

Handbook of Constraint Programming

Edited by F. Rossi, P. van Beek and T. Walsh

Elsevier

Foreword

Constraints are an ubiquitous concept, which in its broader sense pertains to every day experience: they represent the conditions which restrict our freedom of decision. In fact, how much our choices are constrained by the external world is a basic philosophical question. In the formalized reasoning of scientific disciplines, constraints have been employed extensively, from logic to numerical analysis, from mathematical programming to operations research. In computer science, constraints have been with us from the early days, for modeling, representing and reasoning (see the interesting historical remarks in Chapter 2 of this handbook, *Constraint Satisfaction: An Emerging Paradigm*).

I see several good reasons for this ubiquity: one is the conceptually clear separation between the perfectly declarative problem statements and the often cumbersome enumerative efforts for finding solutions. Another reason is the complexity challenge: the classical constraint satisfaction problem is NP-complete and in fact tautology checking in propositional calculus (a constraint problem on Boolean variables) has been the touchstone for this complexity class. A further reason is that large, complex constraint problems often occur in practice, they must be solved in one way or another, and fast, efficient, systematic solutions have an enormous economic value.

What I find surprising about constraints is that within artificial intelligence and computer science a relatively recent, relatively uniform body of knowledge has emerged which often yields decisive advantages over classical, extensively studied and well developed techniques. As for many success stories within computer science, success is largely due to a mixture of structures, algorithms, languages, programming techniques and system implementations. The aim of this handbook is to present this knowledge in all its facets. Different chapters are largely self contained and all contribute to put the subject into focus, similarly to the Hawaii Keck observatory, where the mirror is composed of 36 hexagonal segments.

From the conceptual point of view, the main characteristic features of constraint programming are constraint propagation, and the identification of various special cases which make complexity tractable. The former (see Chapter 3) is an inference technique which makes local constraints stronger without changing the global constraint. The latter issue concerns both the structure (see Chapter 7, *Tractable Structures for Constraint Satisfaction Problems*) and the kind of constraints (see Chapter 8, *The Complexity of Constraint Languages*). Less specific, but still very important issues are as follows: *Backtracking Search Algorithms*, in Chapter 4; *Local Search*, in Chapter 5; *Global Constraints*, in Chapter 6; *Symmetry in Constraint Programming*, in Chapter 10; and *Modelling*, in Chapter 11.

Another surprising fact about constraint theory is the incredibly close relationship with logic programming. In a rather precise sense logic programming is a way of expressing, and solving, certain classes of disjunctive, recursive constraints. Furthermore, logic programming can be very elegantly generalized to constraint logic programming (see Chapter

12), where the ordinary Herbrand constraint system, and its unification algorithm, are complemented with specific constraint solvers. The interaction with the committed choice languages studied in the Japanese projects of the eighties also yielded very interesting models of computation based on constraints. Amalgamation with more common (and efficiently implemented!) programming languages is also possible (see Chapter 13, Constraints in Procedural and Concurrent Languages).

Besides and beyond the beauty of its theoretical foundations, what contributes the most to the practical convenience of constraint programming are: (i) the development of specific results for important classes of constraints; (ii) the ability of extending the basic theory to various additional aspects which are very relevant in practice; and (iii) the flexibility and potential for integration with other modeling and solving methodologies.

About the development of specific results, this handbook includes chapters about constraints on finite (Chapter 14), structured (Chapter 17), temporal (Chapter 19), continuous and interval-based (Chapter 16) domains. The potential to extend the basic theory is evident in the case of soft constraints, considered in Chapter 9. Ordinary constraints are either satisfied or not, namely either true or false. Instead soft constraints return a more informative weight. Interestingly enough, the proposed extensions both accommodate several important cases (fuzzy, hierarchical, optimization, probabilistic constraints), and still often exhibit essentially the same solution algorithms. Extensions to random, changing and distributed/open constraints are treated in Chapters 18, 21 and 20 respectively.

About the last issue, in addition to the seamless integration with logic and imperative programming languages we mentioned already, quite remarkable are the paradigms resulting from the integration of constraint programming with operations research (see Chapter 15), with scheduling and planning (see Chapter 22), with vehicle routing (see Chapter 23), with component configuration (see Chapter 24), with (electricity, water, oil, data) networks (see Chapter 25), and with bioinformatics (see Chapter 26).

The global scenario based on service-oriented computing which is now under development offers additional theoretical and practical challenges to constraint programming. Conditions for service deployment and discovery, both functional and involving different aspects of quality of service, could be expressed in terms of hard and soft constraints, and the negotiation phases should involve substantial constraint solving abilities. Transactions among the various actors could also require partially backtrackable behavior or at least programmable compensations. Some level of real time, distributed, global constraint solving should be implemented in the middleware, since lots of higher level applications will be able to take advantage of, and pay for it.

I think that research and practical development in the area of constraint programming will be very active for quite a while in the future, establishing closer and closer connections with a variety of other design methodologies and even other disciplines. I consider this handbook not only a very nice piece of scientific work, but also a contribution quite instrumental at disseminating advanced knowledge about constraint programming both within the inner constraint community and across the much wider audience of potential users.

UGO MONTANARI
Dipartimento di Informatica
Università di Pisa

Contributors

Rolf Backofen
Albert-Ludwigs-Universität
Germany

Philippe Baptiste
CNRS LIX & École Polytechnique
France

Frédéric Benhamou
Université de Nantes
France

Christian Bessiere
LIRMM-CNRS
France

Kenneth N. Brown
Cork Constraint Computation Centre &
University College Cork, Ireland

Mats Carlsson
SICS AB
Sweden

David Cohen
Royal Holloway, University of London
United Kingdom

Rina Dechter
University of California, Irvine
USA

Boi Faltings
Swiss Federal Institute of Technology
Switzerland

Eugene C. Freuder
Cork Constraint Computation Centre &
University College Cork, Ireland

Thom Frühwirth
Universität Ulm
Germany

Ian P. Gent
University of St. Andrews
Scotland, United Kingdom

Carmen Gervet
Brown University
USA

David Gilbert
University of Glasgow
Scotland, United Kingdom

Carla Gomes
Cornell University
USA

Laurent Granvilliers
Université de Nantes
France

John N. Hooker
Carnegie Mellon University
USA

Holger H. Hoos
University of British Columbia
Canada

Peter Jeavons
University of Oxford
United Kingdom

Ulrich Junker
ILOG SA
France

Irit Katriel
University of Aarhus
Denmark

Philip Kilby
The Australian National University
Australia

Manolis Koubarakis
University of Athens
Greece

Philippe Laborie
ILOG SA
France

Claude Le Pape
ILOG SA
France

Alan K. Mackworth
University of British Columbia
Canada

Kim Marriott
Monash University
Australia

Pedro Meseguer
IIIA-CSIC
Spain

Laurent Michel
University of Connecticut
USA

Ian Miguel
The University of St. Andrews
Scotland, United Kingdom

Wim Nuijten
ILOG SA
France

Karen E. Petrie
University of St. Andrews
Scotland, United Kingdom

Jean-François Puget
ILOG SA
France

Francesca Rossi
Università di Padova
Italy

Thomas Schiex
INRA Toulouse
France

Christian Schulte
KTH - Royal Institute of Technology
Sweden

Paul Shaw
ILOG SA
France

Helmut Simonis
CrossCore Optimization
United Kingdom

Barbara M. Smith
Cork Constraint Computation Centre &
University College Cork, Ireland

Peter J. Stuckey
University of Melbourne
Australia

Edward Tsang
University of Essex
United Kingdom

Peter van Beek
University of Waterloo
Canada

Willem-Jan van Hove
Cornell University
USA

Mark Wallace
Monash University
Australia

Toby Walsh
National ICT Australia &
University of New South Wales
Australia

Contents

Foreword	v
Contributors	vii
Contents	xi
I Foundations	1
1 Introduction	3
Francesca Rossi, Peter van Beek, Toby Walsh	
1.1 Purpose of the Handbook	4
1.2 Structure and Content	4
1.3 Future Research	10
2 Constraint Satisfaction: An Emerging Paradigm	13
Eugene C. Freuder and Alan K. Mackworth	
2.1 The Early Days	13
2.2 The Constraint Satisfaction Problem: Representation and Reasoning . .	16
2.3 Conclusions	23
3 Constraint Propagation	29
Christian Bessiere	
3.1 Background	30
3.2 Formal Viewpoint	33
3.3 Arc Consistency	37
3.4 Higher Order Consistencies	50
3.5 Domain-Based Consistencies Stronger than AC	57
3.6 Domain-Based Consistencies Weaker than AC	62
3.7 Constraint Propagation as Iteration of Reduction Rules	68
3.8 Specific Constraints	70
4 Backtracking Search Algorithms	85
Peter van Beek	
4.1 Preliminaries	86
4.2 Branching Strategies	87
4.3 Constraint Propagation	90
4.4 Nogood Recording	96

4.5	Non-Chronological Backtracking	102
4.6	Heuristics for Backtracking Algorithms	105
4.7	Randomization and Restart Strategies	111
4.8	Best-First Search	116
4.9	Optimization	117
4.10	Comparing Backtracking Algorithms	118
5	Local Search Methods	135
	Holger H. Hoos and Edward Tsang	
5.1	Introduction	136
5.2	Randomised Iterative Improvement Algorithms	142
5.3	Tabu Search and Related Algorithms	144
5.4	Penalty-Based Local Search Algorithms	148
5.5	Other Approaches	154
5.6	Local Search for Constraint Optimisation Problems	155
5.7	Frameworks and Toolkits for Local Search	157
5.8	Conclusions and Outlook	158
6	Global Constraints	169
	Willem-Jan van Hove and Irit Katriel	
6.1	Notation and Preliminaries	170
6.2	Examples of Global Constraints	176
6.3	Complete Filtering Algorithms	182
6.4	Optimization Constraints	189
6.5	Partial Filtering Algorithms	193
6.6	Global Variables	200
6.7	Conclusion	203
7	Tractable Structures for Constraint Satisfaction Problems	209
	Rina Dechter	
7.1	Background	210
7.2	Structure-Based Tractability in Inference	213
7.3	Trading Time and Space by Hybrids of Search and Inference	231
7.4	Structure-Based Tractability in Search	239
7.5	Summary and Bibliographical Notes	241
8	The Complexity of Constraint Languages	245
	David Cohen and Peter Jeavons	
8.1	Basic Definitions	246
8.2	Examples of Constraint Languages	247
8.3	Developing an Algebraic Theory	251
8.4	Applications of the Algebraic Theory	258
8.5	Constraint Languages Over an Infinite Set	263
8.6	Multi-Sorted Constraint Languages	264
8.7	Alternative Approaches	269
8.8	Future Directions	274

9	Soft Constraints	281
	Pedro Meseguer, Francesca Rossi, Thomas Schiex	
9.1	Background: Classical Constraints	282
9.2	Specific Frameworks	283
9.3	Generic Frameworks	287
9.4	Relations among Soft Constraint Frameworks	291
9.5	Search	297
9.6	Inference	300
9.7	Combining Search and Inference	313
9.8	Using Soft Constraints	316
9.9	Promising Directions for Further Research	321
10	Symmetry in Constraint Programming	329
	Ian P. Gent, Karen E. Petrie, Jean-François Puget	
10.1	Symmetries and Group Theory	331
10.2	Definitions	337
10.3	Reformulation	340
10.4	Adding Constraints Before Search	343
10.5	Dynamic Symmetry Breaking Methods	350
10.6	Combinations of Symmetry Breaking Methods	362
10.7	Successful Applications	363
10.8	Symmetry Expression and Detection	364
10.9	Further Research Themes	366
10.10	Conclusions	368
11	Modelling	377
	Barbara M. Smith	
11.1	Preliminaries	378
11.2	Representing a Problem	379
11.3	Propagation and Search	379
11.4	Viewpoints	381
11.5	Expressing the Constraints	382
11.6	Auxiliary Variables	386
11.7	Implied Constraints	387
11.8	Reformulations of CSPs	391
11.9	Combining Viewpoints	394
11.10	Symmetry and Modelling	398
11.11	Optimization Problems	400
11.12	Supporting Modelling and Reformulation	401
II	Extensions, Languages, and Applications	407
12	Constraint Logic Programming	409
	Kim Marriott, Peter J. Stuckey, Mark Wallace	
12.1	History of CLP	411
12.2	Semantics of Constraint Logic Programs	413

12.3	CLP for Conceptual Modeling	425
12.4	CLP for Design Modeling	430
12.5	Search in CLP	437
12.6	Impact of CLP	442
12.7	Future of CLP and Interesting Research Questions	444
13	Constraints in Procedural and Concurrent Languages	453
	Thom Frühwirth, Laurent Michel, and Christian Schulte	
13.1	Procedural and Object-Oriented Languages	454
13.2	Concurrent Constraint Programming	465
13.3	Rule-Based Languages	473
13.4	Challenges and Opportunities	485
13.5	Conclusion	486
14	Finite Domain Constraint Programming Systems	495
	Christian Schulte and Mats Carlsson	
14.1	Architecture for Constraint Programming Systems	496
14.2	Implementing Constraint Propagation	506
14.3	Implementing Search	513
14.4	Systems Overview	517
14.5	Outlook	519
15	Operations Research Methods in Constraint Programming	527
	John N. Hooker	
15.1	Schemes for Incorporating OR into CP	527
15.2	Plan of the Chapter	528
15.3	Linear Programming	530
15.4	Mixed Integer/Linear Modeling	534
15.5	Cutting Planes	536
15.6	Relaxation of Global Constraints	539
15.7	Relaxation of Piecewise Linear and Disjunctive Constraints	545
15.8	Lagrangean Relaxation	547
15.9	Dynamic Programming	550
15.10	Branch-and-Price Methods	554
15.11	Benders Decomposition	556
15.12	Toward Integration of CP and OR	560
16	Continuous and Interval Constraints	571
	Frédéric Benhamou and Laurent Granvilliers	
16.1	From Discrete to Continuous Constraints	574
16.2	The Branch-and-Reduce Framework	575
16.3	Consistency Techniques	577
16.4	Numerical Operators	583
16.5	Hybrid Techniques	587
16.6	First Order Constraints	590
16.7	Applications and Software packages	593
16.8	Conclusion	595

17 Constraints over Structured Domains	605
Carmen Gervet	
17.1 History and Applications	606
17.2 Constraints over Regular and Constructed Sets	609
17.3 Constraints over Finite Set Intervals	613
17.4 Influential Extensions to Subset Bound Solvers	619
17.5 Constraints over Maps, Relations and Graphs	628
17.6 Constraints over Lattices and Hierarchical Trees	631
17.7 Implementation Aspects	631
17.8 Applications	633
17.9 Further Topics	633
18 Randomness and Structure	639
Carla Gomes and Toby Walsh	
18.1 Random Constraint Satisfaction	640
18.2 Random Satisfiability	644
18.3 Random Problems with Structure	648
18.4 Runtime Variability	651
18.5 History	657
18.6 Conclusions	658
19 Temporal CSPs	665
Manolis Koubarakis	
19.1 Preliminaries	666
19.2 Constraint-Based Formalisms for Reasoning About Time	669
19.3 Efficient Algorithms for Temporal CSPs	677
19.4 More Expressive Queries for Temporal CSPs	681
19.5 First-Order Temporal Constraint Languages	683
19.6 The Scheme of Indefinite Constraint Databases	685
19.7 Conclusions	691
20 Distributed Constraint Programming	699
Boi Faltings	
20.1 Definitions	701
20.2 Distributed Search	702
20.3 Improvements and Variants	713
20.4 Distributed Local Search	718
20.5 Open Constraint Programming	721
20.6 Further Issues	724
20.7 Conclusion	726
21 Uncertainty and Change	731
Kenneth N. Brown and Ian Miguel	
21.1 Background and Definitions	732
21.2 Example: Course Scheduling	732
21.3 Uncertain Problems	733
21.4 Problems that Change	738

21.5	Pseudo-dynamic Formalisms	752
21.6	Challenges and Future Trends	753
21.7	Summary	755
22	Constraint-Based Scheduling and Planning	761
	Philippe Baptiste, Philippe Laborie, Claude Le Pape, Wim Nuijten	
22.1	Constraint Programming Models for Scheduling	763
22.2	Constraint Programming Models for Planning	771
22.3	Constraint Propagation for Resource Constraints	778
22.4	Constraint Propagation on Optimization Criteria	785
22.5	Heuristic Search	789
22.6	Conclusions	794
23	Vehicle Routing	801
	Philip Kilby and Paul Shaw	
23.1	The Vehicle Routing Problem	802
23.2	Operations Research Approaches	804
23.3	Constraint Programming Approaches	809
23.4	Constraint Programming in Search	819
23.5	Using Constraint Programming as a Subproblem Solver	823
23.6	CP-VRP in the Real World	825
23.7	Conclusions	828
24	Configuration	837
	Ulrich Junker	
24.1	What Is Configuration?	838
24.2	Configuration Knowledge	844
24.3	Constraint Models for Configuration	853
24.4	Problem Solving for Configuration	863
24.5	Conclusion	868
25	Constraint Applications in Networks	875
	Helmut Simonis	
25.1	Electricity Networks	876
25.2	Water (Oil) Networks	878
25.3	Data Networks	879
25.4	Conclusion	898
26	Bioinformatics and Constraints	905
	Rolf Backofen and David Gilbert	
26.1	What Biologists Want from Bioinformatics	906
26.2	The Central Dogma	907
26.3	A Classification of Problem Areas	908
26.4	Sequence Related Problems	908
26.5	Structure Related Problems	922
26.6	Function Related Problems	935
26.7	Microarrays	937

Part I

Foundations