Quality- and profit-oriented scheduling of flexible resource-constrained projects

Carolin Kellenbrink and Stefan Helber
Department of Production Management
Leibniz Universität Hannover
Königsworther Platz 1, D-30167 Hannover, Germany
carolin.kellenbrink@prod.uni-hannover.de, +49 511 7628002
stefan.helber@prod.uni-hannover.de, +49 511 7625650

December 8, 2014

Abstract

We study the problem of determining both the structure and the schedule of projects subject to capacity constraints. We assume that those projects are flexible in the sense that the activities to be implemented are not entirely known in advance. In such a setting, decisions must be made with respect to the implementation of the optional activities. Such decisions affect the duration, cost, quality and eventual revenue of the project. Examples of this type of problem can often be found when complex capital goods such as aircraft engines are overhauled, when buildings are renovated to meet higher environmental and efficiency standards, or in product-development processes. We describe the problem, develop a mixed-integer optimisation model, explain specific features of a genetic algorithm to solve the problem and report the results of a numerical study.

Keywords: Quality, project scheduling, genetic algorithm, RCPSP, flexible projects

1 Introduction

A standard assumption in the literature on resource-constrained project scheduling problems (RCPSPs) is that the activities to be implemented as well as the precedence relationships between those activities—and hence the project structure—are given. However, there are cases when, in the process of planning and scheduling a project, decisions have to be made with respect to *optional activities*. While this includes the case in which one must choose between alternative modes for given activities in multi-mode RCPSPs, i.e., MRCPSPs, the situation studied in this paper goes far beyond the well-established MRCPSPs. In particular, a RCPSP with a flexible project structure, or RCPSP-PS for short, can model the case in which there is a model-endogenous decision to determine the outcome of the project.

Consider, for example, the process of overhauling jet engines of commercial aircraft. This process is performed by commercial service providers that have to meet rigorous standards and regulations to ensure the airworthiness of the overhauled engines. However, within those rules, there can be a substantial and economically important degree of freedom regarding how to overhaul the engine. This situation gives rise to the question of whether a particular component should be disassembled and overhauled at all, and if so, what should be done about the component's used sub-assemblies. The sub-assemblies could, for example, be used again as they are, be repaired using different available repair procedures, or be replaced, either with used or already repaired parts of the same type or with a new part. Such decisions affect the work content of the project and hence eventually the schedule. However, they also affect qualitative aspects of the project outcome. In the case of an aircraft engine, these aspects could be the energy efficiency and/or the noise emissions of the overhauled engine. Different owners of aircraft engines, particularly airlines with different business models, may have different requirements and objectives and favour different ways to overhaul their engines.

Another example is the renovation or reconstruction of buildings to meet new and more demanding environmental and energy efficiency standards. In European Union countries, those standards are increasingly being imposed on homeowners. There may be alternative ways to meet those efficiency goals, such as installing new windows or heating systems or adding layers or insulation to the building's exterior structures. Substantial public subsidies can be earned by the homeowner in return, depending on the level of efficiency achieved. A particular level of efficiency may be attained by different combinations of measures. Implementing those measures leads to an RCPSP-PS involving quality, duration, cost and revenue aspects.

Hence, it appears to be worthwhile to model RCPSP-PSs with respect to duration, quality and profit. Solving this problem to optimality turns out to be quite demanding. For this reason, we present a specific adaption of a genetic algorithm and evaluate its performance in a numerical study.

The remainder of this paper is structured as follows: In Section 2, the problem and the modelling assumptions are outlined in detail. Furthermore, the related literature is briefly discussed. In Section 3, we describe the model, and in Section 4 we describe the algorithm. Numerical results are presented in Section 5. Section 6 provides a summary of the main results and concluding remarks as well as directions for further research.

2 Problem and literature

2.1 Flexible projects with quality attributes

Figure 1 shows an example of a flexible project; see Kellenbrink and Helber (2013) and Kellenbrink (2014). Activities $j \in \mathscr{J}$ are depicted as squares. They are assumed to be topologically numbered such that $i < j$ implies that there is no direct or indirect precedence relationship from activity j to activity i. Choices $e \in \mathscr{E}$ are depicted as light-grey ovals. They contain a subset of activities $\mathscr{W}_e \subset \mathscr{J}$ from which exactly one activity has to be chosen if that decision is activated. In Figure 1, choice $e = 1$ is activated by the mandatory (start) activity $j = 1$, i.e., $a(e = 1) = j = 1$, and hence, a choice between activities $j = 4$ and $j = 5$ must be made. If in choice $e = 1$ activity $j = 5$ is selected, a second choice $e = 2$ is trigged, the choice between activities $j = 7$ and $j = 8$. Optional activities j can also cause other activities $i \in \mathscr{B}_j \subset \mathscr{J}$. In our example, if in choice $e = 1$ activity $j = 4$ is chosen, the further activity $j = 9$ is caused; see the dark-grey circle in Figure 1. Precedence relationships that lead to or originate from optional activities are depicted by broken arrows in Figure 1. These relationships are only imposed if the optional activities are implemented.

We assume that the activities selected from among the aforementioned sets of optional activities related to the choices affect the quality of the project outcome along potentially multiple quantifiable and independent dimensions. Figure 2 conceptually shows the overall impact on the profit of the project. Implementing specific activities directly affects costs. Because implemented optional activities have a duration, they also affect the overall project duration or makespan. Finally, optional activities determine the quality of the project outcome, and makespan as well as quality have a combined effect on revenues. All of these effects culminate in the profit of the project.

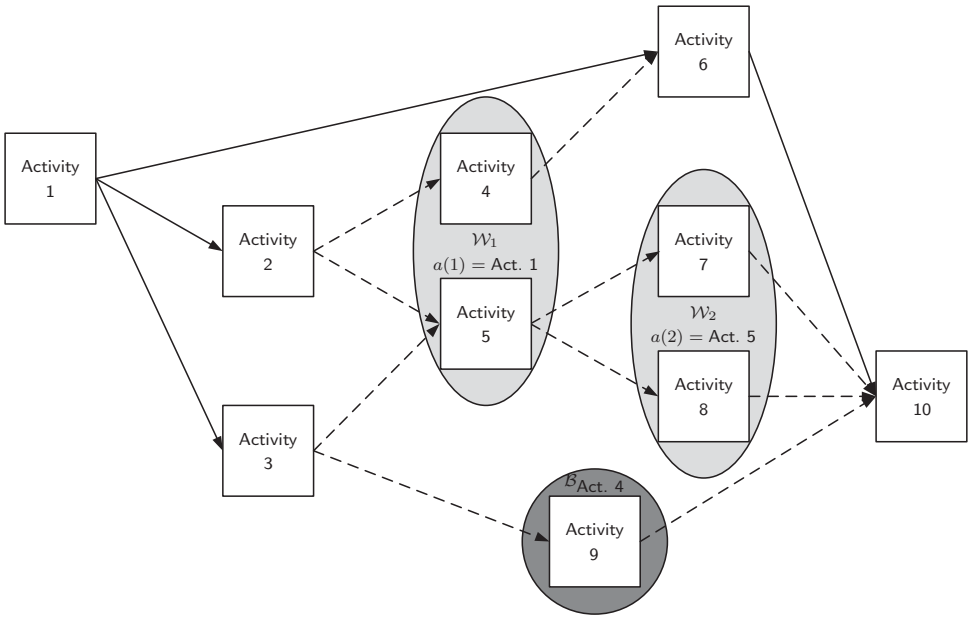

Figure 1: Example of a flexible project

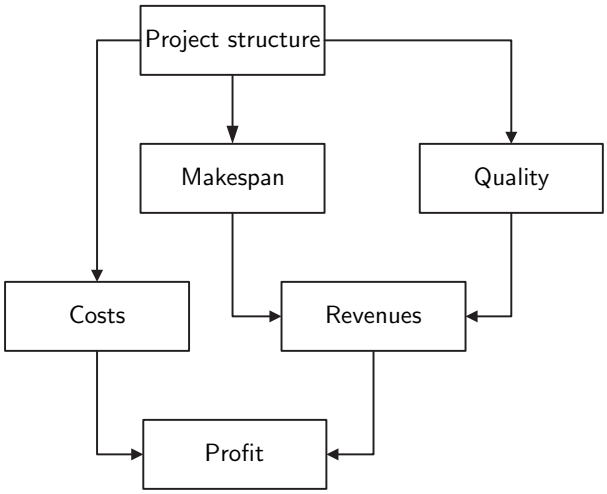

Figure 2: Effect of project structure on profit

In Table 1, the duration d_j, the capacity consumption $k_{j,1}$ of a single renewable resource and the cost c_j of each activity j are given for the sample project in Figure 1.

In our example, we assume that the quality of the project outcome is measured along two different and independent quality dimensions o. Table 2 shows, for example, that the mandatory activities along quality dimension $o = 1$ lead to a base quality of $q_o^{\text{base}} = 20$ and that implementing activity 4, for example, leads to an increase of $q_{4,1} = 10$ units in quality level along quality dimension $o = 1$.

Multi-dimensional quality attributes of goods and services are often condensed in and communicated as quality grades or classes, such as air travel classes (Economy, Business, First). This reduces marketing and operational complexities. Coming back to our initial example of public subsidy programs for the energy-efficient renovation of buildings, we find that those subsidies are often tied to achieved energy efficiency classes. Following these examples, we therefore assume that the multiple quality dimensions are condensed in quality grades. In our example, we assume that there are three quality grades l, denoted "A", "AA", and "AAA". The required minimum quality levels w_{ol} per quality dimension o and grade l are presented in Table 3.

Table 1: Duration, capacity consumption and cost of each activity j

j	1	2	3	4	5	6	7	8	9	10
d_j	0	3	4	3	5	6	4	2	2	0
$k_{j,1}$	0	3	7	5	2	8	6	5	4	0
c_j	0	5	3	2	1	7	10	6	1	0

We assume that the project costs are simply the sum of the costs c_j of the implemented activities (such as wages and material costs); see the last row in Table 1.

In principle, the decision about project structure and schedule in such a setting is a multi-criteria optimisation problem with quality, cost and project duration objectives. One could hence aim at determining the Pareto-efficient front of solutions along those three dimensions. However, finding a single point of that three-dimensional Pareto front already requires to solve a challenging scheduling problem. We hence assume that the decision maker aims at maximising the profit as suggested in Figure 2. To this end, we assume that the project revenue u_{lt} depends on the quality grade ("A" to "AAA" in our example) and the makespan or completion period of the project as follows:

- The revenue u_{lt} increases monotonically with the quality grade l.
- The revenue u_{lt} decreases monotonically with the project completion period t.

The assumptions imply that a customer's willingness to pay and therefore the project revenue does not increase as the quality grade of the project outcome decreases or the makespan of the project increases.

Table 2: Quality dimensions, base levels and quality increases due to optional activities

o	q_o^{base}	$q_{4,o}$	$q_{5,o}$	$q_{7,o}$	$q_{8,o}$	$q_{9,o}$
1	20	10	12	15	5	8
2	0	0	10	15	0	20

Table 3: Minimum quality levels w_{ol} per quality grade and quality dimension

o	$w_{o,\mathrm{A}}$	$w_{o,\mathrm{AA}}$	$w_{o,\mathrm{AAA}}$
1	30	35	40
2	10	15	20

Table 4: Properties of the possible project structures

Project structure	Implemented activities	Quality grade	Duration	Cost
A	1, 2, 3, **4**, 6, **9**, 10	AA	13	18
B	1, 2, 3, **5**, 6, **7**, 10	AAA	14	26
C	1, 2, 3, **5**, 6, **8**, 10	A	12	22

In the algorithm for solving the problem that will be presented in Section 4, this property is used.

For the project in Figure 1, three possible project structures A, B, and C exist. It can easily be shown using the data in Tables 1,2, and 3 and a given (per-period) capacity of the single renewable resource of 10 units that those three project structures lead to the respective quality grades, project durations and costs in Table 4. Note that the lowest quality grade can be reached through project structure C within 12 periods, whereas the highest quality grade is achieved in project structure B within 14 periods.

Which project structure is most profitable depends on the customer's willingness to pay, which in turn may reflect the customer's particular business model. In Table 5, we introduce three hypothetical customers and their willingness to pay—the potential revenues—as a function of quality grade and project duration. For those combinations of project duration and quality grade that cannot be achieved due to the capacity limit of 10 units for the renewable resource, the revenues are presented in brackets. In Table 6, we present the profit per customer and project structure for those combinations of project duration and quality grade that can actually be achieved. The optimal solutions are printed in bold type. Note how the solutions reflect the different customers' preferences. For customer 1, quality matters most, and he should hence be offered project structure B. Customer 2, in contrast, is mostly interested in the duration of the project and should be offered project structure C. The third customer is interested in both quality and project duration and should be offered project structure A. This small example shows how customer segmentation and flexible project scheduling can be used together to maximise profit.

Table 5: Revenue per customer, quality grade, project duration and structure

	Quality grade	Duration 12	13	14	Project structure
Customer 1	AAA	(50)	(49)	48	B
	AA	(40)	39	38	A
	A	30	29	28	C
Customer 2	AAA	(50)	(40)	30	B
	AA	(49)	39	29	A
	A	48	38	28	C
Customer 3	AAA	(40)	(39)	38	B
	AA	(39)	38	37	A
	A	38	37	36	C

Table 6: Profit per customer, quality grade, project duration and structure

	Quality grade	12	Duration 13	14	Project structure
Customer 1	AAA	-	-	**48-26=22**	B
	AA	-	39-18=21	38-18=20	A
	A	30-22=8	29-22=7	28-22=6	C
Customer 2	AAA	-	-	30-26=4	B
	AA	-	39-18=21	29-18=11	A
	A	**48-22=26**	38-22=16	28-22=6	C
Customer 3	AAA	-	-	38-26=12	B
	AA	-	**38-18=20**	37-18=19	A
	A	38-22=16	37-22=15	36-22=14	C

2.2 Related literature

Numerous surveys review the voluminous literature on resource-constrained project scheduling, among them Özdamar and Ulusoy (1995), Herroelen et al. (1998), Brucker et al. (1999), Kolisch and Padman (2001), Demeulemeester and Herroelen (2002) as well as Hartmann and Briskorn (2010). Much of this literature assumes that both the activities and the precedence relationships and hence the project structure are given, whereas we consider below only the case in which there is a substantial degree of project structure flexibility.

A certain degree of flexibility is considered in the multi-mode extension of the RCPSP, the MRCPSP (cf., e.g., Talbot (1982), or for a recent survey Węglarz et al. (2011)). In the MRCPSP, different execution modes can be available for an activity, even though each activity must still be implemented in one of the available modes. Another approach—to depart from the standard assumption of a strictly given project

structure with only mandatory activities and precedence relationships—was introduced by Tiwari et al. (2009), who extended the MRCPSP using rework activities. Here, the basic assumption is that rework is required if the original activity is implemented in a specific predefined mode. However, the underlying modelling assumption is that rework always consists of a single activity that is a direct successor of the original activity causing the rework. This approach is much less general and flexible than the concept of a flexible project as outlined in Section 2.1.

Logical dependencies between some activities are considered in Belhe and Kusiak (1995) who present the so-called "design activity network" approach, whereas alternative process plans are treated explicitly in Čapek et al. (2011). In these approaches, one assumes that a logical dependency among activities always accompanies a precedence constraint among these activities, which is not necessary in our case.

Rescheduling problems in the context of disruption-management problems at airports with alternative process-implementation paths are studied in Kuster et al. (2009) and Kuster et al. (2010). However, this approach concentrates on reactive rescheduling as opposed to structuring a flexible project in the first place. Furthermore, the authors do not present a formal optimisation model of their problem.

In the literature on *stochastic* project networks, cf. Elmaghraby (1964) or Neumann (1990), activities that are not necessarily implemented have also been modelled. There, however, the basic assumption is that the implementation of activities is the result of *random exogenous influences* as the project progresses in contrast to the *endogenous decisions* considered in our setting.

The literature on quality aspects in project scheduling is even more limited, which is not surprising because such considerations require some concept of project flexibility. A rather natural starting point for considering quality aspects is again the MRCPSP. In this type of model, it is possible to tie quality levels to different modes. For example, Icmeli-Tukel and Rom (1997) as well as Erenguc and Icmeli-Tukel (1999) consider quality as a function of rework, which is necessary if specific modes are chosen.

Vanhoucke (2006) considers quality-dependent time slots to model the case in which a deviation from an optimal time window results in a lower quality and thereby higher costs. If quality aspects are considered, constraints can be used as in Tiwari et al. (2009) and in Li and Womer (2008), where the assumption is that a certain given minimum quality must be met. In a different paper, Tareghian and Taheri (2006) consider quality maximisation subject to constraints on costs and project duration.

We are not aware of an approach that simultaneously optimises project structure, schedule and outcome quality to maximise the profit of a project. The work presented

in this paper hence aims at closing that gap. It builds on the concept of a flexible project as introduced in Kellenbrink (2014, Chapter 3) and Kellenbrink and Helber (2013) as well as the quality-related considerations discussed in Kellenbrink (2014, Chapter 6).

3 Quality- and profit-oriented optimisation model

We now formally define the quality- and profit-oriented scheduling problem for resource-constrained flexible projects (RCPSP-PS-Q). The problem is a modification and extension of the well-established RCPSP, cf. Klein (2000), pp. 79-80, and the RCPSP-PS, cf. Kellenbrink and Helber (2013) with respect to the following features:

- We assume that the quality of the project outcome can be quantified along multiple quality dimensions. The outcome is affected by the selected choice of optional activities, i.e., the project structure, which in turn determines the project costs.
- Both the quality outcome and the project duration affect revenues, which leads to profit-maximising decisions regarding both project structure and schedule.

Using the notation in Table 7, the problem can be formulated as follows:

Model RCPSP-PS-Q

$$\max Z \quad = \quad \sum_{l \in \mathscr{L}} \sum_{t=EFT_J}^{LFT_J} u_{lt} \cdot y_{lt} - \sum_{j=1}^{J} c_j \cdot \sum_{t=EFT_j}^{LFT_j} x_{jt} \tag{1}$$

subject to

$$\sum_{t=EFT_j}^{LFT_j} x_{jt} = 1 \qquad j \in \mathscr{V} \tag{2}$$

$$\sum_{j \in \mathscr{W}_e} \sum_{t=EFT_j}^{LFT_j} x_{jt} = \sum_{t=EFT_{a(e)}}^{LFT_{a(e)}} x_{a(e),t} \qquad e \in \mathscr{E} \tag{3}$$

$$\sum_{t=EFT_i}^{LFT_i} x_{it} = \sum_{t=EFT_j}^{LFT_j} x_{jt} \qquad e \in \mathscr{E}; \quad j \in \mathscr{W}_e; \quad i \in \mathscr{B}_j \tag{4}$$

$$\sum_{t=EFT_i}^{LFT_i} t \cdot x_{it} \leq \sum_{t=EFT_j}^{LFT_j} (t - d_j) \cdot x_{jt} + T \cdot (1 - \sum_{t=EFT_j}^{LFT_j} x_{jt}) \qquad j \in \mathscr{J}; \quad i \in \mathscr{P}_j \tag{5}$$

$$\sum_{j=1}^{J} \left(k_{jr} \cdot \sum_{\tau=t}^{t+d_j-1} x_{j\tau} \right) \leq K_r \qquad r \in \mathscr{R}; \quad t \in \mathscr{T} \tag{6}$$

Indices and (ordered) sets

$a(e)$	unique activity causing choice e
$i \in \mathscr{B}_j \subset \mathscr{J}$	activities caused by activity j, with $i, j \notin \mathscr{V}$
$e \in \mathscr{E}$	topologically ordered choices, with $e = 1, ..., E$
$j, i \in \mathscr{J}$	topologically ordered activities, with $j, i = 1, ..., J$
$l \in \mathscr{L}$	quality grades, with $l = 1, ..., L$
$r \in \mathscr{N}$	non-renewable resources
$o \in \mathscr{O}$	quality dimensions/attributes, with $o = 1, ..., O$
$i \in \mathscr{P}_j$	predecessors of activity j
$r \in \mathscr{R}$	renewable resources
$t, \tau \in \mathscr{T}$	periods, with $t, \tau = 1, ..., T$
$j \in \mathscr{V} \subseteq \mathscr{J}$	mandatory activities, including (dummy) activities 1 and J
$j \in \mathscr{W}_e \subset \mathscr{J}$	optional activities of choice e

Parameters

c_j	cost of activity j
EFT_j	earliest finishing time of activity j
d_j	duration of activity j (integer number of periods)
k_{jr}	capacity requirement of resource r by activity j
K_r	available capacity of resource r
LFT_j	latest finishing time of activity j
M	sufficiently large number
q_o^{base}	base quality along quality dimension o
q_{jo}	quality increase along quality dimension o if activity j is implemented
u_{lt}	revenue if the project is finished in period t and quality grade l
w_{ol}	minimum quality level along quality dimension o to reach quality grade l

Decision variables

x_{jt}	$\begin{cases} 1, & \text{if activity } j \text{ is finished in period } t \\ 0, & \text{otherwise} \end{cases}$
y_{lt}	$\begin{cases} 1, & \text{the project is finished in period } t \text{ and quality grade } l \\ 0, & \text{otherwise} \end{cases}$

Table 7: Notation of the RCPSP-PS-Q

$$\sum_{j=1}^{J}\left(k_{jr}\cdot\sum_{t=EFT_j}^{LFT_j}x_{jt}\right)\leq K_r \qquad r\in\mathscr{N} \qquad (7)$$

$$q_o^{\text{Basis}}+\sum_{j=1}^{J}q_{jo}\cdot\sum_{t=EFT_j}^{LFT_j}x_{jt}\geq w_{ol}-M\cdot(1-\sum_{t=1}^{T}y_{lt}) \qquad o\in\mathscr{O};\quad l\in\mathscr{L} \qquad (8)$$

$$\sum_{l\in\mathscr{L}}y_{lt}=x_{Jt} \qquad t\in\mathscr{T} \qquad (9)$$

In the objective function (1), we aim at maximising the profit related to the project. Revenues depend on the quality grade that is reached and the period in which the project is completed, whereas costs are related to the activities that are actually implemented. According to equations (2), all mandatory activities $j\in\mathscr{V}$ have to be performed at least once. In equations (3), if and only if the activity $a(e)$, which triggers choice e, is implemented, one of the optional activities $j\in\mathscr{W}_e$ of choice e is also implemented. Equations (4) state that, for each choice and each implemented activity j of that choice, each other activity $i\in\mathscr{B}_j$ that is caused by activity j is also implemented. The constraints (5) enforce the precedence relationships between those activities that are actually implemented. The capacity limits for renewable and non-renewable resources are enforced in constraints (6) and (7), respectively. The constraints (8) guarantee that the minimum quality level w_{ol}, which corresponds to the resulting quality grade l, is reached along all quality dimensions o, whereas constraints (9) determine the period in which the project is completed. The earliest and latest finishing times EFT_j and LFT_j, respectively, for each activity j are computed in a preprocessing step considering only the mandatory activities and the given time horizon. This computation reduces the number of variables to consider and the computational effort required to solve the model using a standard MIP solver such as CPLEX.

Note that the RCPSP is a special case of the RCPSP-PS-Q in which the set $\mathscr{E}$ is empty (i.e., all activities are mandatory), all revenue parameters are set to $u_{lt}=-t$ and all cost parameters are set to $c_j=0$. Because the RCPSP is an NP-hard optimisation problem (cf., e.g., Blazewicz et al. (1983) or Klein (2000, p. 90)), so is the RCPSP-PS-Q.

4 Genetic algorithm

Given that the RCPSP-PS-Q presented in Section 3 is an NP-hard optimisation problem, it is not surprising that the numerical effort required to fully solve it to optimality using, e.g., an exact standard solver such as CPLEX, increases dramatically with the size of the problem instances. Many heuristic approaches have therefore been proposed in the

context of project scheduling. A promising starting point for our problem is the genetic algorithm for the MRCPSP presented in Hartmann (2001). Genetic algorithms mimic evolutionary selection processes within populations of a species; see, e.g., Goldberg (1989). In the genetic algorithm for the MRCPSP developed by Hartmann (2001), a solution to the problem is essentially represented by an activity list. This activity list serves as a priority ordering of activities to be fed into a serial schedule-generation scheme, cf. Kelley (1963), pp. 352-353, or Kolisch and Hartmann (1999), p. 152.

In Kellenbrink and Helber (2013) and Kellenbrink (2014), a genetic algorithm is described and evaluated to solve an RCPSP-PS with a makespan minimisation objective. The algorithm in this study generalises and extends the ideas presented in Hartmann (2001) to the case of flexible projects. The algorithm for the RCPSP-PS shows attractive performance with respect to both solution quality and computation time. We modified that algorithm presented in Kellenbrink and Helber (2013) to solve the profit- and quality-oriented RCPSP-PS-Q modelled in Section 3 of this paper. In the following, we describe both the central elements of that algorithm and the modifications that are due to the specific features of the profit- and quality-oriented problem treated in this paper. The common implementation details of the genetic algorithm can be found in Kellenbrink and Helber (2013) and Kellenbrink (2014). Those details are not repeated here as they are beyond the scope of this paper.

One of the most important aspects of any genetic algorithm is the *encoding of the individual candidate solutions*. In Kellenbrink and Helber (2013), we used an encoding of an individual solution I based on an activity list λ and an implementation list ν:

$$I = \left(\begin{array}{c|c} \text{Choice list } \alpha & \text{Activity list } \lambda \\ \text{Auxiliary information} & \text{Implementation list } \nu \end{array} \right) \tag{10}$$

For expository purposes, the encoding is augmented by a choice list α and auxiliary information on the left side of equation (10). Consider the project network in Figure 1 with the data in Table 1 and the following individual candidate solution:

$$I = \left(\begin{array}{cc|cccccccccc} 5 & 7 & 1 & 3 & 6 & 4 & 2 & 8 & 5 & 7 & 9 & 10 \\ 1;\{4,5\} & 5;\{7,8\} & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \end{array} \right) \tag{11}$$

This solution indicates that, in the first choice, between activities 4 and 5, activity 5 is selected for implementation. The selection of activity 5 in turn activates the second choice, in which activity 7 is chosen. Consequently, activities 4, 8 and 9 are not

implemented, which is indicated by the "0" entries in the implementation list; see also Figure 3 for a visualisation.

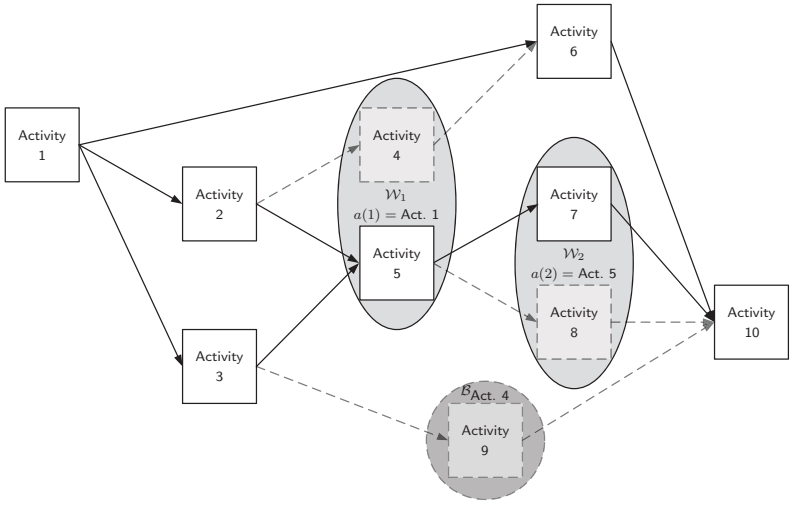

Figure 3: Project with structure B

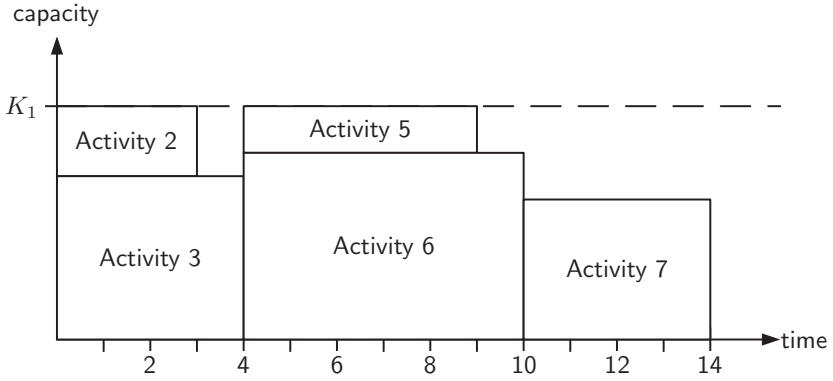

Figure 4: Schedule for individual I

In the standard serial generation scheme, each activity is then scheduled as early as possible with respect to capacity constraints and precedence relationships. In that scheduling process, activities are treated according to the sequence in the activity list λ. Consider the activity list λ in (11) and assume that the single renewable resource has a period capacity of 10 units. Figure 4 shows the resulting schedule for that individual solution I in (11). The first non-dummy activity to be scheduled is activity 3. Then, activity 6 is scheduled. Due to the capacity constraint, activity 6 can only begin after activity 3 is completed. Activity 4 is not implemented. Then, activity 2 is considered and scheduled to start as early as possible, i.e., at time 0. See (Kelley 1963, pp. 352-353), Kolisch and Hartmann (1999) or Kellenbrink and Helber (2013) for details on

that method. Each such individual I hence leads to exactly one project structure and schedule.

In the *fitness function* of the genetic algorithm, the unique schedule associated with each individual and its specific profit is determined. This determination is straightforward because all of the required revenue and cost information on the implemented activities and project termination as well as the resulting quality levels and aggregated quality grade can be found in or be immediately derived from the schedule. For further details on the implementation of the algorithm, such as crossover, mutation, and selection, please see Kellenbrink and Helber (2013).

5 Numerical results

5.1 Overview

The objective of this section is two-fold. On the one hand, we want to assess the performance of the genetic algorithm with respect to solution quality and computational effort. This topic is addressed in Section 5.2. On the other hand, we want to use a fictitious case study to demonstrate the type of effects that can be discovered and studied with respect to the behaviour of the solutions using our model. Section 5.3 is devoted to that latter aspect.

5.2 Performance evaluation of the genetic algorithm

In Kolisch et al. (1995), the ProGen instance generator for RCPSPs was introduced. It was extended in Kellenbrink (2014) and Kellenbrink and Helber (2013) to generate instances of flexible projects in a systematic way. For the purpose of this study, the generator was further extended to cover quality as well as cost and revenue aspects in addition to all of the other aspects originally considered in Kolisch et al. (1995). As a result, the following aspects of a RCPSP-PS-Q were varied systematically in our numerical study:

1. Number of activities
2. Network complexity, defined as the number of non-redundant precedence relationships per activity
3. Resource factor, describing how intensely activities are related to renewable and non-renewable resources

4. Resource strength, measuring the tightness of the capacity constraints of renewable and non-renewable resources
5. Number of choices among alternative activities
6. Number of alternative activities per choice
7. Number of caused activities and number of activities causing other activities

We constructed four problem classes A, B, C, and D with 30, 60, 90, and 120 non-dummy activities, respectively. To this end, the above-mentioned aspects were varied systematically in a full-factorial way to create 1536 test instances for each of the four classes.

We created the cases such that each test instance was feasible with respect to renewable resources, but not all of them were feasible with respect to non-renewable resources. Note that infeasibility with respect to non-renewable resources can be verified easily by solving a relaxed version of the RCPSP-PS-Q in which the precedence constraints as well as the capacity constraints for the renewable resources are omitted. If and only if the relaxed problem is infeasible is the original (non-relaxed) problem also infeasible. It is also reassuring to observe that a substantial fraction of the problem instances are infeasible due to non-renewable resources because this observation implies that the instance generator creates problems in which the non-renewable resources are tight enough to be relevant. Out of those 1536 instances for problem classes A to D, 1174, 1076, 1053, and 1030 instances were shown to be feasible, respectively. The exact parameters of those instances as well as the details regarding the construction process can be obtained from the authors on request, see also Kellenbrink and Helber (2013, Appendix B).

To assess the quality of the solutions found via our genetic algorithm, we compared the results to reference values from a (potentially aborted) branch&bound approach using CPLEX 12 on a 2.00 GHz Intel Xeon machine with 34 GB of RAM and four threads in a central computing cluster at Leibniz Universität Hannover, c.f. `www.rrzn.uni-hannover.de/clustersystem`. However, aggregating the deviations from those reference values is not straightforward due to the level of measurement in the objective function (1). Note that the profit is only interval-scaled but not ratio-scaled. In other words, because the profit has no meaningful absolute zero value (unlike, for example, the makespan), relative deviations, i.e., ratios, from optimal profit values are not meaningful. This particular aspect of the profit objective can easily be understood by considering the relative deviation from the optimal profit in a case in which this optimal profit is zero.

Table 8: Results for the test instances in class A with 30 non-dummy activities

	Genetic Algorithm			**CPLEX**
	1040 schedules	5040 schedules	10000 schedules	
Average gain [%]	99.84	99.89	99.89	100.00
Best [%]	85.09	88.76	89.27	100.00
Time [s]	0.02	0.07	0.14	277.20

Table 9: Results for the test instances in class B with 60 non-dummy activities

	Genetic Algorithm			**CPLEX**
	1040 schedules	5040 schedules	10000 schedules	
Average gain [%]	99.63	99.86	99.87	99.76
Best [%]	59.48	77.42	79.65	88.38
Time [s]	0.05	0.19	0.37	1269.85

Table 10: Results for the test instances in class C with 90 non-dummy activities

	Genetic Algorithm			**CPLEX**
	1040 schedules	5040 schedules	10000 schedules	
Average gain [%]	99.61	99.87	99.90	99.35
Best [%]	52.52	73.98	81.67	80.06
Time [s]	0.09	0.35	0.67	2014.68

Table 11: Results for the test instances in class D with 120 non-dummy activities

	Genetic Algorithm			**CPLEX**
	1040 schedules	5040 schedules	10000 schedules	
Average gain [%]	99.56	99.87	99.91	98.56
Best [%]	38.93	66.12	80.19	73.11
Time [s]	0.15	0.55	1.05	2694.78

For this reason, we defined the *achieved percentage of the possible profit gain* as a ratio-scaled performance measure. To this end, we first determined the smallest possible profit for each instance for the given time horizon by minimising instead of maximising the objective function in (1) in the relaxed problem version mentioned above (without precedence constraints and capacity constraints for renewable resources). Second, we tried to determine for each instance the maximum of the objective function values by i) applying the genetic algorithm and by ii) using CPLEX to determine a profit-maximising solution. We then determined the fraction of the profit interval (or range) for each instance that could be achieved via the genetic algorithm. Assume, for example, that the smallest possible profit is 20 and the largest possible profit (determined for the proven optimal CPLEX solution) is 90, whereas the best solution from the genetic algorithm has a profit of 80. Therefore, $\frac{80-20}{90-20} \approx 85.7\%$ of the possible profit gain can be achieved via the genetic algorithm. Those percentage values can then be averaged to assess the overall solution quality.

Although it was possible to determine the exact smallest possible objective function (profit) values for all of the feasible problem instances with quite limited numerical effort, such solutions were often not possible for the proven largest possible profit values. In our attempt to compute upper reference values, all of the instances of class A were solved to proven optimality. However, we had to limit the CPLEX computation times at the compute cluster to 7200 seconds of CPU time per instance for classes B to D. Within those CPLEX CPU time limits, there were cases in which the best solutions that we know of were found through the genetic algorithm. In such cases, the genetic algorithm hence found 100% of the possible profit interval gain, *as it is known to us*.

As in Kellenbrink and Helber (2013), we identified and eliminated the optional activities that would violate the capacity constraints of the non-renewable resources if they were implemented in a preprocessing step because they could never be part of any feasible solution but could inflate the required computation time. The genetic algorithm was implemented in Delphi XE and executed on a 2.66 GHz Intel Core2 Quad machine with 4 GB of RAM using a single thread. Each generation of the genetic algorithm consisted of $N^I = 80$ individuals (and hence schedules). To illustrate the improvement in the solutions over the generations, we present the results for generations 13, 63, and 125 N^G, i.e., after generating 1040, 5040, and 10000 schedules. The mutation parameters were set to $m^\alpha = 3\%$ and $m^\lambda = 10\%$, respectively.

The aggregated results are presented in Tables 8 to 11. Consider, for example, the results for problem class A in Table 8 for the case of 63 generations, i.e., 5040 generated schedules. On average, over the 1174 feasible instances of that class, 99.89%

of the possible profit gains were achieved via the genetic algorithm, and in 88.8% of the instances, the genetic algorithm found the best known (for this class even optimal) solution when generating those 5040 schedules. On average, this process required 0.07 seconds of CPU time per instance. Note that for all of the classes, the genetic algorithm on average achieved more than 99% of the possible profit gain that is known at present. As the number of activities increases from problem class A to D, the best known solutions were found more frequently using our genetic algorithm to generate 10000 schedules than using CPLEX. However, as the problem instances become larger, more generations of the genetic algorithm are required, i.e., more schedules have to be constructed and evaluated to find good solutions. Given the high overall quality of the results, the algorithm appears to be quite robust with respect to the different problem aspects that were systematically varied in our test bed. Even for the larger instances, the numerical effort of the modified genetic algorithm appears to be negligible compared with the CPLEX computation times. We thus conclude that the genetic algorithm is both fast and reliable.

5.3 Behaviour of solutions and structural insights

In this section, we address the structural aspects of the behaviour of the solutions, i.e., the "economic mechanics" of the problem. To this end, based on the project in network in Figure 5, we study a case that is still manageable yet is substantially richer than the example in Figure 1. We assume that one renewable and one non-renewable resource are required for the activities. Table 12 contains the activity-related data. Note that in this table we only consider the quality impact that is due to the optional activities. Furthermore, we consider two different quality dimensions. Figure 6 shows the relationship between the quality levels along those two quality dimensions and the resulting five-stage quality grade. The quality grade increases as the threshold values for the two quality dimensions are reached. The customer's assumed willingness to pay as a function of quality grade and project duration is depicted in Figure 7. The customer's willingness to pay increases substantially with increasing quality grade but only increases moderately with decreasing project duration.

For our numerical study we determined the optimal solutions for values of the capacity K_{r1} of the renewable resource ranging from 6 to 16 and for the non-renewable resource K_{n1} for values ranging from 65 to 85. The results with respect to profit, project duration and quality grade are summarised in Figures 8, 9, and 10. Figure 8 shows

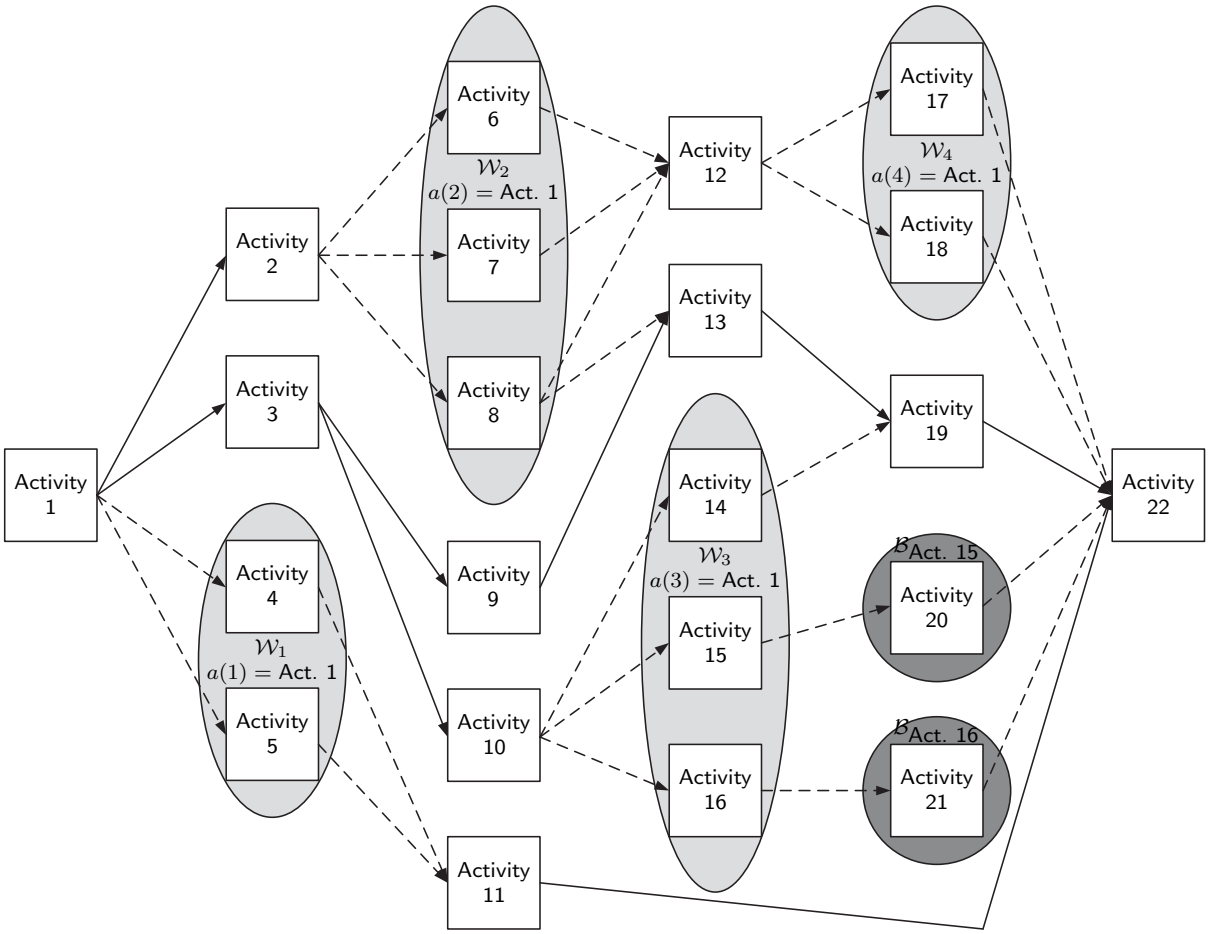

Figure 5: Flexible project with $2 \cdot 3 \cdot 3 \cdot 2 = 36$ project structures

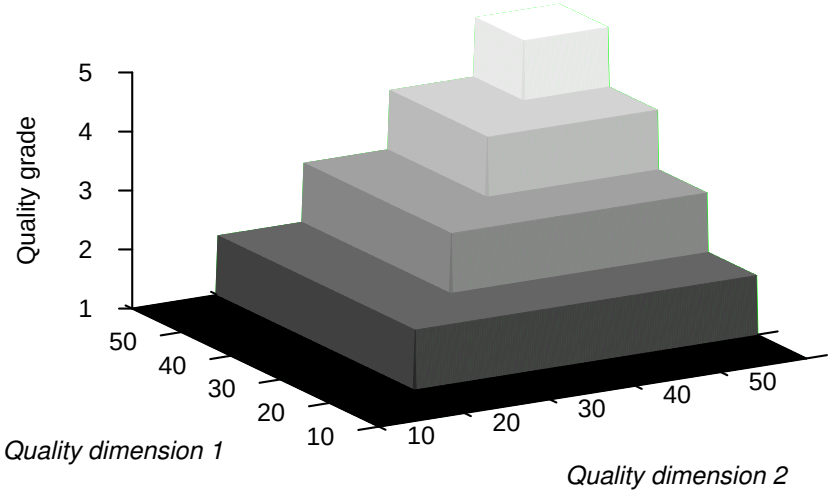

Figure 6: Quality grade as a function of two-dimensional quality levels

Table 12: Activity data

j	d_j	$k_{r1,j}$	$k_{n1,j}$	c_j	$q_{j,o1}$	$q_{j,o2}$
1	0	0	0	0	0	0
2	6	3	8	76	0	0
3	7	2	8	68	0	0
4	6	4	6	78	9	12
5	3	2	2	22	1	2
6	5	5	10	100	12	14
7	6	3	5	61	5	4
8	2	3	2	22	2	2
9	4	5	2	50	0	0
10	3	2	1	17	0	0
11	6	6	7	107	0	0
12	6	3	2	46	0	0
13	4	6	7	83	0	0
14	5	4	8	80	10	12
15	5	1	10	60	6	6
16	3	5	2	40	3	3
17	6	4	7	83	10	8
18	4	8	2	74	3	1
19	7	3	8	82	0	0
20	4	8	5	89	13	10
21	2	3	6	42	1	2
22	0	0	0	0	0	0

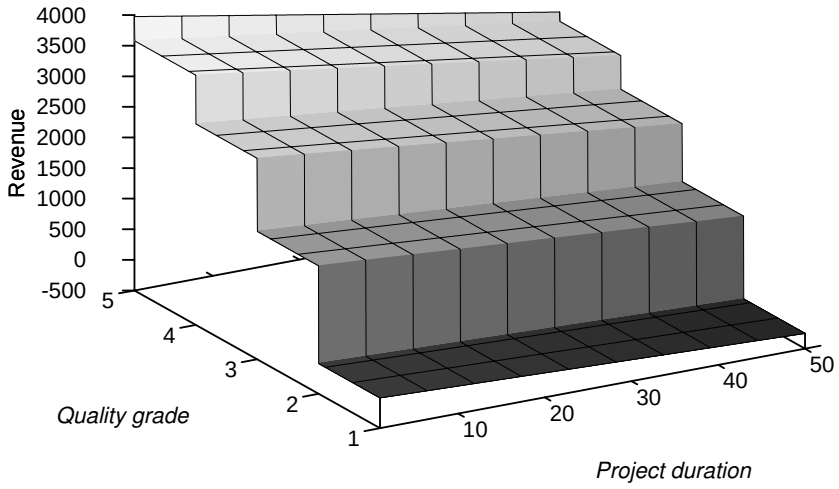

Figure 7: Revenue as a function of project duration and quality grade

that the profit increases as the capacity of both resource types increases. As the capacity of the renewable resource increases, the project duration decreases; see Figure 9. In

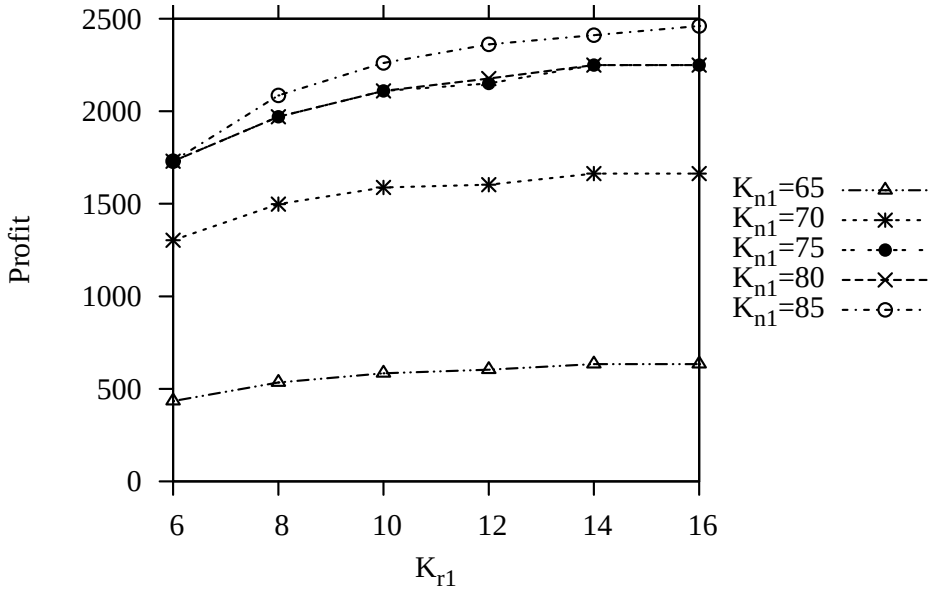

Figure 8: Profit as a function of capacity levels

almost all of the cases, the increase in the capacity of the renewable resource does not affect the quality grade. The single exception is the case with the highest capacity of the non-renewable resource $k_{n1} = 85$, in which an increase in the capacity of the renewable resource from 6 to 8 leads to a change in the quality grade from 4 to 5; see Figure 10. The more substantial profit increase in Figure 8 is due to an increase in the capacity of the non-renewable resource. Due to this capacity increase, optional activities that require a high resource consumption of the non-renewable resource and lead to higher quality and revenues can be selected.

Figure 9 indicates that the project duration decreases as the capacity of the renewable resource increases because more and more activities can be performed in parallel. Note, however, that an increase in the capacity of the non-renewable resource in several cases leads to an increase in the project duration. The reason is that more elaborate activities can now be selected, yielding a substantially higher quality grade and eventually a higher profit, even though the project duration may increase.

These examples show how complex the interactions between project structure, quality, capacity requirements, project duration and profit can be. The model presented in Section 3 captures these relationships, and the algorithm outlined in Section 4 solves these problems quickly and with high solution quality.

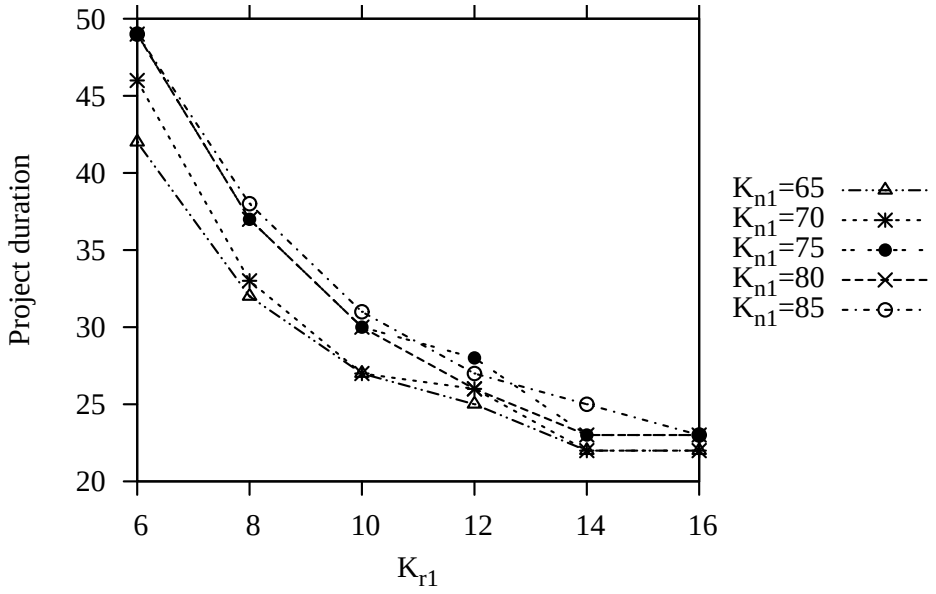

Figure 9: Project duration as a function of capacity levels

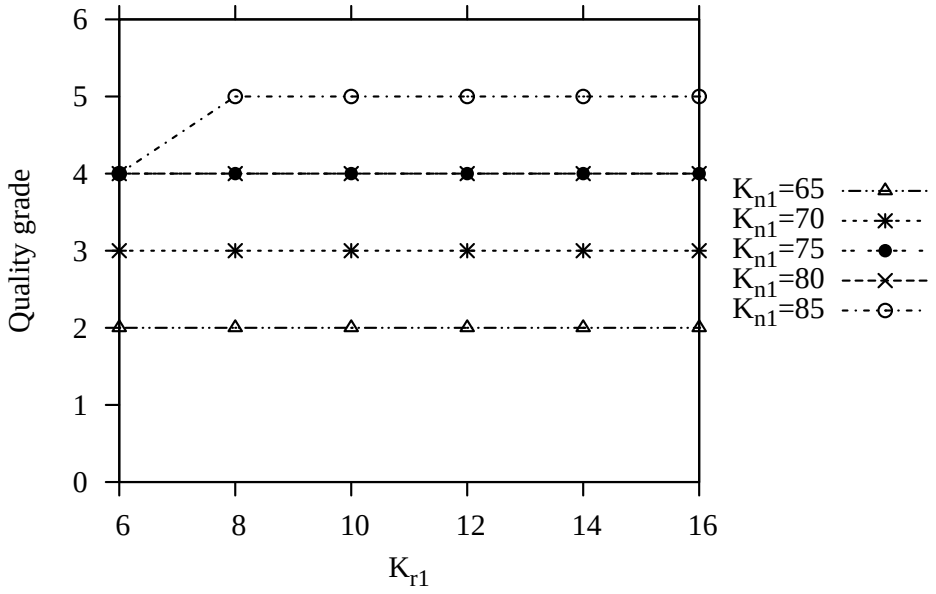

Figure 10: Quality grade as a function of capacity levels

6 Conclusion

We have studied the problem of determining both an optimal project structure and an optimal schedule for a specific type of a flexible project. In such a project, the optional activities affect the quality of the project outcome, the cost, the revenue and the re-

source consumption. We presented a formal model for the problem as well as the core elements of a genetic algorithm to solve the model for problem instances with a large number of activities. In a systematic numerical study on accuracy and performance, the algorithm showed highly satisfactory behaviour. In a further study on structural effects, we investigated and explained the complex interactions between the different aspects of that complex problem. We obtained both intuitive and (perhaps) not so intuitive results. Among the intuitive results are the decreasing rates of return of both renewable and non-renewable resource availabilities with respect to projects' profit and the decrease in project duration due to an increase in renewable resource capacities. A perhaps not so intuitive result is that an increase in the capacity of a non-renewable resource can lead to an increase in project duration. Combining project structure flexibility with quality and profit aspects hence appears to lead to a rich and potentially quite useful project-scheduling approach. A possible direction for further research is the multi-project problem in which different flexible projects compete for resources. Furthermore, given that the genetic algorithm is rather fast, one could modify it to determine an approximation of the front of Pareto-efficient solutions with respect to quality grade, cost and duration of the project. From the algorithmic perspective, one could try to generalize other efficient algorithms for resource-constrained project scheduling to solve the model proposed in this paper, taking advantage of the comparison presented in Kolisch and Hartmann (2006).

References

Belhe, U. and A. Kusiak (1995). Resource Constrained Scheduling of Hierarchically Structured Design Activity Networks. *IEEE Transactions on Engineering Management 42*, 150–158.

Blazewicz, J., J. K. Lenstra, and A. H. G. Rinnooy Kan (1983). Scheduling subject to resource constraints: Classification and complexity. *Discrete Applied Mathematics 5*, 11–24.

Brucker, P., A. Drexl, R. Möhring, K. Neumann, and E. Pesch (1999). Resource-constrained project scheduling: Notation, classification, models, and methods. *European Journal of Operational Research 112*, 3–41.

Čapek, R., P. Šůcha, and Z. Hanzálek (2011). Production scheduling with alternative process plans. *European Journal of Operational Research*.

Demeulemeester, E. L. and W. S. Herroelen (2002). *Project Scheduling. A Research Handbook*. Boston [u.a.]: Kluwer Academic Publishers.

Elmaghraby, S. E. (1964). An algebra for the analysis of generalized activity networks. *Management Science 10*, 494–514.

Erenguc, S. S. and O. Icmeli-Tukel (1999). Integrating quality as a measure of performance in resource-constrained project scheduling problems. In J. Węglarz (Ed.), *Project Scheduling: Recent Models, Algorithms and Applications*, pp. 433–450. Boston [u.a.]: Kluwer Academic Publishers.

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Reading, Massachusetts [u.a.]: Addison-Wesley Publishing Company.

Hartmann, S. (2001). Project Scheduling with Multiple Modes: A Genetic Algorithm. *Annals of Operations Research 102*, 111–135.

Hartmann, S. and D. Briskorn (2010). A survey of variants and extensions of the resource-constrained project scheduling problem. *European Journal of Operational Research 207*, 1–14. Hartmann, Soenke Briskorn, Dirk.

Herroelen, W. S., B. De Reyck, and E. L. Demeulemeester (1998). Resource-constrained project scheduling: A survey of recent developments. *Computers & Operations Research 25*, 279–302.

Icmeli-Tukel, O. and W. O. Rom (1997). Ensuring quality in resource constrained project scheduling. *European Journal of Operational Research 103*, 483–496.

Kellenbrink, C. (2014). *Ressourcenbeschränkte Projektplanung für flexible Projekte*. Springer Gabler.

Kellenbrink, C. and S. Helber (2013). Scheduling resource-constrained projects with a flexible project structure. *Diskussionspapier der wirtschaftwissenschaftlichen Fakultät der Leibniz Universität Hannover Nr. 511*.

Kelley, J. E. (1963). The Critical-path Method: Resources Planning and Scheduling. In J. F. Muth and G. L. Thompson (Eds.), *Industrial Scheduling*, pp. 347–365. Englewood Cliffs: Prentice-Hall.

Klein, R. (2000). *Scheduling of Resource-Constrained Projects*. Boston [u.a.]: Kluwer Acadamic Publishers.

Kolisch, R. and S. Hartmann (1999). Heuristic algorithms for the resource-constrained project scheduling problem: classification and computational analysis. In J. Węglarz (Ed.), *Project Scheduling: Recent Models, Algorithms and Applications*, pp. 147–178. Boston [u.a.]: Kluwer Academic Publishers.

Kolisch, R. and S. Hartmann (2006). Experimental investigation of heuristics for resource-constrained project scheduling: An update. *European Journal of Operational Research 174*(1), 23 – 37.

Kolisch, R. and R. Padman (2001). An integrated survey of deterministic project scheduling. *Omega 29*, 249–272.

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

Kuster, J., D. Jannach, and G. Friedrich (2009). Extending the RCPSP for modeling and solving disruption management problems. *Applied Intelligence 31*, 234–253.

Kuster, J., D. Jannach, and G. Friedrich (2010). Applying Local Rescheduling in response to schedule disruptions. *Annals of Operations Research 180*, 265–282.

Li, H. and K. Womer (2008). Modeling the supply chain configuration problem with resource constraints. *International Journal of Project Management 26*, 646–654.

Neumann, K. (1990). *Stochastic Project Networks*. Berlin [u.a.]: Springer-Verlag.

Talbot, F. B. (1982). Resource-constrained project scheduling with time-resource tradeoffs: The nonpreemptive case. *Management Science 28*, 1197–1210.

Tareghian, H. R. and S. H. Taheri (2006). On the discrete time, cost and quality trade-off problem. *Applied Mathematics and Computation 181*, 1305–1312.

Tiwari, V., J. H. Patterson, and V. A. Mabert (2009). Scheduling projects with heterogeneous resources to meet time and quality objectives. *European Journal of Operational Research 193*, 780–790.

Vanhoucke, M. (2006). Scheduling an R&D project with quality-dependent time slots. In Gavrilova, M and Gervasi, O and Kumar, V and Tan, CJK and Taniar, D and Lagana, A and Mun, Y and Choo, H (Ed.), *Computational science and its applications - ICCSA 2006, PT 3*, Volume 3982 of *Lecture notes in computer science*, pp. 621–630.

Węglarz, J., J. Józefowska, M. Mika, and G. Waligóra (2011). Project scheduling with finite or infinite number of activity processing modes – a survey. *European Journal of Operational Research 208*, 177–205.

Özdamar, L. and G. Ulusoy (1995). A survey on the resource-constrained project scheduling problem. *IIE Transactions 27*, 574–586.