MULTI-AGENT SOFT CONSTRAINT AGGREGATION
A Sequential Approach

Giorgio Dalla Pozza, Francesca Rossi and K. Brent Venable
Dept. of Pure and Applied Mathematics, University of Padova, Padova, Italy

Keywords: Group decision making, Constraint satisfaction, Fuzzy systems.

Abstract: We consider a scenario where several agents express their preferences over a common set of variable assignments, by means of a soft constraint problem for each agent, and we propose a procedure to compute a variable assignment which satisfies the agents' preferences at best. Such a procedure considers one variable at a time and, at each step, asks all agents to express its preferences over the domain of that variable. Based on such preferences, a voting rule is used to decide on which value is the best for that variable. At the end, the values chosen constitute the returned variable assignment. We study several properties of this procedure and we show that the use of soft constraints allows for a great flexibility on the preferences of the agents, compared to similar work in setting where agents model their preferences via CP-nets, where several restrictions on the agents' preferences need to be imposed to obtain similar properties.

1 INTRODUCTION

We consider scenarios in which a set of agents express their preferences over a common set of objects, and the aim is to choose one object which best satisfies the preferences of the agents. We also assume that the set of objects has a combinatorial structure. More precisely, each object is modelled as an elements of the Cartesian product of a set of variable domains. This is often the case in real life, since usually we express preferenecs over objects that are characterized by a set of features (the variables), each of which has some possible instances (the variable domain).

Finally, we also assume that the agents express their preferences over the objects in a compact way. There are many formalisms suitable to do this; in this paper we focus on soft constraints (Meseguer et al., 2005). Thus each agent specifies a set of soft constraints over the variables. Another well-known formalism to do this is CP-nets (Boutilier et al., 2004). The aim of such formalisms to allow one to express in time and space polynomial in the number of variables an ordering over the et of all objects, which may be exponential in such a number.

To define a procedure to aggregate the preferences of such agents, we consider voting theory (Arrow and amd K. Suzumura, 2002), a very wide research area between economy theory and operation research, that deals with elections, where voters (that we would call agents) vote by expressing their preferences over a set of candidates (that we would call objects), and a voting rule decides who the winner candidate is. Voting theory provides many rules to aggregate preferences, that take in input (a part of) the preference orderings of the agents and gives as output the ”winner” object, that is, the object that is considered to be the best according to the rule.

The most naive way to use voting rules in our context consists of choosing a voting rule and giving to it what it needs to know about the preference orderings of the agents, then running the voting rule and see what result comes out. This is however not feasible in general. In fact, if the chosen voting rule needs to know a large part of the preference ordering from the agents, it may take exponential time only to give the input to the rule. A valid alternative is to use the voting rule several times, on each feature of the object set. This approach is certainly more attractive computationally, since usually the number of instances of each feature is small.

What can be done in this situation is to study when, even in presence of dependencies among features, voting can be performed on each single feature at a time rather than on complete objects. We therefore define a sequential voting procedure that, at each step, applies one voting rule to the preferences of the agents over the domain of a single variable. We then study the properties of this sequential voting procedure. In particular, we consider properties such as Condorcet consistency, anynomity, neutrality, consis-

Dalla Pozza G., Rossi F. and Brent Venable K..
MULTI-AGENT SOFT CONSTRAINT AGGREGATION - A Sequential Approach.
DOI: 10.5220/0003156602770282
In *Proceedings of the 3rd International Conference on Agents and Artificial Intelligence* (ICAART-2011), pages 277-282
ISBN: 978-989-8425-40-9

tency, participation, efficiency, and monotonicity, and we relate their presence to the corresponding properties of the voting rules used at each step of the procedure.

This study has been done already for CP-nets (Lang and Xia, 2009), showing that a sequential single-feature voting protocol can find a winner object in polynomial time, and have several other desirable properties, when the CP-nets satisfy certain conditions on their dependencies. We show that the use of soft constraints allows us to avoid imposing many restrictions on the preferences of the agents. In fact, contrarily to CP-nets, soft constraints are not directional, and thus information can flow from one variable of a constraint to another one without a predefined ordering between them. This allows us to not tie the variable ordering used by the sequential procedure to the topology of the constraint graph of each agent. This makes the approach much more generally applicable. In fact, the tractability assumption over the constraint graphs is similar to the assumptions that CP-nets are acyclic. However, we do not need to impose that the constraint graphs are compatible among them and with a graph structure based on the variable ordering.

2 BACKGROUND

Soft Constraints. A soft constraint (Meseguer et al., 2005) involves a set of variables and associates a value from a (totally or partially ordered) set to each instantiation of its variables. Such a value is taken from a c-semiring, which is defined by $\langle A, +, \times, 0, 1\rangle$, where A is the set of preference values, $+$ is a commutative, associative, and idempotent operator, $\times$ is used to combine preference values and is associative, commutative, and distributes over $+$, 0 is the worst element, and 1 is the best element. A c-semiring S induces a partial or total order $\leq_S$ over the preference values, where $a \leq_S b$ iff $a+b=b$. A Soft Constraint Satisfaction Problem (SCSP) is a tuple $\langle V,D,C,A\rangle$ where V is a set of variables, D is the domain of the variables and C is a set of soft constraints over V associating values from c-semiring A.

An instance of the SCSP framework is obtained by choosing a c-semiring. For instance, in classical constraints we want all constraints to be satisfies, thus we may choose the semiring $S_{CSP} = \langle\{false, true\}, \vee, \wedge, false, true\rangle$. If instead we want to maximize the minimum preference, we may choose the semiring $S_{FCSP} = \langle[0,1], max, min, 0, 1\rangle$ and consider the so-called fuzzy CSPs. As an example, consider the following fuzzy CSP where $V = \{X,Y\}$, $D = \{a,b\}$ and $C = \{c_Y, c_{xy}\}$. Soft constraint c_Y is defined over Y and associates preference 0.4 to a and to 0.7 to b. Constraint c_{xy}, instead, is defined over X and Y and associates 0.9, 0.8, 0.7, 0.6 to, respectively, tuples $(X=a, Y=a)$, $(X=a, Y=b)$, $(X=b, Y=a)$ and $(X=b, Y=b)$.

Two main operations are defined on soft constraints: combination, denoted with $\otimes$, and projection, denoted with $\Downarrow$. Combining two constraints means building a new constraint involving all the variables of the original ones, and which associates to each tuple of domain values for such variables a semiring element which is obtained by combining (via $\times$) the elements associated by the original constraints to the appropriate subtuples. In the example of the fuzzy CSP above, $c_Y \otimes c_{XY}$ is a constraint on X and Y associating 0.4, 0.7, 0.4 and 0.6 to, respectively, tuples $(X=a, Y=a)$, $(X=a, Y=b)$, $(X=b, Y=a)$ and $(X=b, Y=b)$.

Projecting a constraint on a subset variables means eliminating the other variables by associating to each tuple over the remaining variables a semiring element which is the sum (via +) of the elements associated by the original constraint to all the extensions of this tuple over the eliminated variables. In the example, constraint $c_Y \otimes c_{XY} \Downarrow_X$ is a constraint defined only over X, which associates 0.7 to a and 0.6 to b.

To solve an SCSP, we just combine all constraints, inducing an ordering over the set of all complete assignments. In the case of fuzzy CSPs, such and ordering is a total order with ties. In the example above, the induced ordering has $(X=a, Y=b)$ at the top with a preference of 0.7, $(X=b, Y=b)$ just below with 0.6 and $(X=b, Y=a)$ and $(X=b, Y=b)$ tied at the bottom with 0.4. An optimal solution of an SCSP is then a complete assignment with an undominated preference. Finding an optimal solution in a set of soft constraints is an NP-hard problem.

Constraint propagation in SCSPs may be very helpful in For some classes of constraints, constraint propagation is enough to solve the problem (Dechter, 2005). This is the case for tree-shaped fuzzy CSPs, where directional arc-consistency (DAC), applied bottom-up on the tree shape of the problem, is enough to make the search for an optimal solution backtrack-free. DAC is also enough to compute the preferences over the values of the root variable, in dependence of the rest of the problem. That is, DAC is equivalent to combining all constraints and projecting over the root variable.

If we project the solution of a SCSP over a single variable, we obtain a total order with ties over the values of that variable, where each value is associated to the preference of the best solution of the SCSP having such variable instantiated to such a value. Given an SCSP P and one of its variables v, we will denote as $top(v,P)$ the set of values of v that are assigned the highest preference in such an ordering. We note that such a preference coincides with the optimal preference value of P. In our running example, if we consider $c_Y \otimes c_{XY} \Downarrow_X$, the induced ordering over the values of the domain of X is $a > b$.

Voting Theory. In classical voting theory (Arrow and amd K. Suzumura, 2002), given a set of candidates C, a *profile* is a sequence of Usually, such orderings are total orders, however several extensions have been studied such as when the orderings are partial orders or total orders with ties. Given a profile, a *voting rule*, also known as social choice function, maps it onto a single winning candidate. In this paper, we will often use a terminology which is more familiar to multi-agent settings, and we will therefore sometimes call "agents" the voters, "solutions" the candidates, and "decision" or "best solution" the winning candidate.

Some examples of widely used voting rules are: *Plurality*: where each voter states who the preferred candidate is, and the candidate who is preferred by the largest number of voters wins; *Borda*: where given m candidates, each voter gives a ranking of all candidates and the i^{th} ranked candidate scores $m-i$; the candidate with the greatest sum of scores wins; *Approval*: where each voter approves between 1 and $m-1$ candidates on m total candidates; the candidate with most votes of approval wins; *Copeland*: where the winner is the candidate that wins the most pairwise competitions against all the other candidates.

The research on voting theory has mainly been concerned with the definition of desirable properties of voting systems. Among them, we recall:

- Condorcet-consistency: every other in pairwise elections (namely, a Condorcet winner) exists, that candidate is always elected; Condorcet winners are unique and may not exist.
- Anonymity: When the results of an election are the same even if it occurs a permutation on the voters' set.
- Neutrality: When it is anonymous w.r.t. the candidates.
- Monotonicity: voter improves his vote in favor of this candidate, then the same candidate still wins.
- Consistency: If, when considering preferences of 2 disjoint sets of voters - who decide over the same issues and have identical final results - the result obtained by a vote of the joint set of voters is the same as the ones obtained by the disjoint set of voters.
- Participation: If, given any profile, and given a new vote over a set of issues by a new voter, the result obtained from the new profile is equally or more preferred by the new voter, who, thus, has an incentive to participate. If, given a winner over an election, there's no candidate who is preferred to the winner by all voters.

All the rules cited above are anonymous and neutral,all but Cup are efficient, only Cup and Copeland are Condorcet consistent, and all but Cup and Copeland are consistent and participative.

3 SEQUENTIAL PREFERENCE AGGREGATION

In the fuzzy scenario, each variable assignment is given a preference between 0 and 1, and assignments with a higher preference value are more preferred.

Assume to have a set of agents, each one expressing his preferences over a common set of objects via some soft constraints. as well as different preferences over the variable domains. We will call a *soft profile* the preference of a set of m agents, identified by a triple (V,D,P): a set of variables V, a sequence D of $|V|$ domains, and a sequence P of m soft constraint problems over variables in V with domains in D. A *fuzzy profile* is a soft profile where the preferences of the agents are fuzzy constraints.

The idea is to sequentially vote on each variable via a voting rule. We do not restrict to use always the same voting rule for all variables, so we will have a sequence of as many voting rules as the variables.

Given a profile (V,D,P), assume $|V| = n$, and consider an ordering of such variables $O = \langle v_1,\ldots,v_n\}$ and a sequence of voting rules $R = \langle r_1,\ldots,r_n\rangle$. The sequential voting procedure we propose is a sequence of n steps, where at each step i:

1. We ask all agents to report their preference ordering over the domain of variable v_i. If we have m agents, let us call such preference orderings $\langle po_i,\ldots,po_m\rangle$.
2. We apply voting rule r_i to this profile, returning a winning assignment for variable v_i, say d_i. If there are ties in the result, the first one following a lexicographical order will be taken.
3. We add the constraint $v_i = d_i$ to the preferences of each agent.

After all n steps have been executed, the tuple $\langle d_1,\ldots,d_n\rangle$ is reported as the chosen assignment for the variables in V. We write $Seq_{O,R}(V,D,P) = \langle d_1,\ldots,d_n\rangle$.

This short description of the sequential voting procedure does not say what it means for an agent to report their preference ordering over the domain of variable v_i. In general, since we do not make any assumption on the voting rules r_i, the agent needs to provide the rule with a preference ordering over the whole domain of v_i. Since this variable can be connected to other parts of the agent's soft constraint problem, in order to report the correct preferences over the domain of v_i, the agent needs to consider the influence of the rest of the problem over v_i. This means that, in general, the agent needs to compute the projection over v_i of its whole soft constraint problem. This task is in general difficult, so it may require exponential time to accomplish it, unless the class of constraint problems used by the agent is tractable. This is for example the case of tree-like shaped soft constraint problems, which are polynomial to solve.

4 CONDORCET CONSISTENCY

It is natural to ask ourselves if the result returned by the sequential voting procedure has some relation with what is considered to be most preferred by the agents.

A Condorcet winner (CW) is a candidate which is preferred to any other candidate by a majority of agents. Given a totally ordered profile, as in classical voting theory, there can be zero or exactly one Condorcet winner. In our context, since we may have ties in the preference orderings of the agents, there could be more than one Condorcet winner, since several variable instantiations could be considered optimal for a majority of agents.

First, we define the notion of sequential Condorcet winner (SCW). Given an SCSP Q, we will denote as $Q|_{v_1=d_1,\cdots,v_h=d_h}$ the problem obtained from Q by fixing variables $v_1,\cdots,v_h$ to the corresponding values. Let P_i denote the fuzzy constraint problem of agent i. Given a soft profile (V,D,P) with m agents and n variables, and an ordering O over V, $\langle d_1,\ldots,d_n\rangle$ is a SCW iff, for all $j=1,\ldots,n$, $|\{i|d_j \in top(v_j,P_i|_{v_1=d_1,\ldots,v_{j-1}=d_{j-1}})\}| > m/2$. In words, a sequential Condorcet winner is the combination of local Condorcet winners.

If all the local rules are Condorcet consistent, the sequential voting procedure returns a SCW by definition. However, to conclude that Seq is Condorcet consistent, we need to prove that SCW = CW. The following results shows that a CW is always an SCW, but unfortunately the opposite does not hold.

Theorem 1. *Given a soft profile (V,D,P) and an ordering O over V, if d is a CW for (V,D,P), it is a SCW for (V,D,P). Thus, if Seq is Condorcet consistent, all local voting rules are so.*

If $d=(d_1,\ldots,d_n)$ is a CW, then a majority of voters prefers it to all other candidates. Thus, at each step i of the sequential voting procedure, the same majority prefers d_i to all other values in the domain of v_i given the values already chosen.

The opposite does not hold in general, even if all voting rules are Condorcet consistent.

Theorem 2. *If all local voting rules are Condorcet consistent, the sequential voting procedure may be not Condorcet consistent.*

Consider a fuzzy profile (V,D,P) where: $V=\{X,Y\}$, $D_X=D_Y=\{a,b\}$ and there are 5 agents. The fuzzy SCSPs of all agents have a single constraint over $\{X,Y\}$. For two agents we have: $def(X=a,Y=b)=0.9$, $def(X=b,Y=b)=0.8$, $def(X=a,Y=a)=0.7$, $def(X=b,Y=a)=0.6$; for one agent: $def(X=a,Y=a)=0.9$, $def(X=a,Y=b)=0.8$, $def(X=b,Y=a)=0.7$, $def(X=b,Y=b)=0.6$; for the other two agents: $def(X=b,Y=a)=0.9$, $def(X=b,Y=b)=0.8$, $def(X=a,Y=a)=0.7$, $def(X=a,Y=b)=0.6$. When each agent solves the problem and projects on variable X, for the first two agents we have $pref_X(a)=0.9$ and $pref_X(b)=0.8$; for the third agent $pref_X(a)=0.9$ and $pref_X(b)=0.7$; and for the last two agents $pref_X(a)=0.7$ and $pref_X(b)=0.9$. Thus, 3 over 5 agents agree that $X=a$ is optimal. Since the voting rule r_X is Condorcet-consistent, this value will be chosen for X. Given $X=a$, the preferences of the agents for Y are: for the first two agents $pref_Y(a)=0.7$ and $pref_Y(b)=0.9$; for the third agent $pref_Y(a)=0.9$ and $pref_Y(b)=0.7$; for the last two agents $pref_Y(a)=0.7$ and $pref_Y(b)=0.6$. Thus $Y=a$ will be chosen, since r_Y is Condorcet consistent, and $(X=a,Y=a)$ will be the SCW. However, $(X=a,Y=a)$ is not a CW, since the majority of the agents prefers $(X=b,Y=b)$.

5 ANONYMITY, NEUTRALITY AND CONSISTENCY

It is also important to make sure that a preference aggregation system does not depend on the names or the order of the agents. This corresponds to saying that the rule is anonymous. In our setting, a permutation of voter set corresponds, basically, to a permutation of the soft constraint problems. It is easy to see that if the sequential voting rule respects anonymity, then also all the local voting rules do so, and, vice versa, if all the local voting rules are anonymous, so is the resulting sequential rule.

Neutrality, on the other hand, is a property that requires for a rule to be insensitive to permutations of the candidates. This means that the result does not depend on the names of the candidates, but only on their position in the preference orderings. We note that the candidates of the local voting rules are the values in the variable domains, while the candidates of the sequential voting rule are the complete assignments to all variables. While a permutation of the values in the domains always corresponds to a permutation of the variable assignments, not all of the permutations of variable assignments can be obtained via permutations of domain values. Thus neutrality of the local voting rules does not imply neutrality of the sequential voting rule, while neutrality of the sequential voting rule implies neutrality of each local voting rule.

As defined above, a voting rule r is consistent if, when considering two profiles P_1 and P_2 with disjoint sets of voters, who vote over the same candidates, such that $r(P_1) = r(P_2)$, we have $r(P_1 \cup P_2) = r(P_1)$.

Theorem 3. *If all the local voting rules in $R <$ The opposite holds as well: if $Seq_{O,R}$ is consistent, then all the local voting rules in R are consistent.*

In fact, if all the local rules are consistent, at every step i of the sequential procedure, applied to profile $P_1 \cup P_2$, the result for variable v_i is the same as the result in profile P_1 (and also in profile P_2), so the overall result $(d_1,\ldots,d_n)$ will be the same as the result obtained by the sequential procedure in profile P_1 and in profile P_2. On the other hand, if one of the local rules

6 PARTICIPATION

Theorem 4. *If the sequential voting procedure is participative, then each local voting rule is so.*

This means that there is a profile over the values of variable v_i, say p_i, and an agent h with preference $p_{i,h}$ over the values of v_i, such that agent h strictly prefers $r_i(p_i)$ to $r_i(p_i \cup p_{i,h})$. for each agent $j = 1,\ldots,m$, the preferences over the values of v_i are as in p_i; all other unary constraints are the same for all agents and associate preference 1 to exactly one value per variable and 0 to all other variables; there are no other constraints. Assume the SCSP P_h of agent h is defined as follows: his preference over variable v_i is $p_{i,h}$, all other unary constraints associate preference 1 to exactly one value per variable and 0 to all other variables; there are no other constraints. Since we have that $Seq_{O,R}(V,D,P) \downarrow v_i = r_i(p_i)$ and $Seq_{O,R}(V,D,(P \cup P_h)) \downarrow v_i = r_i(p_i \cup p_{i,h})$, agent h strictly prefers $Seq_{O,R}(V,D,P)$ to $Seq_{O,R}(V,D,(P \cup P_h))$.

On the other hand, it is possible that all local voting rules are participative, but the sequential voting procedure is not so.

Theorem 5. *If all the local voting rules are participative, the sequential voting procedure may not be participative.*

To see this, consider the profile where $V = \{x,y\}$, $D = (\{a,b,c\},\{a,b\})$, and P is a sequence of two fuzzy SCSPs which coincide and contain a unary constraint on x (associating preference 1 to a, 0.8 to b, and 0.6 to c), a binary constraint on x and y (associating preference 1 to (a,b), 0.9 to (a,a), 0.8 to (b,a), 0.7 to (b,b), 0.6 to (c,a), and 0.5 to (c,b)), and a unary constraint over y (associating preference 1 to both a and b). It is easy to see that this SCSPs are DAC. Assume also that variables are ordered $x \prec_O y$ and that r_1 is the scoring rule with score vector $(3,2,0)$ and r_2 is the majority rule. In this profile, $Seq_{O,R}(V,D,P) = (x = a, y = b)$. We now consider a third voter, with a fuzzy SCSP with a unary constraint on x (associating preference 0.8 to a, 1 to b, and 0.9 to c), a binary constraint on x and y (associating preference 0.8 to (a,b), 0.5 to (a,a), 0.7 to (b,a), 1 to (b,b), 0.6 to (c,a), and 0.5 to (c,b)), and a unary constraint over y (associating preference 1 to both a and b). In this new profile P', $Seq_{O,R}(V,D,P') = (x = b, y = a)$. However, the third voter prefers $(x = a, y = b)$ to $(x = b, y = a)$. Thus the third voter would be better off not participating to the sequential voting process.

7 EFFICIENCY

Theorem 6. *If the sequential voting procedure is efficient, then each local voting rule is so.*

Let us assume that there is a local rule, say r_i that is not efficient. This means that there is a profile over the values of variable v_i, p_i, such that $r_i(p_i) = d$ but all agents prefer another value d'. Let us now consider the soft profile (V,D,P), where for each agent j the preferences over the values of v_i are as in p_i; all other unary constraints are the same for all agents and associate preference 1 to exactly one value per variable and 0 to all other variables and there are no other constraints. We will have that $Seq_{O,R}(V,D,P) = (d_1,d_2,\ldots,d_{i-1},d,\ldots,d_n)$. For each agent j his preference for $Seq_{O,R}(V,D,P)$ corresponds with his local preference for d. Thus each agent will prefer $(d_1,d_2,\ldots,d_{i-1},d',\ldots,d_n)$ to $Seq_{O,R}(V,D,P)$.

On the other hand, it is possible that all local voting rules are efficient, but the sequential voting procedure is not so. However, if we add the condition that there is a single optimal candidate for all agents, then the sequential voting procedure is efficient.

Theorem 7. *If all the local voting rules are efficient, and there is a single candidate which is strictly preferred to all other candidates for all voters, then the sequential voting procedure is efficient.*

In fact, if there is a single candidate, say $d = (d_1, \ldots, d_n)$ that is optimal for all agents, then we have that, after DAC, for each agent i and for each variable j, $top(v_j, P_i|_{v_1=d_1,\ldots,v_j-1=d_j-1}) = d_j$. Thus since each rule is efficient it will elect the value of d assigned to its variable. Thus $Seq_{O,R}(V,D,P) = d$.

We note that for profiles where there is a unique candidate that is optimal for all agents, efficiency coincides with Condorcet consistency. Thus, given such profiles, the Condorcet consistency of the local rules implies that of the the sequential rule.

8 MONOTONICITY

As defined above, a voting rule is monotonic if, when a candidate wins, and one or more voters improve their vote in favor of this candidate, then the same candidate still wins.

Theorem 8. *If the sequential voting procedure is monotonic, then each local voting rule is so.*

Let us assume that there is a local rule, say r_i that is not monotonic. This means that there are two profiles over the values of variable v_i, say p_i and p'_i, such that $r_i(p_i) = d_i$, p'_i is as p_i except that some agents have moved d_i up in their orderings, and $r'_i(p_i) = d'_i \neq d_i$.

Let us now consider the fuzzy profile (V,D,P) where: for each agent, the preferences over the values of v_i are as in p_i; all other unary constraints are the same for all agents and associate preference 1 to exactly one value per variable and 0 to all other variables and there are no other constraints. Assume the result of applying the sequential rule is $Seq_{O,R}(V,D,P) = (d_1, d_2, \ldots, d_i, \ldots, d_n)$.

Now let us consider the fuzzy profile (V,D,P') where P' differs from P only on the preferences on variable v_i that, for each agent, are as in p'_i. We note that in each agent's SCSP, both in P and P', there are as many solutions as the values in the domain of v_i and that their preference coincides with the one of the corresponding value of v_i. Thus P' differs from P only on the preference of the only solution involving value d_i, i.e. $Seq_{O,R}(V,D,P)$, that has been moved up in the ordering of some of the agents. However, the result $Seq_{O,R}(V,D,P')$ will be $(d_1, d_2, \ldots, d'_i, \ldots, d_n)$. Thus Seq is not monotonic.

Theorem 9. *If each local rule is monotonic, so is the sequential rule.*

Let us assume that the sequential rule is not monotonic. This means that there is at least one soft profile (V,D,P) such that $Seq_{O,R}(V,D,P) = d$ and another soft profile (V,D,P'), where d has been moved up in the preference orderings of some agents, but $Seq_{O,R}(V,D,P') = d' \neq d$. Let us denote with d_i, resp. d'_i, the value assigned to variable v_i in d, resp. d'. We note that there must be at least one value on which they differ. For each SCSP in P' and for each variable v_i, d_i has either improved w.r.t. d'_i or remained as in P. Let v_j be any of the variables such that $d_j \neq d'_j$. Then r_j is not monotonic.

The same results can be proven for strong monotonicity, that is, all local voting rules are strongly monotonic iff the sequential tule is so.

9 FUTURE WORK

We just considered a few properties of the sequential voting procedure. We plan to study many others in order to better characterize the result of the procedure in terms of the preferences of the agents. We also plan to develop heuristics for an efficient computing of such a result even when the soft constraint problems are not from a tractable class.

ACKNOWLEDGEMENTS

Research partially supported by the Italian MIUR PRIN project 20089M932N: “Innovative and multi-disciplinary approaches for constraint and preference reasoning”.

REFERENCES

Arrow, K. J. and amd K. Suzumura, A. K. S. (2002). *Handbook of Social Choice and Welfare.* North-Holland, Elsevier.

Boutilier, C., Brafman, R. I., Domshlak, C., Hoos, H. H., and Poole, D. (2004). CP-nets: A tool for representing and reasoning with conditional ceteris paribus preference statements. *J. Artif. Intell. Res. (JAIR)*, 21:135–191.

Dechter, R. (2005). Tractable structures for CSPs. In F. Rossi, P. V. B. and Walsh, T., editors, *Handbook of Constraint Programming*. Elsevier.

Lang, J. and Xia, L. (2009). Sequential composition of voting rules in multi-issue domains. *Mathematical social sciences*, 57:304–324.

Meseguer, P., Rossi, F., and Schiex, T. (2005). Soft constraints. In F. Rossi, P. V. B. and Walsh, T., editors, *Handbook of Constraint Programming*. Elsevier.