THESIS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

Polyhedral and complexity studies in integer optimization, with applications to maintenance planning and location–routing problems

Adam Wojciechowski

CHALMERS | UNIVERSITY OF GOTHENBURG

Department of Mathematical Sciences, Chalmers University of Technology
Department of Mathematical Sciences, University of Gothenburg
SE-412 96 Göteborg, Sweden

Göteborg, 2012

Polyhedral and complexity studies in integer optimization, with applications to maintenance planning and location–routing problems
Adam Wojciechowski
ISBN: 978-91-7385-736-9

Doktorsavhandlingar vid Chalmers Tekniska Högskola
Ny serie nr 3417
ISSN 0346-718X

Department of Mathematical Sciences
Division of Mathematics
Chalmers University of Technology and University of Gothenburg
SE-412 96 Göteborg
Sweden
Telephone +46 (0)31 772 1000

Printed in Göteborg, Sweden 2012

Polyhedral and complexity studies in integer optimization, with applications to maintenance planning and location–routing problems

Adam Wojciechowski

Department of Mathematical Sciences, Chalmers University of Technology
Department of Mathematical Sciences, University of Gothenburg

Abstract

This thesis develops integer linear programming models for and studies the complexity of problems in the areas of maintenance optimization and location–routing. We study how well the polyhedra defined by the linear programming relaxation of the models approximate the convex hull of the integer feasible solutions. Four of the papers consider a series of maintenance decision problems whereas the fifth paper considers a location–routing problem.

In Paper I, we present the opportunistic replacement problem (ORP) which is to find a minimum cost replacement schedule for a multi-component system given a maximum replacement interval for each component. The maintenance cost consists of a fixed/set-up cost and component replacement costs. We show that the problem is *NP*-hard for time dependent costs, introduce an integer linear programming model for it and investigate the linear programming relaxation polyhedron. Numerical tests on random instances as well as instances from aircraft applications are performed.

The stochastic opportunistic replacement problem (SORP) extends the ORP to allow for uncertain component lives/maximum replacement intervals. In Paper II, a first step towards a stochastic programming model for the SORP is taken by allowing for non-identical lives for component individuals. This problem is shown to be *NP*-hard also for time independent costs. A new integer linear programming model for this problem is introduced which reduces the computational time substantially compared to an earlier model.

In Paper III, we study the SORP and present a two-stage stochastic programming solution approach, which aims at — given the failure of one component — deciding on additional component replacements. We present a deterministic equivalent model and a decomposition method; both of which are based on the model developed in Paper II. Numerical tests on instances from the aviation and wind power industries and on two test instances show that the stochastic programming approach performs better than or equivalently good as simpler maintenance policies.

In Paper IV, we study the preventive maintenance scheduling problem with interval costs which again considers a multi-component system with set-up costs. As for the ORP, an optimal schedule for the entire horizon is sought for. Here, the maximum replacement intervals are replaced by a cost on the replacement intervals. The problem is shown to be a generalization of the ORP as well as of the dynamic joint replenishment problem from inventory theory. We present a model for the problem originally introduced for the joint replenishment problem. The model is utilized in three case studies from the railway, aircraft and wind power industries.

Finally, in Paper V we consider the Hamiltonian p-median problem which belongs to the class of location–routing problems. It consists of finding p disjoint minimum weight cycles which cover all vertices in a graph. We present several new and existing models and analyze these from a computational as well as a theoretical point of view. The conclusion is that three models are computationally superior, two of which are introduced in this paper.

The main contribution of this thesis is to develop models for maintenance decisions and thus take an important step towards efficient and reliable maintenance decision support systems.

Keywords: integer linear programming; complexity theory; polyhedral analysis; stochastic programming; maintenance optimization; Hamiltonian p-median problem

Appended papers:

Paper I: *The opportunistic replacement problem: theoretical analyses and numerical tests*, published online in Mathematical Methods of Operations Research DOI: 10.1007/s00186-012-0400-y (with Torgny Almgren, Niclas Andréasson, Michael Patriksson, Ann-Brith Strömberg, and Magnus Önnheim)

Paper II: *The stochastic opportunistic replacement problem, part I: models incorporating individual component lives*, published online in Annals of Operations Research DOI: 10.1007/s10479-012-1131-4 (with Michael Patriksson and Ann-Brith Strömberg)

Paper III: *The stochastic opportunistic replacement problem, part II: a two-stage solution approach*, published online in Annals of Operations Research DOI: 10.1007/s10479-012-1134-1 (with Michael Patriksson and Ann-Brith Strömberg)

Paper IV: *The preventive maintenance scheduling problem with interval costs*, Preprint (with Emil Gustavsson, Magnus Önnheim, Michael Patriksson and Ann-Brith Strömberg)

Paper V: *A comparison of several models for the Hamiltonian p-median problem*, submitted to Networks (with Stefan Gollowitzer, Luis Gouveia, Gilbert Laporte, and Dilson Lucas Pereira)

Publications not appended in the thesis:

A method for simulation based optimization using radial basis functions, Engineering and Optimization, 11 (2010), pp. 501–532 (with Stefan Jakobsson, Michael Patriksson and Johan Rudholm)

An opportunistic maintenance optimization model for shaft seals in feed-water pump systems in nuclear power plants, Proceedings of 2009 IEEE Bucharest PowerTech Conference, June 28th–July 2nd, Bucharest, Romania, pp. 2962–2969 (with Julia Nilsson, Michael Patriksson, Ann-Brith Strömberg, and Lina Bertling)

An optimization framework for opportunistic maintenance of offshore wind power system, Proceedings of 2009 IEEE Bucharest PowerTech Conference, June 28th–July 2nd, Bucharest, Romania, pp. 2970–2976 (with François Besnard, Michael Patriksson, Ann-Brith Strömberg, and Lina Bertling)

A stochastic model for opportunistic maintenance planning of offshore wind farms, Proceedings of 2011 IEEE Trondheim PowerTech Conference, June 19th–June 23rd, Trondheim, Norway, pp. 1–8 (with François Besnard, Michael Patriksson, Ann-Brith Strömberg, Lina Bertling, and Katarina Fischer)

New models for and numerical tests of the Hamiltonian p-median problem , Proceedings of 2011 International Network Optimization Conference, June 13–16, Hamburg, Germany, published in Lecture Notes in Computer Science Vol. 6701 (2011) pp. 385–394 (with Stefan Gollowtizer and Dilson Lucas Pereira)

Acknowledgements

I would like to start by expressing my gratitude towards my supervisor Michael Patriksson and co-supervisor Ann-Brith Strömberg for their generous help and guidance as well as the organization of a great research group. I thank my co-authors Emil Gustavsson and Magnus Önnheim for insightful discussions and a good cooperation. Thanks to all other past and present members of the optimization group (Peter, Karin, Mehdi, Mahmoud, Christoffer, Anna) as well as my co-workers at the Department of Mathematical Sciences (especially Fredrik, Ida and Ragnar) for creating a pleasant working environment.

I also want to thank my research partners at the Department of Energy and Environment at Chalmers Lina Bertling, François Besnard, Katharina Fischer and Julia Nilsson for a fruitful cooperation on maintenance problems in the energy sector. Thanks to Marcus Bohlin and Jan Ekman at SICS (Swedish Institute of Computer Science) for interesting discussions on maintenance and inspections which I am sure will result in future research publications. Great thanks to my coauthors on paper V, Stefan Gollowitzer, Luis Gouveia, Gilbert Laporte, and Dilson Lucas Pereira. A special thanks goes to Luis Gouveia for organizing an excellent Winter school in Estoril during which a first contact and exchange of ideas between the authors of paper V took place.

The research in this thesis has been financially supported by the Swedish Energy Agency (grants 2007-05475 and 2010-004592) and the Swedish Foundation for Strategic Research through the strategic centre GMMC (Gothenburg Mathematical Modelling Centre). This support is gratefully acknowledged.

Finally, I want to thank Emma Molldius and my parents, Elżbieta and Witold, for their encouragement and support during both the difficult and the enjoyable phases of this project. Special thanks goes to Emma for help with the layout of the cover illustration.

Adam Wojciechowski
Göteborg, August 2012

Contents

1 Introduction

The purpose of this thesis is to improve the understanding and solution efficiency of *optimization problems* in the areas of *maintenance optimization* and *location–routing*. Optimization problems can be described as problems where one searches for the best decision with respect to a given objective within the set of allowed, or *feasible*, decisions. The optimization problems that this thesis concern all belong to the family of *combinatorial* optimization problems, in which the feasible set consists of a finite set of objects such as routes in a graph or feasible schedules. Maintenance optimization is to find optimal decisions for replacement and repair actions in a given system during a given time-frame. An important aspect is to balance the cost of preventive maintenance actions with the probability of future component failures or system shutdowns. Location–routing combines two classical fields of combinatorial optimization: facility location and vehicle routing. The problem is to simultaneously decide where to locate facilities, or depots, and how to route vehicles to and from those depots so that all clients are visited, at minimum total cost.

When studying optimization problems, a natural first question to ask is whether the problem is *difficult* or *easy*, i.e., if efficient solution techniques exist or not. The theory designed for such a classification is *complexity theory*. A common approach for solving optimization problems is to introduce a mathematical description or model which belongs to a class of optimization models that can be solved by a general purpose algorithm. In this thesis, we consider the widely used class of *integer linear programming* (ILP) models (also denoted *integer linear optimization*). The models consist of linear constraints, a linear objective, and integrality requirements on the variables. The *linear programming* (LP) relaxation of an ILP model is obtained by removing the integrality requirement. The reason for using ILP models is, on one hand, that they provide a possibility to model a wide variety of problems and, on the other hand, that mature algorithms and a rich theory for these models exist. For a given optimization problem, several mathematical models can be formulated. The solution efficiency of these models greatly depends on how well the LP relaxation approximates the ILP model. Since the LP feasible region is a polyhedron, the relation between the ILP model and the LP relaxation is analyzed by the use of *polyhedral theory*. Finally, in many optimization problems decisions need to be taken without having access to all the data; the future is uncertain, and at best statistical data or probability distributions are available. An important concept developed for optimization under uncertainty is *stochastic programming*.

The following sections are intended to provide a short introduction to the theory used and the results obtained in the appended papers. In Section 2 we introduce basic concepts of and common solution techniques within complexity theory, integer linear programming and stochastic programming. Section 3 gives an introduction to the two areas of application considered, namely maintenance optimization and location–routing. Section 4 introduces the optimization problems considered in the appended papers and presents the most important results therein. Finally, Section 5 summarizes the contributions of the thesis.

2 Theory

This section contains short introductions to the areas of complexity theory, integer linear programming and stochastic programming.

2.1 Complexity

This section is intended to give an informal and short introduction to the topic of complexity theory necessary for understanding the proofs of complexity included in the appended papers. A comprehensive description of complexity theory is found in [14]. Complexity theory classifies a problem according to the number of elementary operations required to solve it by a *Turing machine*, which is a theoretical computer. In order to perform such a formal classification, one needs to define concepts such as an alphabet, language, the Turing machine, etc. We proceed along a more intuitive path that captures the crucial aspects of the theory without the necessity of technical definitions.

Complexity theory was initially developed for *decision problems*. A decision problem consists of an instance definition and a "yes or no" question. As an example, we present two decision versions of classical optimization problems.

Example 1 (The traveling salesman decision problem)**.** *Consider a set of vertices* $V = \{v_1, \ldots, v_n\}$*, distances* $d(v_i, v_j) \in \mathbb{N}$ *for each pair of vertices* $v_i, v_j \in V$*, and a length* $L \in \mathbb{N}$*. Does a cycle of length less than* L *which visits each vertex exactly once exist?*

Example 2 (The shortest path decision problem)**.** *Consider a set* $V = \{v_1, \ldots, v_n\}$ *of vertices, distances* $d(v_i, v_j) \in \mathbb{N}$ *for each pair of vertices* $v_i, v_j \in V$*, and a length* $L \in \mathbb{N}$*. Does a path between node* v_1 *and* v_n *of length less than* L *exist?*

As previously stated, complexity theory groups problems according to their difficulty. We begin by considering the class *P* of polynomially solvable decision problems. For all problems, it is natural to assume that if the instance size (for instance, the number of vertices in the above examples) is increased the computation time required for solving the considered problem will increase as well. A problem is solvable in polynomial time if an algorithm exists such that the solution time is bounded by a polynomial function of the instance size. We consider problems solvable in polynomial time to be "easy". Consider again the problems of Example 1 and 2. Dijkstra's algorithm (see [12]) is a classic polynomial algorithm for the shortest path problem; hence, the problem belongs to *P*. For the travelling salesman problem (TSP) no such algorithm is known to exist.

The second class of problems considered is the class of *non-deterministic polynomial* decision problems, *NP*. It is sometimes described as the class of all reasonable decision problems. It contains all problems for which suggestions or guesses of "yes"-answers are verifiable in polynomial time. The name non-deterministic polynomial stems from the definition of a non-deterministic algorithm consisting of two stages. The first stage contains a non-deterministic guessing which suggests "yes"-answers. The second stage attempts to verify these suggestions. If the guessing stage is prosperous, the problem can be solved in polynomial time. For Examples 1 and 2, a suggestion of a "yes"-answer corresponds to a suggested cycle and path, respectively. The polynomial time verification algorithm is simply to calculate the length of the suggested cycle or path and compare this to L. Note that a suggested "no"-answer does not have to be verifiable in polynomial time in order for a problem to belong to the class *NP*.

The third class considered is that of the *NP-complete* (*NPC*) problems. A problem A is *NPC* if, for every problem B belonging to *NP*, a polynomial reduction from B to A exists. A polynomial reduction is defined as follows. Given an instance of B, construct an instance of A such that the following two properties hold. First, the size of A must be bounded by a polynomial function of the size of B. Second, solving the constructed instance of A implies that a solution to B is obtained. In 1971 S.A. Cook showed that every problem in *NP* is polynomially reducible to the *satisfiability* (SAT) problem, which became the first problem shown to be *NPC*. Since then, proving that a problem A is *NPC* is done by showing that a problem B in *NPC* is polynomially reducible to A. Doing so, one avoids the necessity of constructing a transformation of all *NP* problems to A. Returning once more to Example 1, the TSP problem is shown to be *NPC* by a reduction from the *vertex cover* problem, which in turn is proven to be *NPC* by reduction of the *3SAT* problem (a restricted version of SAT), which finally is shown to be *NPC* by reduction from SAT. Note that a generalized *NPC* problem is *NPC*. However, a restriction of an *NPC* problem might not be *NPC*. For problems in *P*, the opposite is true.

Optimization problems do not belong to *NP*, since they are not decision problems. The TSP is an optimization problem.

Example 3 (The travelling salesman problem)**.** *Consider a set of vertices* $V = \{v_1, \dots, v_n\}$ *and distances* $d(v_i, v_j) \in \mathbb{Z}$ *for each pair of vertices* $v_i, v_j \in V$. *Find the minimum length cycle that visits each vertex in* V *exactly once.*

In order to classify the complexity of optimization problems, we introduce the class of *NP-hard* problems. Problems belonging to this class are no longer required to be decision problems, nor problems that require the verification of suggested solutions; instead the following must hold. A problem A is *NP*-hard if a polynomial reduction of a problem B in *NPC* to A exists. Obviously, by solving the TSP we solve the TSP decision problem. Hence, the TSP is *NP*-hard. Proving that a problem is *NP*-hard can also be done by reduction from another *NP*-hard problem.

If a polynomial algorithm for an *NPC* or an *NP*-hard problem was found, this would imply that *NP*=*P*. "Is *NP*=*P*?" is still today an open and important research question. The usual assumption is that *NP* $\neq$ *P*, which implies that no polynomial algorithms for *NPC* problems exist. The *NPC* and *NP*-hard problems are therefore considered to be "difficult". Proving that a problem is in *P* usually provides an efficient algorithm for the problem. The knowledge that a problem is *NPC* or *NP*-hard motivates further studies and the use of non-polynomial algorithms such as branch–and–bound (see Section 2.2). It should, however, be stated that *NPC* or *NP*-hard problems are generally not impossible to solve. For many of these problems,

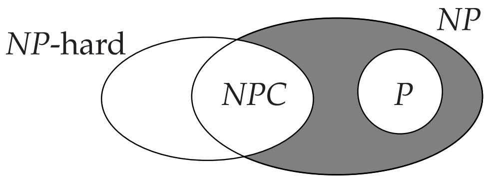


Figure 1: Illustration of the relations between the complexity classes provided that $P \neq NP$.

efficient methods for reasonably sized instances exist. Papers I, II, and V use complexity theory to show that the problems considered are *NP*-hard.

2.2 Integer linear programming

The purpose of this section is to give a short introduction to integer linear programming. For more comprehensive introductions, see [34, 25]. Integer linear programming (ILP), or integer linear optimization, is a class of optimization problems defined as follows. Given $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$ and $c \in \mathbb{R}^n$, we wish to

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & c^T x, && \text{(1a)} \\ \text{subject to} \quad & Ax \geq b, && \text{(1b)} \\ & x \geq 0, && \text{(1c)} \\ & x \in \mathbb{Z}^n. && \text{(1d)} \end{aligned}$$

This problem can be described as finding the best integral point within a polyhedron w.r.t. a linear objective function. The name stems from the fact that relaxing the integrality restrictions (1d) yields a linear program (LP), which is an optimization problem with affine constraints and a linear objective function.

Although the general ILP (1) is *NP*-hard, efficient general-purpose solvers exist for instances of moderate sizes (see Section 2.2.1). Further, ILPs are useful for modeling numerous optimization problems. We demonstrate this fact by presenting ILP models of the shortest path and traveling salesman problems (i.e., the optimization versions of Examples 1 and 2).

Example 4 (Shortest path problem)**.** *Let* $A = \{ (i,j) \mid i,j \in V, i \neq j \}$ *be the set of arcs in the complete graph with the set* V *of vertices. Introduce the binary variables* $x_{ij} = 1$ *if arc* $(i,j) \in A$ *is included in the solution and* $x_{ij} = 0$ *otherwise. An ILP model for the shortest* *path problem is to*

$$
\begin{aligned}
&\text{minimize} && \sum_{(i,j)\in A} d_{ij}x_{ij}, \\
&\text{subject to} && \sum_{i\in V\setminus\{j\}} (x_{ij} - x_{ji}) = \begin{cases} -1, & j = 1, \\ 0, & j = 2,\dots,n-1, \\ 1, & j = n, \end{cases} \\
& && x_{ij} \in \{0,1\},\ (i,j) \in A.
\end{aligned}
$$

Example 5 (Travelling salesman problem)**.** *Let $E = \{(i,j) \mid i,j \in V, i < j\}$ be the set of all edges (undirected) in the complete graph with the set of vertices V. For each proper subset $S \subset V$ let $\delta(S) = \{\, (i,j) \in E \mid i \in S, j \in V\setminus S$ or $j \in S, i \in V\setminus S \,\}$ be the cut set of S; let $\delta(i) := \delta(\{i\})$. An ILP model for the TSP is to*

$$\text{minimize} \sum_{e\in E} d_e x_e, \tag{2a}$$

$$\text{subject to} \sum_{e\in\delta(i)} x_e = 2, \qquad i \in V, \tag{2b}$$

$$\sum_{e\in\delta(S)} x_e \geq 2, \qquad S \subset V, \tag{2c}$$

$$x_e \in \{0,1\}, \quad e \in E. \tag{2d}$$

(2e)

Note that the number of subsets grows exponentially with the instance size of the TSP; hence, so does the size of the ILP.

In the papers included in this thesis, we use ILPs to model and solve problems within the areas of maintenance optimization and location–routing. In Section 2.2.1 we discuss solution approaches for ILPs, and in Section 2.2.2 we use polyhedral analysis to analyze the quality of the models.

2.2.1 Branch–and–bound

Consider the general ILP formulated in (1). By relaxing the integrality restrictions (1d) we obtain a linear program (LP). For LPs efficient solution algorithms, such as the simplex or interior point methods, exist (see [23] and [32], respectively). The optimal solution to the LP relaxation may, however, be non-integral. If this is the case, a common approach is to use the branch–and–bound method, briefly described as follows: Solve the LP relaxation. If the optimal solution of at least one of the variables, x_j^* say, is fractional, introduce two new problems, in which the constraints $x_j \geq \lceil x_j^* \rceil$ and $x_j \leq \lfloor x_j^* \rfloor$, respectively, are included. This is denoted as *branching* on the variable x_j. The two new problems correspond to two new nodes in the branch–and–bound tree. Solve the LPs corresponding to these two nodes. Prune a node if its optimal LP objective is worse than that of the best found integer feasible solution or if the LP is infeasible. The algorithm continues branching on variables in the leaves of the branch–and–bound tree until all leaves either have been pruned or posses optimal solutions of the corresponding LP which are integer. We demonstrate the method on a simple example.

Example 6. *We apply a pure branch–and–bound algorithm on the following problem*

$$\begin{aligned} \text{minimize} \quad & x_1 + x_2 + x_3 + 3x_4, \\ \text{subject to} \quad & 3x_1 + 2x_2 + x_3 + 2x_4 \geq 4, \\ & x_1, \dots, x_4 \in \{0, 1\}. \end{aligned}$$

The resulting branch–and–bound tree is illustrated in Figure 2 and the method is described as follows. Since the optimal LP solution obtained at the root node (node 1) is fractional in x_2, we create two new LPs, one in which $x_2 = 0$ (node 2) and one in which $x_2 = 1$ (node 3). Solve the LPs corresponding to nodes 2 and 3. In node 2 an integer optimal LP solution is obtained. In node 3, we obtain a fractional optimal LP solution with objective value lower than the best found feasible integer solution; hence, we branch again. We obtain two new LPs, one with an integer optimal solution (node 5) and one with a fractional optimal solution (node 4). Prune node 4 since its optimal LP objective value is higher than that of the best integer feasible solution found (node 2). We conclude that two optimal solutions, $(1, 0, 1, 0)$ *and* $(1, 1, 0, 0)$*, are found with the optimal objective value of* 2.

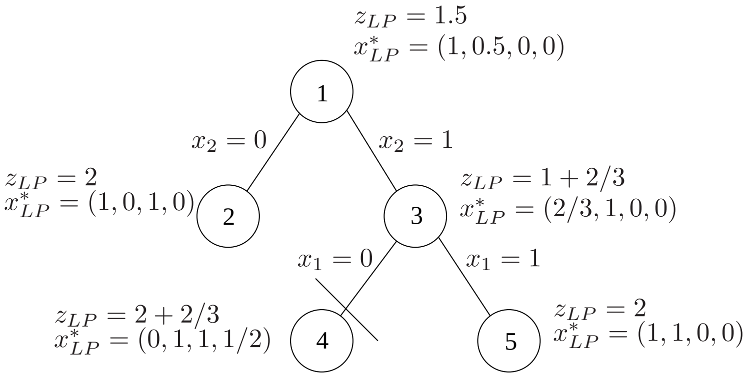


Figure 2: The branch–and–bound tree of Example 6.

The branch–and–bound method for general ILPs was introduced in [20]. The method has since then become increasingly popular for exact solution of combinatorial optimization problems. Several solvers that implement this method successfully exist, both commercial ones such as `Gurobi`, `CPLEX`, and `XPRESS`, and open-source solvers such as `SCIP`, `GLPK`, and `COIN-OR`. Any successful solver also includes additional aspects such as *presolve*, *branching rules*, *heuristics*, and *cutting planes*; see

[22]. Many solvers also allow for a the dynamic generation of constraints through a callback procedure, which enables the implementation of a branch–and–cut method (see Section 2.2.3).

Finally, it should be pointed out that branch–and–bound is a method which in the worst case reduces to a complete enumeration of all feasible solutions, and is thus non-polynomial. The success of the method relies to a large extent on how well the LP relaxation approximates the ILP. Note again that if the optimal solution of the LP relaxation is integral, it is an optimal ILP solution. For the shortest path ILP formulated in Example 4, this holds irrespective of the distances between the nodes. For the ILP model of the TSP formulated in Example 5, this is, however, not the case. Section 2.2.2 discusses, among other topics, the quality of the LP relaxation and sufficient conditions for optimal LP solutions to be integral.

2.2.2 Weak and strong formulations and polyhedral theory

Section 2.2.1 introduced the branch–and–bound method for ILPs based on the sequential solution of LPs. If the optimal value of the LP relaxation is close to the optimal ILP objective value, then it might be possible to prune many nodes at an early stage of the branch–and–bound process. In this section, we investigate conditions on the constraint matrix implying that at least one optimal solution to the LP relaxation is integral, as well as theoretical methods for assessing how well the LP relaxation approximates the ILP.

Recall that an ILP is defined as to find the best integer solution within a polyhedron. Different polyhedra can, however, contain identical sets of integer points (see Figure 3, which illustrates this fact in two dimensions). In general, let $P_1 := \{ x \in \mathbb{R}^n_+ \mid A^1 x \geq b^1 \}$ and $P_2 := \{ x \in \mathbb{R}^n_+ \mid A^2 x \geq b^2 \}$ be such that $P_1 \cap \mathbb{Z}^n = P_2 \cap \mathbb{Z}^n := X$. This implies that P_1 and P_2 together with the integrality restrictions yield the same set of integer feasible solutions X. Assume that $P_2 \subset P_1$. This implies that the LP relaxation of P_2 provides a stronger lower bound than the LP relaxation of P_1, that is, $\min_{x \in P_2} c^T x \geq \min_{x \in P_1} c^T x$; we say that the formulation P_2 is *stronger* than P_1. In some cases, neither $P_1 \subset P_2$ nor $P_1 \subset P_2$ holds. In such cases, $P_3 = \{ x \in \mathbb{R}^n_+ \mid A^1 x \geq b^1, A^2 x \geq b^2 \}$ provides a third formulation which is stronger than both P_1 and P_2. Further, we often obtain ILPs for the same problem which are formulated with different sets of variables. If a linear transformation between these formulations exists we can, however, still compare the strength of the LP relaxation. Papers II and V compare strengths of LP relaxations of different ILP models of the same problem.

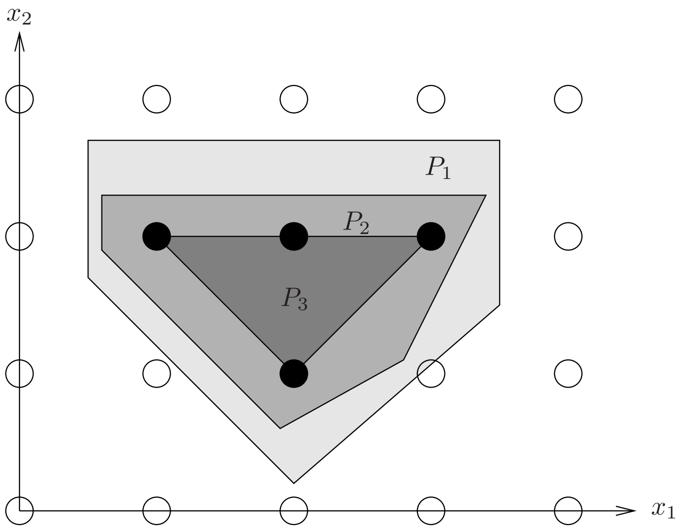


Figure 3: Three polyhedra containing the same integer points.

We proceed by investigating when the LP relaxation is the strongest possible. Consider again a polyhedron $P := \{ x \in \mathbb{R}^n_+ \mid Ax \geq b \}$ and the corresponding integer feasible set $X := P \cap \mathbb{Z}^n$. The smallest convex set which contains X is the convex hull of X and is denoted $\mathrm{conv}(X)$. In Figure 3 the polyhedron P_3 corresponds to $\mathrm{conv}(X)$. In general, if A has rational coefficients, then $\mathrm{conv}(X)$ is a polyhedron. The extreme points of this polyhedron are integral. If all extreme points of the polyhedron P are integral, then $\mathrm{conv}(X) = P$, and we say that the polyhedron P is *integral*. If this is the case, then the set of optimal solutions to the optimization problem $\min_{x \in P} c^T x$ contains the ILP optimal solution. Further, by using the simplex method to solve this LP relaxation we only obtain integer optimal solutions.

We characterize integral polyhedra by studying their constraint matrices. First, however, we need to introduce the concept of *total unimodularity*.

Definition 1 (Total unimodularity)**.** *A matrix A is totally unimodular (TU) if the determinant of each square submatrix of A is equal to -1, 0, or 1.*

We state the following propositions, the proofs of which can be found in [25, Sect. III.2].

Proposition 1. *The polyhedron $P(b) = \{ x \in \mathbb{R}^n_+ \mid Ax \geq b \}$ is either integral or empty for any $b \in \mathbb{Z}^n$ if and only if A is TU.*

Proposition 2. *The following statements are equivalent for a matrix $A \in \mathbb{R}^{m \times n}$ with matrix elements a_{ij}.*

i) The matrix A is TU.

ii) The matrix (A, I) *is TU.*

iii) The matrix A^T *is TU.*

iv) The matrix obtained by duplicating any row or column is TU.

v) The matrix obtained by multiplying any row or column of A *by* -1 *is TU.*

vi) For each subset of columns $J \subseteq \{1, \dots, n\}$*, a partition of* J *into* J_1 *and* J_2 *exists such that*

$$\left| \sum_{j \in J_1} a_{ij} - \sum_{j \in J_2} a_{ij} \right| \leq 1, \quad i \in \{1, \dots, m\}.$$

The constraint matrix of the shortest path model (Example 4) contains exactly one 1 and one -1 in each column. By statement iii) and vi) of Proposition 2 we conclude that the corresponding polyhedron has the integrality property. Papers I, II, and IV use the TU characterization to show that the integrality restrictions on certain variables in the models studied can be relaxed.

If we have access to an LP description of the convex hull polyhedron (i.e., $\operatorname{conv}(X)$), then the corresponding problem is solvable in polynomial time (unless the size of the convex hull LP grows exponentially with instance size). In many cases, however, we have access only to an ILP formulation which yields fractional solutions when relaxing the integrality restrictions. A natural question to ask is then whether the inequalities describing the ILP are necessary for the description of $\operatorname{conv}(X)$ or if these can be improved. In order to formalize this idea, we introduce some further polyhedral theory. We begin by considering a general polyhedron $Q = \{\, x \in \mathbb{R}^n \mid Ax \geq b \,\}$, in which a^i and b^i denote the i:th row of the matrix A and right-hand side b, respectively.

Definition 2 (Affinely independent points)**.** *The points* $x_1, \dots, x_k \in \mathbb{R}^n$ *are affinely independent if the vectors* $x_2 - x_1, \dots, x_k - x_1$ *are linearly independent.*

Definition 3 (Dimension)**.** *A polyhedron* Q *is of dimension* $\dim(Q) = k$ *if a maximum number of* $k + 1$ *affinely independent points belong to* Q*.*

Definition 4 (Proper inequality)**.** *The inequality* $a^i x \geq b^i$ *in the definition of* Q *is a proper inequality if there exists an* $x \in Q$ *such that* $a^i x > b^i$*.*

Definition 5 (Valid inequality)**.** *Given* $\pi \in \mathbb{R}^n$ *and* $\pi_0 \in \mathbb{R}$*, the inequality* $\pi^T x \geq \pi_0$ *is valid for the polyhedron* Q *if it holds for all* $x \in Q$*.*

Definition 6 (Face)**.** *If* (π, π_0) *defines a valid inequality for* Q*, then* $F = \{\, x \in Q \mid \pi^T x = \pi_0 \,\}$ *is called a face of* Q *and* (π, π_0) *represents the face* F*.*

Definition 7 (Facet)**.** *A face* F *of* Q *is called a facet if* $\dim(F) = \dim(Q) - 1$*.*

The following proposition states that the facet inducing inequalities are both necessary and sufficient for representing a polyhedron (see [25, Sect. I.4]).

Proposition 3. *Removing a proper inequality $a^T x \geq b$ which represents a face of dimension less than $dim(Q) - 1$ does not alter the polyhedron Q. For each facet F, a proper inequality representing the facet must be included in each description of the polyhedron.*

We now proceed to consider the polyhedron of interest and let $Q = \mathrm{conv}(X)$. In order to investigate the quality of an LP relaxation, we need to know whether the constraints defining P are facets of Q. We illustrate the principles behind the polyhedral analysis in Figure 4. The inequality constraints 2 and 3 do not represent facets of Q, whereas the constraint 1 does. Constraint 1 must, hence, be included in any description of Q. For a convex hull polyhedron of dimension k, proving that a constraint represents a facet is done by showing that k affinely independent integer feasible points satisfy the constraint with equality. Papers I, II, and IV use this technique to show that the inequality constraints of the ILP models studied therein define facets of the convex hull polyhedron.

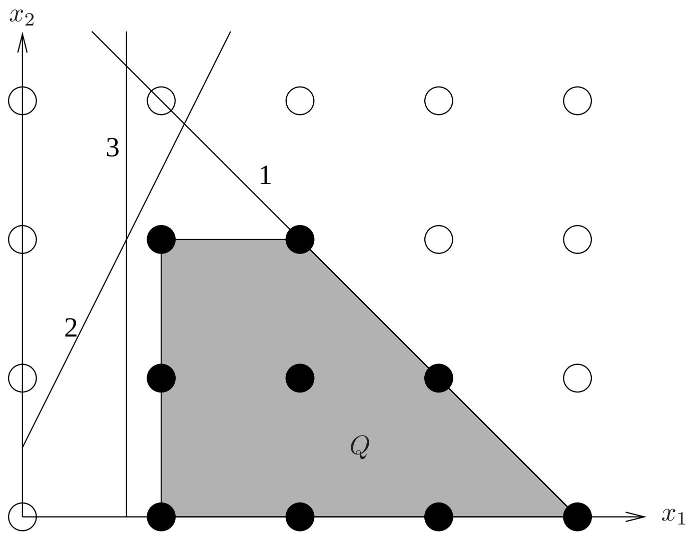


Figure 4: An ILP in two variables and three inequality constraints (in addition to the non-negativity constraints). The black circles represent the integer feasible solutions X and the grey region represents $Q = \mathrm{conv}(X)$.

Returning to the illustration in Figure 4, we see that constraint 3 can be improved by "pushing" it towards the convex hull. The inequality then obtained represents a facet of Q. This corresponds to increasing the right-hand side of the corresponding inequality. In order to obtain the facet-inducing inequality $x_2 \leq 2$ it is, however, not enough to alter the right-hand side of any of the defining inequalities. A general method for generating valid inequalities is Chvátal–Gomory rounding.

Definition 8 (Chvátal-Gomory cut)**.** *Consider an ILP* (1) *with the constraint matrix*

$A \in \mathbb{R}^{m \times n}$ and right hand side vector $b \in \mathbb{R}^m$. Given $u \in \mathbb{R}^m_+$, the inequality

$$\lceil u^T A \rceil x \geq \lceil u^T b \rceil$$

is a Chvátal-Gomory (CG) cut.

The following arguments indicate that a CG cut yields a valid inequality: A linear combination of linear constraints is a valid linear constraint. Increasing the coefficients of the left-hand side is allowed in any valid inequality. Finally, rounding up to the nearest integer of the right-hand side is allowed since all variables and coefficients in the left hand side are integer.

A CG cut with $u_i \in \{0, 1/2\}$ for $i = 1, \ldots, m$ is called a *zero-half cut*. A finite number of repeated applications of zero-half cuts generates a representation of the convex hull polyhedron for bounded integer programs (see [15]). Not all zero-half cuts, however, provide facets of the convex hull. Adding all possible zero-half cuts to an ILP formulation is seldom practical. Instead, one can add these constraints in order to cut away fractional points in the process of a branch–and–bound algorithm (see Section 2.2.3). One can also try to isolate a subset of zero-half cuts which define facets of the convex hull or which are defined by a combinatorial structure that is of help when locating violated constraints. Zero-half cuts are employed in Paper I in order to generate new facets for the ILP model considered.

2.2.3 Branch–and–cut

Consider the ILP model (2) of the TSP in Example 5. The cut-set inequalities (2c) consist of one constraint for each subset of vertices. The number of such constraints grows exponentially with the number of vertices of the graph. The branch–and–bound technique, described in Section 2.2.1, starts with the solution of the LP relaxation. In this case, however, the number of constraints would make the LP problem intractable for moderately sized graphs. Instead, we solve an LP relaxation consisting of the assignment constraints (2b) and bounds $0 \leq x_e \leq 1$ for all $e \in E$. If the optimal LP relaxation obtained does not satisfy a certain cut set inequality (2c), we add the corresponding constraint and resolve. If a solution is fractional, we branch. This is repeated in every node of the branch–and–bound tree and the procedure is called *branch–and–cut*.

A vital step in the branch–and–cut procedure is to find violated inequalities. Consider a (possibly fractional) solution $x^* \in \mathbb{R}^n$ and a set of constraints $a^i x \geq b^i$ for $i \in C$. The *separation problem* is to find a subset $D \subseteq C$ such that $a^i x < b^i$ for $i \in D$, that is, the subset of constraints which are violated. Often, one searches for the most violated constraint. Since the number of such constraints can be large, searching through all the constraints is not an efficient method. If a combinatorial structure exists, one might construct a special-purpose algorithm for the separation problem. For the cut-set inequalities (2c), the separation problem is to find the minimum cut between node 1 and a node $i \in V \backslash \{1\}$. This is equivalent to solving $|V|-1$ maximum flow problems, for which efficient polynomial algorithms exist (for instance, the Edmonds–Karp algorithm introduced in [13]).

Two types of constraints can be added in a branch–and–cut procedure. The first type of constraints, like the cut-set inequalities (2c), remove integer solutions which are feasible with respect to the constraints already included. This implies that an exact separation of integer LP relaxations must be performed before accepting the solution as feasible. The second type of constraints are included solely in order to strengthen a formulation. Such constraints can, for instance, be CG cuts (see Section 2.2.2). These constraints are always satisfied by integer feasible solutions; hence only fractional solutions require separation. The separation of fractional solutions (for both types of constraints) can be made by heuristic methods, although this often leads to weaker formulations.

Paper I uses a branch–and–cut approach for a class of facets obtained by a zero-half cut procedure in order to strengthen the LP formulation. Paper V uses a branch–and–cut approach for including constraints similar to the cut-set inequalities (2c) of the TSP model.

2.2.4 Other solution apporaches

Many solution approaches for ILPs exist; in this section we review two classical methods which are of importance for the appended papers and belong to the class of decomposition methods. *Benders decomposition* (introduced in [4]) is a decomposition approach which is useful when the variables of the problem can be grouped into master and subproblem variables with the following properties. Fixing the master variable values implies that the problem of finding the optimal subproblem variable values is done by solving a series of simpler problems for which efficient solution techniques exist. We use the subproblem solution to include new constraints into a master problem, which is re-optimized in each iteration. An important condition for the classical Benders decomposition to work is that the subproblems are LPs. The second method we consider is *Dantzig–Wolfe* decomposition (introduced in [9]), which replaces complicated constraints by many variables (one variable for each extreme point of the original polyhedron or integer solution). Since the number of variables grows exponentially with instance size, the simplex method is modified such that variables are dynamically generated when needed by a process called *column generation*. If fractional solutions arise, we branch and continue generating new columns in each node of the branch–and–bound tree. This procedure is called *branch–and–price* (see [3]). The integer L-shaped method is a version of the L-shaped method (which equals Benders decomposition in a stochastic programming context, see Section 2.3) for the case of binary subproblems. In Paper III, an extension of the *integer L-shaped* method is presented and successfully applied. In Paper V a formulation based on Dantzig–Wolfe decomposition is considered from a theoretical point of view, although not implemented computationally.

2.3 Stochastic programming

Stochastic programming (SP), also denoted *optimization under uncertainty*, is concerned with decision-making when data is random with a known probability distribution.

We give a short overview of the field with the intention to introduce some key concepts used mainly in Paper III. For a more comprehensive introduction to SP, see for instance [5] or [19].

Consider first a standard linear programming (LP) problem:

$$\begin{aligned} \text{minimize} \quad & c^T x, \\ \text{subject to} \quad & Ax = b, \\ & x \geq 0. \end{aligned}$$

The data consists of the vectors c and b, and the matrix A. If the data is deterministic we obtain the optimal solution by solving the LP. However, if our model includes phenomena such as future outcomes of financial markets, weather forecasts, or the uncertain future demand and price of a product, we may at best obtain probability distributions for the data. A simple approach is then to use the expected values of c, b, and A, and solve the corresponding deterministic LP. The resulting solution is denoted the *expected value* solution. Using this solution often leads to suboptimal decisions since only the expected value of the data is taken under consideration and not the whole probability distribution: the solution might, for instance, perform very poorly for certain outcomes.

When creating a model which takes uncertainty into account, it is common to distinguish between two types of decisions. The *first-stage* decisions are those taken before the realization of the uncertainty (such as to decide the production plan for a product before knowing its demand and/or price). The second stage decisions are taken once the uncertainty has been realized (such as to decide to which customers the product should be offered given the demand and price realized). Let x and y denote the first and second stage variables, respectively, ω a possible realization, or *scenario*, of the uncertain parameters, and Ω the probability space of all possible realizations ω. A standard *two-stage* stochastic linear program is formulated as that to

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & c^T x + \mathbb{E}_{\omega \in \Omega}[Q(x, \omega)], && \text{(3a)} \\ \text{subject to} \quad & Ax = b, && \text{(3b)} \\ & x \geq 0, && \text{(3c)} \end{aligned}$$

where

$$\begin{aligned} Q(x, \omega) = \quad \underset{y}{\text{minimum}} \quad & q(\omega)^T y, && \text{(3d)} \\ \text{subject to} \quad & W(\omega) y = h(\omega) - T(\omega) x, && \text{(3e)} \end{aligned}$$

and $q(\omega)$, $W(\omega)$, $h(\omega)$ and $T(\omega)$ denote the stochastic parameters of the problem realized at ω. The function $\mathcal{Q} : \mathbb{R}^n \to \mathbb{R}$ defined by $\mathcal{Q}(x) = \mathbb{E}_{\omega \in \Omega}[Q(x, \omega)]$, is called the *recourse* function. For a scenario $\omega \in \Omega$ and first stage decision x, evaluating the function $Q(x, \omega)$ corresponds to solving the subproblem (3d)–(3e).

A common assumption in stochastic programs is that the probability space Ω is finite. For a continuous, or finite but large, probability space an approximation of the same is obtained by sampling a finite number of scenarios. This is denoted the *sample average approximation* methodology. For finite probability spaces we can formulate a (large) LP, denoted the *deterministic equivalent*, the solution of which is equivalent to the solution to the stochastic program. Let $p(\omega)$ denote the probability of the scenario $\omega \in \Omega$. The deterministic equivalent of program (3) is to

$$\begin{aligned} \underset{x;y(\omega),\omega\in\Omega}{\text{minimize}} \quad & c^T x + \sum_{\omega\in\Omega} p(\omega) q(\omega)^T y(\omega), \\ \text{subject to} \quad & Ax = b, \\ & W(\omega)y(\omega) + T(\omega)x = h(\omega), \ \omega \in \Omega, \\ & x \geq 0. \end{aligned}$$

An equivalent formulation of this problem is to

$$\underset{x(\omega),y(\omega),\omega\in\Omega}{\text{minimize}} \quad \sum_{\omega\in\Omega} p(\omega)(c^T x(\omega) + q(\omega)^T y(\omega)), \tag{4a}$$

$$\text{subject to} \quad Ax(\omega) = b(\omega), \quad \omega \in \Omega, \tag{4b}$$

$$W(\omega)y(\omega) + T(\omega)x = h(\omega), \quad \omega \in \Omega, \tag{4c}$$

$$x(\omega_1) = x(\omega_2), \ \omega_1, \omega_2 \in \Omega, \tag{4d}$$

$$x(\omega) \geq 0, \qquad \omega \in \Omega. \tag{4e}$$

The constraints (4d) are denoted the *non-anticipativity* constraints. If these are relaxed, then the problem separates into solving one subproblem for each scenario $\omega \in \Omega$. This means that we would anticipate the realization of the stochastic process before determining the optimal value of the first stage variables or, equivalently, that we have information about the realization of the stochastic process before-hand. The solution obtained from solving the program (4a)–(4c), (4e) is denoted the expected *perfect information* solution.

In order to solve instances of stochastic programs for which the deterministic equivalent becomes too large to solve, decomposition techniques have been proven useful. The most common technique is called the *L-shaped* (see [31]) method which is an application of Benders decomposition (see Section 2.2.4). In this thesis, we however consider stochastic *integer* programs, in which integrality constraints are present for both the first and second stage variables. In such a setting the L-shaped method is not applicable since it is based on the assumption that strong duality is satisfied for the subproblems; this is in general not fulfilled for subproblems with integrality constraints. Decomposition techniques for integer stochastic programs exist (see for instance [29]), but they are neither as general, nor as efficient, as the L-shaped method for linear stochastic programs. In Paper III we present a modified version of the *integer L-shaped* method (see [21]) adapted to the stochastic opportunistic replacement problem considered.

In many applications, including the stochastic opportunistic replacement problem studied in this thesis, a sequence of decisions are to be taken. First, a first stage decision is taken (for instance, to buy or sell a stock). Secondly, the outcome of a stochastic process is realized (e.g., the price of the stock is altered) which is followed by a second stage decision (e.g., decide whether or not to sell or buy more of the stock). Again, a realization of a stochastic process occurs (the price is again altered) which is followed by a third stage decision (a new decision on buying or selling the stock). This pattern is repeated for several stages and the problem is then called a *multistage* problem. This type of problems are generally more difficult to solve than two-stage problems. The mathematical formulation of a multistage problem includes a recourse function in the objective of the subproblem (3d)–(3e). The recourse function is evaluated by solving one subproblem for each scenario, which again contains a recourse function, and so on. We can formulate a deterministic equivalent for multistage problems; the number of variables, however, tend to become very large. Multistage problems can be solved by using *nested decomposition* approaches, which means that the subproblems in the decomposition are themselves solved through a decomposition, in a recursive manner (see [5, Ch. 7] for more on multistage programs and nested decomposition). In Paper III we ignore the multistage structure of the problem and approximate it by a two-stage model.

3 Applications

3.1 Maintenance optimization

Maintenance can be described as a set of activities necessary to ensure that a system stays operational. Maintenance costs make up a large proportion of production costs; according to [28], on average 20% of the total plant operating budget consists of maintenance costs. Hence, improving maintenance decisions provides a possibility for substantial cost reductions. The purpose of this section is to give a short overview and introduce important terminology in the field of maintenance optimization.

Maintenance actions are often categorized into preventive maintenance (PM) and corrective maintenance (CM). PM consists mainly of scheduled maintenance tasks performed in order to prevent failure occurrences. CM, on the other hand, are maintenance actions taken after a failure has occurred in order to restore the system to an operational state. Maintenance which is performed due to condition measurements of the system is denoted condition-based maintenance (CBM).

Maintenance activities are also characterized by the state of the system or component after the activity has been performed. A perfect repair returns a component to a state which is equivalent to that of a new component. Perfect repair is therefore equivalent to replacement and also denoted “good as new”. An imperfect repair returns the component to a state which is worse than that of a new component. Minimal repair is a type of imperfect repair which returns the component to the state just before failure; it is also denoted “bad as old”.

Another classification of maintenance problems is with regard to the type of system under consideration. An important distinction is between single-component and multi-component systems. The vast majority of maintenance research until the 1990s was focused on single-component systems. Two classical policies for single component systems, the *age policy* and the *constant-interval policy* or periodic PM policy, were introduced already in 1960; see [2]. The age policy replaces the component after T time units or at failure. T is the parameter of the policy. The constant-interval policy replaces the component at fixed intervals kT for $k = 1, 2, \ldots$ and performs minimal repair in between. The length of the period T is the parameter of the policy. Research on single component systems is still an active field; see the survey [33]. It consists of extensions of the above policies which consider, among others, different types of both repair and failure. Further, policies based on cost and risk level have also been studied.

Most systems, however, consist of several components. If the dependency between these components is negligible, then we can apply single-component models for each of the components individually. If, on the other hand, dependencies between components exist, then optimal maintenance decisions must consider the system as a whole. Surveys of research on multi-component maintenance are presented in [26, 10]. Dependencies are categorized as either *economic*, *structural* or *stochastic*. Positive (negative) economic dependencies imply that maintenance on several components simultaneously is less (more) expensive than maintenance of the same set of components at different times. Structural dependencies imply, for instance, that maintenance of one component enforces the removal of another. Finally, stochastic dependencies arise when the failure of one component is correlated with the failure of another component.

A single-component policy which is easily applied to multi-component systems is the constant-interval policy. Since this policy provides a fixed schedule for each component, these can easily be coordinated in order to save costs. For systems with positive economic dependencies, a well-studied approach for the choice of interval lengths is by *standard indirect grouping* (see [10]) described as follows. The system stops for maintenance every T time units. Component i receives preventive maintenance every $k_i T$ time units. The parameters of the policy are $T \in \mathbb{R}_+$ and $k_i \in \mathbb{N}$.

Another important strategy for systems with positive economic dependencies is opportunistic maintenance (OM). The idea is that CM of one component is considered to be an opportunity for PM of other components. Using OM, maintenance activities on several components can be coordinated. One type of policy, which extends the age policy for single component systems to an OM policy for a multi-component system, is presented in [7]. The policy consists of one hard and one soft age limit for each component. If a failure occurs, or a component reaches its hard age limit, a maintenance stop is enforced. At a maintenance stop, all components which are either failed or have passed their soft lives are replaced. The soft and hard lives are the parameters of the policy.

Finally, different types of horizons are considered. The classical single component policies, among others, consider an infinite horizon approach which enables the use of analytical solution techniques. More recent research often considers finite or rolling horizons which enables the model to take short term information into consideration.

The Papers I–IV consider ILP and SP models for maintenance decisions of multi-component systems with positive economic dependencies. They all consider a finite horizon and perfect repair. The main difference between the problems considered in the papers is the approach to component failures. The use of these models is compared with a constant-interval policy and an age policy.

3.2 Location–routing

Location–routing is a group of problems which combine facility location with vehicle routing. We begin by giving a short description of these two areas.

Facility location is a classical topic in operations research; see [27] and [17]. A well studied problem in this area is the uncapacitated facility location problem (UFLP). It is formulated as follows. Consider a set of customers $I = \{1, \ldots, n\}$ and a set of possible depot locations $J = \{1, \ldots, m\}$. Let d_j denote the cost of opening a depot at site $j \in J$, and c_{ij} the cost of assigning customer $i \in I$ to depot site $j \in J$. Given that each customer must be assigned to one facility, the problem is to decide which facilities to open in order to minimize the sum of fixed costs and assignment costs. Another well-studied problem in the class of facility location problems is the p-median problem. The problem concerns the opening of exactly p facilities (with zero opening cost) among the m locations and assigning each customer in I to one facility, such that the sum of assignment costs between customer and facility is minimized. Both the p-median problem and the UFLP are NP-hard.

Vehicle routing is also an area that has attracted much attention; see [16]. The classical vehicle routing problem was first introduced in [8] and is defined as follows. Consider a graph $G = (V, E)$ with costs c_e on edges $e \in E$. A given vertex corresponds to the depot which contains the fleet of vehicles and the remaining vertices correspond to customers. The problem is to assign a route for each vehicle such that all customers are served by at least one vehicle and such that the transportation costs are minimized. At the end of a route, the vehicle must return to the depot. If the number of vehicles is one, then the problem reduces to the TSP. Many extensions of this problem have been considered, such as introducing time windows, more than one depot, load capacity of trucks, etc.

Facility location problems, in which customers are not individually connected with facilities but instead supplied by a fleet of vehicles, require that the location decisions are taken simultaneously with the routing decisions. The group of problems which consider this aspect are denoted location–routing problems (LRP); see [24] for a survey. Solving LRPs provides a computational and modeling challenge compared to solving the location and routing problems separately. Both exact algorithms and heuristics for LRPs have been studied. Exact algorithms are mainly based on integer linear programming formulations and solved using branch–and–bound or branch–and–cut methods. Heuristics are based on iterating between the location and routing problems using heuristic approaches, such as tabu search, for each subproblem.

Paper V concerns the Hamiltonian p-median problem, which is an LRP. Similarly to the p-median problem, it concerns the location of exactly p facilities of fixed cost zero. The objective is, however, to route exactly one vehicle from each depot so that each customer is visited exactly once. Equivalently stated, the problem is to partition the graph into p subsets and cover each subset by a Hamiltonian tour such that the sum of the tour lengths is minimized.

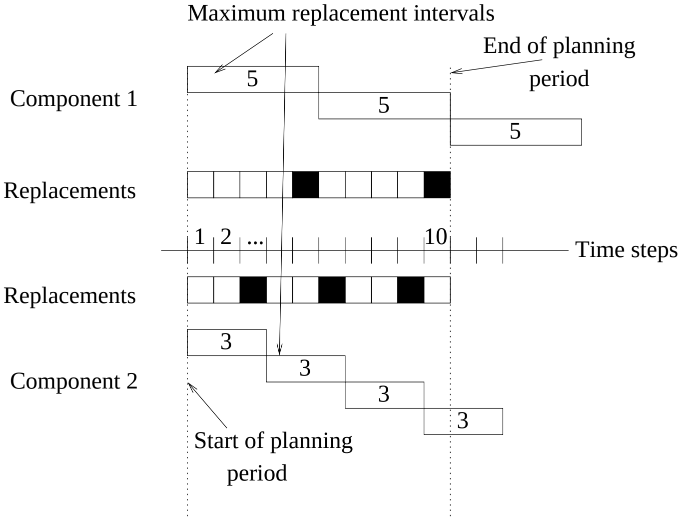


Figure 5: An illustration of the non-opportunistic maintenance schedule for the ORP instance defined in Example 7.

4 Summary of the appended papers

This section contains a summary of the appended papers. The emphasis is on presenting the problems considered in a less formal style than in the papers themselves, and on pointing out the most important contributions.

4.1 Paper I: The opportunistic replacement problem

In this paper we study the scheduling of component replacements for a multi-component system with positive economic dependencies. Each component is assumed to have an a priori given maximum replacement interval or life. The problem is denoted the *opportunistic replacement problem* (ORP), and was introduced in [11] and further studied in [1]. We begin by presenting a small instance of the ORP.

Example 7 (ORP)**.** *Consider a system consisting of two components. Assume that the replacement cost of component 1 is c_1, the replacement cost of component 2 is c_2, the maximum replacement interval of component 1 is 5 time steps, the maximum replacement interval of component 2 is 3 time steps, and that the maintenance occasion/set-up cost is d. We wish to find a minimum cost maintenance schedule over a time period defined by the time steps $1, \ldots, 10$.*

The non-opportunistic schedule is to replace each component at the end of its maximum replacement interval. An illustration of the non-opportunistic maintenance schedule for the system is shown in Figure 5. The resulting maintenance cost is $2c_1 + 3c_2 + 5d$.

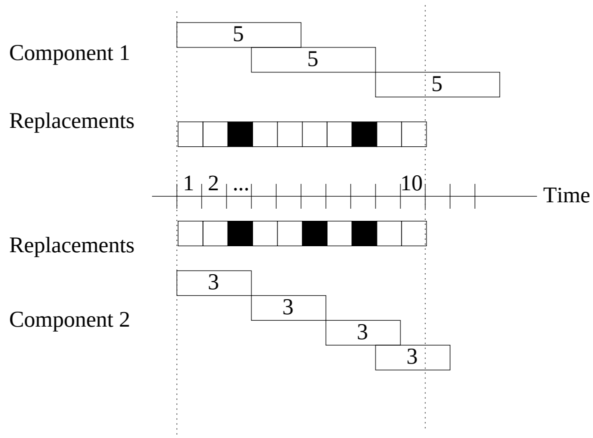


Figure 6: An illustration of the optimal maintenance schedule for the ORP instance defined in Example 7.

Figure 6 illustrates an optimal maintenance schedule for the system with the resulting maintenance cost $2c_1 + 3c_2 + 3d$.

We proceed by presenting a general problem definition, where we allow the replacement and maintenance occasion costs to depend on time. Given a set of components $\mathcal{N} = \{1, \ldots, n\}$ and a time period defined by the set $\mathcal{T} = \{1, \ldots, T\}$ the problem is formally defined as follows.

Definition 9 (opportunistic replacement problem). *Let* d_t *be a fixed cost for a maintenance occasion at time* $t \in \mathcal{T}$, c_{it} *be the cost for replacing a component* $i \in \mathcal{N}$ *at time* $t \in \mathcal{T}$, *and let* T_i *time steps be the maximum replacement interval of component* $i \in \mathcal{N}$. *Find a maintenance schedule over the time period defined by* $\mathcal{T}$ *that minimizes the total maintenance cost and such that each component* $i \in \mathcal{N}$ *is replaced at least once every* T_i *time steps.* □

A major contribution of the paper is the result that the *set covering* problem is polynomially reducible to the ORP, which implies that the ORP is *NP*-hard (see Section 2.1 for an introduction to complexity theory). The reduction relies on the fact that the costs are allowed to be time dependent. The complexity of the ORP with *time independent* costs is still unknown.

We present an ILP model for the ORP. Let

$$z_t = \begin{cases} 1, & \text{if maintenance shall occur at time } t, \\ 0, & \text{otherwise,} \end{cases} \qquad t \in \mathcal{T},$$

and

$$x_{it} = \begin{cases} 1, & \text{if component } i \text{ shall be replaced at time } t, \\ 0, & \text{otherwise,} \end{cases} \qquad i \in \mathcal{N}, \quad t \in \mathcal{T}.$$

The model is then to

$$\begin{aligned}
&\underset{(x,z)}{\text{minimize}} && \sum_{t\in\mathcal{T}} \Big(\sum_{i\in\mathcal{N}} c_{it}x_{it} + d_t z_t \Big), && && \text{(5a)}\\
&\text{subject to} && \sum_{t=\ell+1}^{\ell+T_i} x_{it} \geq 1, && \ell = 0,\dots,T-T_i, \quad i\in\mathcal{N}, && \text{(5b)}\\
& && x_{it} \leq z_t, && t\in\mathcal{T}, \quad i\in\mathcal{N}, && \text{(5c)}\\
& && x_{it} \geq 0, && t\in\mathcal{T}, \quad i\in\mathcal{N}, && \text{(5d)}\\
& && z_t \leq 1, && t\in\mathcal{T}, && \text{(5e)}\\
& && x_{it} \in \{0,1\}, && t\in\mathcal{T}, \quad i\in\mathcal{N}, && \text{(5f)}\\
& && z_t \in \{0,1\}, && t\in\mathcal{T}. && \text{(5g)}
\end{aligned}$$

As discussed in Section 2.2.2, the strength of the LP relaxation is important in order to be able to solve the model efficiently. In Paper I all non-superfluous constraints, out of the constraints (5b)–(5e), are shown to represent facets of the convex hull of the set of feasible solutions. Furthermore, it is shown that the integrality requirements on x_{it} for all $i \in \mathcal{N}$ and $t \in \mathcal{T}$ can be relaxed.

We also use zero-half cuts to produce a new set of facets for the convex hull polytope. Assume that $T_i \geq 2$ for all $i \in \mathcal{N}$ and let $p, q \in \mathcal{N}$ be such that $T_p \geq T_q + 1$. Let $\ell_p \in \{0,\dots,T-T_p\}$, $\ell_q \in \{\ell_p+1,\dots,\ell_p+T_p-T_q\}$, and define $\mathcal{T}^z = \{\ell_q, \ell_q+T_q\}$, $\mathcal{T}^p = \{\ell_p+1,\dots,\ell_p+T_p\}\setminus\mathcal{T}^z$, and $\mathcal{T}^q = \{\ell_q+1,\dots,\ell_q+T_q-1\}$. Then, the following inequality represents a facet of the convex hull polytope:

$$\sum_{t\in\mathcal{T}^p} x_{pt} + \sum_{t\in\mathcal{T}^q} x_{qt} + \sum_{t\in\mathcal{T}^z} z_t \geq 2. \tag{6}$$

The next example illustrates how the inequality (6) is constructed through a zero-half cut procedure.

Example 8. *Let $T_p = 6$, $T_q = 3$, $\ell_p = 2$ $\ell_q = 4$. Consider three of the inequalities* (5b) *corresponding to $i = p$ and $\ell = 2$, $i = q$ and $\ell = 3$, and $i = q$ and $\ell = 4$:*

$$\begin{array}{llllllll}
x_{p3} & +x_{p4} & +x_{p5} & +x_{p6} & +x_{p7} & +x_{p8} & \geq 1, \\
 & x_{q4} & +x_{q5} & +x_{q6} & & & \geq 1, \\
 & & x_{q5} & +x_{q6} & +x_{q7} & & \geq 1.
\end{array}$$

Multiply the above inequalities by $1/2$ and add them together. Use the inequalities (5c) *for $i \in \{p,q\}$ and $t \in \{4,7\}$. We obtain the valid inequality*

$$\frac{1}{2}x_{p3} + z_4 + \frac{1}{2}x_{p5} + x_{q5} + \frac{1}{2}x_{p6} + x_{q6} + z_7 + \frac{1}{2}x_{p8} \geq \frac{3}{2}.$$

Rounding up the coefficients on both sides of the above inequality yields a facet corresponding to the inequality

$$x_{p3} + z_4 + x_{p5} + x_{q5} + x_{p6} + x_{q6} + z_7 + x_{p8} \geq 2.$$

Finally, we introduce constraints which only allow maintenance at time t if, for some component $i \in \mathcal{N}$, a replacement took place T_i time units ago:

$$z_t \leq \sum_{i \in \mathcal{N}} x_{i,t-T_i}, \quad t \in \mathcal{T}. \tag{7}$$

The constraints (7) are denoted *elimination* constraints. For maintenance and replacement costs which decrease with time they eliminate a subset of the optimal solutions such that at least one optimal solution is left. For *strictly* decreasing costs, the constraints eliminate only suboptimal solutions. The purpose of the constraints is to improve the computational speed by reducing the size of the branch–and–bound tree.

The paper contains numerical tests on problem instances from the aircraft industry, as well as tests on random instances. The results for the aircraft case indicate that the use of the model (5) can reduce costs compared to those gained by simple maintenance policies. Tests on random instances indicate that adding the facets (6) in a branch–and–cut framework reduces computational time on instances with time dependent costs. Further, adding the elimination constraints (7) is favourable on instances with time independent costs and up to 10 components.

4.2 Paper II: The stochastic opportunistic replacement problem, part I: models incorporating individual lives

Paper II studies an extension of the ORP (studied in Paper I) which allows different maximum replacement intervals, or lives, for different individuals of the same component. This extended problem is called the *opportunistic replacement problem with individual lives* (ORPIL). The motivation for studying the ORPIL is that it poses a first step towards solving the stochastic ORP (the SORP), which extends the ORP to the case of uncertain component lives. Solving the SORP with perfect information (see Section 2.3) about individual component lives results in solving an instance of the ORPIL. Furthermore, the model for the ORPIL is the basis for a two-stage approach to the SORP presented in Paper III. We begin by presenting a small instance of the ORPIL.

Example 9 (ORPIL)**.** *Consider a system consisting of two components. The first failure of component* 1 *occurs at time* 3*, that is, the life of individual* 1 *of component* 1 *is* 3 *time steps. Individual* 1 *is replaced by individual* 2*, which fails after* 5 *time steps, and is then replaced by individual* 3*, which fails after* 4 *time steps. All individuals* r *with* $r \geq 3$ *have lives of* 4 *time steps. For component* 2*, individual* 1 *has a life of* 2 *time steps, individual* 2 *has a life of* 4 *time steps and individual* r *such that* $r \geq 3$ *has a life of* 3 *time steps. Let* c_1 *and* c_2 *denote the replacement cost of components* 1 *and* 2 *respectively, and let* d *denote the maintenance*

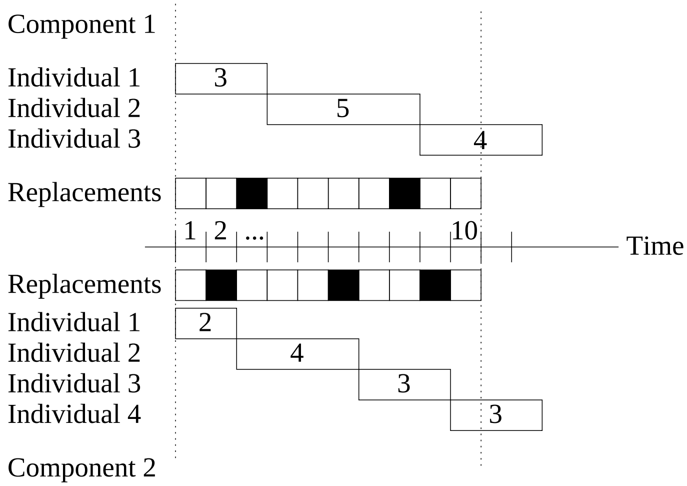


Figure 7: An illustration of a non-opportunistic maintenance schedule for the ORPIL defined in Example 9.

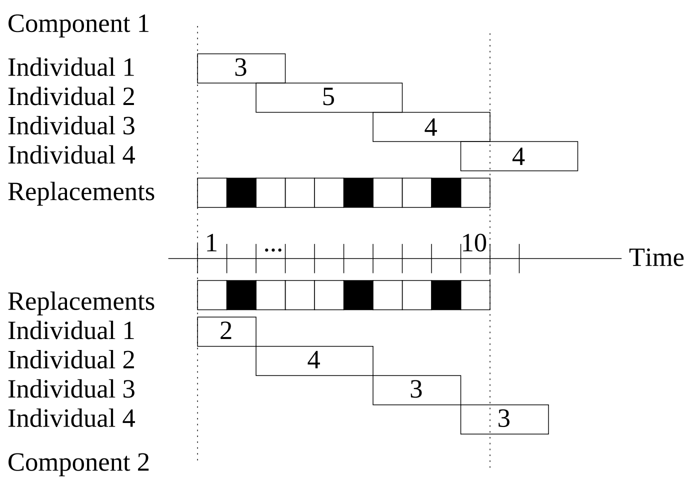


Figure 8: An illustration of an opportunistic maintenance schedule for the ORPIL defined in Example 9.

occasion cost. We wish to find a minimum cost maintenance schedule over the period defined by the time steps $0, \ldots, 10$.

The non-opportunistic maintenance schedule for the system is illustrated in Figure 7 and results in a maintenance cost of $2c_1+3c_2+5d$*. The most opportunistic maintenance schedule for the system is to replace both components at failure of one component. The schedule is illustrated in Figure 8 and results in a maintenance cost of* $3c_1 + 3c_2 + 3d$*. If* $2d \geq c_1$*, this opportunistic schedule is optimal. If* $2d < c_1$*, a schedule which is similar to the non-opportunistic schedule in Figure 7, but in which the first replacement of component* 1 *is made at time* 2 *and the second at time* 7*, is optimal.*

We now present the general definition of the ORPIL. In order to obtain problems that are computationally easier to solve, we only allow the first q individuals to have non-identical lives. Let again $\mathcal{N} = \{1, \ldots, n\}$ be the set of components and $\mathcal{T} = \{0, \ldots, T\}$ represent the time period[1]. To simplify the presentation, we consider a problem with time independent costs, i.e. $c_{it} = c_i$ and $d_t = d$ for all $i \in \mathcal{N}$ and $t \in \mathcal{T}$.

Definition 10 (opportunistic replacement problem with individual component lives (ORPIL)). *Let* d *be the fixed cost for a maintenance occasion,* c_i *the cost for replacing a component* $i \in \mathcal{N}$*,* T_{ir} *the life of individual* $r \in \mathbb{N}$ *of component* $i \in \mathcal{N}$*, and assume that* $T_{ir} = T_i$ *for* $r > q$*. Find a maintenance schedule over the time period defined by* $\mathcal{T}$ *that minimizes the maintenance cost, and such that each individual* $r \in \mathbb{N}$ *of component* $i \in \mathcal{N}$ *is used in the system no more than* T_{ir} *time steps.*

In Example 9, $q = 2$, $T_{11} = 3$, $T_{12} = 5$, and $T_1 = 4$; $T_{21} = 2$, $T_{22} = 4$, and $T_2 = 3$. Note that for $q = 0$ the ORPIL reduces to the ORP with time independent costs and for $q = T$ we obtain a problem in which all individuals may posses non-identical lives. The ORPIL for the cases $q = 1$ and $q = T$ was briefly studied in [1].

A major contribution of Paper II is the result that the ORPIL is NP-hard by reduction from the *vertex cover* problem. This problem reduction and the problem reduction performed in Paper I utilize different properties of the problems analyzed. In Paper II, we utilize the property that the lives of the first two individuals may differ from those of the remaining individuals. In Paper I, we utilize the property that the costs of the component replacement may be time dependent. Hence, the complexity of the ORPIL with $q \in \{0, 1\}$ can neither be determined by the analysis in Paper I nor that in Paper II.

We introduce two ILP models for the ORPIL — model I and model II — and show that in model I the integer requirements on most of the variables may be relaxed. We also show that in model I all the non-superfluous constraints define facets of the convex hull of feasible solutions. Furthermore, we demonstrate that relaxing the integer requirements in model II and in Andréasson's model (the model initially studied in [1]) results in fractional optimal solutions. By projecting the constraints from model I onto the variable space of model II we obtain model II$^+$ which obtains similar properties as model I. Numerical studies show that the solution times

[1] We include time 0, as the model is intended to be used in a stochastic setting to decide on additional replacement decisions at a time of failure, which is denoted "time 0".

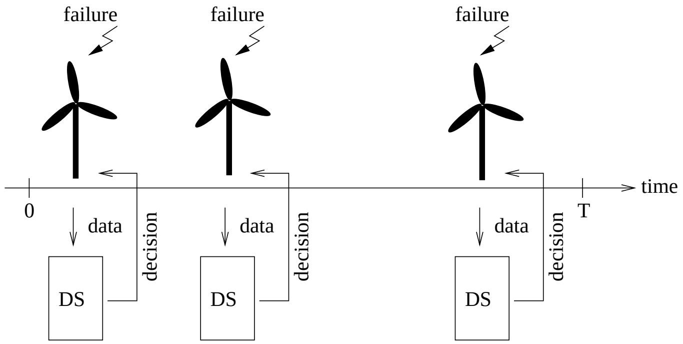


Figure 9: Illustration of a decision support system (DS) for a wind power turbine. At failure, the system data is sent to the DS, which solves the current problem and returns a decision on the type of maintenance action to perform at the current time.

of model I are significantly shorter than those of model II and Andréasson's model, as well as being slightly shorter than those of model II$^+$.

4.3 Paper III: The stochastic opportunistic replacement problem, part II: a two-stage solution approach

In this paper we consider the replacement decisions in a multi-component system similar to that in Paper I. However, we do not assume that maximum replacement intervals are a priori given. Instead, the maximum replacement interval, or life, of each component is uncertain and Weibull distributed. The problem is denoted the *stochastic opportunistic replacement problem* (SORP). As before, the replacement of a component $i \in \mathcal{N}$ generates the replacement cost c_i. We also assume that the maintenance occasion cost of a scheduled PM stop and an unexpected CM stop are both equal to d. This implies that restricting maintenance stops to times when at least one component has failed does not effect optimality.

The time of the maintenance stop is denoted the *current time*. At such a time, the replacement of the failed component is enforced. The stop is, however, an occasion to perform OM consisting of the replacement of the yet non-failed components in order to avoid costly maintenance stops in the near future. We wish to take a decision on which components to replace at the current time[2] s in order to minimize the expected cost over the remaining planing horizon, or contract period, $[s, S]$. This is denoted the *current problem*. The vision, illustrated in Figure 9, is to create a decision support system that, given the system state at a time of failure, returns an optimal (wrt. expected costs) maintenance decision by solving the current problem.

In ILP and SP models, it is common to discretize time. For models of the current

[2]The current time can be a period that stretches from a couple of hours to weeks, months, or years. It is a period after which the decision to perform maintenance or not can not be postponed.

problem, a time discretization δ is introduced such that failures and maintenance decisions are assumed to occur at the time points $\{ s, s+\delta, s+2\delta, \ldots, s+T\delta \}$, where $T = \lceil \frac{S-s}{\delta} \rceil$. These time points will be denoted by $\{0, 1, \ldots, T\}$. Let v denote the state of the system at the current time consisting of the age of each component and the remaining planning period. Let $\xi_i = 1$ if component $i \in \mathcal{N}$ has failed and $\xi_i = 0$ otherwise, and $x_i = 1$ if we decide to replace component $i \in \mathcal{N}$ at the current time and $x_i = 0$ otherwise. The current problem is formally defined as that to

$$\begin{aligned} &\underset{x}{\text{minimize}}\; c^T x + \mathcal{Q}_v(x), \\ &\text{subject to} \quad x_i \geq \xi_i, \qquad i \in \mathcal{N}, \\ &\qquad\qquad\quad\; x_i \in \{0, 1\}, \, i \in \mathcal{N}, \end{aligned}$$

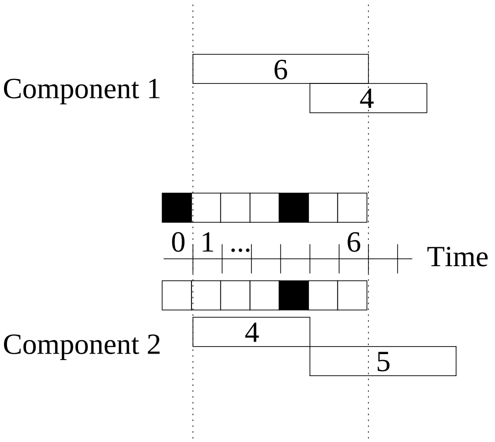
where $\mathcal{Q}_v : \mathbb{B}^{|\mathcal{N}|} \to \mathbb{R}$ is the recourse function (defined in Section 2.3) such that $\mathcal{Q}_v(x)$ is the minimal expected future maintenance cost over the remaining planning period given the current decision x.

The difficulty in solving the current problem stems from the fact that the recourse function is hard to evaluate. However, in Paper III we show that an evaluation of the recourse function given a replacement decision at the current time provides a lower bound on the recourse function of every other replacement decision. The bounds originate from the fact that by replacing less components we can not lower the recourse function value, and by replacing an additional component j we can at most lower the recourse function value by $c_j + d$.

To solve the current problem, we generate a large number of scenarios for the component lives. However, we can not solve each scenario individually; we must decide on one replacement decision at the current time. This decision is common for all the scenarios. The following example illustrates that the solution of ORPILs corresponding to two different scenarios yield different suggestions for the current decision.

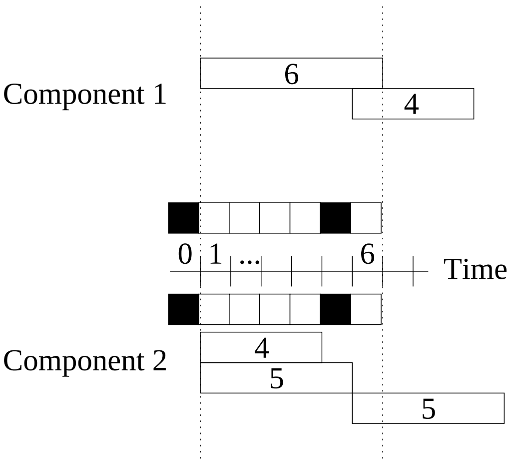
Example 10 (SORP)**.** *Consider an instance of a system with two components, in which component 1 has failed, and two following possible scenarios for the current problem. Using notation from Section 4.2 we can describe the scenarios by the life of each individual of every component. In scenario 1, we have $T_{11} = 6$, $T_1 = 4$, $T_{21} = 4$, and $T_2 = 5$. In scenario 2, we have $T_{11} = 3$, $T_1 = 4$, $T_{21} = 1$, and $T_2 = 5$. In both scenarios, $T = 6$. Figure 10 illustrates the optimal maintenance schedules for the two scenarios (obtained by solving the ORPIL). The optimal maintenance cost in scenarios 1 and 2 become $2c_1 + c_2 + d$ and $2c_1 + 2c_2 + d$, respectively. For scenario 1, the optimal current decision is to replace only the failed component 1, whereas in scenario 2 both components should be replaced.*

In order to impose a common current decision for the two scenarios, we formulate one ORPIL for each scenario and force the decisions at time 0 (i.e., the current time) to be equal (i.e., we impose non-anticipativity at time 0). The decisions at time 0 constitute the first-stage variables, and all other decisions constitute the second-stage variables.



(a) scenario 1 (b) scenario 2

Figure 10: Illustration of optimal replacement schedules for the ORPIL scenarios in Example 10.



(a) scenario 1 - replace both components at time 0 (b) scenario 2 - replace only component 1 at time 0

Figure 11: Illustration of the optimal second stage decisions for scenarios 1 and 2 when replacing both components and only component one, respectively.

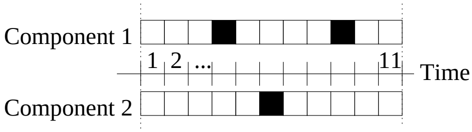
Example 11 (SORP cont.). *After imposing non-anticipativity at time* 0*, we can either replace only the failed component 1 (Figures 10(a) and 11(b)) or both components (Figures 11(a) and 10(b)) in both scenarios at the current time. Let* $p(\omega)$ *denote the probability of scenario* $\omega \in \{1, 2\}$*. The minimal expected cost in the two-stage model is* $p(1)(d + 2c_1 + c_2) + p(2)(2d + 2c_1 + 2c_2)$ *if we replace component 1 only, and* $d + 2c_1 + 2c_2$ *if we replace both components. The example illustrates the fact that by replacing component 2 at the current time, we can at most save* $d + c_2$ *in the future.*

In Paper III we include non-anticipativity only at time zero and therefore solve a two-stage approximation of the current problem. Including non-anticipativity at later times is complicated since the available information depends on earlier replacement decisions. We present a deterministic equivalent model and a decomposition method for the two-stage approximation of the current problem. The deterministic equivalent is based on the ORPIL model. The decomposition is based on the lower bounds of the recourse function and the subproblems are instances of the ORPIL.

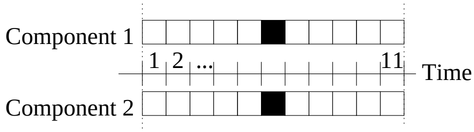
Numerical experiments on problem instances from the aviation and wind power industries, as well as on two smaller test instances, are performed. We compare the use of the two stage approach with the use of a simple age policy on a simulation of the system. Using the stochastic programming approach is favorable for two out of the four instances tested, and performs equivalently with the age policy on the remaining two instances compared. The experiments also show that the decomposition method requires a shorter solution time compared to solving the deterministic equivalent on all four instances considered. The decomposition method reduces the solution time by up to 80% compared to the deterministic equivalent, and the reduction is largest on instances which require long solution times.

4.4 Paper IV: The preventive maintenance scheduling problem with interval costs

In this paper, we consider the preventive maintenance scheduling of a multi-component system with positive economic dependencies. We wish to schedule a set of PM actions over a discrete planning horizon. Similarly as in Papers I–III, we assume a fixed/set-up cost d_t for performing PM on any component at time t. However, instead of enforcing PM at least once within a certain interval, we consider an interval cost c^i_{st} if PM is performed at times s and t, and not in-between. The *preventive maintenance scheduling problem with interval costs* (PMSPIC), introduced here, is to schedule the PM of all components such that the sum of set-up costs and interval costs is minimized. Note that the PMSPIC reduces to the ORP, introduced in Paper I, if we assign $c^i_{st} := c_{is}$ if $t - s \leq T_i$ and $c^i_{st} \gg d_u, c_{ju}$ for all $j \in \mathcal{N}$ and $u \in \mathcal{T}$ otherwise. The PMSPIC is therefore *NP*-hard. We perform case studies on instances with the following time independent cost structure. We assume that $d_t := d$. We introduce a deterioration cost function $M_i : \mathbb{R} \to \mathbb{R}$ and define $c^i_{st} := c^{\mathrm{PM}}_i + M_i(t - s)$ if $s \geq 1$ and $c^i_{0t} := M_i(t)$ otherwise. The following example illustrates an instance of the PMSPIC with time independent costs.



(a) $d = 0$



(b) $d = 2$

Figure 12: The optimal schedules for the instance of the PMSPIC introduced in Example 12

Example 12 (PMSPIC)**.** *Consider a system with $n = 2$ components over a horizon of $T = 11$ time steps. Let the replacement cost for the components be $c_1 = 20$ and $c_2 = 5$, and the set-up cost be $d = 0$. The deterioration cost functions for the system are $M_1(t) := (t/2)^2$ and $M_2(t) := (t/5)^5$. The optimal PM schedule for this problem is displayed in Figure 12(a). If we modify the set-up cost to $d = 2$, we instead obtain the maintenance schedule displayed in Figure 12(b).*

The PMSPIC resembles the standard indirect grouping of PM (see Section 3.1). The main differences are that the PMSPIC does not enforce periodicity, and that it considers a finite discrete horizon and allows time dependent costs. A special case of the PMSPIC is the *dynamic joint replenishment problem* (DJRP) which is a well studied problem in inventory theory. We utilize an ILP model for the PMSPIC which was originally introduced for the DJRP in [18] and is described as follows.[3] Introduce the variables

$$x_{st}^i = \begin{cases} 1, & \text{if component } i \text{ receives PM at times } s \\ & \text{and } t \text{, and not in-between,} \\ 0, & \text{otherwise,} \end{cases} \qquad \begin{array}{l} i \in \mathcal{N}, s \in \{0, \dots T\}, \\ t \in \{s+1, \dots, T+1\}, \end{array}$$

$$z_t = \begin{cases} 1, & \text{if a maintenance occasion occurs at time } t, \\ 0, & \text{otherwise,} \end{cases} \qquad t \in \mathcal{T}.$$

The model is now formulated as that to

$$\text{minimize} \quad \sum_{t \in \mathcal{T}} d_t z_t + \sum_{i \in \mathcal{N}} \sum_{s=0}^{T} \sum_{t=s+1}^{T+1} c_{st}^i x_{st}^i, \tag{8a}$$

$$\text{subject to} \quad \sum_{s=0}^{t-1} x_{st}^i \leq z_t, \qquad i \in \mathcal{N}, t \in \mathcal{T}, \tag{8b}$$

$$\sum_{s=0}^{t-1} x_{st}^i = \sum_{s=t+1}^{T+1} x_{ts}^i, \qquad i \in \mathcal{N}, t \in \mathcal{T}, \tag{8c}$$

$$\sum_{s=1}^{T+1} x_{0s}^i = 1, \qquad i \in \mathcal{N}, \tag{8d}$$

[3]The original model is based on the fact that the DJRP can be interpreted as an instancs of the PMSPIC with a special cost structure, hence the model can be utilized on any instance of the PMSPIC.

$$x^i_{st} \in \{0,1\}, \qquad i \in \mathcal{N}, s \in \{0,\ldots,T\}, \; t \in \{s+1,\ldots,T+1\}, \tag{8e}$$

$$z_t \in \{0,1\}, \qquad t \in \mathcal{T}. \tag{8f}$$

The contribution of Paper IV consists of investigating theoretical properties of the model (8) as well as demonstrating the usefulness of the model in case studies. We show that the integrality restrictions on x_{it} can be relaxed. We also show that the inequality constraints (8b) are facets of the convex hull of the set of feasible solutions.

Further, we demonstrate the usefulness of the ILP model in three case studies originating from the railway, aircraft, and wind turbine industries. In the railway case the deterioration cost corresponds to a cost incurred due to a degradation of rail. In the aircraft and wind turbine cases, the deterioration cost originates from the expected CM cost for components with stochastic failures. To do so, we choose the deterioration cost functions according to $M_i(t) := c_i^{\mathrm{CM}} m_i(t)$, where c_i^{CM} is the corrective maintenance cost for a failed component i and $m_i : \mathbb{R} \to \mathbb{R}$ is the renewal function for the failure distribution of component i. The value $m_i(t)$ is defined as the expected number of failures in the time interval $[0,t]$ if failed components are replaced by new ones. Solving the PMSPIC with this choice of deterioration cost functions yields the optimal PM schedule given that failed components are replaced by new ones and that the PM schedule stays fixed during the whole planning period.

Since Paper IV considers stochastic component failures, there is an overlap with the SORP of Paper III. The difference is that in Paper III we assume that the cost of PM and the cost of CM are equal. Hence, PM occurs only in connection with CM. Further, no schedule for PM exists; instead, all future PM decisions depend on the system state at failure, which makes the recourse function difficult to evaluate. In Paper IV, we instead focus on scheduling and are able to incorporate higher CM costs through the deterioration cost function. The ILP model for the PMSPIC is easier to solve than the current problem of the SORP. Hence, we can also use the model opportunistically by re-optimizing at failure. However, although the deterioration function is modified to compensate for rescheduling effects, the full effects of rescheduling can not be accounted for and the maintenance decision obtained are no longer provably optimal. Nevertheless, results from simulations on case studies suggest that we can reduce costs compared to age and constant interval policies by 5–20%. The computational time for solving the model with were below 10 seconds on most instances, and approximately 200 seconds on the most difficult instance of the case study.

4.5 Paper V: A comparison of several models for the Hamiltonian p-median problem

In this paper, we consider the Hamiltonian p-median (HpMP) problem which is a location–routing problem (see Section 3.2). The problem was introduced in [6]. It consists of placing exactly p facilities in a graph and routing one vehicle from each facility such that each node is on exactly one route and the total length of the routes is minimized. The problem can also be described as that to cover all nodes by p mutually node-disjoint cycles with a minimal total edge weight. Since locating a facility does not generate a cost, any node in a cycle can be considered to be the facility. To simplify modeling, we assume that each cycle has at least three nodes and consider an undirected graph. Figure 13 illustrates an optimal solution on the graph provided by the TSPLIB (see [30]) instance danzig42 for $p = 4$.

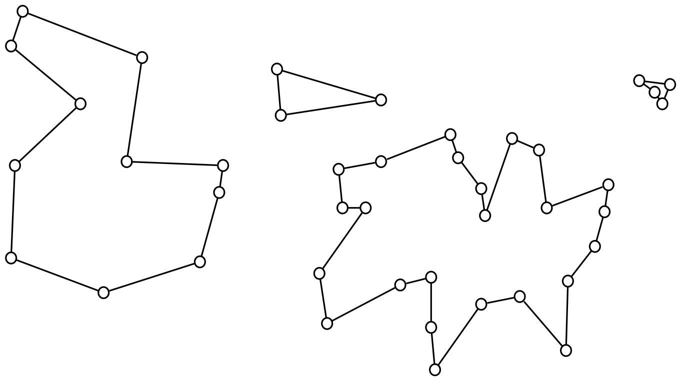


Figure 13: An optimal solution to the HpMP instance danzig42 for $p = 4$.

The literature contains several ILP models of the HpMP; no computational comparison of these has, however, been previously conducted. In this paper, we present new models and perform both a theoretical and a computational comparison of the new and old models. The theoretical analysis is based on comparing the strengths of the respective LP relaxations (see Section 2.2.2).

From a computational point of view, the following model, introduced in Paper V, is shown to be superior. Let us first introduce some notation. We consider the undirected graph $G = (V, E)$, where we label edges such that $(i, j) \in E$ implies that $i < j$. Let $\delta(i) = \{e \in E \mid e = (i, j) \text{ or } e = (j, i) \text{ for } j \in V\}$ be the set of edges connecting to node $i \in V$. Define the partition of the set V of vertices into m subsets with a minimum number of l nodes in each subset: $P = \{S_1, \ldots, S_m\} \in \mathbf{P}^l_m$ such that $|S_i| \geq l$ for all $i \in \{1, \ldots, m\}$, where $\mathbf{P}^l_m$ denotes the set of all such partitions. Let E_P be the m-cut associated with P, that is, the set of edges that connect any two node sets of the partition. Further, let $\mathbf{C}_p = \{C \subseteq E \mid \text{The edges in } C \text{ form exactly } p \text{ disjoint cycles}\}$. Finally, introduce the variables x_e for $e \in E$, defined by $x_e = 1$ if edge $e \in E$ is in the solution and $x_e = 0$ otherwise. The model is to

$$\begin{aligned}
\text{minimize} \quad & \sum_{e \in E} c_e x_e, \\
\text{subject to} \quad & \sum_{e \in \delta(i)} x_e = 2, \qquad i \in V, && \text{(9a)} \\
& \sum_{e \in E_P} x_e \geq 2, \qquad P \in \mathbf{P}^3_{p+1}, && \text{(9b)} \\
& \sum_{e \notin C} x_e \geq 2, \qquad C \in \mathbf{C}_{p-1}, && \text{(9c)} \\
& x_e \in \{0, 1\}, \quad e \in E. && \text{(9d)}
\end{aligned}$$

The objective is to minimize the length of the sum of cycles. The constraint (9a) states that exactly two edges are connected to each vertex. If only constraints (9a) and (9d) are included into the model, the solution obtained corresponds to the optimal two-matching. The two-matching problem can be considered as the HpMP with the number of cycles p being variable and is polynomially solvable. In order to obtain exactly p cycles, we need to introduce additional constraints. The constraints (9b) remove integer solutions with more than p cycles, whereas the constraints (9c) remove integer solutions with less than p cycles. The number of constraints of both types increases exponentially with the instance size; hence they are not explicitly included but dynamically generated in a branch–and–cut procedure. For integer solutions with more than p cycles, a violated constraint (9b) is found by partitioning the nodes according to the cycles obtained. Similarly, for integer solutions with less than p cycles, a violated inequality (9c) is found by choosing C corresponding to the current solution. However, we show that for fractional solutions satisfying (9a) solving the separation problem of (9b) as well as of (9c) is NP-hard. Hence we resort to heuristic separation procedures of fractional solutions. The model (9) is the only model found in the literature in the natural variable space (i.e., only edge variables x_e for each edge $e \in E$). We next present a model with additional variables.

The model (9) combined with the indicated solution procedure performs well with respect to computation time but, according to computational tests, the LP relaxation of the model is not very strong. Instead, the computational tests show that the following p-median based model, also introduced in Paper V, possesses the overall strongest LP relaxation. We introduce some additional notation. For each edge $e \in E$ define the ordered vertex pair $\gamma(e) = \{\, (i,j) \in V \times V \mid (i,j) = e \text{ or } (j,i) = e \,\}$. For each proper node subset $W \subset V$, define the cut set $\delta(W) = E_{\{W, V \setminus W\}}$. Introduce the variables $v_{ij} = 1$ if vertex $i \in V$ is served by depot at vertex $j \in V$ and $v_{ij} = 0$ otherwise. We also introduce $y_i = 1$ if vertex $i \in V$ is a depot and $y_i = 0$ otherwise. Since the choice of depot for a cycle is arbitrary, we assume that the vertex with lowest index in a cycle is the depot of that cycle. This construction reduces the symmetry of the formulation. The model is to

$$
\begin{aligned}
\text{minimize} \quad & \sum_{e \in E} c_e x_e, && && \\
\text{subject to} \quad & \sum_{e \in \delta(i)} x_e = 2, && i \in V, && \text{(10a)}\\
& \sum_{e \in \delta(W)} x_e \geq 2 \sum_{j \in V \setminus W} v_{ij}, && i \in W \subset V \setminus \{j\}, && \text{(10b)}\\
& v_{il} + x_e \leq 1 + v_{jl}, && e \in E, (i,j) \in \gamma(e), l \in V \setminus \{i,j\}, && \text{(10c)}\\
& y_i + x_e \leq 1 + v_{ji}, && e \in E, (i,j) \in \gamma(e), && \text{(10d)}\\
& \sum_{i \in V} y_i = p, && && \text{(10e)}\\
& \sum_{j \in V \setminus \{i\}} v_{ij} + y_i = 1, && i \in V, && \text{(10f)}\\
& v_{ij} \leq y_i, && i \in V, j \in V \setminus \{i\}, && \text{(10g)}\\
& v_{ij} = 0, && i < j, i, j \in V, && \text{(10h)}\\
& x_e \in \{0,1\}, && e \in E, && \text{(10i)}\\
& v_{ij} \in \{0,1\}, && i \in V, j \in V \setminus \{i\}, && \text{(10j)}\\
& y_i \in \{0,1\}, && i \in V. && \text{(10k)}
\end{aligned}
$$

The constraints (10b) ensure that if a node is served by a depot outside the set of nodes W, at least two edges in the cut set of W must be included in the solution. The constraints (10c) and (10d) imply that nodes which are linked must be served by the same depot. The constraint (10e) ensures that p depots are located in the graph. The constraints (10f) ensure that each node is either a depot or served by another depot. The constraints (10g) imply that nodes cannot be served by other nodes if these are not depots. Finally, the constraints (10h) imply that a node may only be served by a depot with a lower index than the node itself.

A variety of other models from the literature are also investigated in the paper. These models are, however, either shown to have a weak LP relaxation or perform poorly in the computational tests. The only exception is a model introduced (without implementation) in [6] based on a Dantzig–Wolfe decomposition approach, wherein each column corresponds to one cycle. We show that the model has a strong LP relaxation. Implementing the model, however, requires a branch–and–price approach in which the subproblem becomes a prize collecting TSP and was considered to be outside of the scope of Paper V.

5 Contributions of the thesis

This section presents a short summary of the main contributions of the thesis, in terms of both the applications considered and mathematical results.

From the point of view of the applications, the following contributions are made. ILP models for several maintenance optimization problems have been developed, implemented, and studied computationally (Papers I, II, and IV). A stochastic programming model (Paper III) together with a decomposition method, which extends the integer L-shaped method, have been developed, implemented and analyzed for the maintenance problem considered. The study of these models constitute important steps towards a decision support system for maintenance decisions. In particular the model with costs based on the replacement interval length (proposed in Paper IV) presents an interesting combination of flexibility and computational tractability to build upon in future research. A connection between the joint replenishment problem from inventory theory and the considered maintenance optimization problems has also been established (Paper IV). The research on the HpMP (Paper V) has contributed to the classification and comparison of new and existing ILP models for the problem. We have demonstrated that problem instances of moderate sizes can be solved by a branch–and–cut method. The conclusion is that three models are of particular interest for future research.

From a mathematical point of view several interesting properties have been investigated. Complexity analyses of two opportunistic replacement problems (ORP in Paper I and ORPIL in Paper II) as well as separation problems for the model of the HpMP (Paper V) have been made. Polyhedral studies of ILP models (Papers I, II, and IV) as well as a comparison of the LP relaxation of different models for the same problem (Paper II and V) were conducted. Finally, lower bounds on the recourse function for the SORP leading to the development of the decomposition method have been obtained (Paper III). All of the above has increased the understanding of the problems as well as lead to ILP models which are efficiently solved.

References

[1] N. ANDRÉASSON, *Optimization of opportunistic replacement activities in deterministic and stochastic multi-component systems*, licentiate thesis, Department of Mathematics, Chalmers University of Technology and Göteborg University, Göteborg, Sweden, 2004.

[2] R. BARLOW AND L. HUNTER, *Optimum preventive maintenance policies*, Operations Research, 8 (1960), pp. 90–100.

[3] C. BARNHART, E. JOHNSON, G. NEMHAUSER, M. SAVELSBERGH, AND P. VANCE, *Branch-and-price: Column generation for solving huge integer programs*, Operations Research, 48 (1998), pp. 316–329.

[4] J. F. BENDERS, *Partitioning procedures for solving mixed-variables programming problems*, Numerische Mathematik, 4 (1962), pp. 238–252.

[5] J. BIRGE AND F. LOUVEAUX, *Introduction to Stochastic Programming*, Springer-Verlag, New York, NY, USA, 1997.

[6] I. BRANCO AND J. COELHO, *The Hamiltonian p-median problem*, European Journal of Operational Research, 47 (1990), pp. 86–95.

[7] J. CROCKER AND U. KUMAR, *Age-related maintenance versus reliability centred maintenance: a case study on aero-engines*, Reliability Engineering & System Safety, 67 (2000), pp. 113–118.

[8] G. DANTZIG AND J. RAMSER, *The truck dispatching problem*, Management Science, 6 (1959), pp. 80–91.

[9] G. DANTZIG AND P. WOLFE, *Decomposition principle for linear programs*, Operations research, 8 (1960), pp. 101–111.

[10] R. DEKKER, R. WILDEMAN, AND F. VAN DER DUYN SCHOUTEN, *A review of multi-component maintenance models with economic dependence*, Mathematical Methods of Operations Research, 45 (1997), pp. 411–435.

[11] B. DICKMAN, S. EPSTEIN, AND Y. WILAMOWSKY, *A mixed integer linear programming formulation for multi-component deterministic opportunistic replacement*, The Journal of the Operational Research Society of India, 28 (1991), pp. 165–175.

[12] E. DIJKSTRA, *A note on two problems in connexion with graphs*, Numerische Mathematik, 1 (1959), pp. 269–271.

[13] J. EDMONDS AND R. M. KARP, *Theoretical improvements in algorithmic efficiency for network flow problems*, Journal of the ACM, 19 (1972), pp. 248–264.

[14] M. R. GAREY AND D. JOHNSON, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W.H. Freeman and Company, New York, NY, USA, 1979.

[15] C. GENTILE, P. VENTURA, AND R. WEISMANTEL, *Mod-2 cuts generation yields the convex hull of bounded integer feasible sets*, SIAM Journal on Discrete Mathematics, 20 (2006), pp. 913–919.

[16] B. GOLDEN, S. RAGHAVAN, AND E. WASIL, *The vehicle routing problem: latest advances and new challenges*, Springer-Verlag, New York, NY, USA, 2008.

[17] T. HALE AND C. MOBERG, *Location science research: a review*, Annals of Operations Research, 123 (2003), pp. 21–35.

[18] D. JONEJA, *The joint replenishment problem: new heuristics and worst case performance bounds*, Operations Research, (1990), pp. 711–723.

[19] P. KALL AND S. WALLACE, *Stochastic Programming*, John Wiley & Sons, New York, NY, USA, 1994.

[20] A. H. LAND AND A. G. DOIG, *An automatic method of solving discrete programming problems*, Econometrica, 28 (1960), pp. 497–520.

[21] G. LAPORTE AND F. LOUVEAUX, *The integer L-shaped method for stochastic integer programs with complete recourse*, Operations Research Letters, 13 (1993), pp. 133–142.

[22] A. LODI, *Mixed integer programming computation*, in 50 Years of Integer Programming 1958-2008, Springer-Verlag, Berlin, Germany, 2008, pp. 619–645.

[23] K. MURTY, *Linear programming*, John Wiley & Sons, New York, NY, USA, 1983.

[24] G. NAGY AND S. SALHI, *Location-routing: Issues, models and methods*, European Journal of Operational Research, 177 (2007), pp. 649–672.

[25] G. NEMHAUSER AND L. WOLSEY, *Integer and combinatorial optimization*, John Wiley & Sons, New York, NY, USA, 1989.

[26] R. P. NICOLAI AND R. DEKKER, *Optimal maintenance of multi-component systems: A review*, in Complex System Maintenance Handbook, K. A. H. Kobbacy and D. N. P. Murthy, eds., Springer Series in Reliability Engineering, Springer-Verlag, Berlin, Germany, 2008, pp. 263–286.

[27] S. OWEN AND M. DASKIN, *Strategic facility location: A review*, European Journal of Operational Research, 111 (1998), pp. 423–447.

[28] R. ROBERTSON AND A. JONES, *Pay day*, Plant Engineering and Maintenance, 28 (2004), pp. 18–25.

[29] S. SEN, *Algorithms for stochastic mixed-integer programming models*, in Handbooks in Operations Research and Management Science: Discrete Optimization, K. Aardal, G. Nemhauser, and R. Weismantel, eds., vol. 12, Elsevier, Amsterdam, Netherlands, 2005, ch. 9, pp. 515–558.

[30] TSPLIB. `http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/`; accessed on September 2nd 2012.

[31] R. M. VAN SLYKE AND R. WETS, *L-shaped linear programs with application to optimal control and stochastic programming*, SIAM Journal on Applied Mathematics, 17 (1969), pp. 638–663.

[32] R. J. VANDERBEI, *Linear Programming: Foundations and Extensions*, Springer, New York, NY, USA, 2008.

[33] H. WANG, *A survey of maintenance policies of deteriorating systems*, European Journal of Operational Research, 139 (2002), pp. 469–489.

[34] L. WOLSEY, *Integer Programming*, John Wiley & Sons, New York, NY, USA, 1998.