From Crisp to Fuzzy Constraint Networks

Massimiliano Giacomin

Università di Brescia
Dipartimento di Elettronica per l'Automazione
Via Branze 38, I-25123 Brescia, Italy
giacomin@ing.unibs.it

Abstract. Several instances of the CSP framework have been proposed to handle specific problems, such as temporal and spatial reasoning formalisms. This paper, standing at a high level of abstraction, studies the problem of extending these specific instances in order to handle fuzzy constraints. The concepts of class of networks, property, algorithm and the relevant fuzzy extensions are defined. Some application examples in the field of temporal reasoning are provided.

1 Introduction

The idea that a fuzzy constraint satisfaction problem (FCSP) can be viewed as a collection of crisp constraint satisfaction problems (CSPs), corresponding to its α-level cuts, is not new in the literature [7, 8]. This link is the reason why basic results in classical CSPs extend right away to FCSPs, such as local consistency concepts and algorithms. For instance, in [5] a general algorithm to compute an optimal solution of a FCSP is proposed, which decomposes the fuzzy problem into a set of classical ones; the arc-consistency algorithm introduced in [7] is a direct generalization of the classical one, and so on.

While CSP (and FCSP as well) is a general framework independent of the semantics of the specific problem, several instances of it have deserved specific attention in the literature, such as the frameworks introduced for temporal reasoning and spatial reasoning. In these frameworks, variables are used to represent temporal (or spatial) entities, they take values in particular domains (e.g. real numbers for points in [6], couples of real numbers for intervals in [1]), and are related by constraints which have a restricted shape (e.g. a set of intervals in [6], a subset of 13 basic relations in [1]). As a consequence, specialized algorithms are developed for these frameworks, which often benefit of specific properties of the CSP instance: an interesting (and useful) question is then to what extent they can be generalized when elastic constraints replace crisp ones.

The aim of this paper is to study this problem standing at a high level of abstraction: frameworks are simply represented by *classes* of CSPs, which implicitly identify their possible *fuzzy extensions*. The concepts of *property* and *algorithm* are introduced for classes of crisp networks with simple and general definitions; by defining the notion of their fuzzy extension, it shown that theorems of CSPs and properties of algorithms hold for FCSPs as well. Finally, some examples of application in the field of temporal reasoning are provided, which allow us to prove some results already known in the literature as well as new ones.

2 Background

We will represent FCSPs by means of *Fuzzy Constraint Networks* (FCNs). A FCN $\mathcal{N}$ is a triple $< \mathcal{X}, \mathcal{D}, \mathcal{C} >$ where $\mathcal{X} = \{x_1, \ldots, x_n\}$ is a set of variables, $\mathcal{D} = \{D_1, \ldots, D_n\}$ is the set of the relevant domains, and $\mathcal{C} = \{C_1, \ldots, C_m\}$ is a set of fuzzy constraints, with $C_i =< V(C_i), \mathcal{R}_i >$, $V(C_i) = \{y_1, \ldots, y_k\} \subseteq \mathcal{X}$, and $\mathcal{R}_i : D'_1 \times \ldots \times D'_k \rightarrow [0, 1]$ a fuzzy relation expressing the level of satisfaction of the relevant constraint, where $\{D'_1, \ldots, D'_k\}$ are the domains of the variables in $V(C_i)$ [7]. Given a partial instantiation $\overline{d}$ of a set of variables $\mathcal{Y} \subseteq \mathcal{X}$, its degree of local consistency, denoted as cons($\overline{d}$), expresses the degree of satisfaction of the involved constraints, and is defined as $\min_{\mathcal{R}_i | V(C_i) \subseteq \mathcal{Y}} \mathcal{R}_i(\overline{d}^{\downarrow V(C_i)})$. Tuples $\overline{d} \in D_1 \times \ldots \times D_n$ are called solutions of $\mathcal{N}$: they satisfy the constraints of $\mathcal{N}$ to the degree $\deg(\overline{d}) = \mathrm{cons}(\overline{d}) = \min_{i=1 \ldots m} \mathcal{R}_i(\overline{d}^{\downarrow V(C_i)})$. The solution set of $\mathcal{N}$ is the fuzzy set $\mathrm{SOL}(\mathcal{N}) : D_1 \times \ldots \times D_n \rightarrow [0, 1]$ defined by the membership function $\deg(\overline{d})$. The consistency degree of $\mathcal{N}$ denotes the satisfaction degree of the best complete instantiations: $\mathrm{CONS}(\mathcal{N}) = \sup_{\overline{d} \in D_1 \times \ldots \times D_n} \deg(\overline{d})$. Optimal solutions are solutions $\overline{d}$ such that $\deg(\overline{d}) = \mathrm{CONS}(\mathcal{N})$. Two fuzzy constraint networks having the same variables and relevant domains are *equivalent* if they have the same solution set.

Constraint propagation algorithms for FCNs are used to render network constraints more explicit, enforcing consistency of sub-networks. The notion of local consistency is expressed by the property of k-consistency: a FCN $\mathcal{N}$ is k-consistent iff any instantiation of $k - 1$ variables can be extended to an instantiation involving any kth variable which has the same degree of local consistency. More formally:

Definition 1. *A FCN $\mathcal{N}$ is k-consistent iff*

$$\forall \mathcal{Y} = \{y_1, \ldots, y_{k-1}\} \subseteq \mathcal{X}, \forall y_k \in \mathcal{X} \text{ such that } y_k \notin \mathcal{Y}, \forall \overline{d} \in D'_1 \times \ldots \times D'_{k-1}, \\ \exists d_k \in D'_k \text{ such that } cons(\overline{d} d_k) = cons(\overline{d})$$

where $D'_1, \ldots, D'_k$ are the domains of $y_1, \ldots, y_k$ respectively.

Path-consistency can be identified with 3-consistency, while strong k-consistency is j-consistency for every $j \leq k$. The minimal network of a FCN $\mathcal{N}$ is the 'most explicit network' among the equivalent ones. While the concept of minimality can be defined for general networks, in this paper, for the sake of simplicity, we will consider minimality of binary FCNs, i.e. those having binary constraints only. In this case, minimality holds iff any instantiation of two variables can be extended to a solution of the network maintaining its degree of satisfaction. More formally:

Definition 2. *A binary FCN $\mathcal{N}$ is minimal iff*

$$\forall \mathcal{Y} = \{x_i, x_j\} \subseteq \mathcal{X}, \forall \overline{d} \in D_i \times D_j,$$
$$\exists \overline{d'} \in D_1 \times \ldots \times D_n,\ \overline{d'}^{\downarrow \mathcal{Y}} = \overline{d}$$
$$\textit{such that } deg(\overline{d'}) = cons(\overline{d})$$

A classical *Crisp Constraint Network* (CCN) can be seen as a FCN whose preference degrees take values in $\{0, 1\}$ only. In particular, relations $\mathcal{R}_i$ are crisp sets expressing admissible tuples, and $\mathrm{SOL}(\mathcal{N})$ is the set of complete instantiations which satisfy all constraints. k-consistency and minimality are defined referring to the possibility of extending *consistent* instantiations preserving consistency.

3 Classes of Constraint Networks: from Crisp to Fuzzy Reasoning Frameworks

Constraint-based models of information and reasoning represent scenarios of interest by means of variables, which denote specific entities, and constraints, which denote possible or desired relations among them. In order to stand at a high level of abstraction, we identify a constraint-based reasoning framework with the set of all possible networks that represent its specific problems:

Definition 3. *A class of crisp (fuzzy) constraint networks $\mathcal{HN}$ ($\mathcal{FN}$) is a possibly infinite set of crisp (fuzzy) constraint networks.*

Example 1. TCSPs [6] model quantitative temporal information by means of binary CCNs: variables are time points whose values are real numbers, and each constraint between x_i and x_j is a set $I = \{[a_1, b_1], \ldots, [a_n, b_n]\}$ of disjoint intervals constraining $x_j - x_i$. Therefore, TCSPs define a class of CCNs, that we call $\mathcal{HN}_{\mathrm{TCSP}}$.

Example 2. In Allen's Interval Algebra [1] the temporal entities are intervals, interpreted as elements of R^2, and their binary constraints are modeled by disjunctions of qualitative basic relations: there are 13 basic relations, such as 'before', 'during', etc., each of them denoting the values in $R^2 \times R^2$ assumed by pairs of intervals which are related by it. We call the relevant class of CCNs $\mathcal{HN}_{\mathrm{IA}}$.

In order to model soft constraints, which characterize almost all real world problems, several classical reasoning frameworks have been generalized to handle fuzzy constraints. As it will be shown in the following, these extensions rely on the notion of α-cut of a fuzzy set:

Definition 4. *Given a fuzzy set defined by the membership function $\mathcal{R}^{fuz} : D_1 \times \ldots \times D_k \to [0, 1]$, and given a real number $\alpha \in [0, 1]$, the α-cut of $\mathcal{R}^{fuz}$ is the crisp set ${\mathcal{R}^{fuz}}_\alpha \subseteq D_1 \times \ldots \times D_k = \left\{\overline{d} \in D_1 \times \ldots \times D_k \mid \mathcal{R}^{fuz}(\overline{d}) \geq \alpha\right\}$*

We extend the definition of α-cut to fuzzy constraints and fuzzy constraint networks. Given a fuzzy constraint $C^{\text{fuz}} =< V(C^{\text{fuz}}), \mathcal{R}^{\text{fuz}} >$ and given a number $\alpha \in [0,1]$, the α-cut of C^{fuz} is the crisp constraint ${C^{\text{fuz}}}_{\alpha} =< V(C^{\text{fuz}}), {\mathcal{R}^{\text{fuz}}}_{\alpha} >$. The α-cut of a fuzzy constraint network is the crisp constraint network obtained from the original one by α-cutting the original constraints:

Definition 5. *Given a fuzzy constraint network $\mathcal{N}^{fuz} =< \mathcal{X}, \mathcal{D}, \mathcal{C}^{fuz} >$, where $\mathcal{C}^{fuz} = \{C_1, \ldots, C_m\}$, the α-cut of $\mathcal{N}^{fuz}$ is the crisp constraint network ${\mathcal{N}^{fuz}}_{\alpha} =< \mathcal{X}, \mathcal{D}, {\mathcal{C}^{fuz}}_{\alpha} >$ where ${\mathcal{C}^{fuz}}_{\alpha} = \{C_{1\alpha}, \ldots, C_{m\alpha}\}$.*

A fundamental property of α-cuts is that they uniquely identify the original set:

Proposition 1. *Let $C_1 =< V(C_1), \mathcal{R}_1 >$ and $C_2 =< V(C_2), \mathcal{R}_2 >$ two fuzzy constraints. If $\forall \alpha \in [0,1]$ $C_{1\alpha} = C_{2\alpha}$ then $C_1 = C_2$.*

Proof. By Definition of α-cut, $V(C_1) = V(C_2)$: let $V(C_1) = \{D_1, \ldots, D_k\}$. We have to prove that $\mathcal{R}_1 = \mathcal{R}_2$, i.e. that $\forall \overline{d} \in D_1 \times \ldots \times D_k$ $\mathcal{R}_1(\overline{d}) = \mathcal{R}_2(\overline{d})$.
We reason by contradiction. Suppose that, for a $\overline{d} \in [0,1]$, $\mathcal{R}_1(\overline{d}) = \alpha$, $\mathcal{R}_2(\overline{d}) = \beta$ and $\alpha \neq \beta$; without loss of generality, assume $\alpha > \beta$. We have that $\overline{d} \in \mathcal{R}_{1\alpha}$ and $\overline{d} \notin \mathcal{R}_{2\alpha}$, therefore $\mathcal{R}_{1\alpha} \neq \mathcal{R}_{2\alpha}$: contradiction!

Proposition 2. *Let $\mathcal{N}_1$ and $\mathcal{N}_2$ be two fuzzy constraint networks. If $\forall \alpha \in [0,1]$ $\mathcal{N}_{1\alpha} = \mathcal{N}_{2\alpha}$ then $\mathcal{N}_1 = \mathcal{N}_2$.*

As it will be shown in the following, the main results provided in the paper rely on the following proposition:

Proposition 3. *Given a fuzzy constraint network $\mathcal{N}$, we have that $[SOL(\mathcal{N})]_{\alpha} = SOL(\mathcal{N}_{\alpha})$.*

Proof. A generic tuple $\overline{d} \in [\mathrm{SOL}(\mathcal{N})]_{\alpha}$ iff $\deg(\overline{d}) \geq \alpha$, i.e. $\min_{i=1\ldots m} \mathcal{R}_i(\overline{d}) \geq \alpha$. The last expression is equivalent to $\forall i = 1, \ldots, m$ $\mathcal{R}_i(\overline{d}) \geq \alpha$, and, by the definition of α-cut, this happens iff $\forall i = 1, \ldots, m$ $\overline{d} \in \mathcal{R}_{i\alpha}$. By the definition of α-cut of a FCN, this is equivalent to $\overline{d} \in \mathrm{SOL}(\mathcal{N}_{\alpha})$, and we are done.

Corollary 1. *Given two fuzzy constraint networks $\mathcal{N}_1$ and $\mathcal{N}_2$, if it is the case that $\forall \alpha \in [0,1]$ $SOL(\mathcal{N}_{1\alpha}) = SOL(\mathcal{N}_{2\alpha})$, then $\mathcal{N}_1$ and $\mathcal{N}_2$ are equivalent.*

We are now in position to introduce the notion of fuzzy extension of a class of CCNs. The meaning of 'extension' is that the domain-dependent entities are maintained in the fuzzy framework, as well as their domains, while the constraints becomes 'elastic', since they are modeled as fuzzy sets. This is captured by the following definitions:

Definition 6. *Given a class of FCNs $\mathcal{FN}$, its* crisp projection *is the class of CCNs defined as*

$$\mathcal{C}(\mathcal{FN}) = \left\{ {\mathcal{N}^{fuz}}_{\alpha} \mid \alpha \in [0,1], \mathcal{N}^{fuz} \in \mathcal{FN} \right\}$$

Definition 7. *Given a class of CCNs $\mathcal{HN}$ and a class of FCNs $\mathcal{FN}$, $\mathcal{FN}$ is a fuzzy extension of $\mathcal{HN}$ iff $\mathcal{C}(\mathcal{FN}) \subseteq \mathcal{HN}$. We refer to the set of all possible fuzzy extensions of $\mathcal{HN}$ as $\mathcal{F}(\mathcal{HN})$.*

The above definition does not constrain a fuzzy extension to be the maximal one; when this occurs the latter is denoted as the proper one:

Definition 8. *$\mathcal{FN}$ is the proper fuzzy extension of $\mathcal{HN}$ iff $\mathcal{C}(\mathcal{FN}) = \mathcal{HN}$*

Example 3. The proper fuzzy extension of $\mathcal{HN}_{\mathrm{TCSP}}$, called $\mathcal{FN}_{\mathrm{TCSP}}$, is the class of binary FCNs whose variables take real values and whose constraints are represented as $< I, f >$, where $f : I \rightarrow [0, 1]$ is a preference function. This is the fuzzy instantiation of the TCSPPs framework defined in [9].

Example 4. The proper fuzzy extension of $\mathcal{HN}_{\mathrm{IA}}$ corresponds to our $\mathrm{IA}^{\mathrm{fuz}}$ algebra [3, 2], where degrees of preference in $[0, 1]$ are attached to Allen's atomic relations.

4 Extending Tractable Classes

The basic task in a crisp constraint reasoning framework is, given a problem, to check its consistency and to find a solution. If this can be done in polynomial time, we say that the relevant class is tractable:

Definition 9. *A class of CCNs $\mathcal{HN}$ is tractable if there exists an algorithm $SOLALG_{\mathcal{HN}}$ that, given a network $\mathcal{N} \in \mathcal{HN}$, $\mathcal{N} =< \mathcal{X}, \mathcal{D}, \mathcal{C} >$, computes in polynomial time the following result:*

$$SOLALG_{\mathcal{HN}}(\mathcal{N}) = \begin{cases} \overline{d} : \overline{d} \in SOL(\mathcal{N}) \text{ if } SOL(\mathcal{N}) \neq \emptyset \\ FAILED \text{ otherwise} \end{cases}$$

Consistency checking in CCN*s* corresponds in FCN*s* to the problem of finding an optimal solution of a given network. There are classes of CCNs, such as the previously defined $\mathcal{HN}_{\mathrm{TCSP}}$ and $\mathcal{HN}_{\mathrm{IA}}$, that are not tractable: of course, their proper fuzzy extensions are not tractable as well. An interesting question concerns the tractability of classes of FCNs when they extend tractable classes of CCNs: in the following, we show that tractability holds if a constraint on the cardinality of the set of used preference degrees is satisfied.

Definition 10. *Let $\mathcal{N} =< \mathcal{X}, \mathcal{D}, \mathcal{C} >$ be a fuzzy constraint network. For any $C_i =< V(C_i), \mathcal{R}_i > \subseteq \mathcal{C}$, $i \in \{1, \ldots, m\}$, let $COD(\mathcal{R}_i)$ be the codomain of $\mathcal{R}_i$, i.e. $COD(\mathcal{R}_i) = \bigcup_{\overline{d} \in D'_1 \times \ldots \times D'_k} \mathcal{R}_i(\overline{d})$, where $D'_1, \ldots, D'_k$ are the domains of the variables in $V(C_i)$. The set of preference degrees of $\mathcal{N}$, $SEDG(\mathcal{N}) \subseteq [0, 1]$, is defined as $SEDG(\mathcal{N}) = \bigcup_{i=1,\ldots,m} COD(\mathcal{R}_i)$.*

Proposition 4. *Let $\mathcal{N}$ be a fuzzy constraint network. If $SEDG(\mathcal{N})$ is finite, then $\mathcal{N}$ has at least an optimal solution.*

Proof. Let $\mathcal{D} = \bigcup_{\overline{d} \in D_1 \times \ldots \times D_m} \{\deg(\overline{d})\}$. Since $\mathcal{D} \subseteq \mathrm{SEDG}(\mathcal{N})$, it follows from the hypothesis that $\mathcal{D}$ is finite. As a consequence, $\mathcal{D}$ has a maximum m which corresponds to (at least) one solution: this is an optimal solution.

Proposition 5. *Let $\mathcal{HN}$ be a tractable class of crisp constraint networks, and $\mathcal{FN} \in \mathcal{F}(\mathcal{HN})$ one of its fuzzy extensions. If, for every network $\mathcal{N}$ of $\mathcal{FN}$, the cardinality of SEDG($\mathcal{N}$) is finite and at most exponential in the number of variables of $\mathcal{N}$, then the problem of finding an optimal solution is tractable in $\mathcal{FN}$.*

Proof. By proposition 4, $\mathcal{N}$ has at least an optimal solution. In particular, the set of optimal solutions is

$$[\mathrm{SOL}(\mathcal{N})]_\beta,\ \beta = \max\{\alpha \mid [\mathrm{SOL}(\mathcal{N})]_\alpha \neq \emptyset\} \ = \\ \mathrm{SOL}(\mathcal{N}_\beta),\ \beta = \max\{\alpha \mid \mathrm{SOL}(\mathcal{N}_\alpha) \neq \emptyset\}$$

where Proposition 3 has been used. As a consequence, an optimal solution can be found in polynomial time performing a binary search on SEDG($\mathcal{N}$) (which has a logarithmic complexity), using the algorithm on $\mathcal{HN}$ to check consistency and find a solution of problems $\mathcal{N}_\alpha \mid \alpha \in \mathrm{SEDG}(\mathcal{N})$.

Example 5. A tractable subclass of TCSPs is that made up of Simple Temporal Problems (STPs), where constraints contain single intervals only [6]: let $\mathcal{HN}_{\mathrm{STP}}$ be the corresponding class of CCNs. Its proper extension $\mathcal{FN}_{\mathrm{STP}}$ is characterized by networks with single-interval constraints and semi-convex preference functions (i.e. $f \mid \forall y\ \{x : f(x) \geq y\}$ is an interval). The tractability of this class, proved in [9], follows from Proposition 5 as well.

Example 6. SA^{c} and SA algebras are tractable subclasses of IA [10]. We have defined the corresponding proper extensions in [3]; since the number of preference degrees in IA networks is quadratic in the set of variables, tractability of these classes holds as well.

5 Fuzzy Extension of Properties and Theorems

Besides computing (optimal) solutions, there are other tasks of interests in (fuzzy) reasoning frameworks, such as computing the most explicit constraints of a given network. In order to perform these task, specific algorithms exploit relations among properties of networks: for instance, constraint propagation algorithms enforce local consistency properties, which for particular classes entail minimality. The aim of this section is to show how properties and theorems of CCNs can be extended to FCNs.

Definition 11. *Given a class of (either fuzzy or crisp) constraint networks $\mathcal{GN}$, a property on $\mathcal{GN}$ is a function $\mathcal{P} : \mathcal{GN} \to \{0, 1\}$. We say that a network $\mathcal{N} \in \mathcal{GN}$ has the property $\mathcal{P}$, written $\mathcal{P}(\mathcal{N})$, iff $\mathcal{P}(\mathcal{N}) = 1$.*

Again, we define the fuzzy extension of properties by means of the notion of α-cut:

Definition 12. *Given a class of crisp constraint networks $\mathcal{HN}$ and a property $\mathcal{P}$ on $\mathcal{HN}$, let $\mathcal{FN}$ be a fuzzy extension of $\mathcal{HN}$, i.e. $\mathcal{FN} \in \mathcal{F}(\mathcal{HN})$. The fuzzy extension to $\mathcal{FN}$ of $\mathcal{P}$ is the property on $\mathcal{FN}$ defined as:*

$$\mathcal{P}^{fuz}(\mathcal{N}) = \begin{cases} 1 \text{ if } \forall\alpha \in [0,1]\ \mathcal{P}(\mathcal{N}_\alpha) = 1 \\ 0 \text{ otherwise} \end{cases}$$

While it is not generally the case that a logical function of fuzzy extensions of properties is the extension of their logical function, the following proposition shows that this holds for conjunction:

Proposition 6. *Let $\mathcal{HN}$ be a class of crisp constraint networks, $\mathcal{FN} \in \mathcal{F}(\mathcal{HN})$ a fuzzy extension of $\mathcal{HN}$, and $\mathcal{P}_1$ and $\mathcal{P}_2$ two properties on $\mathcal{HN}$. If $\mathcal{P}_1^{fuz}$ and $\mathcal{P}_2^{fuz}$ are the fuzzy extensions to $\mathcal{FN}$ of $\mathcal{P}_1$ and $\mathcal{P}_2$ respectively, then $\mathcal{P}_c^{fuz} = \mathcal{P}_1^{fuz} \wedge \mathcal{P}_2^{fuz}$ is the fuzzy extension to $\mathcal{FN}$ of $\mathcal{P}_1 \wedge \mathcal{P}_2$.*

Proof. We have to prove that, given $\mathcal{N} \in \mathcal{FN}$,

$$\mathcal{P}_{\mathrm{c}}^{\mathrm{fuz}}(\mathcal{N}) = \begin{cases} 1 \text{ if } \forall\alpha \in [0,1]\ (\mathcal{P}_1 \wedge \mathcal{P}_2)(\mathcal{N}_\alpha) \\ 0 \text{ otherwise} \end{cases}$$

Assuming that $\forall\alpha \in [0,1]\ (\mathcal{P}_1 \wedge \mathcal{P}_2)(\mathcal{N}_\alpha)$, we have that $\forall\alpha \in [0,1]\ \mathcal{P}_1(\mathcal{N}_\alpha) \wedge \mathcal{P}_2(\mathcal{N}_\alpha)$. This entails that both $\forall\alpha \in [0,1]\ \mathcal{P}_1(\mathcal{N}_\alpha)$ and $\forall\beta \in [0,1]\ \mathcal{P}_2(\mathcal{N}_\beta)$: by Definition 12, $\mathcal{P}_1^{\mathrm{fuz}}(\mathcal{N}) \wedge \mathcal{P}_2^{\mathrm{fuz}}(\mathcal{N})$ is verified, i.e. $\mathcal{P}_{\mathrm{c}}^{\mathrm{fuz}}(\mathcal{N}) = 1$.
On the other hand, assume that $\exists\alpha \in [0,1] : (\mathcal{P}_1 \wedge \mathcal{P}_2)(\mathcal{N}_\alpha) = 0$. We have that $\exists\alpha \in [0,1] : (\mathcal{P}_1(\mathcal{N}_\alpha) = 0 \vee \mathcal{P}_2(\mathcal{N}_\alpha) = 0)$, i.e. $(\exists\alpha \in [0,1] : \mathcal{P}_1(\mathcal{N}_\alpha) = 0) \vee (\exists\beta \in [0,1] : \mathcal{P}_2(\mathcal{N}_\beta) = 0)$. This means that $(\mathcal{P}_1^{\mathrm{fuz}}(\mathcal{N}) = 0) \vee (\mathcal{P}_2^{\mathrm{fuz}}(\mathcal{N}) = 0)$, then $(\mathcal{P}_1^{\mathrm{fuz}} \wedge \mathcal{P}_2^{\mathrm{fuz}})(\mathcal{N}) = 0$ i.e. $\mathcal{P}_{\mathrm{c}}^{\mathrm{fuz}}(\mathcal{N}) = 0$.

The appropriateness of Definition 12 is confirmed by the fact that several local consistency properties, as defined in FCSP framework, are the fuzzy extensions of the corresponding ones in CSPs. In the following, we consider those mentioned in Section 2.

Proposition 7. *Let $\mathcal{HN}$ be a class of crisp constraint networks, and $\mathcal{FN} \in \mathcal{F}(\mathcal{HN})$ one of its fuzzy extensions. Let $\mathcal{P}_{k-cons}$ and $\mathcal{P}^{fuz}{}_{k-cons}$ be the properties of k-consistency defined on $\mathcal{HN}$ and $\mathcal{FN}$, respectively: $\mathcal{P}^{fuz}{}_{k-cons}$ is the fuzzy extension to $\mathcal{FN}$ of $\mathcal{P}_{k-cons}$.*

Proof. In this proof, we will refer to $\mathcal{Y}$ and y_k as the generic set of $k-1$ variables and the generic k-th variable of Definition 1, respectively. Moreover, given a network (either crisp or fuzzy) $\mathcal{N}$, we will denote the subnetwork of $\mathcal{N}$ involving the variables of $\mathcal{Y}$ as $\mathcal{N}^{k-1}$, and that involving the variables of $\mathcal{Y} \cup \{y_k\}$ as $\mathcal{N}^{k+1}$.
We have to prove that, given a FCN $\mathcal{N} \in \mathcal{FN}$, $\mathcal{N}$ is k-consistent iff $\forall\alpha \in [0,1]\ \mathcal{N}_\alpha$

is k-consistent.
First, let us suppose that $\mathcal{N}$ is k-consistent, and let us consider a generic $\alpha \in [0,1]$. Let $\overline{d}$ be a consistent assignment in $\mathcal{N}_\alpha$ of variables in $\mathcal{Y}$, i.e. $\overline{d} \in \mathrm{SOL}({\mathcal{N}_\alpha}^{k-1})$. By Proposition 3, $\overline{d} \in [\mathrm{SOL}(\mathcal{N}^{k-1})]_\alpha$. Since $\mathcal{N}$ is k-consistent, $\overline{d}$ can be extended to an instantiation $\overline{d'}$ involving y_k such that $\overline{d'} \in [\mathrm{SOL}(\mathcal{N}^{k})]_\alpha$. By Proposition 3 again, $\overline{d'} \in \mathrm{SOL}({\mathcal{N}_\alpha}^{k})$, and we are done.
Now, let us suppose that $\forall \alpha \in [0,1]$ $\mathcal{N}_\alpha$ is k-consistent, and let $\overline{d}$ be a generic instantiation in $\mathcal{N}$ of variables in $\mathcal{Y}$ with $\mathrm{cons}(\overline{d}) = \beta$, therefore $\overline{d} \in [\mathrm{SOL}(\mathcal{N}^{k-1})]_\beta$. By Proposition 3, $\overline{d} \in \mathrm{SOL}({\mathcal{N}_\beta}^{k-1})$. Since $\mathcal{N}_\beta$ is k-consistent, $\overline{d}$ can be extended to an instantiation $\overline{d'}$ involving y_k such that $\overline{d'} \in \mathrm{SOL}({\mathcal{N}_\beta}^{k})$. By Proposition 3 again, $\overline{d'} \in [\mathrm{SOL}(\mathcal{N}^{k})]_\beta$, and we are done.

The above proposition holds for path-consistency and strong k-consistency as well: path-consistency is simply k-consistency with $k = 3$, while the claim for strong k-consistency follows by taking into account Proposition 6.

Proposition 8. *Let $\mathcal{HN}$ be a class of binary crisp constraint networks, and $\mathcal{FN} \in \mathcal{F}(\mathcal{HN})$ one of its fuzzy extensions. Let $\mathcal{P}_{min}$ and ${\mathcal{P}^{fuz}}_{min}$ be the properties of minimality defined on $\mathcal{HN}$ and $\mathcal{FN}$, respectively: ${\mathcal{P}^{fuz}}_{min}$ is the fuzzy extension to $\mathcal{FN}$ of $\mathcal{P}_{min}$.*

Proof. We have to prove that, given a FCN $\mathcal{N} \in \mathcal{FN}$, $\mathcal{N}$ is minimal iff $\forall \alpha \in [0,1]$ $\mathcal{N}_\alpha$ is minimal.
First, let us suppose that $\mathcal{N}$ is minimal, and let us consider a generic $\alpha \in [0,1]$. Let $\overline{d} = (d_i, d_j)$ be an assignment of x_i and x_j in $\mathcal{N}_\alpha$ which satisfies the binary constraint $< \{x_i, x_j\}, {\mathcal{R}_{ij}}_\alpha >$ between them, i.e. $\mathrm{cons}(\overline{d}) = \beta \geq \alpha$. Since $\mathcal{N}$ is minimal, $\overline{d}$ can be extended to a solution $\overline{d'} \in D_1 \times \ldots \times D_n$ such that $\deg(\overline{d'}) = \beta \geq \alpha$, therefore $\overline{d'} \in [\mathrm{SOL}(\mathcal{N})]_\alpha$. By Proposition 3, $\overline{d'} \in \mathrm{SOL}(\mathcal{N}_\alpha)$, and we are done.
Now, let us suppose that $\forall \alpha \in [0,1]$ $\mathcal{N}_\alpha$ is minimal, and let $\overline{d} = (d_i, d_j)$ be an assignment of x_i and x_j, with $\mathrm{cons}(\overline{d}) = \beta$, so that $\overline{d}$ satisfies the binary constraint $< \{x_i, x_j\}, {\mathcal{R}_{ij}}_\beta >$ of $\mathcal{N}_\beta$. Since $\mathcal{N}_\beta$ is minimal, $\overline{d}$ can be extended to a solution $\overline{d'} \in D_1 \times \ldots \times D_n$ such that $\overline{d'} \in \mathrm{SOL}(\mathcal{N}_\beta)$. By Proposition 3, $\overline{d'} \in [\mathrm{SOL}(\mathcal{N})]_\beta$, i.e. $\deg(\overline{d'}) \geq \beta$, and we are done.

The concept of fuzzy extension of properties allows us to directly extend theorems that are valid for a class of CCNs to its fuzzy extensions:

Proposition 9. *Let $\mathcal{HN}$ be a class of crisp constraint networks, and let $\mathcal{FN} \in \mathcal{F}(\mathcal{HN})$. If we have a valid theorem in $\mathcal{HN}$ of the form*

$$\textit{For every } \mathcal{N} \in \mathcal{HN}, \textit{ if } \mathcal{P}_1(\mathcal{N}) \textit{ then } \mathcal{P}_2(\mathcal{N})$$

(where $\mathcal{P}_1$ and $\mathcal{P}_2$ are properties on $\mathcal{HN}$) then the following theorem in $\mathcal{FN}$ is valid:

$$\textit{For every } \mathcal{N}^{fuz} \in \mathcal{FN}, \textit{ if } \mathcal{P}_1^{fuz}(\mathcal{N}^{fuz}) \textit{ then } \mathcal{P}_2^{fuz}(\mathcal{N}^{fuz})$$

where $\mathcal{P}_1^{fuz}$ and $\mathcal{P}_2^{fuz}$ are the fuzzy extensions to $\mathcal{FN}$ of $\mathcal{P}_1$ and $\mathcal{P}_2$.

Proof. Assume the hypothesis of the theorem in $\mathcal{FN}$, i.e. $\mathcal{P}_1^{\text{fuz}}(\mathcal{N}^{\text{fuz}})$. By Definition 12 we have that $\forall\alpha \in [0,1]$ $\mathcal{P}_1({\mathcal{N}^{\text{fuz}}}_\alpha)$. By the theorem in $\mathcal{HN}$ it follows that $\forall\alpha \in [0,1]$ $\mathcal{P}_2({\mathcal{N}^{\text{fuz}}}_\alpha)$, i.e. $\mathcal{P}_2^{\text{fuz}}(\mathcal{N}^{\text{fuz}})$.

Example 7. In [6], it is proved that path-consistent STP networks are also minimal. The same theorem is proved in [10] for SA^c networks, while for SA networks it is shown that minimality is entailed by minimality of 4-subnetworks. From Proposition 7, Proposition 8 and Proposition 9, it follows that the same properties hold for $\mathcal{FN}_{\text{STP}}$, $\mathcal{FN}_{\text{SA}^\text{c}}$ and $\mathcal{FN}_{\text{SA}}$, respectively.

6 Extending Algorithms from Crisp to Fuzzy

In this section, we consider the possibility of extending algorithms which compute transformations of networks belonging to a specific class. While the most common examples of these are constraint propagation algorithms, we do not restrict our definition to any particular kind of algorithms:

Definition 13. *Let $\mathcal{GN}$ be a class of (either crisp or fuzzy) constraint networks. A $\mathcal{GN}$-transformation-algorithm $\mathcal{GN}$-T-ALG $\mathcal{A}$ is a function*

$$\mathcal{A} : \mathcal{GN} \to \mathcal{GN}$$

such that if $\mathcal{N} =< \mathcal{X}, \mathcal{D}, \mathcal{C} >$ then $\mathcal{A}(\mathcal{N}) =< \mathcal{X}, \mathcal{D}, \mathcal{C}_{out} >$.

An often desired characteristics of $\mathcal{GN}$-transformation-algorithms is that they compute a network which is always equivalent to the one taken in input (e.g. this is a property of constraint propagation algorithms). Moreover, the function of $\mathcal{GN}$-transformation-algorithms is to enforce a particular property (such as a kind of local consistency) on the output network, sometimes assuming a constraint on possible input networks.

Definition 14. *Given a class of constraint networks $\mathcal{GN}$ and a property $\mathcal{P}$ on $\mathcal{GN}$, a $\mathcal{GN}$-T-ALG $\mathcal{A}$ is equivalence-preserving conditioned on $\mathcal{P}$ ($\mathcal{P}$-EQ) iff $\forall\mathcal{N} \in \mathcal{GN}$ $\mathcal{P}(\mathcal{N}) \to (SOL(\mathcal{A}(\mathcal{N})) = SOL(\mathcal{N}))$*

Definition 15. *Given a class of constraint networks $\mathcal{GN}$ and two properties $\mathcal{P}_1$ and $\mathcal{P}_2$ on $\mathcal{GN}$, a $\mathcal{GN}$-T-ALG $\mathcal{A}$ is $\mathcal{P}_2$-enforcing conditioned on $\mathcal{P}_1$ ($\mathcal{P}_1$-to-$\mathcal{P}_2$) iff $\forall\mathcal{N} \in \mathcal{GN}$ $\mathcal{P}_1(\mathcal{N}) \to \mathcal{P}_2(\mathcal{A}(\mathcal{N}))$*

The notion of fuzzy extension of a $\mathcal{FN}$-transformation-algorithm is captured by the following definition:

Definition 16. *Let $\mathcal{HN}$ be a class of crisp constraint networks, and let $\mathcal{A}$ be a $\mathcal{HN}$-T-ALG. Let $\mathcal{FN}$ be a fuzzy extension of $\mathcal{HN}$. The fuzzy extension to $\mathcal{FN}$ of $\mathcal{A}$ is the $\mathcal{FN}$-T-ALG $\mathcal{A}^{fuz}$ such that*

$$\forall\mathcal{N} \in \mathcal{FN}, \forall\alpha \in [0,1] \quad [\mathcal{A}^{fuz}(\mathcal{N})]_\alpha = \mathcal{A}(\mathcal{N}_\alpha)$$

Proposition 10. *Definition 16 uniquely determines the extension to $\mathcal{FN}$ of a $\mathcal{HN}$-T-ALG.*

Proof. Let us consider the $\mathcal{HN}$-T-ALG $\mathcal{A}$ and its fuzzy extension $\mathcal{A}^{\text{fuz}}$. Definition 16 uniquely determines, given $\mathcal{A}$ and the input network $\mathcal{N}$, $[\mathcal{A}^{\text{fuz}}(\mathcal{N})]_\alpha$ for any $\alpha \in [0,1]$. By Proposition 2, the output $\mathcal{A}^{\text{fuz}}(\mathcal{N})$ is unique.

It turns out that the (possible) property of preserving equivalence and the specific function of a $\mathcal{HN}$-T-ALG are maintained by its fuzzy extension.

Proposition 11. *If a $\mathcal{HN}$-T-ALG $\mathcal{A}$ is $\mathcal{P}$-EQ, then its fuzzy extension $\mathcal{A}^{fuz}$ to $\mathcal{FN} \in \mathcal{F}(\mathcal{HN})$ is $\mathcal{P}^{fuz}$-EQ, where $\mathcal{P}^{fuz}$ is the fuzzy extension to $\mathcal{FN}$ of $\mathcal{P}$.*

Proof. We have to prove that $\forall \mathcal{N} \in \mathcal{FN}\ \mathcal{P}^{\text{fuz}}(\mathcal{N}) \rightarrow (\text{SOL}(\mathcal{A}^{\text{fuz}}(\mathcal{N})) = \text{SOL}(\mathcal{N}))$. By Definition 16, $\forall \alpha \in [0,1]\ [\mathcal{A}^{\text{fuz}}(\mathcal{N})]_\alpha = \mathcal{A}(\mathcal{N}_\alpha)$, therefore in particular $\forall \alpha \in [0,1]\ \text{SOL}([\mathcal{A}^{\text{fuz}}(\mathcal{N})]_\alpha) = \text{SOL}(\mathcal{A}(\mathcal{N}_\alpha))$. Since $\mathcal{P}^{\text{fuz}}$ is the fuzzy extension of $\mathcal{P}$, we have that $\forall \alpha \in [0,1]\ \mathcal{P}(\mathcal{N}_\alpha)$: by the hypothesis that $\mathcal{A}$ is $\mathcal{P}$-EQ, the expression above entails that $\forall \alpha \in [0,1]\ \text{SOL}([\mathcal{A}^{\text{fuz}}(\mathcal{N})]_\alpha) = \text{SOL}(\mathcal{N}_\alpha)$. By Corollary 1, this yields $\text{SOL}(\mathcal{A}^{\text{fuz}}(\mathcal{N})) = \text{SOL}(\mathcal{N})$, and we are done.

Proposition 12. *Let $\mathcal{HN}$ be a class of crisp constraint networks, $\mathcal{A}$ a $\mathcal{HN}$-T-ALG and $\mathcal{P}_1$ and $\mathcal{P}_2$ two properties on $\mathcal{HN}$. Given $\mathcal{FN} \in \mathcal{F}(\mathcal{HN})$, let $\mathcal{A}^{fuz}$, $\mathcal{P}_1^{fuz}$ and $\mathcal{P}_2^{fuz}$ be the fuzzy extensions to $\mathcal{FN}$ of $\mathcal{A}$, $\mathcal{P}_1$ and $\mathcal{P}_2$, respectively. If $\mathcal{A}$ is $\mathcal{P}_1$-to-$\mathcal{P}_2$, then $\mathcal{A}^{fuz}$ is $\mathcal{P}_1^{fuz}$-to-$\mathcal{P}_2^{fuz}$.*

Proof. We have to prove that if $\mathcal{P}_1^{\text{fuz}}(\mathcal{N})$, then $\mathcal{P}_2^{\text{fuz}}(\mathcal{A}^{\text{fuz}}(\mathcal{N}))$. By the hypothesis that $\mathcal{A}$ is $\mathcal{P}_1$-to-$\mathcal{P}_2$, we have that $\forall \alpha \in [0,1]\ (\mathcal{P}_1(\mathcal{N}_\alpha) \rightarrow \mathcal{P}_2(\mathcal{A}(\mathcal{N}_\alpha)))$, therefore in particular $(\forall \alpha \in [0,1]\ \mathcal{P}_1(\mathcal{N}_\alpha)) \rightarrow (\forall \alpha \in [0,1]\ \mathcal{P}_2(\mathcal{A}(\mathcal{N}_\alpha)))$. By Definition 12 concerning $\mathcal{P}_1^{\text{fuz}}$ and Definition 16 concerning $\mathcal{A}$, we obtain $\mathcal{P}_1^{\text{fuz}}(\mathcal{N}) \rightarrow \forall \alpha \in [0,1]\ \mathcal{P}_2([\mathcal{A}^{\text{fuz}}(\mathcal{N})]_\alpha)$. By Definition 12 concerning $\mathcal{P}_2^{\text{fuz}}$, this yields $\mathcal{P}_1^{\text{fuz}}(\mathcal{N}) \rightarrow \mathcal{P}_2^{\text{fuz}}(\mathcal{A}^{\text{fuz}}(\mathcal{N}))$, and we are done.

While Definition 16 is unambiguous, it does not guarantee that the fuzzy extension of a $\mathcal{HN}$-T-ALG always exists. However, this problem does not arise if a finite set of satisfaction degrees is used, which is often sufficient in FCSPs.

Proposition 13. *Let $\mathcal{A}^{fuz}$ be the fuzzy extension to $\mathcal{FN}$ of a $\mathcal{HN}$-T-ALG $\mathcal{A}$. If $\mathcal{N} \in \mathcal{FN}$ is a fuzzy constraint network such that SEDG($\mathcal{N}$) is numerable, then SEDG($\mathcal{A}^{fuz}(\mathcal{N})$) $\subseteq$ SEDG($\mathcal{N}$).*

Proof. Let SEDG($\mathcal{N}$) be $\{\alpha_1, \alpha_2, \ldots\}$; we assume $\alpha_0 = 0 < \alpha_1 < \alpha_2 \ldots$. Moreover, if SEDG($\mathcal{N}$) has a finite number k of elements, let $\alpha_{k+1} = 1$.
We reason by contradiction, assuming that $\exists \beta \in \text{SEDG}(\mathcal{A}^{\text{fuz}}(\mathcal{N}))$ such that $\beta \notin \{\alpha_1, \alpha_2, \ldots\}$: let j be the natural number such that $\alpha_j < \beta < \alpha_{j+1}$. Since $\beta \in$

SEDG($\mathcal{A}^{\text{fuz}}(\mathcal{N})$), we have that $[\mathcal{A}^{\text{fuz}}(\mathcal{N})]_\beta \neq [\mathcal{A}^{\text{fuz}}(\mathcal{N})]_{\alpha_{j+1}}$. However, by Definition 16 $[\mathcal{A}^{\text{fuz}}(\mathcal{N})]_\beta = \mathcal{A}(\mathcal{N}_\beta)$, and since $\mathcal{N}_\beta = \mathcal{N}_{\alpha_{j+1}}$ because of the structure of SEDG($\mathcal{N}$), $\mathcal{A}(\mathcal{N}_\beta) = \mathcal{A}(\mathcal{N}_{\alpha_{j+1}})$. By definition 16, $\mathcal{A}(\mathcal{N}_{\alpha_{j+1}}) = [\mathcal{A}^{\text{fuz}}(\mathcal{N})]_{\alpha_{j+1}}$, therefore we obtain $[\mathcal{A}^{\text{fuz}}(\mathcal{N})]_\beta = [\mathcal{A}^{\text{fuz}}(\mathcal{N})]_{\alpha_{j+1}}$, which contradicts the inequality above.

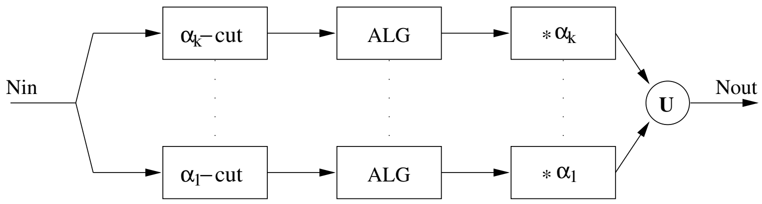

Fig. 1. Fuzzy extension of a crisp algorithm in case of a finite number of satisfaction degrees.

Proposition 13 suggests how to construct $\mathcal{A}^{\text{fuz}}$ for classes $\mathcal{FN}$ whose networks use a finite number of preference degrees: Figure 1 shows that in this case the fuzzy extension of a given algorithm ALG always exists.

7 On the Idempotency Assumption of the Aggregator Operator in FCSPs

As shown in the above sections, several results provided in the paper rely on Proposition 3. The latter in turn relies on the specific choice, in the FCSP framework, of the operator 'min' used to aggregate the satisfaction degrees of FCNs: in particular, the main property of 'min' which is exploited in the proof of Proposition 3 is its idempotency, i.e. the fact that $\forall a \;\min\{a,a\} = a$. If we remove the hypothesis of idempotency, then it is only possible to prove, under the reasonable constraints on the aggregator operator provided e.g. in [4], that $[\text{SOL}(\mathcal{N})]_\alpha \subseteq \text{SOL}(\mathcal{N}_\alpha)$, and this invalidates the main results provided in the subsequent sections.

In order to be more specific, we will refer to the specific framework of 'Weighted CSPs' (WCSPs), in which the degrees of the constraint relations represent costs instead of preferences, and are aggregated by the sum operator instead of min. In this case, optimal solutions are those which minimize the total cost, and the α-cut of a fuzzy set is made up of those values that have a membership degree lower than or equal to α (in other words, in WCSPs the $\leq$ order relation plays the role of $\geq$ in FCSPs, and vice versa). Moreover, the costs of the constraints take values in $\mathcal{R}^+ \cup \{+\infty\}$.

In the framework of WCSPs, the aggregator operator is not idempotent, since $(a+a) \neq a$ in general. As a consequence, the results concerning tractable classes and the fuzzy extension of properties and theorems do not hold in the WCSP framework, as we show now by examples.

First, let us consider Proposition 5 concerning the extension of tractable classes. The algorithm proposed in the relevant proof relies on the possibility of computing all the optimal solutions of a network exactly as the solutions of the consistent α-cut with highest α. But in the WCSP framework only the weak condition $[\mathrm{SOL}(\mathcal{N})]_\alpha \subseteq \mathrm{SOL}(\mathcal{N}_\alpha)$ is ensured, and this does not even guarantee that optimal solutions are among those of the consistent α-cut with highest α. As an example, consider the simple network $\mathcal{N}$ with just two variables x_1 and x_2, both of them taking values in the domain $\{1, 2\}$. Moreover, let us suppose that there are three constraints in the network, namely two unary constraints on x_1 and x_2, and a binary constraint involving both variables: these are represented in Figure 2(a), while the costs of the solutions are shown in Figure 2(b). In this case, the unique optimal solution is $< 2, 2 >$, whose cost is 6. If we consider the significative α-cuts (i.e. with $\alpha \in \{0, 1, 4, 5\}$), we have that $\mathrm{SOL}(\mathcal{N}_0) = \mathrm{SOL}(\mathcal{N}_1) = \emptyset$, $\mathrm{SOL}(\mathcal{N}_4) = \{< 1, 1 >, < 2, 1 >\}$, and $\mathrm{SOL}(\mathcal{N}_5) = \{< 1, 1 >, < 1, 2 >, < 2, 1 >, < 2, 2 >\}$. In this example, the lowest cost α for which there is a consistent α-cut is 4, but the optimal solution is not among the solutions of $\mathcal{N}_4$.

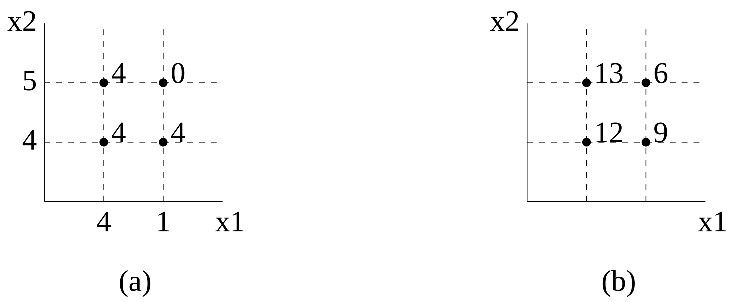

Fig. 2. A WCSP network used as counter-example to the proof of Proposition 5.

Let us now turn to the results provided in Section 5 concerning the fuzzy extension of properties and theorems. In the WCSP framework, Definition 12 is not useful at all, since local consistency and minimality properties are exceptional conditions, and they are not the fuzzy extensions of the corresponding properties of crisp frameworks. As a simple example, consider a trivial WCSP network with two variables x_1 and x_2, such that $D_1 = D_2 = \{1\}$, and two unary constraints on x_1 and x_2 such that $\mathcal{R}_1(1) = \mathcal{R}_2(1) = 2$. It is easy to see that the cost of the unique solution is 4, therefore the network is not minimal. However, the α-cut for $\alpha < 2$ is a network with empty constraints, while the α-cut for $\alpha \geq 2$ is a simple network with two independent unary constraints: in both cases, the α-cut is a minimal network.

Since in this example minimality is equivalent to 2-consistency, the network considered above represents a counter-example also to Proposition 7.

8 Conclusions and Future Work

Several instances of the CSP framework have been proposed, and are being continuously proposed, to handle specific problems. Since constraints in real world problems are seldom hard, fuzzy constraints can be used to encompass preference relations among possible solutions. We believe that the results provided in this paper are interesting, at a practical level, since they allow to directly extend a specific crisp framework to handle fuzzy constraints, reusing theorems and algorithms developed for the original framework. On this respect, it would be interesting to apply the results of this paper in order to prove further properties not yet published in the framework of FCSP; this is left as a topic of future research.

We expect the results provided in the paper to hold (if the aggregator operator $\times$ of the semiring is idempotent) also in the general framework of semiring-based soft constraint formalism [4]: this will be another topic of future research.

Finally, further investigation will be devoted to explicitely handle non finite domains (i.e. infinite sets of satisfaction degrees). Since in this paper no limiting assumption has been made in this respect, the results provided in this work hold in case of infinite domains as well. However, in this case there are some subtleties which prevent the full applicability of our results; one of them is 'hidden' in the definition of minimality that we have adopted, which slightly differs from the usual one. In particular, in Definition 2 we require the existence of a solution which maintains the degree of satisfaction of a generic tuple $\overline{d}$; usually, only the weaker condition $\sup \left\{ \deg \overline{d'} \mid \overline{d'} \in D_1 \times \ldots \times D_n,\ \overline{d'}^{\downarrow \mathcal{Y}} = \overline{d} \right\}$ is required[1]. This definition ensures that, for any FCSP network, there is an equivalent minimal network, while our definition does not guarantee this result. However, our definition is the fuzzy extension of minimality as defined for classical CSPs, while this is not true for the usual one.

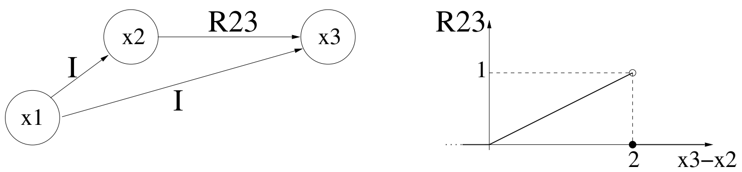

Fig. 3. A problematic FCSP network

[1] Likewise, Definition 1 differs from the notion of k-consistency underlying the definition of 'rule application' of [4]. However, in [4] the domains of the variables are assumed to be finite, therefore the two definitions turn out to be equivalent.

For the sake of clearness, let us refer to the binary network of Figure 3, where variables take real values, I denotes the constraint which assigns the maximum degree of preference 1 to every pair of values, and R_{23} constrains $x_3 - x_2$ as shown in the figure (notice that R_{23} is discontinuous, since $R_{23}(x, x+2) = 0$). It can be easily shown that the network is minimal according to the usual definition of fuzzy minimality, while it is not minimal according to Definition 2, since for instance any assignment of $< x_1, x_2 >$ cannot be extended to a solution with degree of satisfaction equal to 1. Let us now consider the 1-cut of the network: it turns out that $[R_{12}]_1 = [R_{13}]_1 = \mathcal{R}^2$, while $[R_{23}]_1 = \emptyset$, therefore the 1-cut is not a minimal network. On the other hand, according to Definition 2 there is not a minimal network equivalent to that of Figure 3: if we leave R_{12} and R_{13} unchanged then we prevent minimality to hold, since there is no solution with degree of satisfaction equal to 1, while changing for example $R_{12}(a, b)$ to $1 - \epsilon$, with $\epsilon > 0$, lowers the satisfaction degree of all the solutions $< a, b, i >$ such that $2(1 - \epsilon) + b < i < 2 + b$.

In order to settle the matter, we are studying the possibility of extending the FCSP framework in order to explicitely distinguish between those preference degrees that can actually characterize a solution and those ones that can be reached only 'in the limit' (i.e. providing preference degrees of the form a^-, where $a \in (0, 1]$).

References

1. J. F. Allen. Maintaining knowledge about temporal intervals. *Communication of the ACM*, 26(1):832–843, 1983.
2. S. Badaloni and M. Giacomin. Flexible temporal constraints. In *Proc. of IPMU 2000*, pages 1262–1269, Madrid, Spain, 2000.
3. S. Badaloni and M. Giacomin. A fuzzy extension fo allen's interval algebra. In *LNAI 1792*, pages 155–165. Springer-Verlag, 2000.
4. S. Bistarelli, U. Montanari, and F. Rossi. Semiring-based constraint satisfaction and optimization. *Journal of the ACM*, 44(2):201–236, 1997.
5. M. Cooper. On the complexity of fuzzy constraint satisfaction problems. In *H. Fargier, PhD Thesis n. 1760, Univ. P. Sabatier, Toulouse.* 1994.
6. R. Dechter, I. Meiri, and J. Pearl. Temporal constraint networks. *Artif. Intell.*, 49(1–3):61–95, 1991.
7. D. Dubois, H. Fargier, and H. Prade. Possibility theory in constraint satisfaction problems: Handling priority, preference and uncertainty. *Appl. Intell.*, 6:287–309, 1996.
8. H. Prade J. M. Pires. Flexibility as relaxation – the fuzzy set view. In *CP 2000 Workshop on modelling and solving soft constraints*, Singapore, 2000.
9. L. Khatib, P. Morris, R. Morris, and F. Rossi. Temporal constraint reasoning with preferences. In *Proc. of IJCAI 2001*, 2001.
10. P. van Beek. Exact and approximate temporal reasoning. *Comp. Intell.*, 6:132–144, 1990.