Linköping Studies in Science and Technology
Dissertation No. 1854

Computational Complexity of some Optimization Problems in Planning

Meysam Aghighi

Linköping University
Department of Computer and Information Science
SE-581 85 Linköping, Sweden

Linköping 2017

ISBN 978-91-7685-519-5
ISSN 0345-7524
URL `http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-136280/`

Typeset using $\LaTeX$

Printed by LiU-Tryck, Linköping 2017

Computational Complexity of some Optimization Problems in Planning

ABSTRACT

Automated planning is known to be computationally hard in the general case. Propositional planning is **PSPACE**-complete and first-order planning is undecidable. One method for analyzing the computational complexity of planning is to study restricted subsets of planning instances, with the aim of differentiating instances with varying complexity. We use this methodology for studying the computational complexity of planning.

Finding new tractable (i.e. polynomial-time solvable) problems has been a particularly important goal for researchers in the area. The reason behind this is not only to differentiate between easy and hard planning instances, but also to use polynomial-time solvable instances in order to construct better heuristic functions and improve planners.

We identify a new class of tractable cost-optimal planning instances by restricting the causal graph. We also study the computational complexity of oversubscription planning (such as the net-benefit problem) under various restrictions and reveal strong connections with classical planning. Inspired by this, we present a method for compiling oversubscription planning problems into the ordinary plan existence problem. We further study the parameterized complexity of cost-optimal and net-benefit planning under the same restrictions and show that the choice of numeric domain for the action costs has a great impact on the parameterized complexity.

We finally consider the parameterized complexity of certain problems related to partial-order planning. In some applications, less restricted plans than total-order plans are needed. Therefore, a partial-order plan is being used instead. When dealing with partial-order plans, one important question is how to achieve optimal partial order plans, i.e. having the highest degree of freedom according to some notion of flexibility. We study several optimization problems for partial-order plans, such as finding a minimum deordering or reordering, and finding the minimum parallel execution length.

The research presented in this thesis has been partially funded by the National Graduate School of Computer Science in Sweden (CUGS).

Populärvetenskaplig sammanfattning

Huvudtemat i denna avhandling är studiet av beräkningskomplexitet för automatisk planering. På en hög nivå kan man betrakta planeringsproblemet så här; man utgår från en värld eller ett system, som ett obemannat fordon eller en autonom robot, och antar att det finns vissa handlingar som ändrar tillståndet i systemet. Man antar dessutom att det finns ett initial- och ett måltillstånd. Planering är problemet att hitta en sekvens av handlingar som tranformerar initialtillståndet till måltillståndet. En sådan sekvens av handlingar kallas en plan. En planerare är ett datorprogram som löser planeringsproblem, och planerare baseras ofta på sökning genom tillstånden i systemet. Vanligtvis vill man optimera sökningen och hitta en plan som är optimal med avseende på något kriterium. Vad man vill optimera beror på tillämpningen men det är ofta att hitta en kortaste eller billigaste plan.

En viktig fråga när man studerar automatisk planering är hur snabbt man kan hitta lösningar. Ett sätt att analysera denna fråga är att studera beräkningskomplexiteten för olika planeringsproblem. Komplexitetsteori är den gren av datavetenskapen som försöker klassificera hur svåra olika beräkningsproblem är, det vill säga analysera hur mycket resurser (såsom tid och minne) som behövs. Man likställer ofta effektivt lösbara problem med de som kan lösas i *polynomisk tid*: ett problem kan lösas i polynomisk tid om det finns en algoritm vars tidsåtgång begränsas av ett polynom i indatas storlek. Det grundläggande planeringsproblemet är exempelvis känt att vara **PSPACE**-fullständigt. **PSPACE** är den komplexitetsklass som innehåller alla de beräkningsproblem som kan lösas genom att använda polynomiskt mycket minne och de **PSPACE**-fullständiga problemen utgör de svåraste problemen i **PSPACE**. Man tror att det inte finns några **PSPACE**-fullständiga problem som kan lösas i polynomisk tid—detta är ett viktigt olöst problem inom teoretisk datalogi—och man förväntar sig därför inte att det generella planeringsproblemet kan lösas i polynomisk tid.

En metod för att analysera beräkningskomplexitet av planering är att studera begränsade delmängder av planeringsinstanser i syfte att särskilja fall med varierande komplexitet. Vi använder denna metod genomgående i denna avhandling. Så kallad ”oversubscription planning” är en typ av planering där man mäter hur nära ett givet måltillstånd

man kan komma. Vi studerar beräkningskomplexiteten för denna typ av planering under olika restriktioner och avslöjar starka kopplingar med vanlig planering. Kostnadsoptimal planering är planering där varje handling har en kostnad och målet är att hitta en plan med minimal kostnad. Dessa handlingskostnader kan hämtas från olika numeriska domäner såsom de positiva heltalen eller de rationella talen. Vi studerar effekter av valet av numeriska domäner och noterar intressanta komplexitetsmässiga skillnader. För kostnadsoptimal planering identifierar vi dessutom en ny klass av instanser som kan lösas i polynomisk tid. Slutligen studerar vi komplexiteten för olika problem inom så kallad partialordningsplanering där man tillåter planer som inte nödvändigtvis är sekvenser av handlingar.

Acknowledgments

First and foremost, I want to thank my advisors Peter Jonsson and Christer Bäckström. It has been an honor to work with and learn from them for the past three years. They have taught me how well the academic research is done. I am specially grateful to Peter for his great help and Christer for his valuable comments in preparing this thesis.

The members of TCSLAB and IDA contributed immensely to my personal and professional time as a PhD student at LiU. I specifically thank Simon Ståhlberg, who was both a colleague and a friend, Amir Aminifar, Biman Roy, Arian Maghazeh, Adrian Horga, Rouhollah Mahfouzi and Victor Lagerkvist.

I am also grateful to the administration at IDA that facilitated my work in less direct ways. I specially thank Karin Baardsen, Anne Moe, Marie Johansson, Inger Norén and the director of graduate studies, Professor Petru Eles.

Lastly, I would like to thank my family for all their love and encouragement. For my parents who raised me with a love of science and supported me in all my pursuits. For my loving, supporting, encouraging and patient wife Zeinab whose support during my whole PhD is so appreciated. Thank you.

MEYSAM AGHIGHI
Linköping University
May 2017

List of Papers

The thesis is based on the following papers:

1. Meysam Aghighi and Peter Jonsson.
   Oversubscription planning: Complexity and compilability.
   In *Proceedings of the Twenty-Eighth **AAAI** Conference on Artificial Intelligence, Québec City, Québec, Canada, July 27 -31, pages 2221-2227, 2014.*
2. Meysam Aghighi, Peter Jonsson and Simon Ståhlberg.
   Tractable cost-optimal planning over restricted polytree causal graphs.
   In *Proceedings of the Twenty-Ninth **AAAI** Conference on Artificial Intelligence, Austin, Texas, USA, January 25-30, pages 3225-3231, 2015.*
3. Meysam Aghighi and Christer Bäckström.
   Cost-optimal and net-benefit planning - A parameterised complexity view.
   In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence (**IJCAI**), Buenos Aires, Argentina, July 25-31, pages 1487-1493, 2015.*
4. Meysam Aghighi and Christer Bäckström.
   A multi-parameter complexity analysis of cost-optimal and net-benefit planning.
   In *Proceedings of the Twenty-Sixth International Conference on Automated Planning and Scheduling (**ICAPS**), London, UK, June 12-17, pages 2-10, 2016.*
5. Meysam Aghighi and Christer Bäckström.
   Plan reordering and parallel execution – a parameterized complexity view.
   In *Proceedings of the Thirty-First **AAAI** Conference on Artificial Intelligence, San Francisco, California, USA, February 4-9, pages 3540-3546, 2017.*

In addition to the above papers, the author has also contributed to the following publications:

1. Meysam Aghighi, Christer Bäckström, Peter Jonsson and Simon Ståhlberg.
   Analysing Approximability and Heuristics in Planning Using the

Exponential-time Hypothesis.
In *Proceedings of the Twenty-second European Conference on Artificial Intelligence (**ECAI**) , the Hague, Holland, August 29th - September 2nd, pages 184-192, 2016.*

2. Meysam Aghighi, Christer Bäckström, Peter Jonsson and Simon Ståhlberg.
   Refining Complexity Analyses in Planning by Exploiting the Exponential-time Hypothesis.
   In *Annals of Mathematics and Artificial Intelligence (**AMAI**), pages 157-175, 2016.*

Contents

II Papers 37

Part I

Introduction

1 Thesis Overview

1 About this Thesis

The main topic of this thesis is the study of the computational complexity of automated planning problems. Automated planning has been studied by the artificial intelligence community over the past fifty years. The intersection of complexity theory and planning has been studied since the early work by Chapman [21] which proved that first-order planning is undecidable. Since then, the complexity of planning has attracted a lot of interest.

This thesis is a compilation of five papers. It consists of two parts. The first part contains four chapters. This chapter gives an overview of the thesis. Chapters 2 and 3 give an introduction to computational complexity theory and automated planning, respectively. In Chapter 4, the background of each paper and its contributions are summarized. The second part consists of five peer-reviewed conference papers published between the years 2014 and 2017.

In order to read and understand the contents of the thesis, a basic knowledge in discrete mathematics and graph theory is expected from the reader. Also, familiarity with planning and computational complexity theory is very helpful, in order to understand the contributions of each paper, the motivation of the discussed problems and the intuition behind the proofs. Some good references for complexity theory are Arora and Barak [4] and Garey and Johnson [31], for parameterized complexity Flum and Grohe [30] and Downey and Fellows [27], and for planning Ghallab, Nau, and Traverso [33] and Geffner and Bonet [32].

2 Brief Summary of Papers

This section gives a brief summary of the five papers. More information about the background and contributions of each paper can be found in Chapter 4.

Paper 1

This paper provides a complexity analysis of the oversubscription planning problems, many sub-classes are introduced under the PUBS and Bylander restrictions and a detailed complexity map is presented for them. Moreover, a polynomial-time compilation of net-benefit planning problem to plan existence is presented using a novel counter-based reduction.

Paper 2

This paper presents a new class of tractable cost-optimal planning instances. Our algorithms uses a novel concept of variable isomorphism and shows that every planning instance with: (1) bounded domain size, and (2) polytree causal graph with bounded diameter, is polynomial-time solvable.

Paper 3

This paper gives a parameterized complexity analysis of the cost-optimal and net-benefit planning problems. The two problems are studied with different cost domains for action costs and under Bylander and PUBS restrictions. Our results show that cost-optimal planning for positive integers ($\mathbb{Z}_{+}$) is not harder than length-optimal planning while it becomes harder if we use non-negative integers ($\mathbb{Z}_{0}$) or positive rationals ($\mathbb{Q}_{+}$).

Paper 4

This paper continues paper 3, but this time using a multi-parameter complexity analysis. The paper identifies the parameters that affect complexity the most, such as plan length, maximum denominator of action costs and sum of goal utilities.

Paper 5

This paper continues the work of Bäckström [5] on reordering partial-order plans by applying parameterized complexity theory and introducing parameters like the height and width of a partial-order. The paper analyzes plan optimization using two criteria: order size and parallel length of a plan. It also introduces a new problem: finding the minimum parallel length of a partial-order plan when there is a bounded number of processors. We show that this problem is significantly harder than the unbounded case.

The papers appear exactly as in the conference proceedings. However, there has been some minor formatting adjustments. The papers of this thesis can be divided into categories from several points of view. From the goal perspective, the first four papers consider cost-optimal and net-benefit planning while paper 5 is on plan optimization. From the complexity analysis point of view, papers 1 and 2 use traditional complexity while papers 3, 4 and 5 use parameterized complexity. From the planning domain perspective, papers 1, 3 and 4 use syntactic restrictions on classical planning, paper 2 uses both syntactic and semantic restrictions and paper 5 goes beyond classical planning by considering partial-order plans.

2 Computational Complexity Theory

1 Computational Complexity

Computational Problems

Computational complexity theory is a branch of computer science that classifies computational problems according to their inherent difficulty. There are several types of computational problems, such as decision, counting, search, optimization, etc. An optimization problem asks for best solution among the set of all possible solutions and a decision problem asks for a yes or no answer, i.e. is there a solution for a given problem instance? Formally, a problem instance and its solution are considered to be strings of bits (i.e. members of the set $\{0,1\}^*$). Simple encodings can represent general mathematical objects such as integers, graphs, vectors, etc. A **computational problem** is a set of problem instances with a solution for each one of them, i.e. a function $f : \{0,1\}^* \to \{0,1\}^*$. Note that this solution can be an arbitrary mathematical object, like a list of numbers or a set of shortest paths in a graph. A *decision problem* is a special case of computational problem where f is a boolean function $f : \{0,1\}^* \to \{0,1\}$ which is often identified with the set $L_f = \{x : f(x) = 1\}$ that is called a *language*. In complexity theory we are mostly interested in decision problems. We give two concrete examples.

Example I: Satisfiability

Let Φ be a formula in conjunctive normal form (CNF), e.g.

$$\Phi(x_1, x_2, x_3) = (x_1 \vee \neg x_2) \wedge (\neg x_1 \vee x_2 \vee x_3)$$

Φ is satisfiable if there exists a truth assignment to the variables x_1, x_2 and x_3 such that the value of the formula $\Phi(x_1, x_2, x_3)$ is true. For example, $x_1 \mapsto$ TRUE, $x_2 \mapsto$ FALSE and $x_3 \mapsto$ TRUE is a satisfiable assignment for the above formula.

The decision version of SATISFIABILITY (SAT) problem is defined as follows:

> INPUT: A CNF formula Φ with variables in $X = \{x_1, x_2, \ldots, x_n\}$.
> QUESTION: Is there a satisfying truth assignment for Φ?

A truth assignment can be viewed as a function $\alpha : X \to \{\text{TRUE}, \text{FALSE}\}$ and the decision problem as a function $f : \{0,1\}^* \to \{0,1\}$. f takes a string in the input that represents a CNF formula and the set X. Then it returns 1 if the formula is satisfiable, otherwise it returns 0. L_f is the set of all strings on which f returns 1, i.e. the set of all satisfiable CNF formulas.

Next, we present an example based on an optimization problem:

Example II: Traveling Salesperson Problem

The optimization version of TRAVELING SALESPERSON PROBLEM (TSP) is defined as follows:

> INPUT: A graph $G = (V, E)$ and a function $c : E \to \mathbb{N}$.
> QUESTION: Find the minimum value $k \in \mathbb{N}$ for which a Hamiltonian cycle of cost k exists, i.e. there is a permutation $(v_1, v_2, \ldots, v_n)$ of vertices in V such that $(v_n, v_1) \in E$, for every $1 \leqslant i \leqslant n-1$, $(v_i, v_{i+1}) \in E$ and
>
> $$\left(\sum_{i=1}^{n-1} c(v_i, v_{i+1})\right) + c(v_n, v_1) \leqslant k$$

The optimization problem can be viewed as a function $f : \{0,1\}^* \to \{0,1\}^*$. f takes a string in the input that represents a TSP instance, which is a graph and a cost function for the edges. Then it returns a string in the output that represents the minimum integer k for which a Hamiltonian cycle of cost k exists, or a special designated character that means no Hamiltonian cycles exist in the graph.

Model of Computation

In order to measure the complexity of computational problems and analyze the resources required for algorithms and computers, we need to define a model for computation. There are several models for computation such as the random-access machine, the finite-state machine, the Turing machine, the logic circuit, etc. The Turing machine was proposed by Alan Turing in 1936 and is the most commonly used model in computation. This is because of the Church-Turing thesis; a hypothesis stating that every "reasonable" model of computation can be simulated by a Turing machine with only a polynomial slowdown in run-time in terms of the input size. Please note that this was not the original Church-Turing thesis, but a result of the extended Church-Turing hypothesis [4]. It is also believed that for any computer algorithm, a Turing machine can be constructed to simulate the algorithm's logic [70].

A **Turing machine** consists of an infinite memory tape divided into cells, a set of symbols and a finite table of instructions. The machine's head is positioned over a cell and reads the symbol there. Then, based on the read symbol and the current position of the machine in the table of instructions, the head writes a symbol in the cell and then moves either one cell to the left or one cell to the right. Then, it either proceeds to the next instruction or halts the computation.

Formally, a deterministic (one-tape) Turing machine M is described by a 7-tuple $(Q, \Gamma, b, \Sigma, \delta, q_0, F)$ where [45]

- Q is a finite, non-empty set of states that M can be in, including a designated initial state (q_0) and a set of final or accepting states ($F \subseteq Q$).
- Γ is a finite, non-empty set of symbols that M's tape can contain, including a designated blank symbol b (the only symbol allowed to occur on the tape initially often at any step during the computation). Γ is called the *alphabet* of M.
- $\Sigma \subseteq \Gamma \backslash \{b\}$ is the set of input symbols.
- $\delta : (Q \backslash F) \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$ is a function describing the instruction that M uses in every step. L and R denote the left and right movement by the machine's head. δ is called the *transition function* and denotes the table of instructions. If δ is not defined in the current state and the current tape symbol, then the machine halts.

A Turing machine accepts the initial tape contents, if it eventually halts in a state from F.

There are several variations of the Turing machine. The two most common are *deterministic* Turing machine (DTM) and *non-deterministic* Turing machine (NTM). In a non-deterministic Turing machine, the instructions' table allows more than one action in each situation and the transition function is replaced by a transition relation. For instance δ may contain both $((q, a), (q_1, b, L))$ and $((q, a), (q_2, c, R))$. This is to be interpreted as: if the machine is in state q and its head reads the symbol a, it has the choice of going to state q_1, write symbol b and move left, or go to state q_2, write symbol c and move right. An NTM accepts an input if there exists some choice of actions that leads to the designated halting state. These variations are in principle equally powerful but when the resources (such as time and space) are restricted this equality breaks down.

Complexity Measures

The two most common complexity measures are time and space. The running time of an algorithm, intuitively, is the number of basic operations performed by it, where a basic operation takes a fixed amount of time to perform. The run-time of algorithm A on input x is formally defined as the number of state transitions of the corresponding deterministic Turing machine for the algorithm, on input x. The time complexity of an algorithm is the run-time as a function of the input size (number of bits needed to represent the input). Similar to time, the space complexity is the amount of working storage an algorithm needs. The complexity is often reported *asymptotically*, as the input size goes to infinity. For instance, if an algorithm takes $3n^5 + 4n^2 \log n - 2n$ units of time for an input of size n, we say that the time complexity of the algorithm is $O(n^5)$, using the big O notation. This is because for large values of n, the value other terms are insignificant compared to the term $3n^5$. Since the time complexity of an algorithm may vary with different inputs of the same size, we usually consider the complexity as the maximum value over all inputs of size n. This is referred to as the *worst-case* complexity analysis.

Reduction and NP-completeness

Reduction is a way to show that one problem is at least as hard as another problem. Intuitively, problem A is reducible to problem B, if an algorithm that solves problem B, can be modified to solve problem A. There are several types of reductions and the most common form is the polynomial-time many-one (Karp) reduction which is usually referred to as the polynomial-time reduction.

Formally, we say that a language $A \subseteq \{0,1\}^*$ (equivalently, a decision problem) is polynomial-time reducible to a language $B \subseteq \{0,1\}^*$ denoted by $A \leqslant_p B$ if there exists a polynomial-time computable function $f : \{0,1\}^* \to \{0,1\}^*$ such that for every $x \in \{0,1\}^*$, $x \in A$ if and only if $f(x) \in B$.

A complexity class is a set of computational problems that can be computed within a given resource. For instance, **DTIME**$(f(n))$ is the class of all decision problems that can be solved by some DTM running in time $O(f(n))$ where n is the input size and f is some function. The two most well-known complexity classes are **P**, the class of problems solvable in polynomial time by a DTM, and **NP**, the class of problems solvable in polynomial time by an NTM. Informally, a decision problem belongs to the class of **NP** if and only if its instances for which the answer is "yes" have efficiently (polynomial-time) verifiable proofs. The two definitions are equivalent because the first part in both contains a *guess* about the solution, which is generated in a non-deterministic way, while the second part contains a deterministic algorithm that verifies or rejects the guess as a valid solution to the problem.

There are many other complexity classes such as **EXPTIME**, the class of problems solvable in exponential time by a DTM, **PSPACE**, the class of problems solvable in polynomial space by a DTM, **NL**, the class of problems solvable in logarithmic space by an NTM, **#P**, the class of counting problems associated with decision problems in **NP**, etc. The complexity classes are usually distinguished by: (1) The type of the computational problem (decision, counting, etc.), (2) The model of computation (DTM, NTM, Boolean circuits, etc.), and (3) The resource constraints (polynomial time, logarithmic space, etc.).

Let **C** be a complexity class. Problem A is **C**-hard if there is a polynomial-time reduction from every member of **C** to A. This means if there is an efficient algorithm for A, then this algorithm can also solve all problems in **C** with only a polynomial blowup in time. Moreover, A is said to be **C**-complete if it is **C**-hard and additionally in **C**. One may say that the **C**-complete problems are, intuitively, the hardest problems in **C**. Both SAT and TSP decision problems from Examples I and II are proved to be **NP**-complete [31, LO1, GT37]. This results in the optimization of TSP to be **NP**-hard, because a solution to the optimization version of TSP will immediately solve its decision version.

The **P** vs **NP** problem ($\mathbf{P} \overset{?}{=} \mathbf{NP}$) is a major open problem in computer science with a million dollar prize by the Clay Mathematics Institute. It asks if every problem whose solution can be verified in polynomial time by a computer can also be solved in polynomial time by a computer. Most researchers in the area believe that the answer to this question is negative (i.e. $\mathbf{P} \neq \mathbf{NP}$) and this would imply that many common problems cannot be solved efficiently. However, a positive answer would have enormous practical consequences; for instance, many cryptography algorithms rely on the difficulty of certain problems [60].

2 Parameterized Complexity

In traditional complexity analysis, the time complexity is measured only in terms of the input size. However, many problem instances consist of two or more parameters that contribute differently to the complexity of the problem.

Example: Vertex Cover vs Dominating Set

The problems VERTEX COVER and DOMINATING SET are defined as follows:

> VERTEX COVER
> INPUT: A graph $G = (V, E)$ and a positive integer k.
> QUESTION: Does G have a vertex cover of size at most k, i.e. is there a set of vertices $V' \subseteq V$ such that $|V'| \leqslant k$ and for every edge $(u, v) \in E$, either $u \in V'$ or $v \in V'$?
>
> DOMINATING SET
> INPUT: A graph $G = (V, E)$ and a positive integer k.
> QUESTION: Does G have a dominating set of size at most k, i.e.

> is there a set of vertices $V' \subseteq V$ such that $|V'| \leqslant k$ and for every vertex $u \in V \backslash V'$, $(u, v) \in E$ for some $v \in V'$?

Although both problems are **NP**-complete, the input parameter k contributes differently to the complexity of the two problems. Let n be the number of vertices. The best known algorithm for VERTEX COVER solves the problem in time $O(1.2738^k + kn)$ [22], while the best known algorithm for DOMINATING SET is the brute force algorithm of trying all k-subsets which takes $O(n^{k+1})$ time. This is a significant difference; the ratio of n^{k+1} to $1.2738^k + kn$ is more than 10^{16} for $n = 150$ and $k = 10$. □

The observation of such differences between **NP**-complete problems motivated a new and more fine-grained form of complexity analysis: parameterized complexity analysis. Parameterized complexity tries to find how different parameters of a problem contribute to its hardness. For instance, parameter k in VERTEX COVER usually has a much smaller value compared to n in practice. Therefore, even an exponential algorithm that runs in time $O(1.2738^k + kn)$ performs successfully on many real world examples.

A *parameterized problem* is a language $L \subseteq \Sigma^* \times \Sigma^*$ over some finite alphabet Σ. The *instances* of L are tuples $\langle \mathbb{I}, k \rangle$, where k is called the *parameter*. The parameter is often a non-negative integer, but it can be anything, e.g. a rational number, three integers or a graph. For simplicity, we first assume the parameter is a non-negative integer, i.e. $L \subseteq \Sigma^* \times \mathbb{Z}_0$. A parameterized problem is *fixed-parameter tractable (fpt)* if there exists an algorithm that solves every instance $\langle \mathbb{I}, k \rangle$ of size $n = |\mathbb{I}|$ in time $f(k) \cdot n^c$ where f is an arbitrary computable function and c is a constant independent of both n and k. **FPT** is the class of all fixed-parameter tractable decision problems. In contrast to classical tractability, some exponentiality is allowed, but confined to the parameter only, thus better reflecting reality. For instance, the VERTEX COVER problem parameterized by the size of the vertex cover, k, is in **FPT**. Note that $1.2738^k + kn \leqslant 1.2738^k n + kn = (1.2738^k + k)n = f(k)n$.

Hardness for parameterized classes is proven in the usual way, but using fpt reductions instead of ordinary polynomial-time reductions. An *fpt reduction* from a parameterized language $L \subseteq \Sigma^* \times \mathbb{Z}_0$ to another parameterized language $L' \subseteq \Sigma^* \times \mathbb{Z}_0$ is a mapping $R : \Sigma^* \times \mathbb{Z}_0 \to \Sigma^* \times \mathbb{Z}_0$ such that: (1) $\langle \mathbb{I}, k \rangle \in L$ if and only if $\langle \mathbb{I}', k' \rangle = R(\mathbb{I}, k) \in L'$; (2) there is a computable function f and a constant c such that R can be computed in time $f(k) \cdot n^c$, where $n = |\mathbb{I}|$; and (3) there is a computable function g such that $k' \leqslant g(k)$.

Parameterized complexity offers a completeness theory, similar to the theory of **NP**-completeness. It contains the following hierarchy of parameterized complexity classes

$$\mathbf{FPT} \subseteq \mathbf{W}[1] \subseteq \mathbf{W}[2] \subseteq \mathbf{W}[3] \subseteq \cdots \subseteq \mathbf{W}[\mathbf{P}],$$

known as the *W hierarchy*. $\mathbf{W}[i]$ is defined using WEIGHTED CIRCUIT SATISFIABILITY problem: a parameterized problem is in **W**[i] if its every instance $\langle \mathbb{I}, k \rangle$ is fpt-reducible to a boolean circuit of weft i, such that $\langle \mathbb{I}, k \rangle$ has a solution if and only if the circuit has a truth assignment of weight (number

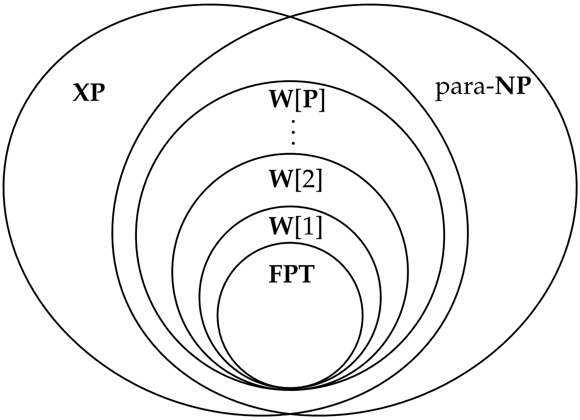

Figure 2.1: Parameterized Complexity Classes

of 1s in the input) at most k. Weft is the maximum number of gates with fan-in more than two on any path from the inputs to the output. The class **W**[**P**] corresponds to arbitrary circuits. For instance, the DOMINATING SET problem parameterized by the size of the dominating set is **W**[2]-complete [25]. There are also classes outside of the W-hierarchy; para-**NP** is the class of problems solvable in *non*-deterministic time $f(k)n^c$ and **XP** is the class of problems solvable in time $n^{f(k)}$; where f is an arbitrary computable function and c is a constant independent of both n and k. It is known that **W**[**P**] is a subset of both para-**NP** and **XP** [30]. Figure 2.1 shows the known relation between the mentioned parameterized complexity classes.

It is known that **P** $\subseteq$ **FPT** and **NP** $\subseteq$ para-**NP**, but otherwise there is not much known about the relations between parameterized complexity classes and the classical ones. For instance, there are **NP**-complete problems that are **W**[**P**]-complete and there are **PSPACE**-complete problems that are in **FPT**. The hardness of a parameterized problem depends a lot on the chosen parameter(s). For instance, consider the **NP**-complete problem GRAPH k-COLORING: this problem is para-**NP**-hard if parameterized by k, the number of available colors (mentioned in Paper 5 of this thesis) while it is in **FPT** if parameterized by n, the number of vertices (using a brute-force algorithm).

3 Automated Planning

1 What is Planning?

> *Planning is the art and practice of thinking before acting.*
>
> Patrik Haslum

In this section we give a high-level description of planning. Imagine a world or a system, like a car, a robotic spacecraft or a classroom with all the students and the teacher in it. There are some actions that change the state of the world. By executing the actions we can move between different states of the world. For instance, entering and leaving a student, giving a lecture or taking a break are all actions that change the state of the classroom. Planning may be defined as the problem of finding a sequence of such actions that lead to a specified goal state from a given initial state. A more abstract definition is given by Ghallab, Nau, and Traverso [34]:

> Planning is the reasoning side of acting. It is an abstract, explicit deliberation process that chooses and organizes actions by anticipating their expected outcomes. This deliberation aims at achieving as best as possible some pre-stated objectives. Automated planning is an area of Artificial Intelligence (AI) that studies this deliberation process computationally.

We will continue with an example, Blocks world, which is one of the most famous planning domains in Artificial Intelligence.

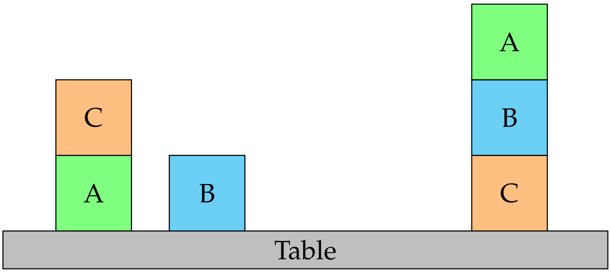

Figure 3.1: A Blocks world example; initial state (left) and goal state (right)

Example: Blocks world

In the Blocks world domain, we are given a set of identical blocks with different letter identifiers on them. The blocks are positioned in several stacks on a table. We have an arm to move the blocks. This arm can only pick blocks from the top of the stacks and put them on top of another stack or on the table. The goal is to reach another specified configuration. This version of Blocks world is known as the Elementary Blocks World (EBW). There are also other variations of this problem [38].

To define the state of the world, we use the following predicate types:

- **ON(A,B):** Block A is on block B.
- **ONTABLE(A):** Block A is on the table.
- **CLEAR(A):** Block A has nothing on it.
- **HOLDING(A):** The arm holds block A.
- **ARMEMPTY:** The arm holds nothing.

We also define the following actions:

- **UNSTACK(A,B):** Pick up clear block A from block B.
- **STACK(A,B):** Place block A using the arm onto clear block B.
- **PICKUP(A):** Lift clear block A with the empty arm from the table.
- **PUTDOWN(A):** Place the held block A onto a free space on the table.

Figure 3.1 shows an example with three blocks. In this example, the initial (I) and goal (G) states contain the following predicates

$$I = \{\textbf{CLEAR(C)}, \textbf{CLEAR(B)}, \textbf{ONTABLE(A)}, \textbf{ONTABLE(B)}, \textbf{ON(C,A)}, \textbf{ARMEMPTY}\}$$

$$G = \{\textbf{CLEAR(A)}, \textbf{ONTABLE(C)}, \textbf{ON(A,B)}, \textbf{ON(B,C)}, \textbf{ARMEMPTY}\}$$

$$\begin{aligned}
&\textbf{UNSTACK(A,B):} && \\
\textit{precondition list} &= \{\textbf{ON(A,B)}, \textbf{CLEAR(A)}, \textbf{ARMEMPTY}\} \\
\textit{add list} &= \{\textbf{CLEAR(B)}, \textbf{HOLDING(A)}\} \\
\textit{delete list} &= \{\textbf{ON(A,B)}, \textbf{ARMEMPTY}\} \\
&\textbf{STACK(A,B):} && \\
\textit{precondition list} &= \{\textbf{HOLDING(A)}, \textbf{CLEAR(B)}\} \\
\textit{add list} &= \{\textbf{ARMEMPTY}, \textbf{ON(A,B)}\} \\
\textit{delete list} &= \{\textbf{HOLDING(A)}, \textbf{CLEAR(B)}\} \\
&\textbf{PICKUP(A):} && \\
\textit{precondition list} &= \{\textbf{CLEAR(A)}, \textbf{ONTABLE(A)}, \textbf{ARMEMPTY}\} \\
\textit{add list} &= \{\textbf{HOLDING(A)}\} \\
\textit{delete list} &= \{\textbf{ONTABLE(A)}, \textbf{ARMEMPTY}\} \\
&\textbf{PUTDOWN(A):} && \\
\textit{precondition list} &= \{\textbf{HOLDING(A)}\} \\
\textit{add list} &= \{\textbf{ONTABLE(A)}, \textbf{ARMEMPTY}\} \\
\textit{delete list} &= \{\textbf{HOLDING(A)}\}
\end{aligned}$$

Figure 3.2: The actions of the Blocks world domain

Every action is defined by three lists of predicates; a *precondition list*, the list of predicates that need to hold for the action to be applicable in a state, an *add list*, the list of predicates that are added after the action is executed in a state, and a *delete list*, the list of predicates that are deleted after the execution of the action. The actions are defined in Figure 3.2.

The objective is to find the shortest sequence of actions (a plan) that bring the initial state (I) to the goal state (G). In this example, the unique optimal plan, ω, is the following list of actions:

$$\begin{aligned}\omega = \big(&\textbf{UNSTACK(C,A)}, \textbf{PUTDOWN(C)}, \textbf{PICKUP(B)}, \\ &\textbf{STACK(B,C)}, \textbf{PICKUP(A)}, \textbf{STACK(A,B)}\big)\end{aligned}$$

□

Given a Blocks world example with n blocks, one can produce in time $O(n)$ a plan (if one exists) that moves no block more that twice. However, the problem of finding the optimal plan is **NP**-hard [38].

Planning is closely connected to search. The state space of the blocks world example can be seen as a directed graph where states (combination of predicates) are the nodes and there is an edge from node u to node v if there is an action that transforms state u to state v. Hence, the problem of finding an optimal plan becomes the problem of finding the shortest path between two nodes. Although this may seem easy using shortest path algorithms, the main challenge is that the number of nodes of this graph is exponential in the input size. For instance, a blocks world instance with n blocks has more than 2^{n^2} states, which means even for $n = 10$ and checking 2 billion states per second, a linear search takes more than 10 billion years. However, the problem can easily be solved by a human in a few seconds.

2 Planning Model and Formalism

There are different ways to model the planning world. The differences come from how the following questions (motivated by Ghallab, Nau, and Traverso [34]) are answered:

- Is the planning process offline (static) or is it online (dynamic)?
- Is the state of the plan fully, partially or not observable?
- Is there only a single agent or are there several agents to execute actions?
- Do variables have discrete or continuous domain? Do they have finite or infinite domain?
- Are the actions deterministic, non-deterministic or stochastic?
- Do the actions have cost or duration?
- Can the actions be executed at the same time or do we only have sequential plans?
- Is the objective to satisfy a goal or to find the optimal plan under some criteria?

The most simple planning model is called **classical planning**. In classical planning the following assumptions are imposed:

- planning is static
- variable domains are discrete and finite
- actions are deterministic
- states are fully observable
- there is a single agent (sequential plan)
- actions have no cost or duration

There are many models and representations for planning in the literature [34]. Two common formalisms that are used to study the complexity of planning are STRIPS and SAS$^+$. This is because they are simple to work with and at the same time expressive enough to be able to represent many complex planning domains.

STRIPS

STRIPS (Stanford Research Institute Problem Solver) was defined by Fikes and Nilsson [29]. It is a planner and a formalism for describing planning instance that is based on first-order logic: the state of the world is viewed as a set of ground and function-free first-order literals. We will not go into the details, but merely note that the Blocks world example in this chapter is an example of a STRIPS planning instance. In many cases, one is content with the simplified variant of STRIPS where the state of the world is viewed as a set of propositional literals. This variant is referred to as propositional STRIPS.

SAS$^+$

SAS$^+$ (Simplified Action Structures) is very similar to propositional STRIPS, with the difference that the two-valued propositions are replaced by variables that take their values from a finite domain. It was introduced by Bäckström and Klein [10]. Another difference is that instead of predicates the domain values are assignments to variables in preconditions, effects, initial and goal state.

We have mostly used SAS$^+$ formalism in the papers of this thesis and will formally define it here (Although depending on the application, SAS$^+$ is defined in slightly different ways in the papers of this Thesis.). A SAS$^+$ planning instance is a 4-tuple $\langle V, A, I, G\rangle$, where

- $V = \{v_1, v_2, \ldots, v_n\}$ is a set of variables over some finite domain $\mathcal{D}$ (in some definitions each variable v has an individual domain $D(v)$). Define $\mathcal{D}_+ = \mathcal{D} \cup \{\mathbf{u}\}$, where $\mathbf{u}$ is an extra value that stands for undefined. $\mathcal{D}^n$ is the set of *total states* where $\mathcal{D}_+^n$ is the set of *partial states*. The value of variable v in state s (partial or total) is denoted by $s[v]$.

- A is a set of actions and each action $a \in A$ is of the form $a : \text{pre}(a) \rightarrow \text{eff}(a)$, where $\text{pre}(a), \text{eff}(a) \in \mathcal{D}_+^n$ and are called precondition and effects (or post-conditions), respectively. An action a is applicable in total state s if and only if every value with a defined value ($\neq \mathbf{u}$) in $\text{pre}(a)$ has the same value in s. If a is applicable in s, the result of a in s is a total state t where $t[v] = s[v]$ if $\text{eff}(a)[v]$ is undefined and $t[v] = \text{eff}(a)[v]$ otherwise.

- $I \in \mathcal{D}^n$ is the initial state.

- $G \in \mathcal{D}_+^n$ is the goal state.

For two total states s_I and s_G, a sequence of actions $(a_1, a_2, \ldots, a_k)$ is called a *plan* from s_I to s_G if and only if there exist total states $\{s_0, s_1, \ldots, s_k\}$ such that $s_0 = s_I$, $s_k = s_G$, a_i is applicable in s_{i-1} and s_i is the result of a_i in s_{i-1}. A plan from I to a total state s is a *solution* plan if $s[v] = G[v]$ for every defined variable in G.

3 Planning Problems

There are many planning problems described in the literature, with respect to different planning models, formalisms, constraints and objectives. In this thesis we investigate the following problems:

PSAT **P**lan **sat**isfiability problem (also known as plan existence problem (PE)) is the most basic planning problem. It asks if a planning instance (e.g. a SAS$^+$ instance) has a solution.

LOP **L**ength-**o**ptimal **p**lanning takes an integer ℓ and asks if the planning instance has a solution of length at most ℓ.

COP **C**ost-**o**ptimal **p**lanning is a generalization of LOP. It assigns a cost to each action ($c : A \to \mathbb{D}$), takes a cost threshold w and asks if there is plan of cost at most w.

PSP **P**artial-**s**atisfaction **p**roblem is a generalization of PSAT. It takes an integer k in the input and asks if there is a plan that satisfies at least k of the goals.

NBP **N**et-**b**enefit **p**lanning assigns a utility to each goal ($u : G \to \mathbb{D}$) and a cost to each action. It takes a net-benefit threshold b and asks if there is a plan with net-benefit at least b. The net-benefit of a plan is the sum of utilities of the satisfied goals minus the plan cost.

We also investigate some problems beyond the boundaries of classical planning. To continue, we first need to define *partial-order* and *parallel* plans. For a more detailed explanation please refer to Paper 5.

Let Π be a planning problem instance. A partial-order plan for Π is a tuple $P = \langle A, \prec \rangle$, where A is a set of action occurrences and $\prec$ is a partial order on A. P is a valid plan for Π if all linearizations of it are valid plans. The size of an order is the number of relation tuples in it. A parallel plan is a tuple $P = \langle A, \prec, \# \rangle$ where $\langle A, \prec \rangle$ is a partial-order plan and # is an irreflexive and symmetric relation on A. The relation # is called the *non-concurrency* relation. For every two actions a and b, $a\#b$ indicates that they cannot be executed in parallel. Let $d : A \mapsto \mathbb{N}$ denote the duration of each action. A *parallel execution* of a parallel plan P is a function $r : A \mapsto \mathbb{N}$, denoting the release time for every action in A, such that: 1) If $a \prec b$ then $r(a) + d(a) \leqslant r(b)$, and 2) If $a\#b$ then either $r(a) + d(a) \leqslant r(b)$ or $r(b) + d(b) \leqslant r(a)$. The length of a parallel execution or $\max_{a \in A}\{r(a) + d(a)\}$ denotes the latest finishing time of any action. A *k-processor parallel execution* of a parallel plan is a parallel execution such that for each time $t \in \mathbb{N}$ there are at most k actions a such that $r(a) \leqslant t < r(a) + d(a)$.

PPL **P**arallel **p**lan **l**ength takes a parallel plan $P = \langle A, \prec, \# \rangle$, a length threshold ℓ and asks if there is a parallel execution for P of length at most ℓ.

PPL$_k$ This problem takes a parallel plan $P = \langle A, \prec, \# \rangle$, an integer k, a length threshold ℓ and asks if there is a k-processor parallel execution for P of length at most ℓ.

MCR **M**inimum **c**onstrained **r**eordering (MCR), takes a partial-order plan, a value k and asks if there is a another valid order for the plan with size k.

MCD **M**inimum **c**onstrained **d**eordering (MCD) is similar to MCR but requires the new order to be a subset of the initial order.

MPR **M**inimum **p**arallel **r**eordering asks if a partial-order plan has a reordering of parallel length ℓ.

MPD **M**inimum **p**arallel **d**eordering is the deordering version of MPR.

Complexity of Planning

Planning problems are known to be computationally hard in the general case. PSAT for propositional STRIPS and SAS$^+$ is **PSPACE**-complete [11, 18] and for STRIPS with predicates over infinite domains is undecidable [21], i.e. it is impossible to construct an algorithm that always generates a correct "yes" or "no" answer.

One method to reduce the computational complexity of planning is to apply restrictions on the planning instance. The complexity of STRIPS planning varies from constant time to undecidable depending on which restrictions we make. These restrictions can be syntactic restrictions, such as restrictions on the number of preconditions and effects [2, 9, 18], or restrictions on action types [7, 10, 11, 36, 49]; semantic restrictions, such as restrictions on the variable dependencies [42, 43, 46, 48, 49].

Finding new tractable problems in planning has received a special attention not only because it helps to distinguish between easy and hard planning instances, but also because one can use the polynomial-time solvable instances in order to construct better heuristic functions and improve planners [42, 43]. Many tractable classes of planning problems have been identified [3, 11, 15, 36, 49, 52].

A recent method to tackle the hardness of planning problems is to get a more fine-grained complexity analysis by using parameterized complexity. Downey, Fellows, and Stege [26] showed that STRIPS planning is **W**[1]-hard parameterized by the plan length (Later, Bäckström, Jonsson, Ordyniak, and Szeider [9] strengthened this result and showed that LOP parameterized by the plan length is **W**[2]-complete). Since then, the parameterized complexity of PSAT, LOP, COP and NBP has been studied under various restrictions and different parameterizations [1, 6, 9, 12, 39, 56].

There are also other complexity results on variants of planning that are out of the scope of this thesis [13, 19, 28, 37, 41, 57, 58, 67].

4 Background and Contributions

This section highlights the background and contributions of each paper in this thesis.

1 Paper 1: Oversubscription Planning

Meysam Aghighi and Peter Jonsson
Oversubscription planning: Complexity and compilability
In *Proceedings of the Twenty-Eighth **AAAI** Conference on Artificial Intelligence, Québec City, Québec, Canada, July 27 -31, 2014.*

In classical planning all goals need to be satisfied at the same time. However, in many real-world applications there are a larger number of possible goals and the aim is to find a feasible subset of them [71]. This type of planning is called partial satisfaction or oversubscription planning. Several custom-made planners and heuristics have been suggested for oversubscription planning [14, 17, 61].

In this paper we study two different problems: the partial satisfaction problem (PSP) where the goal is to maximize the number of achieved goals, and the net-benefit problem (NBP) where the goals is to maximize the total weight of the achieved goals minus the plan cost. The net-benefit problem was first formalized by Briel, Nigenda, Do, and Kambhampati [17] and was shown to be **PSPACE**-complete. After that, there has not been a more detailed complexity analysis of this problem until paper 1 of this thesis. We consider two types of restrictions in our complexity analysis: (1) restrictions on the number of preconditions and effects that were first proposed by Bylander [18], and (2) the PUBS restrictions proposed by Bäckström and Klein [10].

The results indicate that there is a high correlation between PSAT, COP, PSP and NBP. For instance, if PSAT for some set of instances X (under mild additional assumptions) is **NP**-complete, then PSP for X is **NP**-complete, too. We also propose a polynomial-time reduction from NBP into PSAT, using a novel counter to keep track of plan cost during the plan execution and calculate the net-benefit after plan execution. The importance of this compilation is that the number of variables increase slowly in the size of the original instance, and this means that the algorithms for PSAT are able to solve NBP instances with a limited slow-down. Compiling different planning problems into each other [55, 64, 72] and into other problems [16, 20, 54] have been active research areas for quite some time.

At the end of the paper we suggested a continuation of the work by imposing restrictions on the causal graph. Later, Katz and Mirkis [53] discovered several tractable fragments of net-benefit planning by restricting the causal graph.

2 Paper 2: Causal Graphs for Tractability

Meysam Aghighi, Peter Jonsson, and Simon Ståhlberg
Tractable cost-optimal planning over restricted polytree causal graphs
In *Proceedings of the Twenty-Ninth **AAAI** Conference on Artificial Intelligence, Austin, Texas, USA, January 25-30, 2015.*

A causal graph is a directed graph with the domain variables as nodes. Node x in the graph is connected to node y if there exists some action a where $\mathrm{eff}(a)[y] \neq \mathbf{u}$ and either $\mathrm{eff}(a)[x] \neq \mathbf{u}$ or $\mathrm{pre}(a)[x] \neq \mathbf{u}$. There has been many studies on the causal graph of a planning problem [15, 36, 51]. Giménez and Jonsson [35] showed that plan generation is **NP**-hard for instances whose causal graph is a directed path and have domain size at least 5. Katz and Domshlak [48, 50] have studied planning instances whose causal graphs are forks and inverted-forks. Katz and Keyder [52] have shown that cost-optimal planning instances with bounded domain size and hourglass causal graphs are in **P**. In this paper, we generalize this result and prove that cost-optimal planning instances with bounded domain size and polytree causal graphs with bounded diameter, are in **P**.

The algorithm in this paper is built on a novel notion of isomorphic variables. Intuitively, two variables are isomorphic if the structure of their domain transition graphs are isomorphic with regard to the initial value, goal value and actions. This results in the same behavior of these variables in every optimal solution, which allows us to combine them into a single variable. By repeating this process, we gradually reduce the size of the initial instance and then solve it in polynomial time.

Brafman and Domshlak [15] have presented a polynomial-time algorithm for instances with binary variables and where the causal graph is a polytree such that the in-degree of every vertex is bounded by a constant. Giménez and Jonsson [36] later generalized this result by giving an algorithm that, instead of restricting the in-degree, restricts the number of prevail-conditions (a bound on the in-degree implies a bound on the number of prevail-conditions but not the other way round). Our algorithm works differently: it allows not only binary domains, but any domain size bounded by a constant and it allows the actions to have an unbounded number of prevail-conditions. However, this flexibility comes at the cost of bounding the diameter by a constant.

3 Papers 3, 4: The Effect of Cost Domain in Cost-optimal and Net-benefit Planning

Meysam Aghighi and Christer Bäckström
Cost-optimal and net-benefit planning - A parameterised complexity view
In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, **IJCAI** 2015, Buenos Aires, Argentina, July 25-31, 2015.*

Meysam Aghighi and Christer Bäckström
A multi-parameter complexity analysis of cost-optimal and net-benefit planning
In *Proceedings of the Twenty-Sixth International Conference on Automated Planning and Scheduling, **ICAPS** 2016, London, UK, June 12-17, 2016.*

Although cost-optimal and net-benefit planning have been vastly studied in the literature, very little attention has been paid to motivate and discuss the choice of numeric domain for the action costs. For instance, Katz and Domshlak [49] and Helmert, Haslum, Hoffmann, and Nissim [44] use non-negative reals; Bäckström and Jonsson [8] and Yang, Culberson, Holte, Zahavi, and Felner [77] use non-negative integers; Cooper, Roquemaurel, and Régnier [24] use positive integers; Coles, Fox, Long, and Smith [23] use non-negative costs with no specified type; and Thayer, Stern, Felner, and Ruml [73] disregard cost types.

A traditional computational complexity analysis does not differentiate between different cost types; planning remains **PSPACE**-complete regardless of whether action costs are positive or non-negative, integer or rational. In paper 3, we use parameterized complexity to draw a line between different cost domains: cost-optimal planning is **W**[2]-complete for positive integer ($\mathbb{Z}_+$) costs while it is para-**NP**-hard if the costs are non-negative integers ($\mathbb{Z}_0$) or positive rationals ($\mathbb{Q}_+$). This is a strong indication that the latter cases are substantially harder. The paper continues by providing a detailed complexity map of both cost-optimal and net-benefit planning problems, with PUBS restrictions and restrictions on the number of preconditions and effects.

This result may seem in contradiction with the usual assumption that the use of rational costs is the same as integer costs: by multiplying all action costs and plan cost threshold by a sufficiently large number. By using this transformation, from any rational-cost planning instance we can get an integer-cost instance and solve it by using an integer-cost planner. Although this method is correct and the new instance with integer costs has a solution if and only if the initial rational-cost instance has a solution, the transformation does not preserve the parameterized complexity (it is not an fpt-reduction).

The work is continued by a multi-parameter complexity analysis of the same problems in paper 4. In this work we consider a total of thirteen parameters including the plan length, ℓ, the plan cost, k, the largest denominator of the action costs, d, and the sum of variable utilities, t. The latter parameter only applies to NBP. Two new cost domains were also considered in this paper, the non-negative rationals ($\mathbb{Q}_0$) and rationals greater than or equal to one ($\mathbb{Q}_1$). Our results show that COP remains **W**[2]-hard for all cost domains even when combining all parameters. We also show the **W**[2]-membership of COP for the following parameter combinations and domains:

- $\{k, \ell\}$ for $\mathbb{Z}_0$
- $\{k, d\}$ for $\mathbb{Q}_+$
- $\{k, d, \ell\}$ for $\mathbb{Q}_0$

Another interesting and important result of paper 4 is the effect of parameter t, the sum of all utilities. We show, through an fpt reduction, that NBP parameterized by t is reducible to COP parameterized by k over the same numeric domain. This reduction is both related to the method used by Keyder and Geffner [55] to compile away soft goals, and to the compilation of NBP to PSAT by Aghighi and Jonsson [2] in paper 1 of this thesis. However, if we use the second method and apply the counter reduction from paper 1, we have to limit ourselves only to integer costs.

4 Paper 5: Plan Optimization

Meysam Aghighi and Christer Bäckström
Plan reordering and parallel execution – a parameterized complexity view
In *Proceedings of the Thirty-First **AAAI** Conference on Artificial Intelligence, San Francisco, California, USA, February 4-9, 2017.*

Some real-world applications require a more flexible structure than a sequential plan. For instance the applications that involve real-time execution or multi-agent taskability [75, 76]. One method to model this flexibility is to use partial-order plans. This allows not only different ways to linearize a plan, but also the possibility to postpone making decisions for linearization to run-time. Another important feature of a partial-order plan is its ability to be used as a parallel plan, where mutually unordered actions can be executed in parallel. Some planners directly produce a partial-order plan. At the same time, there has been attempts to produce a partial-order plan from the output of a total-order planner. The core idea is to remove the unnecessary ordering relations and several algorithms have been proposed to do this, both in the early 1990s and recently [47, 62, 63, 66, 74]. These algorithms have been applied to various planning domains including workflow planning [68], automatic generation of narratives [40], composition of web services [59], plan execution and monitoring [62], plan repair [69], and distributed control systems [65].

Bäckström [5] theoretically approached the problem and defined deordering and reordering of a partial-order plan. Deordering of a plan is to replace the plan order by a subset of the order such that the new plan remains a valid partial-order plan. Reordering is to replace the plan order with any other order such that the result is a valid partial-order plan. He showed that it is **NP**-complete to optimize the size (number of relation tuples) of both deordering and reordering. There can be other criteria for plan optimization as well. One example is the length of the parallel execution of the plan. Bäckström [5] has also looked into this problem and showed that both finding the shortest parallel execution of a parallel plan (PPL) and optimization of the parallel length by reordering (MPR) are **NP**-complete problems.

In paper 5 of this thesis, we initiate a parameterized complexity investigation of the problems proposed by Bäckström [5] in order to shed more light on them and get more refined results. Two important parameters of a partial-order are its width (the size of the largest anti-chain) and height (the length of the longest chain). We study the effect of these parameters on the parameterized complexity of finding the optimal deordering (MCD) and reordering (MCR). We show that both MCD and MCR are **W**[2]-hard and in **W**[P] if parameterized by the desired order size, and MCD is in **FPT** if parameterized by the original order size.

We further show that the length of the parallel execution of a plan is highly related to both the height of the partial order and the chromatic number of the non-concurrency graph. Moreover, we prove that finding the minimum parallel execution of a plan is in **FPT** if parameterized by the size of the non-concurrency relation.

Finally, we propose a more realistic criterion for plan optimization: the length of the k-processor parallel execution. We study this criterion through the problem PPL$_k$. We prove that PPL$_k$ is in **FPT** if parameterized by the combination of: (1) the size of the non-concurrency relation, (2) the length of the k-processor parallel execution and (3) the number of processor respectively. On the other hand, the problem remains para-**NP**-hard if only parameterized by $\{n_\#, l_p\}$; i.e. significantly harder than the case with unlimited number of processors (PPL). This is consistent with the intuition that the bound on the number of processors makes the problem much harder.

References

[1] Meysam Aghighi and Christer Bäckström. "Cost-Optimal and Net-Benefit Planning - A Parameterised Complexity View". In: *Proc. 24th Int'l Joint Conf. Artif. Intell. (IJCAI-15), Buenos Aires, Argentina*. 2015, pp. 1487–1493.

[2] Meysam Aghighi and Peter Jonsson. "Oversubscription Planning: Complexity and Compilability". In: *Proc. 28th AAAI Conf. Artif. Intell. (AAAI-14), Québec City, QC, Canada*. 2014, pp. 2221–2227.

[3] Meysam Aghighi, Peter Jonsson, and Simon Ståhlberg. "Tractable Cost-Optimal Planning over Restricted Polytree Causal Graphs". In: *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA*. 2015, pp. 3225–3231.

[4] Sanjeev Arora and Boaz Barak. *Computational complexity: a modern approach*. Cambridge University Press, 2009.

[5] Christer Bäckström. "Computational Aspects of Reordering Plans". In: *J. Artif. Intell. Res.* 9 (1998), pp. 99–137.

[6] Christer Bäckström. "Parameterising the Complexity of Planning by the Number of Paths in the Domain-transition Graphs". In: *ECAI 2014 - 21st European Conference on Artificial Intelligence, 18-22 August 2014, Prague, Czech Republic - Including Prestigious Applications of Intelligent Systems (PAIS 2014)*. 2014, pp. 33–38.

[7] Christer Bäckström, Yue Chen, Peter Jonsson, Sebastian Ordyniak, and Stefan Szeider. "The Complexity of Planning Revisited—A Parameterized Analysis". In: *Proc. of the 26th AAAI Conference on Artificial Intelligence (AAAI-2012)*. 2012.

[8] Christer Bäckström and Peter Jonsson. "Bridging the Gap Between Refinement and Heuristics in Abstraction". In: *Proc. 23rd Int'l Joint Conf. Artif. Intell. (IJCAI-13), Beijing, China*. 2013, pp. 2261–2267.

[9] Christer Bäckström, Peter Jonsson, Sebastian Ordyniak, and Stefan Szeider. "A complete parameterized complexity analysis of bounded planning". In: *J. Comput. Syst. Sci.* 81.7 (2015), pp. 1311–1332.

[10] Christer Bäckström and Inger Klein. "Planning in polynomial time: the SAS-PUBS class". In: *Computational Intelligence* 7.3 (1991), pp. 181–197.

[11] Christer Bäckström and Bernhard Nebel. "Complexity Results for SAS+ Planning". In: *Computational Intelligence* 11.4 (1995), pp. 625–655.

[12] Christer Bäckström. "Some Fixed Parameter Tractability Results for Planning with Non-Acyclic Domain-Transition Graphs". In: *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA.* 2015, pp. 3232–3238.

[13] Chitta Baral, Vladik Kreinovich, and Raúl A Trejo. "Computational complexity of planning with temporal goals". In: (2001).

[14] J. Benton, M. Do, and S. Kambhampati. "Anytime heuristic search for partial satisfaction planning". In: *Artif. Intell.* 173.5-6 (2009), pp. 562–592.

[15] Ronen I. Brafman and Carmel Domshlak. "Structure and complexity in planning with unary operators". In: *Journal of Artificial Intelligence Research (JAIR)* (2003), pp. 315–349.

[16] Menkes van den Briel and Subbarao Kambhampati. "Optiplan: Unifying IP-based and Graph-based Planning". In: *J. Artif. Intell. Res.* 24 (2005), pp. 919–931.

[17] Menkes van den Briel, Romeo Sanchez Nigenda, Minh Binh Do, and Subbarao Kambhampati. "Effective Approaches for Partial Satisfaction (Over-Subscription) Planning". In: *Proc. 19th National Conference on Artificial Intelligence (AAAI-2004)*. 2004, pp. 562–569.

[18] Tom Bylander. "The computational complexity of propositional STRIPS planning". In: *Artificial Intelligence* 69.1 (1994), pp. 165–204.

[19] John Canny. *The complexity of robot motion planning*. MIT press, 1988.

[20] Michael Cashmore, Maria Fox, and Enrico Giunchiglia. "Planning as Quantified Boolean Formula". In: *Proc. 20th European Conference on Artificial Intelligence (ECAI-2012)*. 2012, pp. 217–222.

[21] David Chapman. "Planning for conjunctive goals". In: *Artificial intelligence* 32.3 (1987), pp. 333–377.

[22] Jianer Chen, Iyad A. Kanj, and Ge Xia. "Improved Parameterized Upper Bounds for Vertex Cover". In: *Mathematical Foundations of Computer Science 2006, 31st International Symposium, MFCS 2006, Stará Lesná, Slovakia, August 28-September 1, 2006, Proceedings*. 2006, pp. 238–249.

[23] Andrew Coles, Maria Fox, Derek Long, and Amanda Smith. "Additive-Disjunctive Heuristics for Optimal Planning". In: *Proc. 18th Int'l Conf. Aut. Planning and Scheduling (ICAPS-08), Sydney, Australia*. 2008, pp. 44–51.

[24] Martin C. Cooper, Marie de Roquemaurel, and Pierre Régnier. "A weighted CSP approach to cost-optimal planning". In: *AI Commun.* 24.1 (2011), pp. 1–29.

[25] Rod G Downey and Michael R Fellows. *Fixed-parameter tractability and completeness*. Cornell University, Mathematical Sciences Institute, 1992.

[26] Rodney G. Downey, Michael R. Fellows, and Ulrike Stege. "Parameterized complexity: A framework for systematically confronting computational intractability". In: *Contemporary Trends in Discrete Mathematics: From DIMACS and DIMATIA to the Future, Proceedings of a DIMACS Workshop, Stirín Castle, Czech Republic, May 19-25, 1997*. 1997, pp. 49–100.

[27] Rodney G Downey and Michael Ralph Fellows. *Parameterized complexity*. Springer Science & Business Media, 2012.

[28] Kutluhan Erol, Dana S Nau, and Venkatramana S Subrahmanian. "Complexity, decidability and undecidability results for domain-independent planning". In: *Artificial intelligence* 76.1 (1995), pp. 75–88.

[29] Richard Fikes and Nils J. Nilsson. "STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving". In: *Proceedings of the 2nd International Joint Conference on Artificial Intelligence. London, UK, September 1-3, 1971.* 1971, pp. 608–620.

[30] Jörg Flum and Martin Grohe. *Parameterized Complexity Theory, volume XIV of Texts in Theoretical Computer Science. An EATCS Series*. 2006.

[31] Michael R Garey and David S Johnson. *Computers and intractability*. Vol. 174. Freeman New York, 1979.

[32] Hector Geffner and Blai Bonet. *A Concise Introduction to Models and Methods for Automated Planning*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, 2013.

[33] Malik Ghallab, Dana Nau, and Paolo Traverso. *Automated Planning and Acting*. Cambridge University Press, 2016.

[34] Malik Ghallab, Dana Nau, and Paolo Traverso. *Automated planning: theory & practice*. Elsevier, 2004.

[35] Omer Giménez and Anders Jonsson. "Planning over Chain Causal Graphs for Variables with Domains of Size 5 Is NP-Hard". In: *Journal of Artificial Intelligence Research (JAIR)* (2009), pp. 675–706.

[36] Omer Giménez and Anders Jonsson. "The influence of k-dependence on the complexity of planning". In: *Artificial Intelligence* 177-179 (2012), pp. 25–45.

[37] Judy Goldsmith, Jérôme Lang, Miroslaw Truszczynski, and Nic Wilson. "The computational complexity of dominance and consistency in CP-nets". In: *Journal of Artificial Intelligence Research* 33 (2008), pp. 403–432.

[38] Naresh Gupta and Dana S. Nau. "On the Complexity of Blocks-World Planning". In: *Artif. Intell.* 56.2-3 (1992), pp. 223–254.

[39] Ronald de Haan, Martin Kronegger, and Andreas Pfandler. "Fixed-Parameter Tractable Reductions to SAT for Planning". In: *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*. 2015, pp. 2897–2903.

[40] Patrik Haslum. "Narrative Planning: Compilations to Classical Planning". In: *J. Artif. Intell. Res.* 44 (2012), pp. 383–395.

[41] Patrik Haslum and Peter Jonsson. "Some Results on the Complexity of Planning with Incomplete Information". In: *Recent Advances in AI Planning, 5th European Conference on Planning, ECP'99, Durham, UK, September 8-10, 1999, Proceedings*. 1999, pp. 308–318.

[42] Malte Helmert. "A Planning Heuristic Based on Causal Graph Analysis". In: *14th International Conference on Automated Planning & Scheduling (ICAPS)*. 2004, pp. 161–170.

[43] Malte Helmert, Patrik Haslum, and Jörg Hoffmann. "Flexible abstraction heuristics for optimal sequential planning". In: *17th International Conference on Automated Planning & Scheduling (ICAPS)*. 2007, pp. 176–183.

[44] Malte Helmert, Patrik Haslum, Jörg Hoffmann, and Raz Nissim. "Merge-and-Shrink Abstraction: A Method for Generating Lower Bounds in Factored State Spaces". In: *J. ACM* 61.3 (2014), 16:1–63.

[45] John E. Hopcroft and Jeffrey D. Ullman. *Introduction to Automata Theory, Languages and Computation*. Addison-Wesley, 1979.

[46] Peter Jonsson and Christer Bäckström. "State-Variable Planning Under Structural Restrictions: Algorithms and Complexity". In: *Artif. Intell.* 100.1-2 (1998), pp. 125–176.

[47] Subbarao Kambhampati and Smadar Kedar. "A Unified Framework for Explanation-Based Generalization of Partially Ordered and Partially Instantiated Plans". In: *Artif. Intell.* 67.1 (1994), pp. 29–70.

[48] Michael Katz and Carmel Domshlak. "Implicit Abstraction Heuristics". In: *Journal of Artificial Intelligence Research* 39 (2010), pp. 51–126.

[49] Michael Katz and Carmel Domshlak. "New Islands of Tractability of Cost-Optimal Planning". In: *J. Artif. Intell. Res.* 32 (2008), pp. 203–288.

[50] Michael Katz and Carmel Domshlak. "Structural Patterns Heuristics via Fork Decomposition". In: *18th International Conference on Automated Planning & Scheduling (ICAPS)*. 2008, pp. 182–189.

[51] Michael Katz and Carmel Domshlak. "Structural Patterns of Tractable Sequentially-Optimal Planning". In: *17th International Conference on Automated Planning & Scheduling (ICAPS)*. 2007, pp. 200–207.

[52] Michael Katz and Emil Keyder. "Structural Patterns Beyond Forks: Extending the Complexity Boundaries of Classical Planning". In: *AAAI Conference on Artificial Intelligence* (2012).

[53] Michael Katz and Vitaly Mirkis. "In Search of Tractability for Partial Satisfaction Planning". In: *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*. 2016, pp. 3154–3160.

[54] Henry A. Kautz and Bart Selman. "Planning as Satisfiability". In: *Proc. 10th European Conference on Artificial Intelligence (ECAI-2012)*. 1992, pp. 359–363.

[55] Emil Keyder and Hector Geffner. "Soft Goals Can Be Compiled Away". In: *J. Artif. Intell. Res.* 36 (2009), pp. 547–556.

[56] Martin Kronegger, Andreas Pfandler, and Reinhard Pichler. "Parameterized Complexity of Optimal Planning: A Detailed Map". In: *Proc. 23rd Int'l Joint Conf. Artif. Intell. (IJCAI-13), Beijing, China*. 2013, pp. 954–961.

[57] Michael L. Littman. "Probabilistic Propositional Planning: Representations and Complexity". In: *Proceedings of the Fourteenth National Conference on Artificial Intelligence and Ninth Innovative Applications of Artificial Intelligence Conference, AAAI 97, IAAI 97, July 27-31, 1997, Providence, Rhode Island.* 1997, pp. 748–754.

[58] Michael L Littman, Judy Goldsmith, and Martin Mundhenk. "The computational complexity of probabilistic planning". In: *Journal of Artificial Intelligence Research* 9.1 (1998), pp. 1–36.

[59] Therani Madhusudan and Naveen Uttamsingh. "A declarative approach to composing web services in dynamic environments". In: *Decision Support Systems* 41.2 (2006), pp. 325–357.

[60] Fabio Massacci and Laura Marraro. "Logical Cryptanalysis as a SAT Problem". In: *J. Autom. Reasoning* 24.1/2 (2000), pp. 165–203.

[61] V. Mirkis and C. Domshlak. "Abstractions for Oversubscription Planning". In: *Proc. 23rd International Conference on Automated Planning and Scheduling (ICAPS-2013)*. 2013.

[62] Christian J. Muise, Sheila A. McIlraith, and J. Christopher Beck. "Monitoring the Execution of Partial-Order Plans via Regression". In: *Proc. 22nd Int'l Joint Conf. Artif. Intell. (IJCAI 2015), Barcelona, Catalonia, Spain*. 2011, pp. 1975–1982.

[63] Christian J. Muise, Sheila A. McIlraith, and J. Christopher Beck. "Optimally Relaxing Partial-Order Plans with MaxSAT". In: *Proc. 22nd Int'l Conf. Automated Planning and Scheduling, (ICAPS 2012), Atibaia, São Paulo, Brazil*. 2012.

[64] Héctor Palacios and Hector Geffner. "Compiling Uncertainty Away in Conformant Planning Problems with Bounded Width". In: *J. Artif. Intell. Res.* 35 (2009), pp. 623–675.

[65] Jürgen Pannek. "Parallelizing a state exchange strategy for noncooperative distributed NMPC". In: *Systems & Control Letters* 62.1 (2013), pp. 29–36.

[66] Pierre Régnier and Bernard Fade. "Complete Determination of Parallel Actions and Temporal Optimization in Linear Plans of Action". In: *Proc. European Workshop on Planning, (EWSP '91), Sankt Augustin, FRG.* 1991, pp. 100–111.

[67] Jussi Rintanen. "Complexity of Planning with Partial Observability". In: *Proceedings of the Fourteenth International Conference on Automated Planning and Scheduling (ICAPS 2004), June 3-7 2004, Whistler, British Columbia, Canada*. 2004, pp. 345–354.

[68] María Dolores Rodríguez-Moreno, Daniel Borrajo, Amedeo Cesta, and Angelo Oddi. "Integrating planning and scheduling in workflow domains". In: *Expert Syst. Appl.* 33.2 (2007), pp. 389–406.

[69] Enrico Scala and Pietro Torasso. "Deordering and Numeric Macro Actions for Plan Repair". In: *Proc. 24th Int'l Joint Conf. Arfif. Intell. (IJCAI-2015), Buenos Aires, Argentina*. 2015, pp. 1673–1681.

[70] Michael Sipser. *Introduction to the Theory of Computation*. Vol. 2. Thomson Course Technology Boston, 2006.

[71] D. Smith. "Choosing Objectives in Over-Subscription Planning". In: *Proc. 14th International Conference on Automated Planning and Scheduling (ICAPS-2004)*. 2004, pp. 393–401.

[72] Ran Taig and Ronen I. Brafman. "Compiling Conformant Probabilistic Planning Problems into Classical Planning". In: *Proc. 23rd International Conference on Automated Planning and Scheduling (ICAPS-2013)*. 2013.

[73] Jordan Tyler Thayer, Roni Stern, Ariel Felner, and Wheeler Ruml. "Faster Bounded-Cost Search Using Inadmissible Estimates". In: *Proc. 22nd Int'l Conf. Aut. Planning and Scheduling (ICAPS-12), Atibaia, São Paulo, Brazil*. 2012, pp. 270–278.

[74] Manuela Veloso, Alicia Pérez, and Jaime Carbonell. "Nonlinear Planning with Parallel Resource Allocation". In: *Workshop on Innovative Approches to Planning, Scheduling and Control, San Diego, CA, USA, 1990*. 1990, pp. 207–212.

[75] Manuela M. Veloso, Martha E. Pollack, and Michael T. Cox. "Rationale-Based Monitoring for Planning in Dynamic Environments". In: *Proceedings of the Fourth International Conference on Artificial Intelligence Planning Systems, Pittsburgh, Pennsylvania, USA, 1998*. 1998, pp. 171–180.

[76] Daniel S. Weld. "An Introduction to Least Commitment Planning". In: *AI Magazine* 15.4 (1994), pp. 27–61.

[77] Fan Yang, Joseph C. Culberson, Robert Holte, Uzi Zahavi, and Ariel Felner. “A General Theory of Additive State Space Abstractions”. In: *J. Artif. Intell. Res.* 32 (2008), pp. 631–662.

Papers

Department of Computer and Information Science
Linköpings universitet

Dissertations

Linköping Studies in Science and Technology
Linköping Studies in Arts and Science
Linköping Studies in Statistics
Linköping Studies in Information Science

Linköping Studies in Science and Technology

No 14 **Anders Haraldsson:** A Program Manipulation System Based on Partial Evaluation, 1977, ISBN 91-7372-144-1.

No 17 **Bengt Magnhagen:** Probability Based Verification of Time Margins in Digital Designs, 1977, ISBN 91-7372-157-3.

No 18 **Mats Cedwall**: Semantisk analys av processbeskrivningar i naturligt språk, 1977, ISBN 91- 7372-168-9.

No 22 **Jaak Urmi:** A Machine Independent LISP Compiler and its Implications for Ideal Hardware, 1978, ISBN 91-7372-188-3.

No 33 **Tore Risch:** Compilation of Multiple File Queries in a Meta-Database System 1978, ISBN 91- 7372-232-4.

No 51 **Erland Jungert:** Synthesizing Database Structures from a User Oriented Data Model, 1980, ISBN 91-7372-387-8.

No 54 **Sture Hägglund:** Contributions to the Development of Methods and Tools for Interactive Design of Applications Software, 1980, ISBN 91-7372-404-1.

No 55 **Pär Emanuelson:** Performance Enhancement in a Well-Structured Pattern Matcher through Partial Evaluation, 1980, ISBN 91-7372-403-3.

No 58 **Bengt Johnsson, Bertil Andersson:** The Human-Computer Interface in Commercial Systems, 1981, ISBN 91-7372-414-9.

No 69 **H. Jan Komorowski:** A Specification of an Abstract Prolog Machine and its Application to Partial Evaluation, 1981, ISBN 91-7372-479-3.

No 71 **René Reboh:** Knowledge Engineering Techniques and Tools for Expert Systems, 1981, ISBN 91-7372-489-0.

No 77 **Östen Oskarsson:** Mechanisms of Modifiability in large Software Systems, 1982, ISBN 91- 7372-527-7.

No 94 **Hans Lunell:** Code Generator Writing Systems, 1983, ISBN 91-7372-652-4.

No 97 **Andrzej Lingas:** Advances in Minimum Weight Triangulation, 1983, ISBN 91-7372-660-5.

No 109 **Peter Fritzson:** Towards a Distributed Programming Environment based on Incremental Compilation, 1984, ISBN 91-7372-801-2.

No 111 **Erik Tengvald:** The Design of Expert Planning Systems. An Experimental Operations Planning System for Turning, 1984, ISBN 91-7372- 805-5.

No 155 **Christos Levcopoulos:** Heuristics for Minimum Decompositions of Polygons, 1987, ISBN 91-7870-133-3.

No 165 **James W. Goodwin:** A Theory and System for Non-Monotonic Reasoning, 1987, ISBN 91-7870-183-X.

No 170 **Zebo Peng:** A Formal Methodology for Automated Synthesis of VLSI Systems, 1987, ISBN 91-7870-225-9.

No 174 **Johan Fagerström:** A Paradigm and System for Design of Distributed Systems, 1988, ISBN 91-7870-301-8.

No 192 **Dimiter Driankov:** Towards a Many Valued Logic of Quantified Belief, 1988, ISBN 91-7870-374-3.

No 213 **Lin Padgham:** Non-Monotonic Inheritance for an Object Oriented Knowledge Base, 1989, ISBN 91-7870-485-5.

No 214 **Tony Larsson:** A Formal Hardware Description and Verification Method, 1989, ISBN 91-7870-517-7.

No 221 **Michael Reinfrank:** Fundamentals and Logical Foundations of Truth Maintenance, 1989, ISBN 91-7870-546-0.

No 239 **Jonas Löwgren:** Knowledge-Based Design Support and Discourse Management in User Interface Management Systems, 1991, ISBN 91-7870-720-X.

No 244 **Henrik Eriksson:** Meta-Tool Support for Knowledge Acquisition, 1991, ISBN 91-7870-746-3.

No 252 **Peter Eklund:** An Epistemic Approach to Interactive Design in Multiple Inheritance Hierarchies, 1991, ISBN 91-7870-784-6.

No 258 **Patrick Doherty:** NML3 - A Non-Monotonic Formalism with Explicit Defaults, 1991, ISBN 91-7870-816-8.

No 260 **Nahid Shahmehri:** Generalized Algorithmic Debugging, 1991, ISBN 91-7870-828-1.

No 264 **Nils Dahlbäck:** Representation of Discourse-Cognitive and Computational Aspects, 1992, ISBN 91-7870-850-8.

No 265 **Ulf Nilsson:** Abstract Interpretations and Abstract Machines: Contributions to a Methodology for the Implementation of Logic Programs, 1992, ISBN 91-7870-858-3.

No 270 **Ralph Rönnquist:** Theory and Practice of Tense-bound Object References, 1992, ISBN 91-7870-873-7.

No 273 **Björn Fjellborg:** Pipeline Extraction for VLSI Data Path Synthesis, 1992, ISBN 91-7870-880-X.

No 276 **Staffan Bonnier:** A Formal Basis for Horn Clause Logic with External Polymorphic Functions, 1992, ISBN 91-7870-896-6.

No 277 **Kristian Sandahl:** Developing Knowledge Management Systems with an Active Expert Methodology, 1992, ISBN 91-7870-897-4.

No 281 **Christer Bäckström:** Computational Complexity of Reasoning about Plans, 1992, ISBN 91-7870-979-2.

No 292 **Mats Wirén:** Studies in Incremental Natural Language Analysis, 1992, ISBN 91-7871-027-8.

No 297 **Mariam Kamkar:** Interprocedural Dynamic Slicing with Applications to Debugging and Testing, 1993, ISBN 91-7871-065-0.

No 302 **Tingting Zhang:** A Study in Diagnosis Using Classification and Defaults, 1993, ISBN 91-7871-078-2

No 312 **Arne Jönsson:** Dialogue Management for Natural Language Interfaces - An Empirical Approach, 1993, ISBN 91-7871-110-X.

No 338 **Simin Nadjm-Tehrani:** Reactive Systems in Physical Environments: Compositional Modelling and Framework for Verification, 1994, ISBN 91-7871-237-8.

No 371 **Bengt Savén:** Business Models for Decision Support and Learning. A Study of Discrete-Event Manufacturing Simulation at Asea/ ABB 1968-1993, 1995, ISBN 91-7871-494-X.

No 375 **Ulf Söderman:** Conceptual Modelling of Mode Switching Physical Systems, 1995, ISBN 91-7871-516-4.

No 383 **Andreas Kågedal:** Exploiting Groundness in Logic Programs, 1995, ISBN 91-7871-538-5.

No 396 **George Fodor:** Ontological Control, Description, Identification and Recovery from Problematic Control Situations, 1995, ISBN 91-7871-603-9.

No 413 **Mikael Pettersson:** Compiling Natural Semantics, 1995, ISBN 91-7871-641-1.

No 414 **Xinli Gu:** RT Level Testability Improvement by Testability Analysis and Transformations, 1996, ISBN 91-7871-654-3.

No 416 **Hua Shu:** Distributed Default Reasoning, 1996, ISBN 91-7871-665-9.

No 429 **Jaime Villegas:** Simulation Supported Industrial Training from an Organisational Learning Perspective - Development and Evaluation of the SSIT Method, 1996, ISBN 91-7871-700-0.

No 431 **Peter Jonsson:** Studies in Action Planning: Algorithms and Complexity, 1996, ISBN 91-7871-704-3.

No 437 **Johan Boye:** Directional Types in Logic Programming, 1996, ISBN 91-7871-725-6.

No 439 **Cecilia Sjöberg:** Activities, Voices and Arenas: Participatory Design in Practice, 1996, ISBN 91-7871-728-0.

No 448 **Patrick Lambrix:** Part-Whole Reasoning in Description Logics, 1996, ISBN 91-7871-820-1.

No 452 **Kjell Orsborn:** On Extensible and Object-Relational Database Technology for Finite Element Analysis Applications, 1996, ISBN 91-7871-827-9.

No 459 **Olof Johansson:** Development Environments for Complex Product Models, 1996, ISBN 91-7871-855-4.

No 461 **Lena Strömbäck:** User-Defined Constructions in Unification-Based Formalisms, 1997, ISBN 91-7871-857-0.

No 462 **Lars Degerstedt:** Tabulation-based Logic Programming: A Multi-Level View of Query Answering, 1996, ISBN 91-7871-858-9.

No 475 **Fredrik Nilsson:** Strategi och ekonomisk styrning - En studie av hur ekonomiska styrsystem utformas och används efter företagsförvärv, 1997, ISBN 91-7871-914-3.

No 480 **Mikael Lindvall:** An Empirical Study of Requirements-Driven Impact Analysis in Object-Oriented Software Evolution, 1997, ISBN 91-7871-927-5.

No 485 **Göran Forslund:** Opinion-Based Systems: The Cooperative Perspective on Knowledge-Based Decision Support, 1997, ISBN 91-7871-938-0.

No 494 **Martin Sköld:** Active Database Management Systems for Monitoring and Control, 1997, ISBN 91-7219-002-7.

No 495 **Hans Olsén:** Automatic Verification of Petri Nets in a CLP framework, 1997, ISBN 91-7219-011-6.

No 498 **Thomas Drakengren:** Algorithms and Complexity for Temporal and Spatial Formalisms, 1997, ISBN 91-7219-019-1.

No 502 **Jakob Axelsson:** Analysis and Synthesis of Heterogeneous Real-Time Systems, 1997, ISBN 91-7219-035-3.

No 503 **Johan Ringström:** Compiler Generation for Data-Parallel Programming Languages from Two-Level Semantics Specifications, 1997, ISBN 91-7219-045-0.

No 512 **Anna Moberg:** Närhet och distans - Studier av kommunikationsmönster i satellitkontor och flexibla kontor, 1997, ISBN 91-7219-119-8.

No 520 **Mikael Ronström:** Design and Modelling of a Parallel Data Server for Telecom Applications, 1998, ISBN 91-7219-169-4.

No 522 **Niclas Ohlsson:** Towards Effective Fault Prevention - An Empirical Study in Software Engineering, 1998, ISBN 91-7219-176-7.

No 526 **Joachim Karlsson:** A Systematic Approach for Prioritizing Software Requirements, 1998, ISBN 91-7219-184-8.

No 530 **Henrik Nilsson:** Declarative Debugging for Lazy Functional Languages, 1998, ISBN 91-7219-197-x.

No 555 **Jonas Hallberg:** Timing Issues in High-Level Synthesis, 1998, ISBN 91-7219-369-7.

No 561 **Ling Lin:** Management of 1-D Sequence Data - From Discrete to Continuous, 1999, ISBN 91-7219-402-2.

No 563 **Eva L Ragnemalm:** Student Modelling based on Collaborative Dialogue with a Learning Companion, 1999, ISBN 91-7219-412-X.

No 567 **Jörgen Lindström:** Does Distance matter? On geographical dispersion in organisations, 1999, ISBN 91-7219-439-1.

No 582 **Vanja Josifovski:** Design, Implementation and Evaluation of a Distributed Mediator System for Data Integration, 1999, ISBN 91-7219-482-0.

No 589 **Rita Kovordányi:** Modeling and Simulating Inhibitory Mechanisms in Mental Image Reinterpretation - Towards Cooperative Human-Computer Creativity, 1999, ISBN 91-7219-506-1.

No 592 **Mikael Ericsson:** Supporting the Use of Design Knowledge - An Assessment of Commenting Agents, 1999, ISBN 91-7219-532-0.

No 593 **Lars Karlsson:** Actions, Interactions and Narratives, 1999, ISBN 91-7219-534-7.

No 594 **C. G. Mikael Johansson:** Social and Organizational Aspects of Requirements Engineering Methods - A practice-oriented approach, 1999, ISBN 91-7219-541-X.

No 595 **Jörgen Hansson:** Value-Driven Multi-Class Overload Management in Real-Time Database Systems, 1999, ISBN 91-7219-542-8.

No 596 **Niklas Hallberg:** Incorporating User Values in the Design of Information Systems and Services in the Public Sector: A Methods Approach, 1999, ISBN 91-7219-543-6.

No 597 **Vivian Vimarlund:** An Economic Perspective on the Analysis of Impacts of Information Technology: From Case Studies in Health-Care towards General Models and Theories, 1999, ISBN 91-7219-544-4.

No 598 **Johan Jenvald:** Methods and Tools in Computer-Supported Taskforce Training, 1999, ISBN 91-7219-547-9.

No 607 **Magnus Merkel:** Understanding and enhancing translation by parallel text processing, 1999, ISBN 91-7219-614-9.

No 611 **Silvia Coradeschi:** Anchoring symbols to sensory data, 1999, ISBN 91-7219-623-8.

No 613 **Man Lin:** Analysis and Synthesis of Reactive Systems: A Generic Layered Architecture Perspective, 1999, ISBN 91-7219-630-0.

No 618 **Jimmy Tjäder:** Systemimplementering i praktiken - En studie av logiker i fyra projekt, 1999, ISBN 91-7219-657-2.

No 627 **Vadim Engelson:** Tools for Design, Interactive Simulation, and Visualization of Object-Oriented Models in Scientific Computing, 2000, ISBN 91-7219-709-9.

No 637 **Esa Falkenroth:** Database Technology for Control and Simulation, 2000, ISBN 91-7219-766-8.

No 639 **Per-Arne Persson:** Bringing Power and Knowledge Together: Information Systems Design for Autonomy and Control in Command Work, 2000, ISBN 91-7219-796-X.

No 660 **Erik Larsson:** An Integrated System-Level Design for Testability Methodology, 2000, ISBN 91-7219-890-7.

No 688 **Marcus Bjäreland:** Model-based Execution Monitoring, 2001, ISBN 91-7373-016-5.

No 689 **Joakim Gustafsson:** Extending Temporal Action Logic, 2001, ISBN 91-7373-017-3.

No 720 **Carl-Johan Petri:** Organizational Information Provision - Managing Mandatory and Discretionary Use of Information Technology, 2001, ISBN-91-7373-126-9.

No 724 **Paul Scerri:** Designing Agents for Systems with Adjustable Autonomy, 2001, ISBN 91 7373 207 9.

No 725 **Tim Heyer:** Semantic Inspection of Software Artifacts: From Theory to Practice, 2001, ISBN 91 7373 208 7.

No 726 **Pär Carlshamre:** A Usability Perspective on Requirements Engineering - From Methodology to Product Development, 2001, ISBN 91 7373 212 5.

No 732 **Juha Takkinen:** From Information Management to Task Management in Electronic Mail, 2002, ISBN 91 7373 258 3.

No 745 **Johan Åberg:** Live Help Systems: An Approach to Intelligent Help for Web Information Systems, 2002, ISBN 91-7373-311-3.

No 746 **Rego Granlund:** Monitoring Distributed Teamwork Training, 2002, ISBN 91-7373-312-1.

No 757 **Henrik André-Jönsson:** Indexing Strategies for Time Series Data, 2002, ISBN 917373-346-6.

No 747 **Anneli Hagdahl:** Development of IT-supported Interorganisational Collaboration - A Case Study in the Swedish Public Sector, 2002, ISBN 91-7373-314-8.

No 749 **Sofie Pilemalm:** Information Technology for Non-Profit Organisations - Extended Participatory Design of an Information System for Trade Union Shop Stewards, 2002, ISBN 91-7373-318-0.

No 765 **Stefan Holmlid:** Adapting users: Towards a theory of use quality, 2002, ISBN 91-7373-397-0.

No 771 **Magnus Morin:** Multimedia Representations of Distributed Tactical Operations, 2002, ISBN 91-7373-421-7.

No 772 **Pawel Pietrzak:** A Type-Based Framework for Locating Errors in Constraint Logic Programs, 2002, ISBN 91-7373-422-5.

No 758 **Erik Berglund:** Library Communication Among Programmers Worldwide, 2002, ISBN 91-7373-349-0.

No 774 **Choong-ho Yi:** Modelling Object-Oriented Dynamic Systems Using a Logic-Based Framework, 2002, ISBN 91-7373-424-1.

No 779 **Mathias Broxvall:** A Study in the Computational Complexity of Temporal Reasoning, 2002, ISBN 91-7373-440-3.

No 793 **Asmus Pandikow:** A Generic Principle for Enabling Interoperability of Structured and Object-Oriented Analysis and Design Tools, 2002, ISBN 91-7373-479-9.

No 785 **Lars Hult:** Publika Informationstjänster. En studie av den Internetbaserade encyklopedins bruksegenskaper, 2003, ISBN 91-7373-461-6.

No 800 **Lars Taxén:** A Framework for the Coordination of Complex Systems´ Development, 2003, ISBN 91-7373-604-X

No 808 **Klas Gäre:** Tre perspektiv på förväntningar och förändringar i samband med införande av informationssystem, 2003, ISBN 91-7373-618-X.

No 821 **Mikael Kindborg:** Concurrent Comics - programming of social agents by children, 2003, ISBN 91-7373-651-1.

No 823 **Christina Ölvingson:** On Development of Information Systems with GIS Functionality in Public Health Informatics: A Requirements Engineering Approach, 2003, ISBN 91-7373-656-2.

No 828 **Tobias Ritzau:** Memory Efficient Hard Real-Time Garbage Collection, 2003, ISBN 91-7373-666-X.

No 833 **Paul Pop:** Analysis and Synthesis of Communication-Intensive Heterogeneous Real-Time Systems, 2003, ISBN 91-7373-683-X.

No 852 **Johan Moe:** Observing the Dynamic Behaviour of Large Distributed Systems to Improve Development and Testing – An Empirical Study in Software Engineering, 2003, ISBN 91-7373-779-8.

No 867 **Erik Herzog:** An Approach to Systems Engineering Tool Data Representation and Exchange, 2004, ISBN 91-7373-929-4.

No 872 **Aseel Berglund:** Augmenting the Remote Control: Studies in Complex Information Navigation for Digital TV, 2004, ISBN 91-7373-940-5.

No 869 **Jo Skåmedal:** Telecommuting's Implications on Travel and Travel Patterns, 2004, ISBN 91-7373-935-9.

No 870 **Linda Askenäs:** The Roles of IT - Studies of Organising when Implementing and Using Enterprise Systems, 2004, ISBN 91-7373-936-7.

No 874 **Annika Flycht-Eriksson:** Design and Use of Ontologies in Information-Providing Dialogue Systems, 2004, ISBN 91-7373-947-2.

No 873 **Peter Bunus:** Debugging Techniques for Equation-Based Languages, 2004, ISBN 91-7373-941-3.

No 876 **Jonas Mellin:** Resource-Predictable and Efficient Monitoring of Events, 2004, ISBN 91-7373-956-1.

No 883 **Magnus Bång:** Computing at the Speed of Paper: Ubiquitous Computing Environments for Healthcare Professionals, 2004, ISBN 91-7373-971-5

No 882 **Robert Eklund:** Disfluency in Swedish human-human and human-machine travel booking dialogues, 2004, ISBN 91-7373-966-9.

No 887 **Anders Lindström:** English and other Foreign Linguistic Elements in Spoken Swedish. Studies of Productive Processes and their Modelling using Finite-State Tools, 2004, ISBN 91-7373-981-2.

No 889 **Zhiping Wang:** Capacity-Constrained Production-inventory systems - Modelling and Analysis in both a traditional and an e-business context, 2004, ISBN 91-85295-08-6.

No 893 **Pernilla Qvarfordt:** Eyes on Multimodal Interaction, 2004, ISBN 91-85295-30-2.

No 910 **Magnus Kald:** In the Borderland between Strategy and Management Control - Theoretical Framework and Empirical Evidence, 2004, ISBN 91-85295-82-5.

No 918 **Jonas Lundberg:** Shaping Electronic News: Genre Perspectives on Interaction Design, 2004, ISBN 91-85297-14-3.

No 900 **Mattias Arvola:** Shades of use: The dynamics of interaction design for sociable use, 2004, ISBN 91-85295-42-6.

No 920 **Luis Alejandro Cortés:** Verification and Scheduling Techniques for Real-Time Embedded Systems, 2004, ISBN 91-85297-21-6.

No 929 **Diana Szentivanyi:** Performance Studies of Fault-Tolerant Middleware, 2005, ISBN 91-85297-58-5.

No 933 **Mikael Cäker:** Management Accounting as Constructing and Opposing Customer Focus: Three Case Studies on Management Accounting and Customer Relations, 2005, ISBN 91-85297-64-X.

No 937 **Jonas Kvarnström:** TALplanner and Other Extensions to Temporal Action Logic, 2005, ISBN 91-85297-75-5.

No 938 **Bourhane Kadmiry:** Fuzzy Gain-Scheduled Visual Servoing for Unmanned Helicopter, 2005, ISBN 91-85297-76-3.

No 945 **Gert Jervan:** Hybrid Built-In Self-Test and Test Generation Techniques for Digital Systems, 2005, ISBN: 91-85297-97-6.

No 946 **Anders Arpteg:** Intelligent Semi-Structured Information Extraction, 2005, ISBN 91-85297-98-4.

No 947 **Ola Angelsmark:** Constructing Algorithms for Constraint Satisfaction and Related Problems - Methods and Applications, 2005, ISBN 91-85297-99-2.

No 963 **Calin Curescu:** Utility-based Optimisation of Resource Allocation for Wireless Networks, 2005, ISBN 91-85457-07-8.

No 972 **Björn Johansson:** Joint Control in Dynamic Situations, 2005, ISBN 91-85457-31-0.

No 974 **Dan Lawesson:** An Approach to Diagnosability Analysis for Interacting Finite State Systems, 2005, ISBN 91-85457-39-6.

No 979 **Claudiu Duma:** Security and Trust Mechanisms for Groups in Distributed Services, 2005, ISBN 91-85457-54-X.

No 983 **Sorin Manolache:** Analysis and Optimisation of Real-Time Systems with Stochastic Behaviour, 2005, ISBN 91-85457-60-4.

No 986 **Yuxiao Zhao:** Standards-Based Application Integration for Business-to-Business Communications, 2005, ISBN 91-85457-66-3.

No 1004 **Patrik Haslum:** Admissible Heuristics for Automated Planning, 2006, ISBN 91-85497-28-2.

No 1005 **Aleksandra Tešanovic:** Developing Reusable and Reconfigurable Real-Time Software using Aspects and Components, 2006, ISBN 91-85497-29-0.

No 1008 **David Dinka:** Role, Identity and Work: Extending the design and development agenda, 2006, ISBN 91-85497-42-8.

No 1009 **Iakov Nakhimovski:** Contributions to the Modeling and Simulation of Mechanical Systems with Detailed Contact Analysis, 2006, ISBN 91-85497-43-X.

No 1013 **Wilhelm Dahllöf:** Exact Algorithms for Exact Satisfiability Problems, 2006, ISBN 91-85523-97-6.

No 1016 **Levon Saldamli:** PDEModelica - A High-Level Language for Modeling with Partial Differential Equations, 2006, ISBN 91-85523-84-4.

No 1017 **Daniel Karlsson:** Verification of Component-based Embedded System Designs, 2006, ISBN 91-85523-79-8

No 1018 **Ioan Chisalita:** Communication and Networking Techniques for Traffic Safety Systems, 2006, ISBN 91-85523-77-1.

No 1019 **Tarja Susi:** The Puzzle of Social Activity - The Significance of Tools in Cognition and Cooperation, 2006, ISBN 91-85523-71-2.

No 1021 **Andrzej Bednarski:** Integrated Optimal Code Generation for Digital Signal Processors, 2006, ISBN 91-85523-69-0.

No 1022 **Peter Aronsson:** Automatic Parallelization of Equation-Based Simulation Programs, 2006, ISBN 91-85523-68-2.

No 1030 **Robert Nilsson:** A Mutation-based Framework for Automated Testing of Timeliness, 2006, ISBN 91-85523-35-6.

No 1034 **Jon Edvardsson:** Techniques for Automatic Generation of Tests from Programs and Specifications, 2006, ISBN 91-85523-31-3.

No 1035 **Vaida Jakoniene:** Integration of Biological Data, 2006, ISBN 91-85523-28-3.

No 1045 **Genevieve Gorrell:** Generalized Hebbian Algorithms for Dimensionality Reduction in Natural Language Processing, 2006, ISBN 91-85643-88-2.

No 1051 **Yu-Hsing Huang:** Having a New Pair of Glasses - Applying Systemic Accident Models on Road Safety, 2006, ISBN 91-85643-64-5.

No 1054 **Åsa Hedenskog:** Perceive those things which cannot be seen - A Cognitive Systems Engineering perspective on requirements management, 2006, ISBN 91-85643-57-2.

No 1061 **Cécile Åberg:** An Evaluation Platform for Semantic Web Technology, 2007, ISBN 91-85643-31-9.

No 1073 **Mats Grindal:** Handling Combinatorial Explosion in Software Testing, 2007, ISBN 978-91-85715-74-9.

No 1075 **Almut Herzog:** Usable Security Policies for Runtime Environments, 2007, ISBN 978-91-85715-65-7.

No 1079 **Magnus Wahlström:** Algorithms, measures, and upper bounds for Satisfiability and related problems, 2007, ISBN 978-91-85715-55-8.

No 1083 **Jesper Andersson:** Dynamic Software Architectures, 2007, ISBN 978-91-85715-46-6.

No 1086 **Ulf Johansson:** Obtaining Accurate and Comprehensible Data Mining Models - An Evolutionary Approach, 2007, ISBN 978-91-85715-34-3.

No 1089 **Traian Pop:** Analysis and Optimisation of Distributed Embedded Systems with Heterogeneous Scheduling Policies, 2007, ISBN 978-91-85715-27-5.

No 1091 **Gustav Nordh:** Complexity Dichotomies for CSP-related Problems, 2007, ISBN 978-91-85715-20-6.

No 1106 **Per Ola Kristensson:** Discrete and Continuous Shape Writing for Text Entry and Control, 2007, ISBN 978-91-85831-77-7.

No 1110 **He Tan:** Aligning Biomedical Ontologies, 2007, ISBN 978-91-85831-56-2.

No 1112 **Jessica Lindblom:** Minding the body - Interacting socially through embodied action, 2007, ISBN 978-91-85831-48-7.

No 1113 **Pontus Wärnestål:** Dialogue Behavior Management in Conversational Recommender Systems, 2007, ISBN 978-91-85831-47-0.

No 1120 **Thomas Gustafsson:** Management of Real-Time Data Consistency and Transient Overloads in Embedded Systems, 2007, ISBN 978-91-85831-33-3.

No 1127 **Alexandru Andrei:** Energy Efficient and Predictable Design of Real-time Embedded Systems, 2007, ISBN 978-91-85831-06-7.

No 1139 **Per Wikberg:** Eliciting Knowledge from Experts in Modeling of Complex Systems: Managing Variation and Interactions, 2007, ISBN 978-91-85895-66-3.

No 1143 **Mehdi Amirijoo:** QoS Control of Real-Time Data Services under Uncertain Workload, 2007, ISBN 978-91-85895-49-6.

No 1150 **Sanny Syberfeldt:** Optimistic Replication with Forward Conflict Resolution in Distributed Real-Time Databases, 2007, ISBN 978-91-85895-27-4.

No 1155 **Beatrice Alenljung:** Envisioning a Future Decision Support System for Requirements Engineering - A Holistic and Human-centred Perspective, 2008, ISBN 978-91-85895-11-3.

No 1156 **Artur Wilk:** Types for XML with Application to Xcerpt, 2008, ISBN 978-91-85895-08-3.

No 1183 **Adrian Pop:** Integrated Model-Driven Development Environments for Equation-Based Object-Oriented Languages, 2008, ISBN 978-91-7393-895-2.

No 1185 **Jörgen Skågeby:** Gifting Technologies - Ethnographic Studies of End-users and Social Media Sharing, 2008, ISBN 978-91-7393-892-1.

No 1187 **Imad-Eldin Ali Abugessaisa:** Analytical tools and information-sharing methods supporting road safety organizations, 2008, ISBN 978-91-7393-887-7.

No 1204 **H. Joe Steinhauer:** A Representation Scheme for Description and Reconstruction of Object Configurations Based on Qualitative Relations, 2008, ISBN 978-91-7393-823-5.

No 1222 **Anders Larsson:** Test Optimization for Core-based System-on-Chip, 2008, ISBN 978-91-7393-768-9.

No 1238 **Andreas Borg:** Processes and Models for Capacity Requirements in Telecommunication Systems, 2009, ISBN 978-91-7393-700-9.

No 1240 **Fredrik Heintz:** DyKnow: A Stream-Based Knowledge Processing Middleware Framework, 2009, ISBN 978-91-7393-696-5.

No 1241 **Birgitta Lindström:** Testability of Dynamic Real-Time Systems, 2009, ISBN 978-91-7393-695-8.

No 1244 **Eva Blomqvist:** Semi-automatic Ontology Construction based on Patterns, 2009, ISBN 978-91-7393-683-5.

No 1249 **Rogier Woltjer:** Functional Modeling of Constraint Management in Aviation Safety and Command and Control, 2009, ISBN 978-91-7393-659-0.

No 1260 **Gianpaolo Conte:** Vision-Based Localization and Guidance for Unmanned Aerial Vehicles, 2009, ISBN 978-91-7393-603-3.

No 1262 **AnnMarie Ericsson:** Enabling Tool Support for Formal Analysis of ECA Rules, 2009, ISBN 978-91-7393-598-2.

No 1266 **Jiri Trnka:** Exploring Tactical Command and Control: A Role-Playing Simulation Approach, 2009, ISBN 978-91-7393-571-5.

No 1268 **Bahlol Rahimi:** Supporting Collaborative Work through ICT - How End-users Think of and Adopt Integrated Health Information Systems, 2009, ISBN 978-91-7393-550-0.

No 1274 **Fredrik Kuivinen:** Algorithms and Hardness Results for Some Valued CSPs, 2009, ISBN 978-91-7393-525-8.

No 1281 **Gunnar Mathiason:** Virtual Full Replication for Scalable Distributed Real-Time Databases, 2009, ISBN 978-91-7393-503-6.

No 1290 **Viacheslav Izosimov:** Scheduling and Optimization of Fault-Tolerant Distributed Embedded Systems, 2009, ISBN 978-91-7393-482-4.

No 1294 **Johan Thapper:** Aspects of a Constraint Optimisation Problem, 2010, ISBN 978-91-7393-464-0.

No 1306 **Susanna Nilsson:** Augmentation in the Wild: User Centered Development and Evaluation of Augmented Reality Applications, 2010, ISBN 978-91-7393-416-9.

No 1313 **Christer Thörn:** On the Quality of Feature Models, 2010, ISBN 978-91-7393-394-0.

No 1321 **Zhiyuan He:** Temperature Aware and Defect-Probability Driven Test Scheduling for System-on-Chip, 2010, ISBN 978-91-7393-378-0.

No 1333 **David Broman:** Meta-Languages and Semantics for Equation-Based Modeling and Simulation, 2010, ISBN 978-91-7393-335-3.

No 1337 **Alexander Siemers:** Contributions to Modelling and Visualisation of Multibody Systems Simulations with Detailed Contact Analysis, 2010, ISBN 978-91-7393-317-9.

No 1354 **Mikael Asplund:** Disconnected Discoveries: Availability Studies in Partitioned Networks, 2010, ISBN 978-91-7393-278-3.

No 1359 **Jana Rambusch**: Mind Games Extended: Understanding Gameplay as Situated Activity, 2010, ISBN 978-91-7393-252-3.

No 1373 **Sonia Sangari**: Head Movement Correlates to Focus Assignment in Swedish,2011,ISBN 978-91-7393-154-0.

No 1374 **Jan-Erik Källhammer**: Using False Alarms when Developing Automotive Active Safety Systems, 2011, ISBN 978-91-7393-153-3.

No 1375 **Mattias Eriksson**: Integrated Code Generation, 2011, ISBN 978-91-7393-147-2.

No 1381 **Ola Leifler**: Affordances and Constraints of Intelligent Decision Support for Military Command and Control – Three Case Studies of Support Systems, 2011, ISBN 978-91-7393-133-5.

No 1386 **Soheil Samii**: Quality-Driven Synthesis and Optimization of Embedded Control Systems, 2011, ISBN 978-91-7393-102-1.

No 1419 **Erik Kuiper**: Geographic Routing in Intermittently-connected Mobile Ad Hoc Networks: Algorithms and Performance Models, 2012, ISBN 978-91-7519-981-8.

No 1451 **Sara Stymne**: Text Harmonization Strategies for Phrase-Based Statistical Machine Translation, 2012, ISBN 978-91-7519-887-3.

No 1455 **Alberto Montebelli**: Modeling the Role of Energy Management in Embodied Cognition, 2012, ISBN 978-91-7519-882-8.

No 1465 **Mohammad Saifullah**: Biologically-Based Interactive Neural Network Models for Visual Attention and Object Recognition, 2012, ISBN 978-91-7519-838-5.

No 1490 **Tomas Bengtsson**: Testing and Logic Optimization Techniques for Systems on Chip, 2012, ISBN 978-91-7519-742-5.

No 1481 **David Byers**: Improving Software Security by Preventing Known Vulnerabilities, 2012, ISBN 978-91-7519-784-5.

No 1496 **Tommy Färnqvist**: Exploiting Structure in CSP-related Problems, 2013, ISBN 978-91-7519-711-1.

No 1503 **John Wilander**: Contributions to Specification, Implementation, and Execution of Secure Software, 2013, ISBN 978-91-7519-681-7.

No 1506 **Magnus Ingmarsson**: Creating and Enabling the Useful Service Discovery Experience, 2013, ISBN 978-91-7519-662-6.

No 1547 **Wladimir Schamai**: Model-Based Verification of Dynamic System Behavior against Requirements: Method, Language, and Tool, 2013, ISBN 978-91-7519-505-6.

No 1551 **Henrik Svensson**: Simulations, 2013, ISBN 978-91-7519-491-2.

No 1559 **Sergiu Rafiliu**: Stability of Adaptive Distributed Real-Time Systems with Dynamic Resource Management, 2013, ISBN 978-91-7519-471-4.

No 1581 **Usman Dastgeer**: Performance-aware Component Composition for GPU-based Systems, 2014, ISBN 978-91-7519-383-0.

No 1602 **Cai Li**: Reinforcement Learning of Locomotion based on Central Pattern Generators, 2014, ISBN 978-91-7519-313-7.

No 1652 **Roland Samlaus**: An Integrated Development Environment with Enhanced Domain-Specific

Interactive Model Validation, 2015, ISBN 978-91-7519-090-7.

No 1663 **Hannes Uppman**: On Some Combinatorial Optimization Problems: Algorithms and Complexity, 2015, ISBN 978-91-7519-072-3.

No 1664 **Martin Sjölund**: Tools and Methods for Analysis, Debugging, and Performance Improvement of Equation-Based Models, 2015, ISBN 978-91-7519-071-6.

No 1666 **Kristian Stavåker**: Contributions to Simulation of Modelica Models on Data-Parallel Multi-Core Architectures, 2015, ISBN 978-91-7519-068-6.

No 1680 **Adrian Lifa**: Hardware/Software Codesign of Embedded Systems with Reconfigurable and Heterogeneous Platforms, 2015, ISBN 978-91-7519-040-2.

No 1685 **Bogdan Tanasa**: Timing Analysis of Distributed Embedded Systems with Stochastic Workload and Reliability Constraints, 2015, ISBN 978-91-7519-022-8.

No 1691 **Håkan Warnquist**: Troubleshooting Trucks – Automated Planning and Diagnosis, 2015, ISBN 978-91-7685-993-3.

No 1702 **Nima Aghaee**: Thermal Issues in Testing of Advanced Systems on Chip, 2015, ISBN 978-91-7685-949-0.

No 1715 **Maria Vasilevskaya**: Security in Embedded Systems: A Model-Based Approach with Risk Metrics, 2015, ISBN 978-91-7685-917-9.

No 1729 **Ke Jiang**: Security-Driven Design of Real-Time Embedded System, 2016, ISBN 978-91-7685-884-4.

No 1733 **Victor Lagerkvist**: Strong Partial Clones and the Complexity of Constraint Satisfaction Problems: Limitations and Applications, 2016, ISBN 978-91-7685-856-1.

No 1734 **Chandan Roy**: An Informed System Development Approach to Tropical Cyclone Track and Intensity Forecasting, 2016, ISBN 978-91-7685-854-7.

No 1746 **Amir Aminifar**: Analysis, Design, and Optimization of Embedded Control Systems, 2016, ISBN 978-91-7685-826-4.

No 1747 **Ekhiotz Vergara**: Energy Modelling and Fairness for Efficient Mobile Communication, 2016, ISBN 978-91-7685-822-6.

No 1748 **Dag Sonntag**: Chain Graphs – Interpretations, Expressiveness and Learning Algorithms, 2016, ISBN 978-91-7685-818-9.

No 1768 **Anna Vapen**: Web Authentication using Third-Parties in Untrusted Environments, 2016, ISBN 978-91-7685-753-3.

No 1778 **Magnus Jandinger**: On a Need to Know Basis: A Conceptual and Methodological Framework for Modelling and Analysis of Information Demand in an Enterprise Context, 2016, ISBN 978-91-7685-713-7.

No 1798 **Rahul Hiran**: Collaborative Network Security: Targeting Wide-area Routing and Edge-network Attacks, 2016, ISBN 978-91-7685-662-8.

No 1813 **Nicolas Melot**: Algorithms and Framework for Energy Efficient Parallel Stream Computing on Many-Core Architectures, 2016, ISBN 978-91-7685-623-9.

No 1823 **Amy Rankin**: Making Sense of Adaptations: Resilience in High-Risk Work, 2017, ISBN 978-91-7685-596-6.

No 1831 **Lisa Malmberg**: Building Design Capability in the Public Sector: Expanding the Horizons of Development, 2017, ISBN 978-91-7685-585-0.

No 1854 **Meysam Aghighi**: Computational Complexity of some Optimization Problems in Planning, 2017, ISBN 978-91-7685-519-5.

Linköping Studies in Arts and Science

No 504 **Ing-Marie Jonsson:** Social and Emotional Characteristics of Speech-based In-Vehicle Information Systems: Impact on Attitude and Driving Behaviour, 2009, ISBN 978-91-7393-478-7.

No 586 **Fabian Segelström:** Stakeholder Engagement for Service Design: How service designers identify and communicate insights, 2013, ISBN 978-91-7519-554-4.

No 618 **Johan Blomkvist:** Representing Future Situations of Service: Prototyping in Service Design, 2014, ISBN 978-91-7519-343-4.

No 620 **Marcus Mast:** Human-Robot Interaction for Semi-Autonomous Assistive Robots, 2014, ISBN 978-91-7519-319-9.

No 677 **Peter Berggren:** Assessing Shared Strategic Understanding, 2016, ISBN 978-91-7685-786-1.

No 695 **Mattias Forsblad:** Distributed cognition in home environments: The prospective memory and cognitive practices of older adults, 2016, ISBN 978-91-7685-686-4.

Linköping Studies in Statistics

No 9 **Davood Shahsavani:** Computer Experiments Designed to Explore and Approximate Complex Deterministic Models, 2008, ISBN 978-91-7393-976-8.

No 10 **Karl Wahlin:** Roadmap for Trend Detection and Assessment of Data Quality, 2008, ISBN 978-91-7393-792-4.

No 11 **Oleg Sysoev:** Monotonic regression for large multivariate datasets, 2010, ISBN 978-91-7393-412-1.

No 13 **Agné Burauskaite-Harju:** Characterizing Temporal Change and Inter-Site Correlations in Daily and Sub-daily Precipitation Extremes, 2011, ISBN 978-91-7393-110-6.

Linköping Studies in Information Science

No 1 **Karin Axelsson:** Metodisk systemstrukturering- att skapa samstämmighet mellan informationssystem-arkitektur och verksamhet, 1998. ISBN-9172-19-296-8.

No 2 **Stefan Cronholm:** Metodverktyg och användbarhet - en studie av datorstödd metodbaserad systemutveckling, 1998, ISBN-9172-19-299-2.

No 3 **Anders Avdic:** Användare och utvecklare - om anveckling med kalkylprogram, 1999. ISBN-91-7219-606-8.

No 4 **Owen Eriksson:** Kommunikationskvalitet hos informationssystem och affärsprocesser, 2000, ISBN 91-7219-811-7.

No 5 **Mikael Lind:** Från system till process - kriterier för processbestämning vid verksamhetsanalys, 2001, ISBN 91-7373-067-X.

No 6 **Ulf Melin:** Koordination och informationssystem i företag och nätverk, 2002, ISBN 91-7373-278-8.

No 7 **Pär J. Ågerfalk:** Information Systems Actability - Understanding Information Technology as a Tool for Business Action and Communication, 2003, ISBN 91-7373-628-7.

No 8 **Ulf Seigerroth:** Att förstå och förändra systemutvecklingsverksamheter - en taxonomi för metautveckling, 2003, ISBN91-7373-736-4.

No 9 **Karin Hedström:** Spår av datoriseringens värden – Effekter av IT i äldreomsorg, 2004, ISBN 91-7373-963-4.

No 10 **Ewa Braf:** Knowledge Demanded for Action - Studies on Knowledge Mediation in Organisations, 2004, ISBN 91-85295-47-7.

No 11 **Fredrik Karlsson:** Method Configuration method and computerized tool support, 2005, ISBN 91-85297-48-8.

No 12 **Malin Nordström:** Styrbar systemförvaltning - Att organisera systemförvaltningsverksamhet med hjälp av effektiva förvaltningsobjekt, 2005, ISBN 91-85297-60-7.

No 13 **Stefan Holgersson:** Yrke: POLIS - Yrkeskunskap, motivation, IT-system och andra förutsättningar för polisarbete, 2005, ISBN 91-85299-43-X.

No 14 **Benneth Christiansson, Marie-Therese Christiansson:** Mötet mellan process och komponent - mot ett ramverk för en verksamhetsnära kravspecifikation vid anskaffning av komponentbaserade informationssystem, 2006, ISBN 91-85643-22-X.