Constraint Satisfaction with Preferences

PhD Thesis Summary

Hana Rudová
Faculty of Informatics, Masaryk University
Brno, Czech Republic
http://www.fi.muni.cz/~hanka

January 2001

First works on *constraint satisfaction problems (CSP)* can be traced in the 70'th but their widespread study and application started since the end of 80'th when the extension of unification algorithm of logic programming led to the proposal of *constraint logic programming (CLP)* scheme. Introductory part of the thesis introduces both proposals as approaches allowing to express properties of the problem declaratively by means of constraints and to search for its solution via specialized algorithms. Presented thesis is focused on CSPs on finite domains because of their practical applicability. Generally 95 % of all industrial constraint-based applications are formulated as CSPs on finite domains. *Constraint programming* is applied to solve wide range of problems, among them scheduling, configuration, hardware verification, graphical interfaces, or molecular biology. However, in most real-life situations we need to express fuzziness, possibilities, probabilities, costs, ... While problems may become *over-constrained* it does not make sense to say there is no solution. Many problems require finding of best or optimal solution wrt. one or even more *optimization criteria*. Others may be ill-defined and we need to include *uncertainty* of the problem definition into final solution. All these problems need to apply some type of *preferences* which have to be included into both declarative and control part of the solution. That led us to the study of *CSPs with preferences* from both theoretical (proposal of new methodologies and uniform view to the existing ones) and practical (their application in *timetabling* area) points of view within this thesis.

1 Frameworks

After short introduction of CSP & CLP schemes, we may proceed through description of particular frameworks for solving CSPs with preferences. As we intended to give a unifying view to particular approaches, we have proposed a uniform structure which is followed by all frameworks. Each of them elucidates notions of *constraint* (natural extension of constraint definition by a specific preference), *problem* (how it is extended to handle given preferences), *distribution* (a function aimed to evaluate particular assignments based on preferences), *solution* (defined with help of distribution via its optimal value), and *consistency* (up to which extent may be given problem satisfied).

Starting from the basic frameworks over particular types of preferences (*weighted, probabilistic, possibilistic, fuzzy CSPs*) we continue through meta-frameworks (*partial, valued, semiring-based CSPs*). Basic frameworks may be obtained by specification of a general algebraic *structure* of meta-frameworks. Weighted CSP applies costs to optimize satisfaction of particular constraints (e.g., minimize sum of weights of unsatisfied constraints). Probabilities allow to express the degrees how much each constraint is valid in some ill-defined problem. Here we need to search for a definition of the real problem together with its best solution. Possibilistic CSP assigns a possibility degree to each constraint expressing how desirable its satisfaction is. This problem together with fuzzy CSP may be handled via min-max optimization. Fuzzy CSP associates the same type of preference to each tuple of values of constraint, however.

Partial CSP introduces the first attempt towards the generalization of basic approaches via problem relaxation. Later proposed valued CSP (VCSP) and semiring-based CSP (SCSP) build axiomatic theories which enable the construction of non trivial generic properties and algorithms. Particular formalisms define monoid and lattice structures to catch particular preferences. While the monoid structure is able to handle only *totally ordered preferences*, SCSP with its lattice structure may represent *partial ordering* also. Multi-criteria optimization belongs to a typical representatives which require this property.

Even if SCSP is able to handle partial ordering of preferences, relation of equivalence between different SCSPs is not able to consider them. This led us to definition of a new equivalence and refinement which are able to compare SCSP classes with partial ordering of preferences too. This definition remains in accordance with basic idea behind the refinement: if some problem is a refinement of another one, then its set of solutions is included in the set of the later problem.

Together with the comparison of meta-frameworks, final part of this chapter summarizes relationships between particular basic frameworks which may be partitioned according to the *idempotency of unifiable operator* of both algebraic structures. Classical CSP, possibilistic and fuzzy CSP may be transformed via polynomial refinement into weighted CSP but the opposite transformation doesn't exist. It means that problem of finding optimal solution of weighted CSP can not be reduced to fuzzy CSP.

2 Constraint Hierarchies

Constraint hierarchies (CHs) and namely hierarchical constraint logic programming belong to traditional frameworks for handling of over-constrained problems. They allow to express *hard constraints* which has to be satisfied and several preference levels of *soft constraints* which violations are minimized level by level subsequently. CHs define the so called *comparators* aimed to select solutions (the best assignment of values to particular variables) via minimizing *errors* of violated constraints. *Global* comparators aggregate errors of violated constraints at each level, basically we may compare weighted sum of errors or error of worst satisfied constraint (*weighted-sum-better* and *worst-case-better* comparators). *Local* (*locally-better*) comparator considers each constraint individually, *regional* comparator is able to select among assignments by individual comparison of constraints at some lower level even if they are incomparable at higher levels.

We have enhanced classical local comparator such that it is able to apply weights similar to those existing for weighted-sum-better comparator and proposed the so called *ordered-better* comparator [1]. We have studied its relationship with locally-better comparator —

based on it and on existing algorithm for locally-better comparator we have proposed a new extended algorithm which is able to find ordered-better solution of CH. Our further study was concentrated on a more accurate proposal of the worst-case comparator which evokes following contra-intuitive behavior: violating a constraint with large weight we are not able to differ assignments violating or satisfying constraints with any smaller weight. This disadvantage may be eliminated with help of lexicographic ordering which led us to the definition of the *lexicographic-better* comparator.

Even if the CHs are extensively studied and diverse algorithms were discussed to solve particular comparators, any correspondence of CH's comparators with existing algebraic meta-frameworks wasn't shown with exclusion of vague statements about their relationship with enhanced versions of fuzzy CSPs. This opens not only a theory gap, but also practical problems which disable the applicability of any general algorithm already proposed. Our effort was devoted to the proposal of instances of particular algebraic structures which would capture most of the discussed comparators. Remaining of them are shown to be incompatible with some of the required properties of algebraic structures. As the second step, we have applied our extended definitions of equivalence and refinement between classes of SCSPs and we have classified all comparators of CHs which have an existing counterpart in some class of SCSP.

Basically we have constructed three classes of SCSP: weighted-SCSP corresponding to comparators derived from weighted-sum-better comparator, lexicographic-SCSP for lexicographic-better comparator, and local-SCSP for locally-better and ordered-better comparators. Worst-case-better comparator does not have its SCSP's counterpart due to above described contra-intuitive behavior — this makes the lexicographic-better comparator even more important as its equivalent SCSP class does exist. The property which forbids a proposal of SCSP class for regional comparator is just its ability to compare assignments which are incomparable at higher levels.

Weighted-SCSP together with lexicographic-SCSP are shown to be equivalent with the weighted CSP via polynomial time refinement. This relationship might be expected because all of them have non-idempotent operator and total ordering of preferences. The total ordering implies that all comparators dedicated to weighted-SCSP and lexicographic-SCSP classes may be defined as a classes of VCSP. However, this doesn't hold for local-SCSP having partial ordering of preferences. This class may be polynomially transformed into the weighted CSP but the opposite transformation may not be constructed as the preferences within local-SCSP are partially ordered.

3 Variables' Annotations

Above discussed frameworks associate preferences either with constraints or with each tuple of values of constraint. An original contribution of the thesis concerns another still not investigated possibility, an idea of *assigning preferences to particular variables* in constraint. Such preferences may express different levels of importance for particular variables (e.g., time of events within temporal scheduling may be classified by participant's interest, order of job within jobs' sequence is influenced by owner's priority, prior placement of objects may depend on their properties).

We propose an *annotation triple*, a structure over annotations defining ordering over annotations and function for combining particular annotations. Local annotations of variables within constraints are combined to compute *global annotations* of variables and constraints via specified combining function. Examples of its instances are discussed to be later applied for computing solution via *fuzzy annotations*, *hierarchical annotations*, and *variable ordering* [6].

Fuzzy and hierarchical annotations [2, 4, 3] present interpretation of variable's annotations based on existing frameworks with preferences, on fuzzy CSP and constraint hierarchies. For fuzzy annotations, global constraint annotations define preferences for selection of solutions via min-max optimization. A correspondence of constraint system with fuzzy annotation with both fuzzy and possibilistic CSPs is proven. For hierarchical annotations [1], *annotated constraint hierarchy (ACH)* is constructed with help of global annotations which define weights of particular constraints. Solution of ACH is computed by different methods inherited by weighted-sum, worst-case, and least-square principles. Another method is presented via individual comparison of particular global annotations at each level. Mapping of applied methods to particular comparators of CH is presented in accordance with selection method for global comparators. Solutions obtained via individual comparison are compatible with the proposed ordered-better comparator of CHs. Based on early mentioned algorithm for computing the ordered-better solution of CH, we have implemented a solver for systems of inequalities with annotations.

Another interpretation of annotations considers them as a source for computing variable ordering (VO) in CSPs with preferences. Several methods are proposed including different annotation triples and static versus dynamic VO. While static VO is computed with help of global annotations within initial solving step, dynamic VO recomputes global annotation each time a variable should be instantiated. Such global annotations take into account those variables only which remain to be free wrt. current partial assignment of variables in given computation step.

4 Timetabling

The final part of the thesis concerns practical application of constraint-based approach with preferences in the area of timetabling [7, 5]. Our intent is devoted to the *course timetabling problem* which is a typical representative of scheduling and resource allocation application where various type of preferences need to be involved to obtain any acceptable solution.

First we have introduced general problem considering particular *courses*, *teachers*, *classrooms*, and *students* as the basic objects. Having courses with too large number of students, they may be split to *course sections*. The main tasks to be solved are *section assignment* (assign students to particular course sections), *classroom allocation* (assign classrooms to course sections), and *time assignment* (assign time to course sections).

Subsequently our attention is devoted to *current solution methods* of the classroom allocation and time assignment problems via constrained-based approach. We have distinguished possible models of problem solution including classroom and time assignment variables wrt. problem properties. We have also studied methods for search of solution space wrt. preferences within the problem. Such search might be processed with no global optimization criteria via *labelling heuristics* or *non-systematic constraint relaxation* methods. Labelling heuristics influence solution search by an ordering of variables and values in their domains. Both orderings may reflect preferences within the problem. Constraint relaxation methods select some of the problem constraints to be violated. Here selection of violated constraints depends on their preferences. More sophisticated approaches define some objective function over preferences within the problem with aim to optimize its value. This definition often leads to the *weighted CSP* and relaxing the constraints wrt. their weights or costs.

We have studied *classroom allocation problem* and proposed representation of problem with help of *global constraints* for distinguished problem instances. The most simple problem definition allows to allocate classrooms of the same capacity to courses such that all requirements towards classrooms can be specified over time variables and overall classroom allocation might be processed after completing the time assignment. We have defined sufficient property of the problem together with its representation by global constraints to allow such separation of classroom allocation and time assignment also for classrooms of various capacities and types. To solve general problem instance, we partitioned classroom allocation to two steps which are processed before and after time assignments.

Next section is devoted to construction of *student-oriented schedules* representing problem which still was not solved via CP methodology. The first part defines an objective function over generated timetable minimizing the number of courses which student is not able to attend wrt. parallel schedule of his (her) other course(s). Incremental definition of approximation of this objective function allows its inclusion into labelling. The second part defines *soft disjunctive scheduling problem (SDSP)* which instance is also designated student conflict minimization problem. Solution of such SDSP is defined via *weighted and fuzzy constraint satisfaction* as the basic representative of different complexity classes of CSPs with preferences.

Preferences are also included into the problem with help of *variables' annotations* as overall problem is stated in variables having their own natural preferences. Annotations are derived from seniority of teachers (dean, professors, assistants, ...), type of classrooms, or interest of students expressed by their requirements in course pre-enrollment. These annotations are applied via variable ordering methods. Examples of hard and soft constraints with annotations are described finally.

Proposed methods are applied for *Faculty of Informatics timetabling problem* which solution was implemented in CLP library of ILOG Scheduler. Overall problem is characterized by requirements of scheduling courses for individual students (more than 1 000 students) and taking into account preferential requirements of teachers towards any acceptable timetable. Achieved computational results conclude this part, within them the most interesting one is a quality of generated timetable satisfying more than 94 % students' requirements.

Our solution methods together with achieved results are compared with other approaches and solutions of comparable problem instances at the end of chapter.

References

[1] Hana Rudová. Constraints with variables' annotations and constraint hierarchies. In Branislav Rovan, editor, *SOFSEM'98: Theory and Practice of Informatics*, pages 409–418. Springer-Verlag LNCS 1521, 1998.

[2] Hana Rudová. Constraints with variables' annotations. In Henri Prade, editor, *13th European Conference on Artificial Intelligence Proceedings*, pages 261–262. John Wiley & Sons, Ltd., 1998.

[3] Hana Rudová. Constraints with variables' annotations. Technical Report FIMU-RS-98-04, Faculty of Informatics Masaryk University, 1998.

[4] Hana Rudová. Over-constrained systems. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence and the Eleventh Innovative Applications of Artificial Intelligence Conference*, page 954. AAAI Press/MIT Press, 1999. Presented in 1999 SIGART/AAAI Doctoral Consortium.

[5] Hana Rudová and Luděk Matyska. Timetabling with annotations. Technical Report FIMU-RS-99-09, Faculty of Informatics Masaryk University, 1999.

[6] Hana Rudová and Luděk Matyska. Uniform framework for solving over-constrained and optimization problems. In *CP'99 Post-Conference Workshop on Modelling and Solving Soft Constraints*, 1999.

[7] Hana Rudová and Luděk Matyska. Constraint-based timetabling with student schedules. In Edmund Burke and Wilhelm Erben, editors, *PATAT 2000 — Proceedings of the 3rd international conference on the Practice And Theory of Automated Timetabling*, pages 109–123, 2000.