Optimal Project Portfolio Execution

Using analytical and simulation models with realistic project layouts and resource behaviour

OLA JOHANNESSON
CHRISTOFER JOHANSSON HIITTI

Master of Science Thesis
Stockholm, Sweden 2014

Optimal Project Portfolio Execution

Using analytical and simulation models with realistic project layouts and resource behaviour

OLA JOHANNESSON
CHRISTOFER JOHANSSON HIITTI

Master's Thesis in Systems Engineering (30 ECTS credits)
Master Programme in Aerospace Engineering (120 credits)
Royal Institute of Technology year 2014
Supervisor at KTH was Per Enqvist
Examiner was Per Engvist

TRITA-MAT-E 2014:04
ISRN-KTH/MAT/E--14/04--SE

Royal Institute of Technology
School of Engineering Sciences

KTH SCI
SE-100 44 Stockholm, Sweden

URL: www.kth.se/sci

Optimal Project Portfolio Execution

Using analytical and simulation models with realistic project layouts and resource behaviour

OLA JOHANNESSON 890502-0635
CHRISTOFER JOHANSSON HIITTI 891018-8534

Master's Thesis at SCI
Supervisor: Per Enqvist
Examiner: Per Enqvist

TRITA xxx yyyy-nn

Abstract

This project aims to increase the knowledge in the area of resource allocation in R&D portfolios, a topic interesting to both the industry and academic research. The thesis investigates how to optimally allocate resources to projects in a R&D portfolio, with focus on how many projects that are optimal to run in parallel. A complex mixed-integer nonlinear programming model with a realistic stage-gate project layout and advanced resource behaviour, including project resource learning and other efficiency losses, is developed and also proven unsolvable in realistic time. A simplified model handling learning is proposed and solved using a mixed integer solver for a small portfolio. A simulation framework implementing all complexities is developed, and used to find portfolio parameters affecting the optimal number of projects to run in parallel, using a Monte Carlo method.

From the simplified mathematical model optimal allocations from small portfolios are presented, and from the simulation several results from large portfolios using different resource allocation strategies are presented. From these results it is argued that there exists an optimal number of projects to run in a portfolio, and that a portfolio run with either larger or smaller number of running projects produces a lower gain. This number is however found to be highly dependent on the size and specification of the portfolio, and how resources are allocated to projects.

Sammanfattning

Detta masterarbete har som mål att utöka kunskapen kring resursallokering inom portföljer med utvecklingsprojekt, ett område som genererat intresse både inom industrin och inom tidigare akademiska arbeten. Rapporten undersöker hur resurser optimalt ska allokeras till projekt inom utvecklingsportföljer, med huvudsakligt fokus på hur många projekt som är optimalt att köra på samma gång. Ett ickelinjärt mixed-integer optimeringsproblem, som bygger på en realistisk stage-gate projektmodell samt avancerade approximationer av resusers beteende, ställs upp; modellen inkluderar bland annat lärotider vid allokering till nya projekt och andra effektivitetsförluster. Denna modell bevisas olösbar med realistisk beräkningskraft. En förenklad modell som tar hänsyn till läran-deeffekterna tas fram och löses för små portföljer. Ett simuleringsprogram tas också fram vilket inkluderar all komplexitet från den fullständiga modellen, och detta används för att med en Monte Carlo-metod undersöka vilka portföljparametrar som påverkar hur många projekt som optimalt ska köras parallellt.

Från den förenklade matematiska modellen tas optimala resursfördelningar fram för små portföljer, och från simuleringarna presenteras data från realistiskt stora portföljer vilka har simulerats med olika resursallokeringsstrategier. Dessa resultat visar på att det finns ett optimalt antal projekt att köra parallellt i en portfölj, och att om en portfölj drivs med antingen färre eller fler projekt parallellt ger det en lägre vinst. Exakt vad detta optimala antal projekt är beror dock mycket på storleken och detaljer i portföljen, och hur resurser allokeras till projekt.

Contents

0.1 Acknowledgements

We would like to express our very great appreciation to Mattias Nordin in Atlas Copco, Lars Cederblad in Level 21 and Per Enquist in KTH for making this thesis happen and guiding us through the project in their roles as our supervisors. We would also like to extend our thanks to Bengt Hallberg in Level 21 and Jan Kalander in Atlas Copco for their constructive inputs on many parts of this project.

Introduction

In a Research and Development (R&D) department, as in every project-driven environment, the question of how to select and run a project portfolio arises. How this question is answered highly affects the output from the portfolio. The problem can be divided into two parts, one concerned with optimal portfolio selection and one treating project scheduling and resource allocation in such an optimal portfolio. In optimal portfolio theory simplified economic models are developed for the projects, often describing expected profit and associated risk, and then the combination of these giving optimal profit under certain boundary conditions are calculated. The allocation problem is in its basic form a relatively simple problem with well-known solutions [7]. Models that combine both these steps have been proposed, but have had the downside of being highly simplified in order to be solvable [4].

In practice the different project portfolio selection tools and theories are widely used and implemented as company policies. However the use of advanced models for portfolio execution and staff allocation is limited, instead strategies such as maximizing resource utilization instead of R&D output are widely used, according to experts in *Atlas Copco* and *Level 21 Management*. A major problem with the models come from the simplifications done when describing projects, the portfolio and the environment that it is executed in. These simplifications most notably include only looking at one competence, or a project description with fixed run time and fixed work demand over time. Models with these limitations provide limited use in real-life multi-competence environments with flexible deadlines. They also lack the possibility to study one of the main portfolio management variables, the effect of the number of projects running simultaneously.

In order to further understand the execution of portfolios in a complex environment a model capable of describing realistic projects, with demands for multiple competences, will be presented in this paper. The project model makes use of a task based internal model to give support for advanced project descriptions; it is focused on project following a stage-gate model, a model widely used in R&D projects [8]. The model presented optimises for maximum future profit from the project portfolio. From this model both an optimisation problem and a simulation framework are formed, and a combination of these are used to examine the behaviour of optimal solutions. The existence of a global optimal solution is examined for small problems, and proposals and evaluation of practically usable allocation strategies are made.

The model is based on information from the large multinational manufacturing

company *Atlas Copco* and knowledge from *Level 21 Management*, and the model is most suited for similar organizations with a large project-base and several projects running at any given time.

1.1 Cooperation

The project and research on which this report builds was made in a team consisting of the two authors of this report and Elhabib Moustaid. A report titled *Optimal Project Portfolio Execution - Computer Implementation of Models and Simulation Framework* [9] is also available covering the same models and implementations, but with further focus on the computer implementations and solutions.

Bibliography

Solving the portfolio execution and resource allocation problem is highly desired by companies and is thus subject to research by the scientific community. The research in this field is extensive and broad since there are many different ways to model the problem, and several special cases that can be considered. Earlier works and publications in the area are mostly focused on choosing and scheduling projects for a portfolio, however the resource allocation problem has been studied as well. Studies have been done handling the individual resources in respect to knowledge and learning associated with being allocated to certain projects, and others considering resources as a source of costs and thus treating budget allocations over time.

Ghasemzadeh et. al [4] propose a way to select possible projects to a project portfolio to be executed. They also propose the scheduling of those projects within the portfolio with a varying consumption of resources. They solve this using a Zero-one integer linear programming model, however they are scheduling projects with a fixed run time.

Solak et. al [1] looks at the problem differently and argues that the problem needs to be modeled as a multistage stochastic integer problem. They introduce uncertainty about the running projects which needs to be progressively learnt during project execution, the unknown variable is the work required. They also introduce project dependencies, that one project may not be started before a knowledge in its technology is filled by another project. They identify the full multistage problem to be NP-hard and reduce it by going from multistage to a two-stage problem which they later solve by a sample average approximation method of Lagrangian relaxation and a new lower bounding heuristic. However they limit the problem to one competence.

Kira et. al [6] investigates a neighbouring project selection and allocation problem to R&D projects, namely IS projects (Information system, software and hardware decisions) under risk. They use linear programming to solve the problem and conducts extensive analysis about the effect of uncertainty due to unknown variables, project work demand etc, using Monte Carlo simulation. They also develop a Specific Decision Support System to aid management decision making guided from their results.

Matt Bassett [2] investigates how to optimally assign project tasks to people in respect to utilization of their expertise and hiring outside resources. He uses heuris-

tic to quickly give a feasible solution to the problem of globally assigning people to the available tasks and also a mathematical program which is more rigorous. However he is assuming a fixed consumption of resources.

Certa et. al [5] focuses on fixed-time-frame completion of multiple projects and treats learning effects and social interaction effects. They use multi-skilled resources in their model. This leads to the problem being multi-objective since knowledge and development of the resources is as desired as pure monetary output from a portfolio execution.

The stage gate model for project execution, first described by Cooper in 1990 [3], describes how a project can be separated into a number of sequentially executed parts (stages) with control checkpoints between them (gates). He argues that an important factor in the model is parallel execution of activities involving different functions in the firm. Through case-studies in multiple companies Högman et. al. finds that the stage gate model is widely used mainly in product development, but also is implemented in technology development with some slight modifications [8].

2.1 Conclusion

In previous work the focus has been mainly on project selection and scheduling [1][6][5], and in some cases simple allocation models. The resource allocation problem has however been studied previously, both in its basic form [1][6] and incorporating extensions such as knowledge accumulation [5] and hiring external resources [2]. All of these does however work with a simple model for projects, and by investigating the behaviour of a portfolio with multiple competences and an advanced project model [3] this paper seeks to contribute new knowledge to the field.

Problem formulation

As presented in Section 2.1 the main focus on previous research in portfolio execution has been that of allocating resources to projects with fixed run time. This is however a simplification that limits the optimal solution. A dynamic project execution time allows for further freedom in the model and thus constitutes an approach to the optimal solution to the portfolio execution and allocation problem. With this model that gives a higher degree of freedom in portfolio execution the possibility arises to study phenomena previously hidden in simplifications.

In both *Atlas Copco* and *Level 21 Management* there has been expressed concern that there might be a tendency in R&D departments to start too many projects, and for this reason the report aims to extend the knowledge of portfolio execution and resource allocation in ways that helps answer this question.

3.1 Research question

Based in this a main research question and two important sub questions are identified:

- Given a R&D portfolio, does it exist an *optimal number* of running projects, in the sense that gives the highest profit?
- If so, what is this number for a given R&D portfolio?
- How does the portfolio composition affect this number?

Definition of concepts

In the description of a R&D department and the running of projects a number of concepts are necessary, in this chapter all major concepts are defined for use in this paper.

4.1 The R&D department

A Research and Development (R&D) department is consistently in this paper modelled as one with a predefined constant work force. The department is assumed to be a multi competence environment, where resources are divided into groups based on their main competence, and a project might need resources from several different competence groups to be completed.

4.2 Resource

A resource is one person working in the R&D department; a resource has a fixed amount of available work hours per week, to be allocated to projects. The resource is defined to work in only one area, and thus belongs to one single competence group. Dividing resources by competence is a simplification not done in all previous research [5], however it is valid for the R&D department in *Atlas Copco* and many companies with them.

4.3 Competence

A competence is a skill type, or a knowledge area, and a resource of a specific competence can only produce work on a project task with matching competence.

4.4 Project

A project is a description of a combination of work that together produces an output (new product, improvement) with monetary value. It is built up of several project tasks, put together to create a stage-gate project description [3], see Figure 4.1. In each stage a number of project tasks represents the demand for different competences, with only one task per competence and stage.

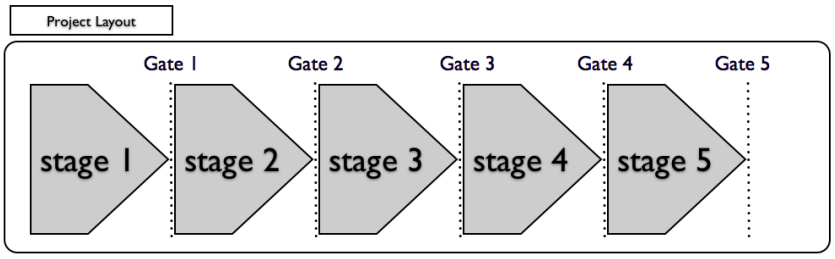

Figure 4.1. State gate project

4.4.1 Project task

A project task is a part of a project, with demand of one *single competence*. It is defined to be finished when the work done on the tasks exceeds its demand. A task may have dependencies on other project tasks in such a way that it cannot be started before its dependency tasks are done.

4.5 Demand

Demand is the amount of work required for a project task, and is measured in *work hours* with perfect efficiency. Because the efficiency is usually lower, the amount of allocated man work hours required to finish a task usually exceeds its demand.

4.6 Portfolio

A portfolio is a grouping of multiple projects that are desired to be run in the R&D department. They are assumed to be preselected using portfolio selection techniques and to be ordered in the desired run order by some other part of the company, for example marketing or higher management. However it is assumed that no project dependency exists so the projects execution order is free. In the portfolios used all projects share the same basic layout of tasks and is described by a stage-gate model, see A.1. The total demand for all competences are scaled differently to create projects of different sizes.

4.7 Portfolio execution strategy

Portfolio execution strategies are heuristic algorithms for allocating resources to projects.

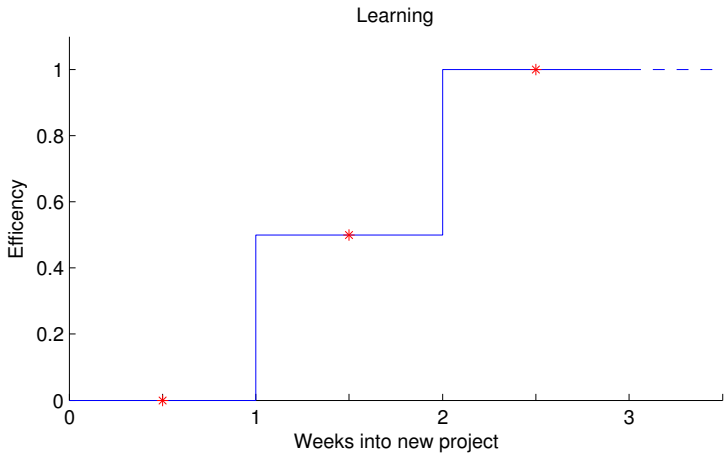

Figure 4.2. Effect of learning

4.8 Allocation

An allocation is a link between a resource and a project, it describes how many hours a resource will devote to a specific project task in a week. Because of efficiency this is however not necessarily the amount of work the resource do on the task.

4.9 Efficiency

Even if a resource is working fully in a project task, the actual work produced will not necessarily be the same as what is allocated. The amount of work produced will vary with a number of efficiency functions, described below. These functions have been developed in cooperation with the R&D portfolio management team in *Atlas Copco* and experts in *Level 21 Management*. The total efficiency is the product of all efficiency functions evaluated for a resource-task combination. As several of these functions are non-convex the total efficiency function is also non-convex.

4.9.1 Learning

A resource will not work fully if it does not know the project, it needs to learn it, thus efficiency loss, see Figure 4.2. If a resource have not been working on a project for a while it will start to lose its knowledge and thus needing to learn again, see Figure 4.3.

4.9.2 Ramping

When new resource are allocated to a project old resources will need to take time from their productive work time to teach the new resources, thus efficiency loss on

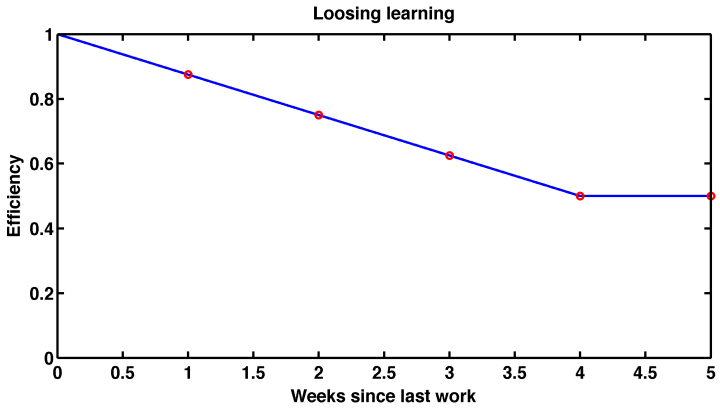

Figure 4.3. Effect of loosing learning

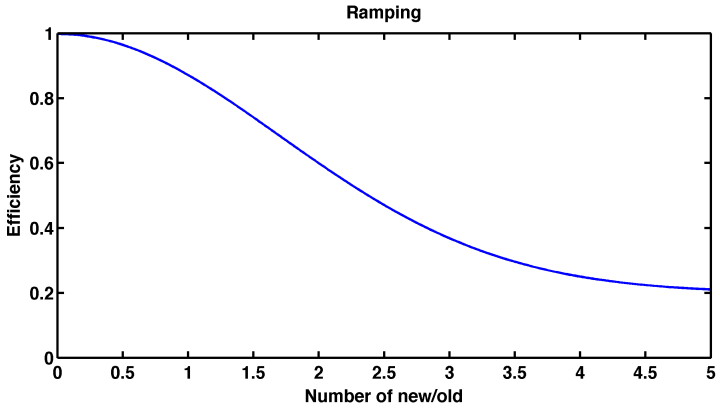

Figure 4.4. Effect of ramping

all old resources. The efficiency loss is proportional to the ratio of new and old resources, see Figure 4.4.

4.9.3 Multiple projects

When a resource is working on multiple projects its overall efficiency will be affected. This is due to the fact that the resource needs switch projects during the working week which introduces some overhead work. It can also be a positive change in efficiency since if one project is stuck the resource can work on another, see Figure 4.5.

4.9.4 Multiple resources

If there are many resources working on a project the project needs more overhead, such as team leaders and groups, and also more time will be spent in group meetings.

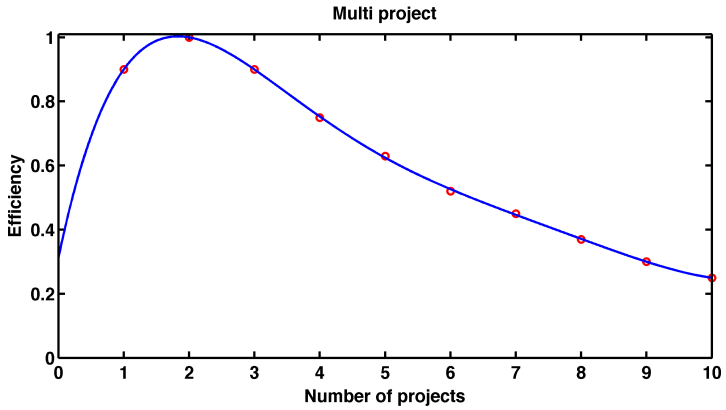

Figure 4.5. Effect of multiple projects

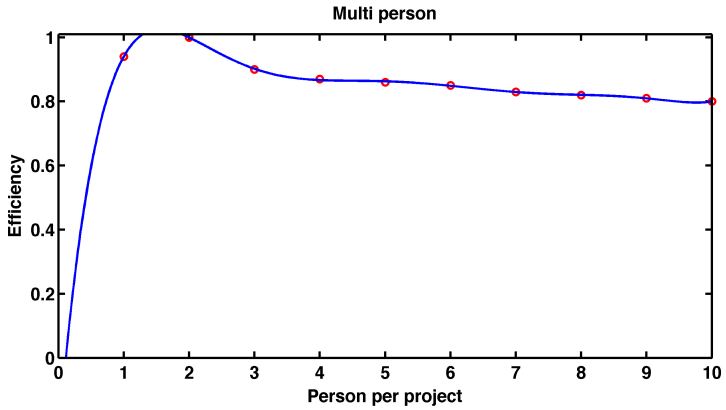

Figure 4.6. Effect of multiple resources

This will reduce the overall efficiency on that project, see Figure 4.6.

Valuation of a portfolio

An important aspect of optimising a project portfolio is deciding the criteria for optimality. In this paper the chosen measurement is purely economical, in other words the optimal portfolio is the one that gives the company the highest future monetary value.

5.1 Objective function

An important factor for the behaviour of the solution is the chosen objective function. Net Present Value (NPV) is the chosen economic measure in this model, as it is a number that represents the entire future of both a project, and when combined, a portfolio. The NPV is the accumulated sum of all future cash flows, positive and negative, discounted with a discount rate to compensate for the time value of money. Negative cash flows include cost of work force and prototyping, while positive cash flows originates in product sales upon project completion. An example can be seen in Figure 5.1. The NPV is defined as

$$NPV = NV_{k=0} = \sum_k c_k \frac{1}{(1+r)^k} \tag{5.1}$$

where c_k is the cash flow in time period k and r is the discount rate. $NV_{k=0}$ is the Net Value at time $k = 0$. If the NV is calculated for another time step, it can be moved in time using the formula

$$NV_{k=n} = NV_{k=m} \cdot \frac{1}{(1+r)^{m-n}}. \tag{5.2}$$

Typically this is the case when all gains in a project are discounted from the finishing date of the project prior to optimisation, and then this value is shifted to an NPV when the finishing date of the project is known.

However as the environment studied is that of a constant size R&D department, the cost of work force will always sum up to the same amount. Further prototyping cost will be the same regardless of execution time. For this reason negative cash flows are not considered in the objective function, instead the Net Present Profit (NPP) is introduced as

$$NPP = NP_{k=0} = \sum_k c_k^+ \frac{1}{(1+r)^k} \tag{5.3}$$

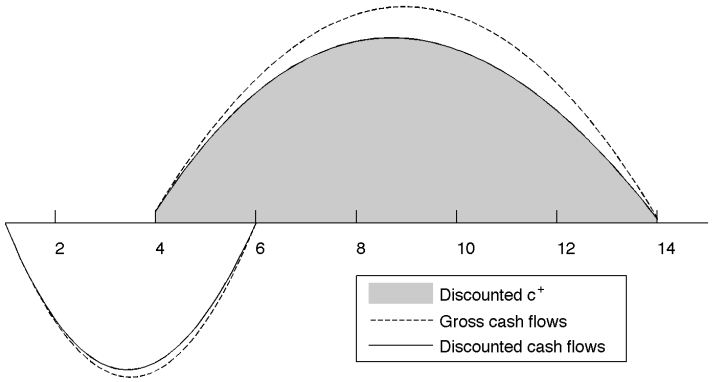

Figure 5.1. Cash flows over time for a project

where c_k^+ is the positive cash flow in period k. The cash flows can be either those for one project, or the accumulated cash flow for a full portfolio. The NP can be shifted as described in function (5.2).

5.2 Dynamics of the portfolio

The project portfolio of a R&D department is constantly changing and lacks natural start- and endpoints; the possibility to value a portfolio that contains half-done projects is crucial. If a fixed time-period is considered both projects that is started before the period starts and produces gain in the period, projects that are started in the period but produces gain after it and project started before and finishing after the period have to be handled. This is handled by making only a fraction of the profit from a project counted if it is not fully done in the period. The fraction used is the fraction of demand done in the time period, and in the case of equal cost for all resources, the fraction of costs for executing the project. An illustration of this can be found in Figure 5.2, that shows the considered cash flows for a project that is started before the time period.

Projects not finished in the time period are treated similarly, with the addition that as the project finishing date is unknown. To resolve this a simple approximation is used:

$$k_f = round((k_e - k_s)\frac{w_l}{w}) \tag{5.4}$$

where k_f is the time step the project is expected to finish, k_e is the end step of the period, k_s is the step the project started in, w is the total demand for the project

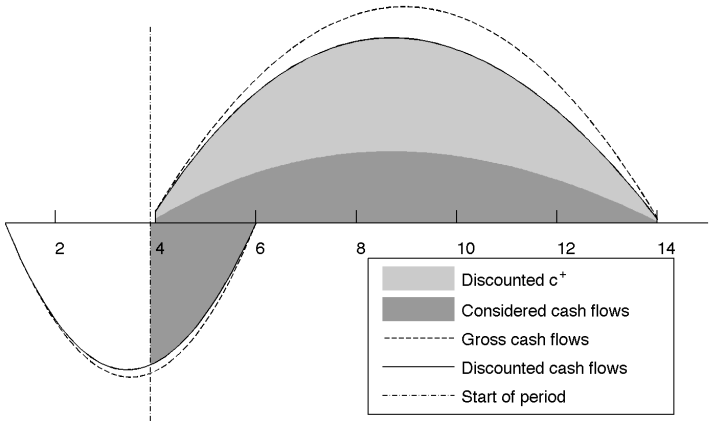

Figure 5.2. Cash flows over time for a project, with start time

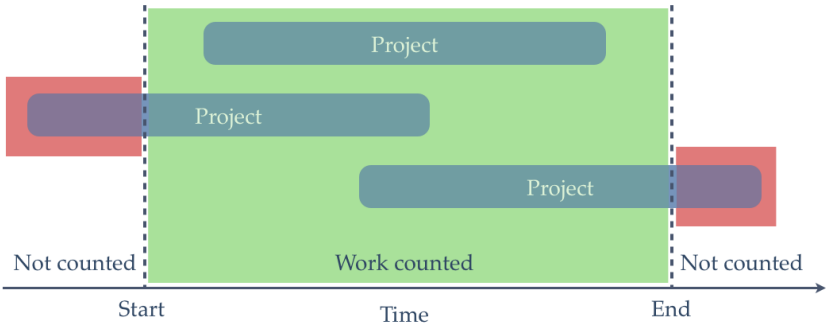

Figure 5.3. A portfolio with unstarted and unfinished projects

and w_l is the demand left at the end of the period. A schematic of this can be seen in Figure 5.3.

Mathematical model

In order to gain understanding of the problem mathematical problems are formulated for both the full problem with all nonlinearities and arbitrary task dependencies. A simplified model that is practically solvable is also presented.

6.1 Nonlinear model

The model aims to capture the full complexity of the R&D portfolio allocation problem, and thus include all concept previously discussed. That includes arbitrary task dependencies and all nonlinearities. First the notation will be presented, then the equations are presented and explained.

6.1.1 Notations

Notations used in the equations.
G_p: A gain value.
λ: A positive constant for the discount rate d_r, $\lambda = \frac{1}{1+d_r}$.
g_i^k: Gain of project i in time period k, $g_i^k = G_p\lambda^k$
C_r: The cost of the resource r.
c_r^k: The cost of the resource r in time period k, $c_r^k = C_r\lambda^k$
T_0: The first time step and T_f the last time step.
Q: Set of tasks.
Q_i: The set of task belonging the the project i. $Q = \cup_i Q_i$ and $\cap_i Q_i = \emptyset$, the last condition means that a task belongs to only one project.
D_j: Set of task demanding competence j. $Q = \cup_j D_j$ and $\cap_j D_j = \emptyset$, the last condition means that a task demands only one competence.
R_j: Set of resources of competence j. $\cap_j R_j = \emptyset$, the last condition means that a resource has only one competence.
di_q the initial demand of the task q.
Γ_q: set of dependencies of the the task q. i.e the set of tasks on which q is dependent.
N_p: Number of projects, N_r number of resources and N_c number of competences.
Used sets: $R = r \in 1 \dots N_r$, $K = k \in T_0 \dots T_f$, $J = j \in 1 \dots N_c$, $I = i \in 1 \dots N_p$
M : Big constant

We also denote by $c(r)$ the competence of the resource r, and $c(q)$ the competence demanded by task q.

Variables

d_q^k: Demand of the task q in time step k.

$a_{q,r}^k$: Allocated amount of the resource r to task q in time step k.

x_i^k: A binary variable describing the project i state,

$$x_i^k = \begin{cases} 1 & \text{Project } i \text{ is finished in time step } k \\ 0 & \text{otherwise} \end{cases}.$$

y_q^k: A binary variable describing the task q state,

$$y_q^k = \begin{cases} 1 & \text{task } q \text{ is finished in time step } k \\ 0 & \text{otherwise} \end{cases}.$$

s_q^k a binary variable describing if the task q state,

$$s_q^k = \begin{cases} 1 & \text{task } q \text{ is running in time step } k \\ 0 & \text{otherwise} \end{cases}.$$

$\xi(r, p, k)$: Efficiency function, the efficiency of the resource r inside the project p in the time period k.

o_i^k a binary variable to set order between projects.

$$o_i^k = \begin{cases} 1 & \text{project } i \text{ started/is running in time step } k \\ 0 & \text{otherwise} \end{cases}.$$

6.1.2 Equations

The global nonlinear problem:

$$\min \sum_{k=T_0}^{T_f} \sum_{r=1}^{N_r} \sum_{q \in Q} a_{q,r}^k c_r^k - \sum_{k=T_0}^{T_f} \sum_{i=1}^{N_p} g_i^k x_i^k \tag{6.1}$$

$$\text{s.t.} \sum_{q \in Q} a_{q,r}^k \leq 1 \qquad \forall r \in R, k \in K \tag{6.2}$$

$$d_q^k = d_q^{k-1} - \sum_{r \in R_j} a_{q,r}^{k-1} \cdot \xi(r,p,k) \qquad \forall q \in Q, j \in J, k \in T_0 + 1 \ldots T_f \tag{6.3}$$

$$d_q^k \leq M(1 - y_q^k) \qquad \forall q \in Q, k \in K \tag{6.4}$$

$$x_i^k \leq \frac{\sum_{q \in Q_i} y_q^k}{card(Q_i)} \qquad \forall i \in I, k \in K \tag{6.5}$$

$$s_q^k \leq \frac{\sum_{q \in \Gamma_q} y_q^k + 1}{card(\Gamma_q) + 1} \qquad \forall q \in Q, k \in K \tag{6.6}$$

$$a_{q,r}^k \leq M s_q^k \qquad \forall q \in Q, r \in R, k \in K \tag{6.7}$$

$$\sum_{k=T_0}^{T_f} x_i^k \leq 1 \qquad \forall i \in I \tag{6.8}$$

$$x_i^k, y_q^k, o_i^k, s_q^k \in \{0,1\} \tag{6.9}$$

$$d_q^1 = di_q \qquad \forall q \in Q \tag{6.10}$$

$$a_{q,r}^k \geq 0 \tag{6.11}$$

Equation 6.1 is the value function which is to minimize the cost of using resources and maximize the final gain of the portfolio. A resource cannot be over allocated, the maximum allocation is here one unit, which equation 6.2 enforces. The dynamics of the model is captured in Equation 6.3 where the demands for project tasks are decreased for allocations on the same competence as the project task. Here is also where the non-linear non-convex efficiency (Section 4.9) function is included. Equation 6.4 updates information regarding if a task is done or not, similarly the status of a project is updated in Equation 6.5. A task cannot be set as running before all its dependency tasks are done, this is forced by Equation 6.6. Allocations may not be set on projects that are not running, see Equation 6.7. The gain from a finished project is only to be counted once as defined by NPP (Equation 5.3), realised by Equation 6.8. Equation 6.9 forces the binary variables to take only one or zero as value and Equation 6.11 forces the allocations to be positive. The initial demand is set by Equation 6.10.

6.1.3 Complexity

The problem is NP-hard which can be proven by restriction to the bin packing problem [1]. The amount of binary variables arising from a realistic problem is also

unmanageable, the problem is unsolvable for all but trivial problems.

6.2 Simplified model

By simplifying the stage-gate model so that a project is only one gate, but still having one task per competence, the optimisation problem can be reduced. As there are no dependencies in this model all the tasks can be run in parallel, and the amount of binary values is greatly reduced. To further reduce the complexity the non convex efficiency function is reshaped to only include what the empirical method indicates to be the most crucial part, learning. The learning will be included as a penalty in the objective function, suppressing unnecessary moving of resources between projects. Non finished projects is included in the profit to give a result closer to what real steady state would give. To simplify even further the gain is independ on project, but it is still discounted over time.

6.2.1 Notations

This is the notation needed to simplify the model

Constants
g_i^k: gain of project i in time period k, discounted as in the previous model.
r_j: number of available resources of competence j.
N_p: Number of projects, N_r: Number of resources.
T_0: The first time step and T_f the last time step.
$di_{i,j}$: Initial demand of the project i of competence j.
c: The learning cost.
P: Profit value used to evaluate the work done during the allocation period.
M : big constant

Variables
$d_{i,j}^k$: Demand of the project i to the competence j at the time step k.
$a_{i,j}^k$: Allocated amount of resources of the competence j to project i at the time step k.
$p_{i,j}^k$: Amount of resources of competence j *newly* allocated to project i in time step k.
$\zeta_{i,j}^k$: Slack variable used while calculating $p_{i,j}^k$.
x_i^k: a binary variable describing the project i state $x_i^k = \begin{cases} 0 & \text{Project } i \text{ is finished in time step } k \\ 1 & \text{otherwise} \end{cases}$

6.2.2 Equations

The mixed integer model taking the simplifications into account is then as follows:

Let

$$z = \sum_{k=T_0}^{T_f-1} \sum_{i=1}^{N_p} g^k x_i^k + P \sum_{i=1}^{N_p} \sum_{j=1}^{Nr} g_{T_f}(1 - \frac{d_{i,j}^{T_f}}{di(i,j)}) - c \sum_{k=T_0}^{T_f} \sum_{i=1}^{N_p} \sum_{j=1}^{Nr} p_{i,j}^k$$

$$\begin{aligned}
&\min \ -z && && (6.12)\\
&\text{s.t.} \ \sum_j a_{ij}^k \leq r_j^k && \forall i \in I, k \in K && (6.13)\\
&\quad d_{ij}^{T_0} = di_{ij} && \forall i \in I, j \in J && (6.14)\\
&\quad d_{ij}^{k+1} = d_{ij}^k - a_{ij}^k && \forall i \in I, j \in J, k \in T_0 \dots T_f - 1 && (6.15)\\
&\quad \sum_{j=1}^{j=Nr} d_{i,j} \leq M(1 - x_i^k) && \forall i \in I && (6.16)\\
&\quad p_{i,j}^{T_0} = a_{i,j}^{T_0} + \zeta_{i,j}^{T_0} && \forall i \in I, j \in J && (6.17)\\
&\quad p_{i,j}^k = a_{i,j}^k - a_{i,j}^{k-1} + \zeta_{i,j}^k && \forall i \in I, j \in J, k \in T_0 + 1 \dots T_f && (6.18)\\
&\quad \sum_{k=T_0}^{k=T_f} x_i^k \leq 1 && \forall i \in I && (6.19)\\
&\quad a_{ij}^k, d_{ij}^k, p_{i,j}^k, \zeta_{i,j}^k \geq 0 && \forall i \in I, j \in J, k \in K && (6.20)\\
&\quad x_i^k \in \{0, 1\} && \forall i \in I, j \in J, k \in K && (6.21)\\
& && && (6.22)
\end{aligned}$$

Where equation 6.12 is the objective function to maximise which penalize moving resources around or adding unnecessary resources to projects. One project produces gain only once as forced by equation 6.19. Since the resources are already in the portfolio, no cost of using them is included. Equation 6.13 forces the solution so that no more than the available resources in the portfolio are allocated. The dynamics of the model, the update of the demands, is done by equation 6.15. Equation 6.16 updates the variable x when a project is done. The initial demands is set by equation 6.14. Equations 6.17 and 6.18 is to take learning into account, the variable $p_{i,j}^k$ penalizes the objective function with a unit cost c when learning is present on a resource. When the allocation on a task is higher than the previous week, a new resource is added, and p is then how many new resources there is that week. If there is less resources than the previous week the slack variable ζ is populated instead due to equation 6.20, leaving no penalization on the objective function. Equation 6.20 forces some variables to be non negative and equation 6.21 is the boolean constraint.

6.2.3 Complexity

The problem is still NP-hard, however the model seems less complicated to solve, and it will give an idea of the behaviour of optimal allocations. It is still expensive

and will prove solvable only for portfolios with low number of projects, competences and weeks.

Empirical model

Since the full problem is unsolvable with realistic computer power another way to examine the problem is necessary, and for this a simulation framework was produced and implemented. The simulation framework enables testing of different resource allocation strategies to see how they influence the value of a certain portfolio of projects and resources. As the simulation time is short, a matter of seconds, the effects of changing nonlinearity functions and other model parameters can be investigated.

7.1 Features

Non-linear efficiency used in simulation framework:

- Knowledge learning.
- Knowledge loss.
- Project task resource ramping.
- Multiple resource per project task.
- Multiple project task per resource.

The corresponding efficiency contributions can be seen in Section 4.9.

7.2 Output

The main analysed output from a simulation is the Net Present Profit (NPP) for the portfolio, as discussed in Section 5.3.

7.3 Steady state

It is important to properly handle start and end time and its effect on NPP, see Section 5.2. When the simulation starts no project in the portfolio is started and they all start in the same stage, as the stages are very similar in the different projects this creates unrealistic bottlenecks. The chosen approach to remove these

bottlenecks is to run the simulation on the portfolio until the running projects are no longer in phase and thus have a realistic start for the NPP calculation, with some projects already started and running; this is called steady state. The amount of steps to run before calculating NPP is called pre-run. A project that is not finished always has a NPP, which is defined to be the portion of how much work that has been produced in respect to the initial demand and its expected gain at the extrapolated finish time from the current work per time step ratio, see Equation (5.4).

7.4 Portfolio execution strategies

To examine the behaviour of a full model in different scenarios different strategies were implemented into the simulation. A strategy is responsible to set allocation of resources to projects in a fixed time step, based on the current state of the portfolio. As one of the decision variables in portfolio execution is the number of projects run simultaneously this appears as an input variable in all strategies. Some strategies make a distinction between prioritized and running projects, which means that there can be projects running that is not normally being allocated but instead is allocated resources not used by the prioritized projects (these are called slack projects).

7.4.1 Equal sharing

The available resources is shared equally between the prioritized projects, regardless of their sizes and the stage that they are in. Resources not usable in the prioritized projects can be allocated to various slack projects.

7.4.2 Equal sharing, prioritise running

The equal share of the available resources is calculated. Then iterating through the projects in the priority order, the project receives the equal share if it is higher than the present allocation. This might make it so that some projects does not get their equal share, but the first projects in priority order will get a higher allocation.

7.4.3 Execute allocations

In order to reduce complexity in designing strategies they do not, in the phase of calculation desired allocations for all project, consider the individual resources but instead only look at the amount of work that has to be done. A function then exists to move the individual resources to fulfil these allocations. If the allocations for the resources are done badly it will have a huge impact on efficiency and will distort the results. To solve this problem a unified way to go from allocations of units of resources to individual resources is used. There are then multiple variants of this function to choose the resource to fill a allocation on a task.

Resource dividing allocation

When resources are taken arbitrarily from available resources in the portfolio with no sensible prioritization the resources will end up with many allocations. They will start to split into small allocations on many projects which will drastically reduce the overall efficiency of the resources.

No limits resource queueing allocation

The way resources are chosen to fill a required allocation is as following, take from free resources in the following order:

- Already assigned to the project.
- Last worked on project.
- Allocated to least number of projects.
- Low availability left.

If there are not enough available resources it take as much as possible. This will still split resources, but not as much as the Resource dividing allocation does.

Resource queueing allocation

Using the same queuing principle as no limits resource queueing allocation but adding a vital limit to the allocations. By setting limits on the minimum allocation from a resource to allocate, by taking all that is left from that resource the unnecessary splitting of resources is suppressed to a minimum.

7.5 Weaknesses

If the strategy makes good decisions on what a project should have as allocations they can easily be diminished by bad choices of resources for those allocations, which can distort the results. A resource might be split too much and working on many projects and thus become inefficient on all those projects. Or one might reallocate a resource from a project and later fill that spot with another resource that have no knowledge at the project, thus introducing unnecessary lead time.

Results

8.1 Analytical

In order to study the behaviour of an optimal solution a small linear portfolio execution problem was implemented using the GAMS (General Algebraic Modeling System) modelling language. To solve these models the computer solver SNOPT (see [10]) was used. Different problems were solved, all using the same portfolio; the initial problem was the simplest possible linear execution, and in turn the effect of learning and that of steady state were added. Description of the portfolio used can be found in Appendix A.1, most notable is that project 1 and 2 are both partially done at the beginning of the period. From the solution it is possible to extract a general trend in how projects are allocated, and what effect learning has.

In Figure 8.1 the results from solving the problem defined in Section 6.2, without learning or cost of resource usage, can be seen. The figure presents the allocation of resources per project, competence and week. The gray diagonal lines represents finished projects. Notably large numbers are used in order to avoid rounding errors that might otherwise influence the solution in the particular solver used. In this solution the main trend is to prioritize only one single project, and if all resources can not be used in that project transfer them to the next project to be started.

When a simplified version of learning is added, described by Equations 6.17 and 6.18, the allocations change and can now be seen in Figure 8.2. Learning is implemented as penalty cost, instead of a lowered efficiency, for implementation reasons. Because of the learning the results now differ, and instead prioritises two projects. Notable is that no allocations are made in the end of the period, as only finished projects are rewarded.

With steady state considered (see Section 5.2) the objective function changes further, and now gives gain for projects that are partially finished at the end of the run period. With this the solution, presented in Figure 8.3, contains allocations in projects that does not finish during the period.

8.2 Empirical

With the simulation framework described in Chapter 7 different strategies can be run with a set of different settings in order to give an understanding of the be-

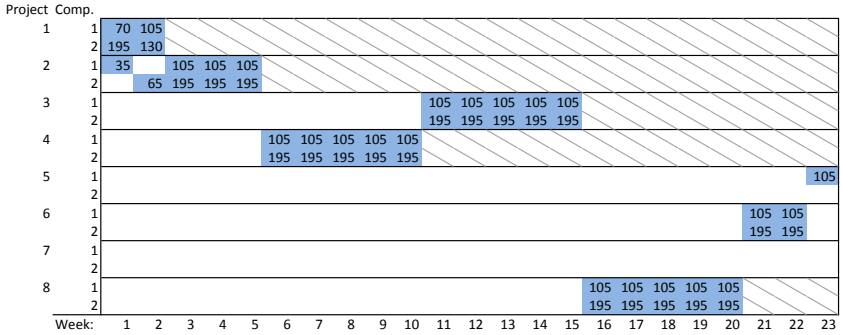

Figure 8.1. Optimal execution of a small portfolio, without learning or steady state handling

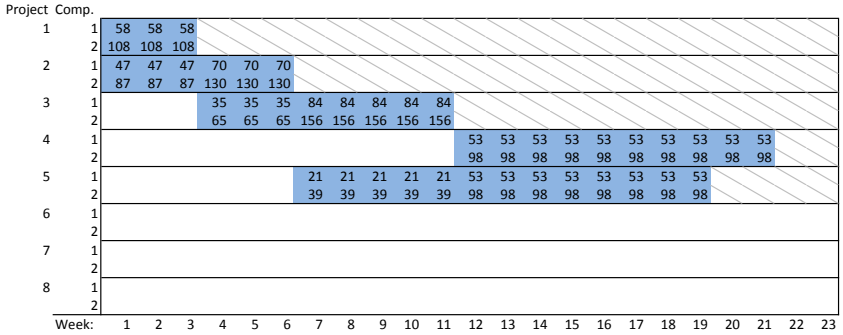

Figure 8.2. Optimal execution of a small portfolio, with learning but without steady state handling

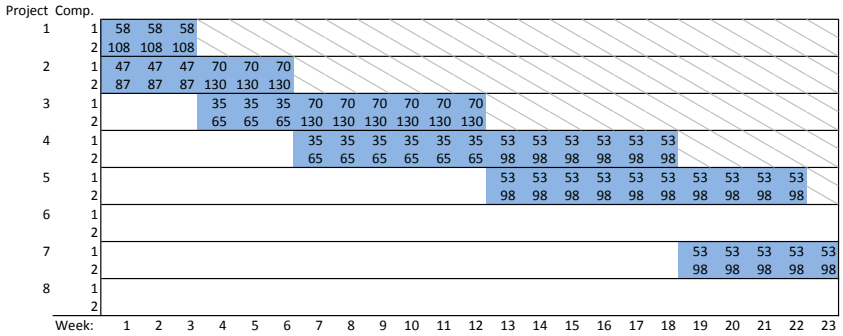

Figure 8.3. Optimal execution of a small portfolio, with learning and steady state handling.

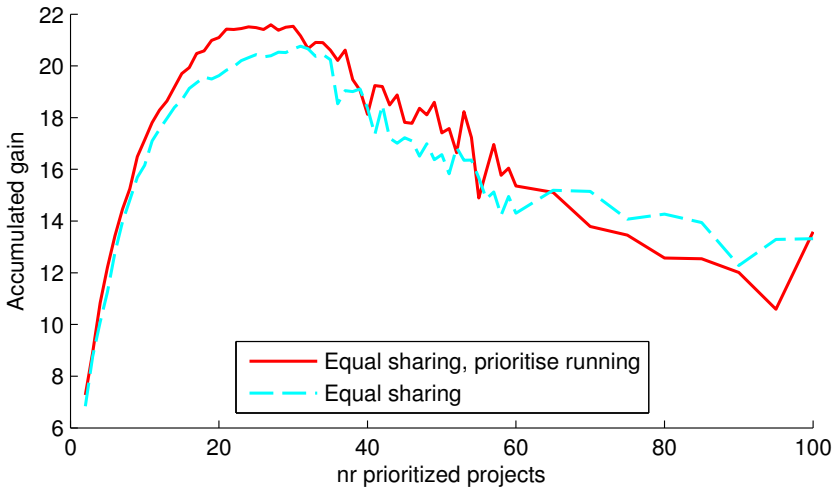

Figure 8.4. *Equal sharing* and *Equal sharing, prioritise running*

haviour of more complex portfolios, impossible to solve analytically. The main parameters varied in the different simulations with different strategies are the number of projects run by the R&D department. All simulations are run using a Monte Carlo method, they are run multiple times with randomly generated portfolios to generate approximate average values and confidence intervals.

8.2.1 Portfolio investigated

The portfolio used consists of 15 competences and 600 projects with an average of 1000 man week demands, 300 resources. Each project follows the stage gate model with six stages. The amount of each resource is proportional to the mean demand per competence. See Appendix A.1.

8.2.2 Portfolio execution strategies

The portfolio execution strategy *Equal sharing, prioritise running*, performs better than *Equal sharing* over the interesting simulation interval with the described portfolio, see Figure 8.4. Therefore the simulations was conducted using *Equal sharing, prioritise running*.

8.2.3 Steady-state effects

As described in Section 7.3 the steady state has an big impact on the results. The definition of steady state is clear, however finding a large-enough pre-run (see Section 7.3) to produce this is not trivial. An infinite pre-run is desired but not feasible. Since a long pre-run takes much computational time the shortest but sufficiently long pre-run is desired. To find the minimum pre-run required to produce

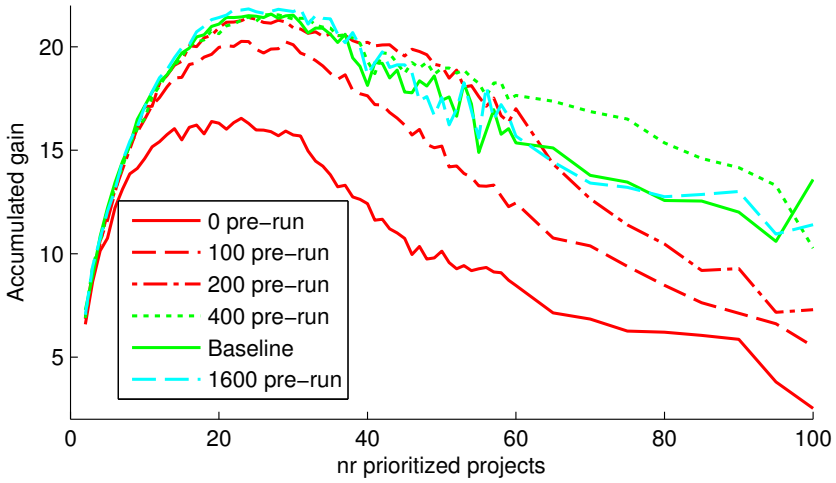

Figure 8.5. Test of pre-run where baseline is 800 pre-run

steady-state results the pre-run was gradually increased until the results where no longer changing. see Figure 8.5. What was found is that a pre-run of 800 produces almost the same as 1600, and is thus enough. The amount of pre-run needed is dependent on the maximum number of running projects simulated, as for example pre-runs of 200 and higher all are about the same up to 35 projects prioritised.

The breakdown of the gain, the contributions from already started and unfinished projects, can be seen in Figure 8.6 which shows that portfolio is in steady-state. The accumulated gain coming in to the steady-state, already started projects, is close to the one going out, coming from unfinished projects. The big difference when prioritising around 40 projects is due to some projects being started before the steady-state and not finished during the steady-state, since the portfolio does not finish as many projects.

8.2.4 Sensitivity to input portfolio

To get more accurate results the portfolios are generated with random distributions of demands and resources. To be able to study the behaviour of a generic portfolio a Monte Carlo method is used, multiple simulation runs with different randomization seeds are used. Based on their outcome the avarage gain for a certain strategy can be approximated, and the sensitivity to portfolio composition be examined. The results presented in this report are always an average over several randomized portfolios. Figure 8.7 shows the individual seeds' results and their average with a 95% confidence interval.

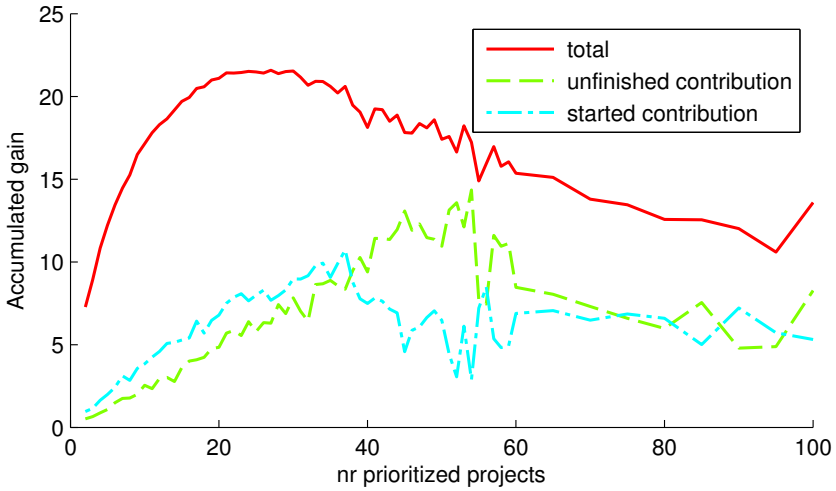

Figure 8.6. The gain from started and non-finished projects during the simulation

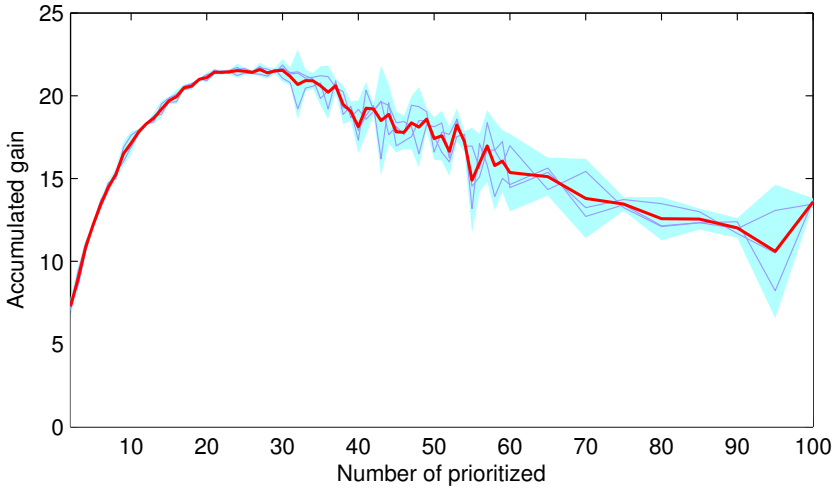

Figure 8.7. Confidence interval

Total work \ Resources	150	225	300	375	450
500	21	28	35	45	56
750	16	22	29	37	43
1000	15	20	26	31	38
1250	14	19	24	29	34
1500	13	18	23	26	32

Table 8.1. Optimal nr projects to prioritize, total work in man weeks

Comp	2	3	4	5	6	7	8	9	10	
Projects	21	36	35	32	31	30	28	28	28	
Comp	11	12	13	14	15	16	17	18	19	20
Projects	27	27	28	27	26	25	25	24	24	25

Table 8.2. Optimal nr projects to prioritize, changing the amount of competences with constant portfolio size

8.2.5 Size of portfolio

The size of the portfolio is greatly affecting the results. As the portfolio resources increases the number of projects to prioritize also increases. The opposite is true for when the work increases in the portfolio, which gives fewer prioritized projects. See Figure 8.8 and Table 8.1

8.2.6 Number of competences

The amount of competences in the portfolio, or competences per portfolio size ratio, changes how many projects one should run in parallel. With a low amount of competences many parallel projects should be run, the number of parallel projects to run decreases when the amount of competences increases. See Table 8.2.

8.2.7 Bottleneck

The optimal number of project to prioritize is decreasing when a bottleneck is present, see Figure 8.9.

8.2.8 Resource allocation

As mentioned in 7.4.3 the ways to go from desired allocation to individual resource allocation is not trivial. The way used for the produced graphs in this report is *Resource queueing allocation* since it handles the resource dividing problem the best, see Figure 8.10.

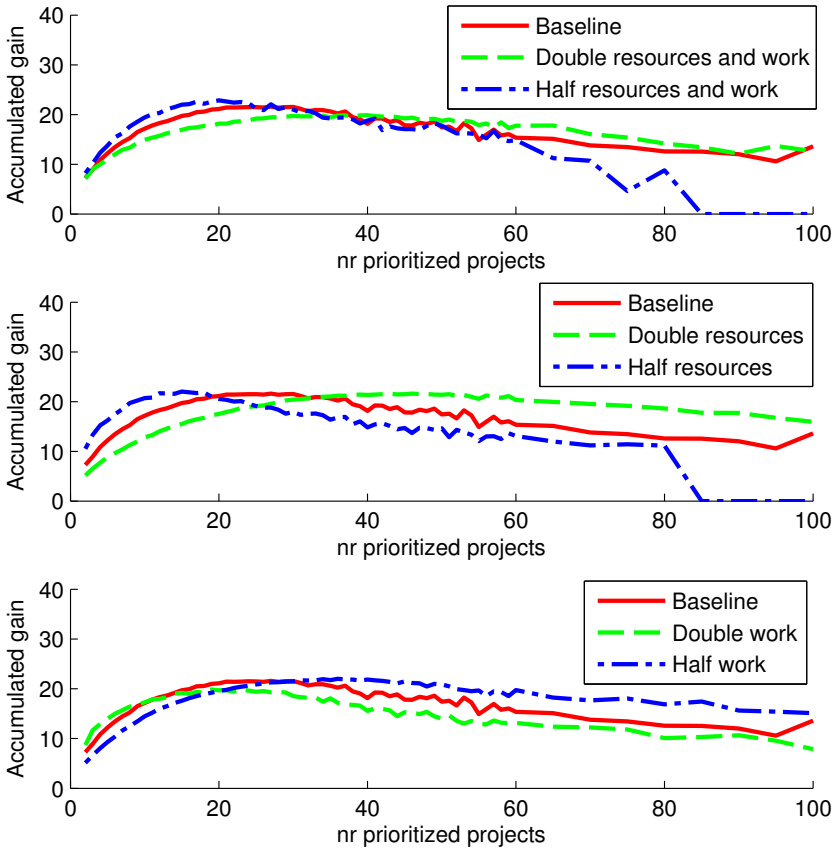

Figure 8.8. Different sizes of portfolios, results scaled

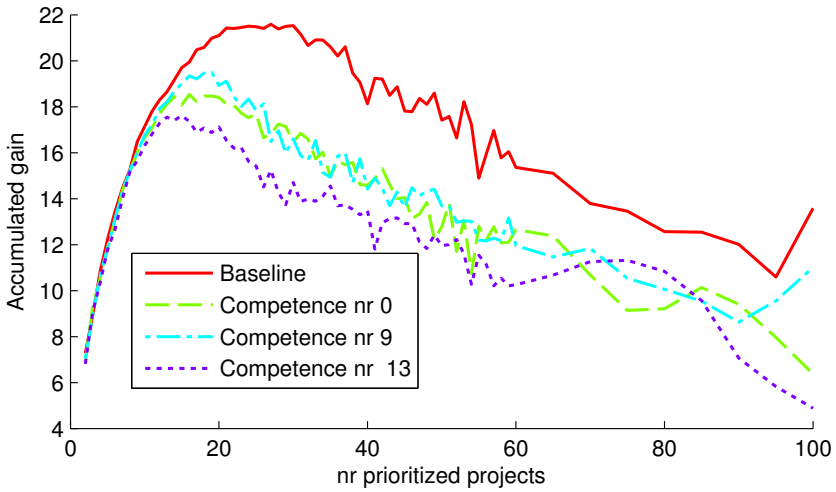

Figure 8.9. Bottleneck of 75% on different competences and baseline

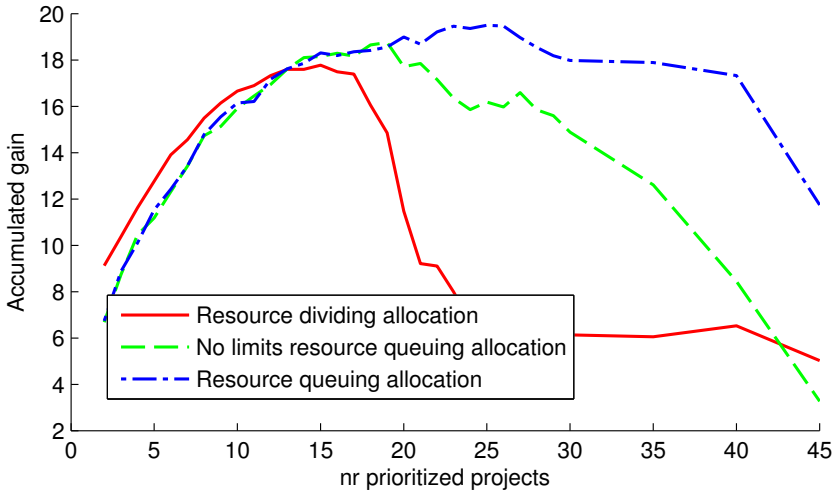

Figure 8.10. Different ways to allocate resources, using equal sharing strategy

Discussion

9.1 Analytical

The analytical solution of the simplified problem presented in Section 6.2 gives interesting information about the behaviour of the optimal solution. The results are interesting not only in their own but also as a motivation for the chosen way of representing the problem when simulating portfolio executions.

The general trend that can be seen is that a limited number of projects are run in parallel, with stepwise constant allocations only changing when other projects are finished. The number of prioritized projects are changed with the cost of switching, where a higher cost of switching leads to more projects running in parallel.

However the results from this part should be looked at knowing the limitations of the model that produced them. Several severe limitations have been made in the simplified model. The first one is the absence of the stage-gate model project description, it is instead replaced by a linear demand execution. This mainly takes away the desire to run several projects in different stages in order to reduce the bottlenecking several projects in the same stage produces. The linear demand execution also creates problem when bottlenecking occurs, as one demand can be finished without the others being started.

The other big simplification is in the efficiency functions, where only learning is considered, and only in a simplified form. As learning is introduced as a cost in the objective function instead of as an increase in the required demand for a project, the project execution time is shorter than it should, especially for short project execution times.

Viewed in the light of its shortcomings the model still gives results that are interesting to study, the general trend of allocating to a limited number of projects and reallocating only when other projects are finished are both interesting results. In a real-world environment with delays, unknown demands and different project stages the results become less usable.

Future study in this area would be to expand the simplified model to support multiple stages. With access to a solver on a more powerful computer that problem would be solvable in reasonable time, and give more definitive answers on how to run realistic projects optimally. Another area of the model is the treatment of learning, a function closer to the desired behaviour that allows for realistic solution times would increase the quality of the results.

9.2 Empirical

The results from the simulation of the portfolio execution strategies is in line with and strengthens the findings in the analytical analysis, that there is an number of running projects that is the optimal for profit, and that starting more projects after this point is counter-productive. That number is of high interest, however it is hard to find a rule of what it should be; the simulations indicate that it is strongly dependent on the size and composition of the portfolio, and it would also depend on the portfolio execution strategy used. The two portfolio execution strategies tested, *equal sharing* and *equal sharing, prioritise running*, are not enough to draw conclusions about where that point lies even for the tested portfolio, even when using realistic heuristics. The simulations does however produce an order of magnitude for this number and behaviour of the value.

As pointed out in Section 7.5 there is no certainty that the results from the simulator is representative of the portfolio execution strategies it is set to investigate, since the implementation of strategies is complex and small changes in the implementation can have big effects.

The superior results of *Equal sharing, prioritise running* over *Equal sharing* might indicate that prioritising to finish started tasks, not removing previous allocations, is the way to go, or it might be due to the fact that *Equal sharing* looses extra efficiency since more resources are moved around.

The efficiency functions used are not validated by extensive research, but rather deduced from the extensive knowledge accumulated in the companies *Atlas Copco* and *Level 21 Management*. The exact shape of these function might therefore be subject to debate, but as long as the shapes of the functions are agreed on the general trends of the solutions should remain the same.

Since there is no cost on using resources the portfolio execution strategies will use as much resources as possible and thus always have near full utilization. As there is no cost of being allocated to a project nor a penalty for going directly from one project to another, this should thus not impact the execution of the portfolio negatively.

The extensive results achieved in the empirical study indicates that this kind of simulation would be worth further investigation. As it simulates the portfolio execution with much higher level of detail than the mathematical models, it is possible to study more complex scenarios. The main area of improvement would be further research and testing of portfolio execution strategies, both to find more realistic representation of strategies currently used in companies to serve as a baseline and to formulate other heuristic strategies that brings the results closer to the real optima. Such strategies could also provide more information such as the impact on profits from maximizing utilization.

Conclusion

With both the analytical and the empirical studies of the resource allocation problem three main results can be extracted from the accumulated knowledge and results.

The first one concerns the problem formulation and the possibility for optimisation. That the allocation problem is NP hard had been proved before the work on this project began [1]; in this project it was further found that for a realistic modelling of execution of projects in a full R&D department the amount of variables become unmanageable. In conclusion the problem is too big to solve analytically and to complex to draw final conclusions about optimality using heuristics.

The second major result is that every model points to the existence of an optimal value for the number of project to run. However the overall complexity to allocate resources to projects indicates that the point will be hard to find, it is only clear that a too big or too small number of running projects results in a notably lower profit. The answer to how many project to run in parallel is unanswered but some of its depending variables have been found; it is highly dependent of the size of the portfolio and the ways to allocate the resources.

This leads to the third main result, that the way resources are allocated to projects are of great importance, in the empirical studies the allocation strategies impacted the results massively. Keeping track of past allocations and avoiding to split a resource unnecessarily had great impact. This seems to mostly be a problem for numbers of running projects, but also the optimal value is affected and pushed down by a bad allocation strategy.

Bibliography

[1] Senay Solaka & John-Paul B. Clarke & Ellis L. Johnsonc & Earl R. Barnes. Optimization of r&d project portfolios under endogenous uncertainty. *European Journal of Operational Research*, May 2010.

[2] Matt Basset. Assigning projects-to optimize the utilization of employees' time and expertise. *Computers and Chemical Engineering*, 2000.

[3] Robert G. Cooper. Stage-gate systems: A new tool for managing new products. *Business Horizon*, May-June 1990.

[4] N. Archer F. Ghasemzadeh and P. Iyogun. A zero-one model for project portfolio selection and scheduling. *The Journal of the Operational Research Society*, July 1999.

[5] Antonella Certa & Mario Enea & Giacomo Galante & Concetta Manuela La Fata. Multi-objective human resources allocation in r&d projects planning. *International Journal of Production Research*, July 2009.

[6] Dennis S. Kira & Martin I. Kusy & David H. Murray & Barbara J. Goranson. A specific decision support system (sdss) to develop an optimal project portfolio mix under uncertainty. *IEEE TRANSACTIONS ON ENGINEERING MANAGEMENT*, August 1990.

[7] Frederick S. Hillerman and Gerald J. Lieberman. *Introduction to Operations Research*. McGraw-Hill, New York, ninth international edition edition, 2010.

[8] Ulf Högman and Hans Johannesson. Applying stage-gate processes to technology development - experience from six hardware-orientated companies. *Journal of Engineering and Technology Management*, 2013.

[9] Elhabib Moustaid. Optimal project portfolio execution - computer implementation of models and simulation framework. Master Thesis Report.

[10] Walter Murray Philip E. Gill and Michael A. Saunders. Snopt: An sqp algorithm for large-scale constrained optimization. *Society for Industrial and Applied Mathematics*, 2005.

Appendix

A.1 Portfolio investigated in simulations

The portfolio consists of 600 projects, where the projects demand is a uniform distribution between 500 man weeks and 1500 man weeks, distributed into the stages and tasks as described in Table A.1. The resources is a total of 300 man weeks, that is 300 individual persons, distributed proportional to the mean demands in one project for each competence.

Competence \ Stage	0	1	2	3	4	5	
0	0.020	0.015	0.020	0.003	0.013	0.013	Management
1	0.015	0.025	0.036	0.018	0.015	0.005	
2	0	0.005	0.032	0.005	0.003	0	
3	0	0.005	0.032	0.005	0.003	0	
4	0	0.005	0.032	0.005	0.003	0	
5	0	0.005	0.032	0.005	0.003	0	Engineering
6	0	0.005	0.032	0.005	0.003	0	
7	0	0.005	0.032	0.005	0.003	0	
8	0	0.005	0.032	0.005	0.003	0	
9	0	0.002	0.020	0.008	0.038	0	
10	0	0.002	0.020	0.030	0.025	0	
11	0	0.002	0.020	0.030	0.025	0	
12	0.005	0.007	0.032	0.018	0.088	0.005	Bring to market
13	0.005	0.002	0.008	0.003	0.013	0.020	
14	0.005	0.010	0.020	0.009	0.018	0.008	

Table A.1. Project composition, tasks demand fractions of total project demand

A.2 Portfolio investigated using GAMS

The portfolio is 8 projects with 2 competences for a running period of 23 weeks. The Gain is discounted with a discount rate of $\frac{1}{1.05}$ per week. The start gain, in week 1, is 1000 units. To have a behaviour close to that in steady state, the projects

are in different phases of the demands, other than that they are similar. The initial demands are as following:

Demand	P 1	P 2	P 3	P 4	P 5	P 6	P 7	P 8
Competence 1	175	350	525	525	525	525	525	525
Competence 2	325	650	975	975	975	975	975	975

Table A.2. Table of demands

The amount of available resources are directly proportional to the demands, with a total available 300 man work weeks, consisting of 105 man work weeks for competence 1 and 195 for competence 2.

TRITA-MAT-E 2014:04
ISRN-KTH/MAT/E—14/04-SE

www.kth.se