Tito Homem-de Mello. 2002. The Sample Average Approximation Method for Stochastic Discrete Optimization. *SIAM Journal on Optimization* **12**(2) 479–502.

Knuth, Donald Ervin. 1968. *The Art of Computer Programming*. Addison-Wesley Series in Computer Science and Information Processing, Addison-Wesley, Reading.

Koç, Ali, David P. Morton, Elmira Popova, Stephen M. Hess, Ernie Kee, Drew Richards. 2009. Prioritizing Project Selection. *The Engineering Economist* **54**(4) 267–297.

Kolisch, Rainer. 1996. Serial and Parallel Resource-Constrained Project Scheduling Methods Revisited: Theory and Computation. *European Journal of Operational Research* **90**(2) 320–333.

Kolisch, Rainer, Jens Brunner, Thomas Fliedner. 2012. Project Portfolio Planning with Project Interdependencies and Multiple Criteria: A Practical Application. Erik L. Demeulemeester, Willy S. Herroelen, eds., *Proceedings of the 13th International Conference on Project Management and Scheduling*. Leuven, 183–186.

Kolisch, Rainer, Christian Heimerl, Sven Hausen. 2008. Projektportfolio- und Multiprojektplanung: Modellierung, prototypische Implementierung und Einsatz in der Finanzdienstleistungsbranche. *Zeitschrift für Betriebswirtschaft* **78**(6) 591–610.

Kolisch, Rainer, Arno Sprecher, Andreas Drexl. 1995. Characterization and Generation of a General Class of Resource-Constrained Project Scheduling Problems. *Management Science* **41**(10) 1693–1703.

Könnölä, Totti, Ville Brummer, Ahti Salo. 2007. Diversity in Foresight: Insights from the Fostering of Innovation Ideas. *Technological Forecasting and Social Change* **74**(5) 608–626.

Kouvelis, Panos, Gang Yu. 1997. *Robust Discrete Optimization and Its Applications*, *Nonconvex Optimization and Its Applications*, vol. 14. Springer, Boston.

Langholtz, Harvey J., Christopher T. Ball, Barron Sopchak, Jacqueline Auble. 1997. Resource-Allocation Behavior in Complex but Commonplace Tasks. *Organizational Behavior and Human Decision Processes* **70**(3) 249–266.

Langholtz, Harvey J., Charles Gettys, Bobbie Foote. 1993. Resource-Allocation Behavior under Certainty, Risk, and Uncertainty. *Organizational Behavior and Human Decision Processes* **54**(2) 203–224.

Langholtz, Harvey J., Charles Gettys, Bobbie Foote. 1994. Allocating Resources over Time in Benign and Harsh Environments. *Organizational Behavior and Human Decision Processes* **58**(1) 28–50.

Langholtz, Harvey J., Charles Gettys, Bobbie Foote. 1995. Are Resource Fluctuations Anticipated in Resource Allocation Tasks? *Organizational Behavior and Human Decision Processes* **64**(3) 274–282.

Leus, Roel, Willy S. Herroelen. 2004. Stability and Resource Allocation in Project Planning. *IIE Transactions* **36**(7) 667–682.

Li, Zongzhi. 2009. Stochastic Optimization Model and Solution Algorithm for Highway Investment Decision Making under Budget Uncertainty. *Journal of Transportation Engineering* **135**(6) 371–379.

Liesiö, Juuso, Pekka Mild, Ahti Salo. 2007. Preference Programming for Robust Portfolio Modeling and Project Selection. *European Journal of Operational Research* **181**(3) 1488–1505.

Liesiö, Juuso, Pekka Mild, Ahti Salo. 2008. Robust Portfolio Modeling with Incomplete Cost Information and Project Interdependencies. *European Journal of Operational Research* **190**(3) 679–695.

Liesiö, Juuso, Antti Punkka. 2014. Baseline Value Specification and Sensitivity Analysis in Multiattribute Project Portfolio Selection. *European Journal of Operational Research* **237**(3) 946–956.

Liesiö, Juuso, Ahti Salo. 2012. Scenario-Based Portfolio Selection of Investment Projects with Incomplete Probability and Utility Information. *European Journal of Operational Research* **217**(1) 162–172.

Loch, Christoph H. 2000. Tailoring Product Development to Strategy: Case of a European Technology Manufacturer. *European Management Journal* **18**(3) 246–258.

Loch, Christoph H., Stylianos Kavadias. 2002. Dynamic Portfolio Selection of NPD Programs Using Marginal Returns. *Management Science* **48**(10) 1227–1241.

Loch, Christoph H., Michael T. Pich, Christian Terwiesch, Michael Urbschat. 2001. Selecting R&D Projects at BMW: A Case Study of Adopting Mathematical Programming Models. *IEEE Transactions on Engineering Management* **48**(1) 70–80.

Loch, Christoph H., Yaozhong Wu. 2007. *Behavioral Operations Management*, *Foundations and Trends in Technology, Information and Operations Management*, vol. 1, issue 3. Now Publishers, Boston.

Lourenço, João Carlos, Alec Morton, Carlos A. Bana e Costa. 2012. PROBE—a Multicriteria Decision Support System for Portfolio Robustness Evaluation. *Decision Support Systems* **54**(1) 534–550.

Lova, Antonio, Concepción Maroto, Pilar Tormos. 2000. A Multicriteria Heuristic

Method to Improve Resource Allocation in Multiproject Scheduling. *European Journal of Operational Research* **127**(2) 408–424.

Martello, Silvano, Paolo Toth. 1990. *Knapsack Problems: Algorithms and Computer Implementations*. Wiley-Interscience Series in Discrete Mathematics and Optimization, J. Wiley & Sons, New York.

Martinsuo, Miia. 2013. Project Portfolio Management in Practice and in Context. *International Journal of Project Management* **31**(6) 794–803.

Martinsuo, Miia, Tuomas Korhonen, Teemu Laine. 2014. Identifying, Framing and Managing Uncertainties in Project Portfolios. *International Journal of Project Management* **32**(5) 732–746.

Marx, Frederik, Jörg H. Mayer, Robert Winter. 2011. Six Principles for Redesigning Executive Information Systems: Findings of a Survey and Evaluation of a Prototype. *ACM Transactions on Management Information Systems* **2**(4) 1–19.

Melchiors, Philipp. 2013. Dynamic Stochastic Multi-Project Scheduling without Preemptions: Models, Methodology and Computational Results. Ph.D. thesis, Technische Universität München.

Melchiors, Philipp, Rainer Kolisch. 2009. Scheduling of Multiple R&D Projects in a Dynamic and Stochastic Environment. Bernhard Fleischmann, Karl-Heinz Borgwardt, Robert Klein, Axel Tuma, eds., *Operations Research Proceedings 2008*. Springer, Berlin, 135–140.

Melchiors, Philipp, Rainer Kolisch. 2010. Markov-Decision-Process-Based Scheduling of Multiple Projects in a Stochastic Dynamic Environment. Vincent Tkindt, ed., *Proceedings of the 12th International Conference on Project Management and Scheduling*. Tours.

Miller, George A. 1994. The Magical Number Seven, plus or minus Two: Some Limits on Our Capacity for Processing Information. *Psychological Review* **101**(2) 343–352.

Möhring, Rolf H. 2000. Scheduling under Uncertainty: Optimizing against a Randomizing Adversary. Klaus Jansen, Samir Khuller, eds., *Approximation Algorithms for Combinatorial Optimization*, *Lecture Notes in Computer Science*, vol. 1913. Springer, Berlin, 15–26.

Möhring, Rolf H. 2001. Scheduling under Uncertainty: Bounding the Makespan

Distribution. Helmut Alt, ed., *Computational Discrete Mathematics*, *Lecture Notes in Computer Science*, vol. 2122. Springer, Berlin, 79–97.

Möhring, Rolf H., Franz Josef Radermacher, Gideon Weiss. 1984. Stochastic Scheduling Problems I: General Strategies. *Zeitschrift für Operations Research* **28**(7) 193–260.

Möhring, Rolf H., Franz Josef Radermacher, Gideon Weiss. 1985. Stochastic Scheduling Problems II: Set Strategies. *Zeitschrift für Operations Research* **29**(3) 65–104.

Möhring, Rolf H., Frederik Stork. 2000. Linear Preselective Policies for Stochastic Project Scheduling. *Mathematical Methods of Operations Research (ZOR)* **52**(3) 501–515.

Ouelhadj, Djamila, Sanja Petrovic. 2009. A Survey of Dynamic Scheduling in Manufacturing Systems. *Journal of Scheduling* **12**(4) 417–431.

Payne, John H. 1995. Management of Multiple Simultaneous Projects: A State-Of-The-Art Review. *International Journal of Project Management* **13**(3) 163–168.

Phillips, Lawrence D., Carlos A. Bana e Costa. 2007. Transparent Prioritisation, Budgeting and Resource Allocation with Multi-Criteria Decision Analysis and Decision Conferencing. *Annals of Operations Research* **154**(1) 51–68.

Pisinger, David. 2005. Where Are the Hard Knapsack Problems? *Computers & Operations Research* **32**(9) 2271–2284.

Poss, Michael. 2013. Robust Combinatorial Optimization with Variable Budgeted Uncertainty. *4OR* **11**(1) 75–92.

Poss, Michael. 2014. Robust Combinatorial Optimization with Variable Cost Uncertainty. *European Journal of Operational Research* **237**(3) 836–845.

Reyck, Bert de, Yael Grushka-Cockayne, Martin Lockett, Sergio Ricardo Calderini, Marcio Moura, Andrew Sloper. 2005. The Impact of Project Portfolio Management on Information Technology Projects. *International Journal of Project Management* **23**(7) 524–537.

Reyck, Bert de, Willy S. Herroelen. 1996. On the Use of the Complexity Index as a Measure of Complexity in Activity Networks. *European Journal of Operational Research* **91**(2) 347–366.

Roy, Bernard. 2010. Robustness in Operational Research and Decision Aiding: A Multi-Faceted Issue. *European Journal of Operational Research* **200**(3) 629–638.

Saliby, Eduardo. 1990. Descriptive Sampling: A Better Approach to Monte Carlo Simulation. *Journal of the Operational Research Society* **41**(12) 1133–1142.

Salo, Ahti, Raimo P. Hämäläinen. 1992. Preference Assessment by Imprecise Ratio Statements. *Operations Research* **40**(6) 1053–1061.

Salo, Ahti, Raimo P. Hämäläinen. 1995. Preference Programming through Approximate Ratio Comparisons. *European Journal of Operational Research* **82**(3) 458–475.

Salo, Ahti, Jeffrey M. Keisler, Alec Morton. 2011. An Invitation to Portfolio Decision Analysis. Ahti Salo, Jeffrey M. Keisler, Alec Morton, eds., *Portfolio Decision Analysis*, *International Series in Operations Research & Management Science*, vol. 162. Springer, New York, 3–27.

Salo, Ahti, Antti Punkka. 2005. Rank Inclusion in Criteria Hierarchies. *European Journal of Operational Research* **163**(2) 338–356.

Santhanam, Radhika, George J. Kyparisis. 1996. A Decision Model for Interdependent Information System Project Selection. *European Journal of Operational Research* **89**(2) 380–399.

Sarabando, Paula, Luís C. Dias. 2010. Simple Procedures of Choice in Multicriteria Problems without Precise Information about the Alternatives' Values. *Computers & Operations Research* **37**(12) 2239–2247.

Schweitzer, Maurice E., Gérard P. Cachon. 2000. Decision Bias in the Newsvendor Problem with a Known Demand Distribution: Experimental Evidence. *Management Science* **46**(3) 404–420.

Schwindt, Christoph. 1998. Generation of Resource Constrained Project Scheduling Problems Subject to Temporal Constraints. URL `http://www.wiwi.tu-clausthal.de/abteilungen/produktion/forschung/schwerpunkte/project-generator`.

Seale, Darryl A., Amnon Rapoport. 1997. Sequential Decision Making with Relative Ranks: An Experimental Investigation of the Secretary Problem. *Organizational Behavior and Human Decision Processes* **69**(3) 221–236.

Shakhsi-Niaei, Majid, S. Ali Torabi, Seyed H. Iranmanesh. 2011. A Comprehensive Framework for Project Selection Problem under Uncertainty and Real-World Constraints. *Computers & Industrial Engineering* **61**(1) 226–237.

Smith-Miles, Kate, Leo Lopes. 2012. Measuring Instance Difficulty for Combinatorial Optimization Problems. *Computers & Operations Research* **39**(5) 875–889.

Sprecher, Arno, Rainer Kolisch, Andreas Drexl. 1995. Semi-Active, Active, and Non-Delay Schedules for the Resource-Constrained Project Scheduling Problem. *European Journal of Operational Research* **80**(1) 94–102.

Steuer, Ralph E. 1986. *Multiple Criteria Optimization: Theory, Computation, and Application*. Wiley Series in Probability and Mathematical Statistics. Applied Probability and Statistics, J. Wiley & Sons, New York.

Stork, Frederik. 2001. Stochastic Resource Constrained Project Scheduling. Ph.D. thesis, Technische Universität Berlin.

Stummer, Christian, Kurt Heidenberger. 2003. Interactive R&D Portfolio Analysis with Project Interdependencies and Time Profiles of Multiple Objectives. *IEEE Transactions on Engineering Management* **50**(2) 175–183.

Tavana, Madjid, Mariya A. Sodenkamp. 2009. A Fuzzy Multi-Criteria Decision Analysis Model for Advanced Technology Assessment at Kennedy Space Center. *Journal of the Operational Research Society* **61**(10) 1459–1470.

Vredeveld, Tjark. 2012. Stochastic Online Scheduling. *Computer Science - Research and Development* **27**(3) 181–187.

Walker, Derek, Christopher J. Dart. 2011. Frontinus - A Project Manager from the Roman Empire Era. *Project Management Journal* **42**(5) 4–16.

Ward, Stephen C., Chris B. Chapman. 2003. Transforming Project Risk Management into Project Uncertainty Management. *International Journal of Project Management* **21**(2) 97–105.

Weber, Martin. 1987. Decision Making with Incomplete Information. *European Journal of Operational Research* **28**(1) 44–57.

Weiss, Gideon, Michael L. Pinedo. 1980. Scheduling Tasks with Exponential Service Times on Non-Identical Processors to Minimize Various Cost Functions. *Journal of Applied Probability* **17**(1) 187.

Wilcoxon, Frank. 1945. Individual Comparisons by Ranking Methods. *Biometrics Bulletin* **1**(6) 80–83.