Fuzzy Constraint Satisfaction

Zs. Ruttkay

AI Group, Dept. of Mathematics and Computer Science, Vrije Universiteit Amsterdam
De Boelelaan 1081a, 1081HV Amsterdam, The Netherlands
e-mail: zsofi@cs.vu.nl

***Abstract* — In this paper the issue of soft constraint satisfaction is discussed from a fuzzy set theoretical point of view. A fuzzy constraint is considered as a fuzzy relation. Different possible definitions for the degree of joint satisfaction of a set of fuzzy constraints are given, covering specific other soft constraint satisfaction problem (CSP) types such a partial and hierarchical CSP. It is shown that the classical CSP solving heuristics based on variable and value evaluations can be generalised and used to guide the solution construction process for solving fuzzy CSPs, and that the heuristic search can be replaced by branch-and-bound search. The solution process is illustrated with an example from the CSP literature. Finally, research issues are discussed.**

1. Introduction

In the recent years there has been a growing interest in soft constraint satisfaction. In general, in a soft CSP not all the given constraints need to be satisfied — either because all of them cannot be met, theoretically, or in practice it would require too much time to find a solution, or simply, the real problem to be modelled is such that soft constraints are adequate and should be introduced, e.g. to express possible alternative requirements. Different kinds of soft CSPs and solution techniques have been discussed in the CSP literature: *partial constraint satisfaction* [8], *hierarchical constraint satisfaction* [1, 7], *structured constraint satisfaction* [16], *soft constraint relaxation* [9] and *constrained heuristic search* [6]. In all these approaches constraints (or sets of constraints) are compared, e.g. on the basis of weights (e.g. indicating importance) assigned to the given crisp constraints.

Another approach to deal with soft constraints is to generalise the notion of crisp constraint. A crisp constraint is characterised by the set of tuples for which the constraint holds. Hence, it is a natural extension that a fuzzy set characterises a fuzzy constraint. This possibility has recently received attention [3, 4, 11, 12, 17]. In the case of a fuzzy constraint, different tuples satisfy the given constraint to a different degree. We find the fuzzy set theoretical approach to deal with soft constraints appealing for two reasons. From a technical point of view, the existing concepts and techniques of fuzzy set theory can be adapted for defining and solving fuzzy CSPs. From a conceptual point of view, by comparing tuples the preferences are expressed in an explicit way, which may facilitate the knowledge acquisition and model building process.

In this paper we discuss possible definitions for fuzzy solutions of fuzzy CSPs. Then we show how the basic solution technique for solving CSPs — constructive heuristic search — can be adapted for solving fuzzy CSPs. We discuss appropriate generalisations of variable and value evaluations which can be used to guide the search, and the applicability of the branch and bound technique to prune the search space. Finally, we formulate some research issues.

2. Basic definitions

The fuzzy relation, cardinality, domain of a fuzzy relation and fuzzy operators common in fuzzy set theory [2] can be used to generalise essential concepts of crisp CSPs for fuzzy CSPs.

2.1 Fuzzy constraint

A *fuzzy constraint* is a mapping form a $D=D_1 x...xD_k$ domain (the direct product of the finite domain of the variables referred by the constraint) to the [0,1] interval. For a fuzzy constraint c the number $c(v_1,...v_k)$ denotes 'how well' the tuple $(v_1,...v_k)$ satisfies the constraint. A fuzzy constraint corresponds to the membership function of a fuzzy set. For a $(v_1,...v_k)$ we call $c(v_1,...v_k)$ the *degree of satisfaction* of the constraint c by the instantiation $(v_1,...v_k)$. If $c(v_1,...v_k)=1$ then we say that $(v_1,...v_k)$

0-7803-1896-X/94 $4.00 ©1994 IEEE

fully satisfies c, while if $c(v_1,...v_k)=0$ then we say that $(v_1,...v_k)$ *fully violates* c. The $(v_1,...v_k)$ elements for which $c(v_1,...v_k)>0$ form the so-called *support set*, denoted by S. One may be interested in elements which satisfy a constraint better than a given threshold α, where $1\geq\alpha>0$. The *α-support set*, denoted by S_α, is the set of all $(v_1,...v_k)\in D_1 x...xD_k$ for which

$$c(v_1,...v_k)\geq\alpha.$$

The *cardinality* of c is

$$\| c \| = \sum_{v_1...v_k} c(v_1,...v_k).$$

2.2 Fuzzy CSP

A *fuzzy CSP* (FCSP) is given as a list of variables $(x_1,...x_K)$, a list of finite domains of possible values for the variables $(D_1,...D_K)$, and a list of fuzzy constraints $(c_1,...c_N)$, each of them referring to some of the given variables. We will consider binary FCSPs only. We will also assume that there is at most one constraint referring to any two variables. These restrictions are made only to simplify the discussion. All that will be introduced below can be easily generalised for FCSPs with non-binary constraints, and with possible multiple constraints referring to the same variables.

We will use the notation $\underline{x}_i$ for some (x_{i_1}, x_{i_2}), that is the pair of variables referred to by the constraint c_i, and $\underline{v}_i$ for some $(v_{i_1}, v_{i_2})\in D_{i_1}xD_{i_2}$. We will also use the notation c_{ij} for a constraint referring to the variables x_i and x_j, $\underline{x}_{ij}$ for (x_i, x_j), and $\underline{v}_{ij}$ for $(v_i, v_j)\in D_i x D_j$. If a $\underline{v}\in D$ is given, then it satisfies the individual $c_1,...c_N$ constraints to the degree $c_1(\underline{v}_1),...c_N(\underline{v}_N)$, respectively. The *domain* of a constraint c_{ij} is a fuzzy set, with the membership function dom_{ij}, where

$$dom_{ij}(v)=\max \{c_{ij}(v, w) \mid w\in D_j\} \ \forall \ v\in D_i.$$

2.3 Fuzzy solution

The solution of an FCSP is also fuzzy: for any instantiation the degree of satisfaction indicates how well the constraints are satisfied jointly. There are several ways to define the degree of joint satisfaction in terms of the degree of satisfaction of the individual constraints. We will introduce three plausible and simple definitions.

As an individual constraint can be considered as the membership function of a fuzzy set, the problem of defining the degree of satisfaction of several constraints can be considered as the problem of defining the membership function for intersection of fuzzy sets. Adapting two common fuzzy set theoretical intersection definitions [2], we get two possible ways to define the degree of joint satisfaction of a set of fuzzy constraints, and hence, of an FCSP.

Definition 1: Based on the *conjunctive combination* principle, the degree of joint satisfaction of the constraints $c_1,...c_N$ by the instantiation $\underline{v} \in D$ is the minimum of the satisfaction of the individual constraints. That is,

$$C_{min}((c_1,...c_N), v) = \min \{c_i(v_i) \mid i=1,...N\}.$$

•

Definition 2: Based on the *productive combination* principle, the degree of joint satisfaction of the constraints $c_1,...c_N$ by the instantiation $\underline{v} \in D$ is the product of the satisfaction of the individual constraints. That is,

$$C_{pro}((c_1,...c_N), \underline{v}) = \prod_{i=1}^{N} c_i(\underline{v}_i).$$ •

However, form a CSP perspective it also makes sense to consider the joint satisfaction as the cumulated satisfaction of the individual constraints, related to the fuzzy-set theoretical concept of union of fuzzy sets.

Definition 3: Based on the *average combination* principle, the degree of joint satisfaction of the constraints $c_1,...c_N$ by the instantiation $\underline{v} \in D$ is the average of the satisfaction of the individual constraints. That is,

$$C_{ave}((c_1,...c_N), \underline{v}) = \frac{1}{N} \sum_{i=1}^{N} c_i(\underline{v}_i).$$ •

As it is discussed in [4], the first definition is too rough, not discriminating instantiations which are equally 'bad' concerning the least satisfied constraint, but differ much in the degree of satisfaction of the rest of the constraints. In [4], two refinements of the conjunctive combination principle are discussed based on inclusion and lexicographical ordering.

Our last two definitions are also such that the above mentioned 'drowning effect' does not occur. The second definition does not differentiate between instantiations which fully violate at least one constraint, while the third definition does. All the definitions above can be modified in such a way that importance of the individual constraints (relative or absolute) is taken into account. It also can be shown that partial constraint satisfaction corresponds to the case when the constraints are crisp, but the notion of joint satisfaction is defined on the average combination principle.

Without respect to the specific definition of the degree of joint satisfaction — further on denoted by C, in general — , the concepts of perfect and best solution can be defined (the same concepts were defined for the specific partial CSP case in [8]). An instantiation $\underline{v}^* \in D$ is a *perfect solution* if all the individual constraints are satisfied in the best possible way, that is

$$c_{ij}(\underline{v}^*_{ij}) = \max \{ c_{ij}(v^*_i, v_j) \mid v_j \in D_j \} \text{ for all } c_{ij}.$$

An instantiation $\underline{v}^* \in D$ is a *best solution* if the degree of joint satisfaction of all the constraints is the maximal possible, that is

$$C((c_1,...c_N)\, \underline{v}^*) = \max \{ C((c_1,...c_N), \underline{v}) \mid \underline{v} \in D \}.$$

When an FCSP is given, one can be interested in finding a best solution, or in a *globally good enough solution* — a solution for which the joint satisfaction exceeds a given minimum —, or in a *locally good enough* solution — an instantiation for which certain individual constraints are satisfied better than a given minimum level. The threshold can be given in an absolute way, or as a percentage of the degree of satisfaction of the (theoretical) perfect solution.

3. Constructing a best solution

The solution mechanism for finding a best or good enough solution to be discussed below can be applied for FCSPs with any of the above definitions of fuzzy solutions. More precisely, we will exploit only two characteristics of the definition of the degree of joint satisfaction:

(I) The degree of joint satisfaction can be computed on the basis of the satisfaction of the individual constraints, independent of each other.

(II) The joint satisfaction is monotone with respect to the satisfaction of the individual constraints, that is, if $\underline{v}' \in D$ and $\underline{v}'' \in D$ are such that

$c_i(\underline{v}'_i) = c_i(\underline{v}''_i)$ for $i=1,...N$, $i \neq j$, and
$c_j(\underline{v}'_j) \geq c_j(\underline{v}''_j)$, then
$C((c_1,...c_N), \underline{v}') \geq C((c_1,...c_N), \underline{v}'')$.

3.1 Heuristic search

The most common way to solve a CSP is to construct a solution by extending a partial solution, that is instantiating variables one after the other in such a way that all the constraints which can be evaluated are satisfied. If a dead-end situation occurs (for the variable on turn no appropriate value can be found), backtracking has to be performed. As extensively discussed in the CSP literature [13], the order in which the variables are instantiated and the order in which the possible values are tried have an essential effect on the number of steps required to construct a solution. The critical selections (of a variable to be instantiated and of a value to be assigned) are often performed on the basis of the general idea of 'instantiating the most constrained variables first', and 'assigning the least constraining values first'. For possible measures for these concepts, see [13, 14].

The same principles can be used when solving FCSPs. There can be different measures used for guiding the variable and value selection when exploring the search space in the case of a FCSP solving as well. Below we discuss the most straightforward possibilities.

The *appropriateness of a value* $v \in D_i$ for a variable x_i is evaluated on the basis of the degree of the best possible joint satisfaction of the constraints referring to x_i. That is,

$$a_i(v) = \max \{ C((c_{il_1},...c_{il_i}), \underline{v}) \mid \underline{v} \in D_{il_1} x ... x D_{il_{k-1}} x \{v\} x D_{il_{k+1}} x ... x D_{il_i} \}.$$

Because of (I), $a_i(v)$ can be computed on the basis of $dom_{il_1}(v),...dom_{il_i}(v)$, that is the best possible satisfaction of the individual constraints referring to x_i.

The *difficulty of a variable* x_i is evaluated on the basis of the evaluation of the possible values for the variable. That is,

$$d_i = \sum_{v \in D_i} a_i(v).$$

The search for a best solution of a given FCSP, using the above evaluations for variable and value selection, considers variables first for which few good values exist, and selects one of the best values for the variable. Note, however, that in general the algorithm does not guarantee that the constructed solution is a best one. Another problem is that backtracking takes place only if it is known that any extension of the current partial solution fully violates the given constraints. This can be proven for a partial solution if the definition of the joint satisfaction is such that the full violation of a subset of the constraints entails the full violation of the entire set of constraints. For the joint satisfactions C_{min} and C_{pro} this is true, but for C_{ave} not.

3.2 Pruning not good enough solutions

The above mentioned anomalies can be cured by extending the solution scheme with pruning the search space, on the basis of the common discrete optimisation technique branch-and-bound [5]. When exploring the search space, an upper bound for the degree of satisfaction of the best possible extension of a given partial solution is computed. As a consequence of (II), the degree of satisfaction of a partial solution provides an upper bound for the degree of satisfaction of any extension of the given partial solution.

All partial instantiations (not yet explored leaves of the search tree) for which the computed upper bound is lower than the degree of satisfaction of the best solution found so far are excluded from further considerations. Hence, as soon as a solution is found, some (ideally, all) other possible alternatives may be pruned. The upper limit can be used also to guide the search, not only to prune the search space. Then the partial instantiations with the highest estimate are explored first.

4 An example

The above described solution mechanism will be illustrated on an somewhat revised example taken from [8]. The problem a robot is facing is to get dressed in such a way that the pieces he puts on match. The robot has a choice of sneakers or Cordovans for foot-wear, a choice of denim, blue and grey trousers, and a choice of white or light-grey shirts. The problem can be modelled as an FCSP, with variables f, t and s with domains {S, C}, {D, B, G} and {W, L}, respectively. The matching degree of the possible pairs is expressed as degree of satisfaction of the appropriate binary fuzzy constraints (given in Fig. 1). The definition of joint satisfaction is based on the 'production combination' principle.

It can be seen that this problem has no perfect solution, and the only best solution is f=S, t=D, s=W with a joint satisfaction degree 0.7. There are 12 possible instantiations of the variables, and only two of them violate fully the prescribed constraints. In general, there are 72 different ways to construct a full instantiation (each complete instantiation can be constructed in 6 different ways, depending on the order of instantiating the variables). By applying the heuristics described above, the best solution is constructed at first, in 3 steps.

constraints	satis-faction	f	t	s
c_1	1	S	D	
	0.4	S	B	
	0.2	S	G	
	0.8	C	G	
	0.5	C	B	
c_2	1	S		L
	0.7	S		W
	1	C		W
	0.1	C		L
c_3	1		D	W
	0.7		D	L
	1		B	W
	0.4		B	L
	1		G	L
	0.6		G	W

Fig.1
The fuzzy CSP model of the robot-dressing problem (Not indicated pairs fully violate the appropriate constraint.)

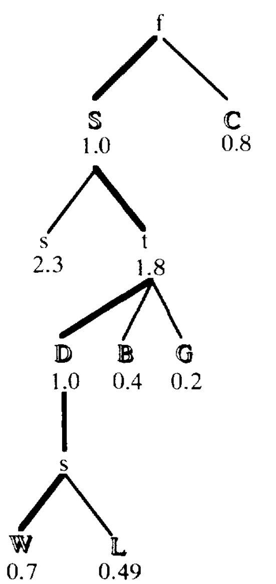

Fig. 2.
The construction of the first solution, based on the evaluation of possible values and uninstantiated variables

The solution process is illustrated in Fig .2. In order to decide which variable to instantiate first, the possible values for each variable are evaluated. It can be seen that

$a_f(S)=1$, $a_f(C)=0.8$, and thus $d_f=1.8$;

$a_t(D)=1$, $a_t(B)=0.5$, $a_t(G)=0.8$, and thus $d_t=2.3$;

$a_s(L)=1$, $a_s(W)=1$, and thus $d_s=2$.

Hence, the most critical variable, f is instantiated first to the best possible value S. For selecting the second variable to be instantiated, the evaluations are as follows (note that some of them have decreased due to the fact that f=S):

$a_t(D)=1$, $a_t(B)=0.4$, $a_t(G)=0.2$, and thus $d_t=1.6$;

$a_s(L)=1$, $a_s(W)=0.7$, and thus $d_s=1.7$.

As a result, t is the next variable to be instantiated, and the best possible value, D is assigned to it. Finally, for the remaining variable s the best value W is chosen. The degree of satisfaction of the solution is 0.7.

Though the found solution is quite good, it has to be checked if it is a best one. At each value selection step, the value evaluations —multiplied by the degree of satisfaction of those constraints which can be evaluated due to earlier instantiations — serve as an upper bound for the degree of satisfaction of the best possible extension, assuming the value in question.

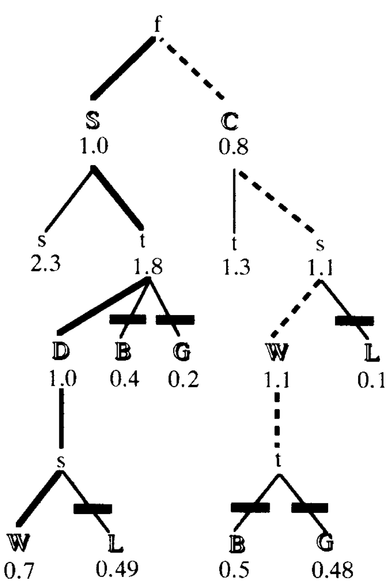

Fig. 3.
Pruning poor possible values (— symbol), and exploring the remaining choices (- - - line).

As the estimates for alternative values for t and s are all lower than the degree of satisfaction of the found solution, these alternatives need not to be explored, the search space can be pruned (see Fig. 3).

The alternative value for the variable f has to be considered, as $a_f(C)=0.8$. Exploring this assignment, it can be seen that the next variable to be instantiated is s, and the best assignment is s=W. The other alternative for s, L is pruned as $a_s(L)=0.1$, hence no better solution than the already found one exists for which s=L. However, calculating the estimates for the possible values for the remaining, not yet instantiated variable t, it is obvious that for any of the possible values the degree of satisfaction is below 0.7. We need not explore alternatives corresponding to different variable orders, as for both possible values for f we have shown that there is at most one candidate for a better solution than the already constructed one.

5. Discussion

We have shown that fuzzy set theory provides a good and general mathematical framework to define fuzzy constraints, FCSPs and the degree of satisfaction of solutions, and that the classical constructive solution method can be adapted for the FCSP case. The proposed heuristics can be used with any definition of the degree of solution for which the independent computation (I) and the monotonicity (II) criteria hold.

Space did not allow us to discuss in this paper further generalisations of classical CSP concepts and solution principles. For a-priory analysis of the FCSP and filtering of the domains, local consistency can be generalised on the basis of the α-support sets, and can be used to estimate the degree of satisfaction of the best solution. However, while in the classical case local consistency is necessary, but not sufficient condition for the existence of a solution, in the case of a FCSP the necessity depends on the definition of the degree of joint satisfaction. In our definitions, only for the conjunctive combination is local α-consistency necessary for the existence of a solution with a degree of satisfaction not smaller than α.

The topology of the constraint graph can play a similar role as in the case of crisp CSP in characterising the difficulty of the problem. It is interesting to investigate if the number and length of cycles in a fuzzy graph representing an FCSP can be used to characterise the difficulty of the problem, similarly to the case of crisp CSPs.

Further fuzzy set theorethical concepts, such the as cardinality of a constraint or the different definitions to characterise the fuzziness of the support sets may also be used to estimate the difficulty of an FCSP. The outlined branch-and-bound search could be further improved by introducing heuristics for the order of checking the constraints.

Domains for variables with a given, natural metric are special, as the degree of satisfaction for tuples for a fuzzy constraint can be defined on the basis of the distance of the tuple from the set of tuples which fully satisfy a given crisp constraint. Hence, the fuzzy relaxation of a crisp constraint could be generated automatically, even in an interwoven way with the solution process. The distance of fuzzy sets [15] could be used to indicate the distance of the relaxed problem — an FCSP — from the original CSP.

A challenging and from a modelling point of view very useful further generalisation of the crisp CSP would be to add one more factor of fuzziness, by considering fuzzy constraints over fuzzy domains.

Acknowledgement

Thanks for Paul ten Hagen and Wojtek Kowalczyk for his constructive comments on this paper. The paper was written in the framework of the project SKBS A3.

References

[1] Borning, A., Maher, M., Martindale, A., Wilson, M.: Constraint hierarchies and logic programming, Proc. of the Sixth International Logic Programming Conference, Lisbon, 1989. pp 149-164.

[2] Dubois, D. and Prade, H.: Fuzzy Sets and Systems, Academic Press, New York, 1980.

[3] Dubois, D., Fargier, H. and Prade, H.: Propagation and satisfaction of flexible constraints, in: Yager, R.R. and Zadeh, L. A. (eds.): Fuzzy Sets, Neural Networks and Soft Computing, Kluwer Acad. Publ. 1993.

[4] Fargier, H., Lang, J. and Schiex, T.: Selecting preferred solutions in fuzzy constraint satisfaction problems, Proc. of the First European Congress on Fuzzy and Intelligent Technologies (EUFIT'93), Aachen, Germany, 1993.

[5] Foulds, L.R.: Optimization Techniques, Springer-Verlag, 1981, pp. 150-162.

[6] Fox, M. S., Sadeh, N., Baykan, C.: Constrained Heuristic Search, 11th IJCAI, Detroit, 1989, pp 309-315.

[7] Freeman-Benson, B. N., Maloney, J., and Borning, A.: An incremental constraint solver, Communications of ACM, Vol. 33. no. 1. 1990. pp 54-63.

[8] Freuder, E. C. and Wallace, R. J.: Partial constraint satisfaction, Artificial Intelligence 59 (1992) pp 21-70.

[9] Hower, W.: Proper Constraint Relaxation, Proc. of the 4th Portugese Conference on AI, Lisboa, Portugal, 1989.

[11] Martin-Clouraire, R.: Dealing with soft constraints in a constraint satisfaction problem, Proc. of the 4th International Conf. on Information Processing of Uncertainty in Knowledge-Based System, Palma de Mallorca, 1992, pp 37-40.

[12] Martin-Clouraire, R.: CSP techniques with fuzzy linear constraints: practical issues, Proc. of the First European Congress on Fuzzy and Intelligent Technologies (EUFIT'93), Aachen, Germany, 1993.

[13] Meseguer, P.: Constraint Satisfaction Problems: An Overview, AICOM, Vol. 2. no. 1, March 1989, pp. 3-17.

[14] Nudel, B.: Consistent-Labeling Problems and their Algorithms: Expected Complexities and Theory Based Heuristics, Artificial Intelligence, 21. 1983. pp 135-178.

[15] Rosenfeld, A.: Distances between fuzzy sets, Computer Science Technical Report Series, CS-TR-1448 University of Maryland, Maryland, 1984.

[16] Ruttkay, Zs., Sturctured Constraint Satisfaction, Technical Report IR-316, Dept. of Mathematics and Comp. Science, Free University Amterdam.

[17] Schiex, T.: Possibilistic constraint satisfaction problems or how to handle soft constraints, Proc. of the 8th Con. on Uncertainity in AI, Stanford, 1992. pp 268-275.