Cognitive Technologies

Managing Editors: D.M. Gabbay J. Siekmann

For further volumes:
www.springer.com/series/5216

Stanislav Živný

The Complexity of Valued Constraint Satisfaction Problems

Springer

Stanislav Živný
University College
University of Oxford
Oxford, UK

Managing Editors
Prof. Dov M. Gabbay
Augustus De Morgan Professor of Logic
Department of Computer Science
King's College London
London, UK

Prof. Dr. Jörg Siekmann
Forschungsbereich Deduktions- und
Multiagentensysteme
DFKI
Saarbrücken, Germany

ISSN 1611-2482 Cognitive Technologies
ISBN 978-3-642-33973-8 ISBN 978-3-642-33974-5 (eBook)
DOI 10.1007/978-3-642-33974-5
Springer Heidelberg New York Dordrecht London

Library of Congress Control Number: 2012951706

ACM Computing Classification (1998): F.2.2, G.2.1, I.2

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

This book is dedicated to the memory of my mother, Marta Živná (1954–2010)

Preface

Computer science is no more about computers than astronomy is about telescopes.
Edsger Dijkstra

The topic of this book is the following optimisation problem: given a set of discrete variables and a set of functions, each depending on a subset of the variables, minimise the sum of the functions over all variables. This fundamental research problem has been studied within several different contexts of computer science and artificial intelligence under different names: Min-Sum Problems, inference in Markov Random Fields (MRFs) and Conditional Random Fields (CRFs), Gibbs energy minimisation, valued constraint satisfaction problems (VCSPs), and (for two-state variables) pseudo-Boolean optimisation. We present general techniques for analysing the structure of such functions and the computational complexity of the minimisation problem.

This book could not have been written without the support of Oxford's University College, which funded me through a Stipendiary Junior Research Fellowship in Mathematical and Physical Sciences for 3 years.

Many results in this book are joint work with Dave Cohen and Pete Jeavons, from whom I have learnt the ropes of academic work. Pete has also served as my mentor and Ph.D. supervisor at Oxford. I am grateful to both for their advice, support, and friendship. Some results from Chap. 2 are joint work with Páidí Creed. Some results described in Chap. 3 are joint work with Bruno Zanuttini. The results described in Chap. 7 were obtained in collaboration with Vladimir Kolmogorov, whom I met at the Tractability Workshop in Microsoft Research Cambridge in 2010, when I was a research intern there. The results presented in Chap. 8 are joint work with Johan Thapper, whom I met at the Algebraic CSP Workshop at the Fields Institute for Research in Mathematical Sciences. The Fields Institute kindly funded my attendance at the workshop. The UK Engineering and Physical Sciences Research Council (EPSRC), the Royal Society (RS), and the French National Research Agency (ANR) financed my trips to the University of Toulouse III. Chapter 9 briefly summarises some of the results that have come out of these very productive research visits to

Toulouse and have given me the opportunity to work with and learn from Martin Cooper. I am grateful to all the above-mentioned collaborators for the time spent together, fun we had, and everything I have learnt from them.

I am grateful to my sister, Radka, and my parents-in-law for their support. Last but not least, I would like to express my gratitude to my wonderful wife, Biying, for her love, encouragement, and inspiration.

Oxford, UK
July 2012

Stanislav Živný

Contents

Introduction

We can only see a short distance ahead,
but we can see plenty there that needs to be done.
Alan Turing

The main topic of this book is the following optimisation problem: given a set of discrete variables and a set of functions, each depending on a subset of the variables, minimise the sum of the functions over all variables. This fundamental research problem has been studied within several different contexts of artificial intelligence, computer science, and combinatorial optimisation under different names: Min-Sum Problems [268], MAP inference in Markov Random Fields (MRFs) and Conditional Random Fields (CRFs) [203, 267], Gibbs energy minimisation [128], valued constraint satisfaction problems [97], and (for two-state variables) pseudo-Boolean optimisation [34, 84].

We start off with several examples of interesting and well-studied problems that can be modelled in this way.

Example 1 (Satisfiability) The standard propositional SATISFIABILITY problem for ternary clauses, 3-SAT [125], consists in determining whether it is possible to satisfy a Boolean formula given as a conjunction of ternary clauses, where each clause is a set of three literals and each literal is either a variable or the negation of a variable. A generalisation of 3-SAT, the 3-MAX-SAT problem [125] consists in finding an assignment of 0s and 1s (representing FALSE and TRUE, respectively) to all variables in the given formula such that the number of satisfied clauses is maximised, or equivalently (with respect to exact solvability), finding an assignment of 0s and 1s to all variables such that the number of unsatisfied clauses is minimised.

Any 3-SAT instance can be easily seen as a minimisation problem with the objective function given by a sum of ternary $\{0, \infty\}$-valued functions, one function for each clause. For instance, clause $(x \vee \neg v \vee z)$ yields a ternary function f defined by $f(0, 1, 0) = \infty$ and $f(x, y, z) = 0$ otherwise.

Similarly, any 3-MAX-SAT instance can be easily seen as a minimisation problem with the objective function given by a sum of ternary $\{0, 1\}$-valued functions, one function for each clause. For instance, clause $(x \vee \neg v \vee z)$ yields a ternary function f defined by $f(0, 1, 0) = 1$ and $f(x, y, z) = 0$ otherwise.

Example 2 (Graph colouring) The k-COLOURABILITY problem [125] consists in determining whether it is possible to assign k colours to the vertices of a given graph so that adjacent vertices are assigned different colours. This can be viewed as a minimisation problem with the objective function given by a sum of binary functions; each edge in the graph yields a binary function f defined by $f(x, y) = 0$ if $x \neq y$ and $f(x, y) = \infty$ otherwise.

Example 3 (Digraph acyclicity) Given a directed graph G, the question of whether G is acyclic can be modelled as follows: variables correspond to the vertices of G, the domain of every variable is the set of natural numbers $\mathbb{N}$,[1] and every arc (x, y) of G yields a binary function f that represents the standard "smaller than" ordering on natural numbers; that is, $f(x, y) = 0$ if $x < y$ and $f(x, y) = \infty$ otherwise.

Example 4 (Diophantine equations) Hilbert's tenth problem asks for an algorithm that decides whether a given system of polynomial equations with integer coefficients (a *diophantine* equation system) has an integer solution. This problem can be modelled with variables $x_1, \ldots, x_n$, each with the domain $\mathbb{Z}$, and constraints of the form $ax_i + bx_j + c = x_k$, or $x_i * x_j = x_k$, for $a, b, c, d \in \mathbb{Z}$ and $i, j, k \in \{1, 2, \ldots, n\}$; this can be easily represented by $\{0, \infty\}$-valued ternary functions. Matiyasevič has shown that this problem is undecidable [216].

Example 5 (Min-Cost-Hom) Given a graph G, we denote by $V(G)$ the set of vertices of G and by $E(G)$ the set of edges of G. Given two (directed or undirected) graphs G and H, a mapping $h : V(G) \to V(H)$ is a *homomorphism* from G to H if h preserves edges; that is, $(u, v) \in E(G)$ implies $(h(u), h(v)) \in E(H)$. The homomorphism problem for graphs asks for the existence of a homomorphism from G to H [150]. Let $c_v(u)$ be a nonnegative rational cost for all $u \in V(G)$ and $v \in V(H)$. The cost of a homomorphism f from G to H is defined by $\sum_{u \in V(G)} c_{f(u)}(u)$. The MINIMUM-COST HOMOMORPHISM problem, MIN-COST-HOM [143, 145], asks for a homomorphism of minimum cost between two given graphs. This problem can be cast as a minimisation problem of a sum of binary $\{0, \infty\}$-valued functions and unary rational-valued functions.

Example 6 (Max-Cut) Given a graph G with the vertex set V, the MAXIMUM CUT problem, MAX-CUT [125], consists in finding a subset $S \subseteq V$ of the vertices of G that maximises the number of edges between vertices in S and $V \setminus S$. The polynomial-time equivalent (with respect to exact solvability) problem is MINIMUM UNCUT, MIN-UNCUT; that is, the problem of finding a subset $S \subseteq V$ that minimises the number of edges in S and $V \setminus S$.

This problem can be seen as a minimisation problem with the objective function being a sum of binary functions defined on $\{0, 1\}$. In particular, if we define

[1] In fact, only the set $\{1, 2, \ldots, n\}$, where n is the number of vertices of G, would suffice as the domain of each variable.

$\bar{\lambda}(x, y) = 0$ if $x \neq y$ and $\bar{\lambda}(x, y) = 1$ if $x = y$, then the objective function is a sum of $\bar{\lambda}$'s, each edge of G corresponding to one $\bar{\lambda}$.

Example 7 ((s,t)-Min-Cut) Given a graph G with the vertex set V and two specified vertices $s, t \in V$, the (s, t)-MIN-CUT problem consists in finding a subset $S \subseteq V$ of the vertices G with $s \in V$ and $t \notin V$ that minimises the number of edges between vertices in S and $V \setminus S$.

This problem can be seen as a minimisation problem with the objective function being a sum of unary and binary functions defined on $\{0, 1\}$. In particular, let $\lambda(0, 1) = 1$ and $\lambda(x, y) = 0$ otherwise, and for any $d \in D$, let $\mu^d(d) = \infty$ and $\mu^d(x) = 0$ otherwise. Now the objective function is a sum of λ's, each edge of G corresponding to one λ, and $\mu^1(s) + \mu^0(t)$, which enforces the inclusion of s and exclusion of t.

It is straightforward to generalise (s, t)-MIN-CUT to graphs with edge weights. Both versions of the problem are solvable in polynomial time [131].

Example 8 (Submodularity) Let $D = \{1, 2, \ldots, d\}$ for some fixed d. Let Γ be a set of functions $f : D^k \to \mathbb{Q}_{\geq 0}$ satisfying $f(\min(s, t)) + f(\max(s, t)) \leq f(s) + f(t)$ for any $s, t \in D^k$, where k is the arity of f, and min and max are binary functions returning the smaller and larger, respectively, of its two arguments with respect to the usual order on integers. (Note that min and max are applied componentwise on tuples s and t.) Any sum of functions from Γ over n variables, each with domain D, can be minimised in polynomial time in n due to its submodularity property [159, 257].

We finish our list with some more applied examples.

Example 9 (Timetabling) In timetabling exams at a university [248], variables can represent the times and locations of the different exams, and the functions can model the capacity of each examination room (for example, we cannot assign more students to take exams in a given room at any one time than the room's capacity) and prevent certain exams from being scheduled at the same time (for example, we cannot schedule two exams at the same time if they share students in common). The objective function can also take into account teachers' and students' preferences.

Example 10 (Texture-based segmentation) Given a set of distinct textures, such as a dictionary of RGB patches, together with their object class labels, the goal is to segment a given image; that is, the pixels of the image should be labelled as belonging to one of the object classes. This problem can be formulated using discrete variables, one for each pixel, where the domain of each variable is the set of distinct object classes. The binary functions are usually defined such that they encourage contiguous segments whose boundaries lie on image edges [37]. Similar approaches can be used for 3D reconstruction or object recognition [36].

Example 11 (Office assignment) Each of n staff members, represented by n variables, must be assigned an office. There are m offices, each of which can be assigned at most u_j people. Unary functions express personal preferences of each staff member for each office. There are also nonoverlapping groups of people $G_1, \ldots, G_g$ whom we would prefer to assign to different offices (such as married couples, for example). What is the best solution?

While some of the problems from the examples above are tractable, such as Examples 7 and 8, some of them are (NP-)hard, or even undecidable, such as Example 4. *Our main interest is in the question of what makes these problems hard and what the special cases are that are tractable.*

Focus of This Book

The focus of this book is on *exact solvability*; that is, we are interested in solving the problem in hand *optimally* (as opposed to approximately). Furthermore, a class of problems is considered *tractable* if any instance from it can be solved in *polynomial time* (as opposed to other notions of tractability such as moderate exponential-time tractability or fixed-parameter tractability). Finally, we will consider problems with discrete variables on *finite domains* only. For some (classes of) problems, the domains will be fixed; for others, the domains will be part of the input (and thus unbounded), but always finite.

Since all the above-mentioned frameworks are equivalent with respect to exact solvability and given that this book is based on several papers that talk about valued constraint satisfaction problems, we will use the terminology of *valued constraint satisfaction problems* (VCSPs) [97, 253].

A special case of VCSPs are so-called *constraint satisfaction problems* (CSPs), first identified in the seminal work of Montanari [218]. CSPs deal only with the feasibility (rather than the optimisation) problem, as is Examples 2 and 3, but we will pay some attention to them in this book as they have been studied in other contexts as well, such as homomorphisms between relational structures [111, 150] and conjunctive query evaluation [111, 136, 183, 251]. Apart from the tractability notion studied in this book, defined by polynomial-time solvability, there is an alternative and well-studied approach to solving CSPs in practice, which consists in interleaving a backtracking search with a series of heuristics and polynomial-time propagation, which significantly prune the exponential search space. We refer the reader interested in this research area, known as *constraint programming*, to the standard textbooks [3, 97, 248], the proceedings of the Annual International Conference on Principles and Practice of Constraint Programming (CP), and the website of the Association for Constraint Programming (ACP).[2]

[2] http://4c.ucc.ie/a4cp/.

Structure of This Book

Apart from the introduction, this book consists of three parts. Part II is based on the author's doctoral thesis from the University of Oxford [274], which won the 2011 Association for Constraint Programming (ACP) Doctoral Award.

Part II investigates the expressive power of various classes of functions. Chapter 2, based on [63, 68, 274], presents an algebraic theory for the expressive power of languages. Chapter 3, based on [70] (preliminary version [69]) and [272], investigates the expressive power of fixed-arity languages. Chapter 4, based on [280] (preliminary version [277]), is concerned with submodular languages. Chapter 5, based on [275] (preliminary version [276]), shows that not all submodular languages are expressible by binary submodular languages.

Part III deals with the tractability of valued constraint satisfaction problems. Chapter 6 surveys known tractable languages and is based on [168]. Chapter 7, based on [188] (full version [187]), briefly presents the complexity of conservative languages. Chapter 8, based on [264], presents recent results on the power of linear programming for valued constraint satisfaction problems. Chapter 9, based on [80] and [82] (preliminary versions [78, 79, 81]), surveys known results on hybrid tractability. Finally, Chap. 10 concludes with some open problems.