Multi-agent soft constraint aggregation via sequential voting

Paper 985

Abstract

We consider scenarios where several agents must aggregate their preferences over a large set of candidates with a combinatorial structure. That is, each candidate is an element of the Cartesian product of the domains of some variables. We assume agents compactly express their preferences over the candidates via soft constraints. We consider a sequential procedure that chooses one candidate by asking the agents to vote on one variable at a time. While some properties of this procedure have been already studied, here we focus on independence to irrelevant alternatives, non-dictatorship, and strategy proofneess. Also, we perform an experimental study that shows that the proposed sequential procedure yields a considerable saving in time with respect to a non-sequential approach, while returning candidates that satisfy the agents just as well, independently of the variable ordering and of the presence of coalitions of agents.

1 Introduction

We consider scenarios where a set of agents needs to select a common decision from a set of possible decisions, over which they express their preferences. We also assume that such a decision set has a combinatorial structure, that is, that each decision can be seen as the combination of certain features, where each feature has a set of possible instances. This occurs in several AI applications, such as combinatorial auctions, web recommender systems, and configuration systems. Even if the number of features and instances is small, the number of possible decisions can be very large. Fortunately, in the presence of such a combinatorial structure, agents may describe their preference in a compact and efficient way, using one of the several formalisms available in the literature, such as soft constraints [8], CP-nets [3], and graphical utility models [2]. The goal is to aggregate such preferences and to select the most preferred decision. While such compact preference formalisms are suitable for aggregating the preferences of a single agent, they may lead to undesirable results if used for multi-agent preference aggregation. For example, in soft constraints, the negative opinion of an agent over a specific instance of a feature would impact negatively on any decision containing such an instance, even if all other agents like it. An alternative could be to search for a decision belonging to the Pareto frontier of the multi-agent preferences, as it is done in [6] for preferences expressed via GAI utility functions. Another option, that we study in this paper, is to aggregate the preferences via voting rules.

Voting theory [1] is a wide research area that considers similar scenarios: in an election, voters (that we call agents) vote by expressing their preferences over a set of candidates (that we call objects or decisions), and a voting rule decides who the winner candidate is. Voting theory provides many rules to aggregate preferences, each taking in input (a part of) the preference orderings of the agents and giving in output the ”winner” object, that is, the object that is considered to be the best according to the rule. In our context, it may take exponential time just to provide a voting rule with what it needs of the preference orderings of the agents. A valid alternative is to use a voting rule several times, on each feature of the object set (possibly using a different voting rule for each feature). That is, the voting rule asks the agents to provide their preferences on each feature at a time, and at each step a winner instance for a certain feature will be returned. At the end, all the winner instances will constitute the winning object.

This approach is certainly more attractive computationally, since usually the number of instances of each feature is small. However, when features are interdependent, it is not clear if the result of this sequential approach is useful at all. In this paper we consider this issue, assuming agents express their preferences via soft constraints.

As an example of the setting we consider, in a combinatorial auction, the features are the various items to be sold, and each bid consists of the agent's preferences over single items or combinations of them. In this context, a sequential approach would consider one item at a time, and at each step it would exploit the bids to allocate this item.

A similar approach has been considered for CP-nets in [7] and partially investigated for soft constraints in [9]. With respect to CP-nets, soft constraints allow one to avoid imposing many restrictions on the agents' preferences, since they 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 one to not tie the variable ordering used by the sequential procedure to the topology of the constraint graph of each agent. Moreover, soft constraints can model also strict requirements, which are often necessary in multi-issue settings where one needs to rule out some combinations of feature instances.

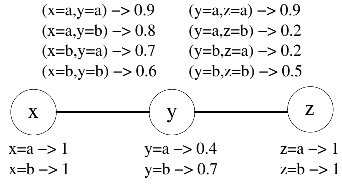
In this paper we study some desirable properties of the considered voting rules, namely independence to irrelevant alternatives (IIA), non-dictatorship, and strategy proofness. We prove that IIA of the sequential procedure implies IIA of all the voting rules used in for the features (also called 'local rules'), while the opposite does not hold. The same results hold for strategy proofness. On the other hand, non-dictatoriality of the sequential procedure implies non-dictatoriality of at least one local rule, and viceversa. We also consider the computational complexity of deciding if a coalition of agents can manipulate the result by forcing the selection of a certain object (aka coalitional constructive manipulation): if it is easy to do this for all local rules, then it is also easy for the sequential procedure, and if it is NP-complete for at least one local rule, then it is so also for the sequential procedure.

We also show experimentally that the sequential approach is convenient in terms of computation time, while satisfying the agents just as much as the non-sequential approach. Also, the quality of the returned solution is not affected by the ordering of the features, nor by the presence of coalitions of voters (i.e., voters who vote in the same way) in the profile.

2 Background

Soft Constraints. A soft constraint [8] 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, $+$ induces an ordering over A (where $a \leq_S b$ iff $a+b = b$), $\times$ is used to combine preference values, and 0 and 1 are respectively the worst and best element. 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 (each one involving a subset of V) associating values from A.

An instance of the SCSP framework is obtained by choosing a specific c-semiring. For instance, choosing $S_{FCSP} = \langle [0,1],\ max, min,\ 0,\ 1\rangle$ means that preference are in [0,1] and we want to maximize the minimum preference. This is the setting of fuzzy CSPs (FCSPs) [8], that we will use in the examples of this paper. Consider the following figure, which shows the constraint graph of an FCSP where $V = \{x, y, z\}$, $D = \{a, b\}$ and $C = \{c_x, c_y, c_z, c_{xy}, c_{yz}\}$. Each node models a variable and each arc models a binary constraint, while unary constraints define variables' domains. For example, c_y associates preference 0.4 to $y = a$ and 0.7 to $y = b$. Default constraints such as c_x and c_z will often be omitted in the following examples.



Solving an SCSP means finding the ordering induced by the constraints over the set of all complete variable assignments. In the case of FCSPs, such an ordering is a total order with ties. In the example above, the induced ordering has $(x = a, y = b, z = b)$ and $(x = b, y = b, z = b)$ at the top with a preference of 0.5, $(x = a, y = a, z = a)$ and $(x = b, y = a, z = a)$ just below with 0.4, and all others tied at the bottom with preference 0.2. An optimal solution, say s, of an SCSP is then a complete assignment with an undominated preference (thus $(x = a, y = b, z = b)$ or $(x = b, y = b, z = b)$ in this example). Given a variable x, we write $s \downarrow x$ to denote the value of x in s. Finding an optimal solution is an NP-hard problem, unless certain restrictions are imposed, such as a tree-shaped constraint graph.

Constraint propagation may improve the search for an optimal solution. Given a variable ordering o, a FCSP is directional arc-consistent (DAC) if, for any two variables x and y linked by a fuzzy constraint, such that x precedes y in the ordering o, we have that, for each a in the domain of x, $f_x(a) = max_{b\in D(y)}(min(f_x(a), f_{xy}(a, b), f_y(b)))$, where f_x, f_y, and f_{xy} are the preference functions of c_x, c_y and c_{xy}. This definition can be generalized to any instance of the SCSP approach by replacing max with $+$ and min with $\times$.

DAC is enough to find the preference level of an optimal solution when the problem has a tree-shape and the variable ordering is compatible with the father-child relation of the tree [8]. In fact, such an optimum preference level is the best preference level in the domain of the root variable. To find an optimal solution, it is then enough to perform a backtrack-free search which instantiates variables in the same order used for DAC (thus, from the root of the tree to its leaves). In our running example, if we choose the variable ordering $\langle x, y, z\rangle$, achieving DAC means first enforcing the property over y and z and then over x and y. The first phase modifies the preference of $y = b$ to 0.5, while the second phase sets the preferences of both $x = a$ and $x = b$ to 0.5. We note that, by achieving DAC w.r.t ordering o, we obtain a total order with ties over the values of the first variable in o, where each value is associated to the preference of the best solution having such a variable instantiated to such a value. In our running example, achieving DAC brings both values of x in a tie, with 0.5 that is the preference of an optimal solution.

Voting Rules. A voting rule allows a set of voters to choose one among a set of candidates. Voters need to submit their vote, that is, their preference ordering over the set of candidates (or part of it), and the voting rule aggregates such votes to yield a final result, usually called the winner. In the classical setting [1], given a set of candidates C, a *profile* is a collection of total orderings over the set of candidates, one for each voter. Given a profile, a *voting rule* maps it onto a single winning candidate (if necessary, ties are broken appropriately). In this paper, we will often use a terminology which is more familiar to multi-agent settings: we will 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, the i^{th} ranked candidate scores $m - i$, and the candidate with the greatest sum of scores wins; *Approval*, where each voter approves between 1 and $m - 1$ candidates on m total candidates, and 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.

Voting theory has considered many desirable properties of voting rules. In this paper we will focus on independence to irrelevant alternatives (IIA), non-dictatorship, and strategy proofness. A voting rule is IIA [10] if, whenever a candidate y loses to some winner x, and the relative ranking of x and y does not change in the profile, then y cannot win (independently of any possible change w.r.t. other irrelevant alternatives). A voting rule is *non-dictatorial* if there is no voter such that the winner is always a top-ranked alternative of this voter (the dictator). A voting rule is *strategy proof* if it is not manipulable, that is, no voter can get better off by lying on its preference ordering. All the above rules are non-dictatorial and manipulable in general, while only Approval is IIA.

Recently, the computational complexity of several kinds of manipulation has been studied. For example, coalitional constructive manipulation, denoted by $CCM(d, C, r, p)$, is the problem of deciding if coalition C can make d win in profile p via rule r, and it has been shown to be in P for Copeland with three candidates and Plurality [4], and NP-complete for Copeland [5].

Sequential preference aggregation.

Assume to have a set of agents, each one expressing his preferences over a common set of objects via an SCSP whose variable assignments correspond to the objects. Since the objects are common to all agents, this means that all the SCSPs have the same set of variables and the same variable domains but they may have different constraints, as well as different preferences over the variable domains. In [9] this is the notion of *soft profile*, which is formally defined as a triple (V, D, P) where V is a set of variables (also called issues), D is a sequence of $|V|$ lexicographically ordered finite domains, and P a sequence of m SCSPs over variables in V with domains in D[1]. A *fuzzy profile* is a soft profile with fuzzy soft constraints. An example of a fuzzy profile where $V = \{x, y\}$, $D_x = D_y = \{a, b, c, d, e, f, g\}$), and P is a sequence of seven FCSPs, is shown in Fig. 1.

The idea proposed in [9] is to sequentially vote on each variable via a voting rule, possibly using a different one for each variable. Given a soft profile (V, D, P), assume $|V| = n$, and consider an ordering of such variables $O = \langle v_1, \ldots, v_n \rangle$, and a corresponding sequence of voting rules $R = \langle r_1, \ldots, r_n \rangle$ (that will be called "local"). The sequential procedure is a sequence of n steps, where at each step i,

1. All agents are first asked for their preference ordering over the domain of variable v_i, yielding profile p_i over such a domain. To do this, the agents will perform DAC on their SCSP, considering the ordering O[2].
2. Then, the voting rule r_i is applied to profile p_i, returning a winning assignment for variable v_i, say d_i. If there are ties, the first one following the given lexicographical order will be taken.
3. Finally, the constraint $v_i = d_i$ is added to the preferences of each agent and DAC is applied to propagate its effect considering the reverse order of O.

[1]Notice that a soft profile consists of a collection of SCSPs over the same set of variables, while a profile (as in the classical social choice setting) is a collection of total orderings over a set of candidates.

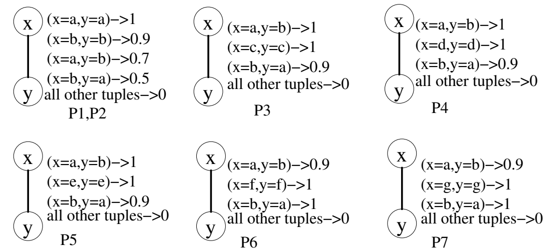


Figure 1: A fuzzy profile.

After all n steps have been executed, the winning assignments are collected in the tuple $\langle d_1, \ldots, d_n \rangle$, i.e., the winner of the election. This is denoted by $Seq_{O,R}(V, D, P)$.

In the soft profile above, assume the variable ordering is $\langle x, y \rangle$ and $r_i =$ Approval for all $i = 1, ..., n$. In step 1, agents apply DAC. This changes the preferences of the agents over x. For example, in P_1 and P_2, $x = a$ maintains preference 1, $x = b$ gets preferences 0.9, and all other domain values get preference 0, while in P_3, $x = a$ and $x = c$ maintain preference 1, $x = b$ gets preference 0.9, while all others get preference 0. Then, Approval is applied over the domain of x where the sets of approved values are: $\{a\}$ for the first two voters and respectively $\{c, a\}$, $\{d, a\}$, $\{e, a\}$, $\{f, b\}$, and $\{g, b\}$ for the others. Thus, $x = a$ is chosen and the constraint $x = a$ added to all SCSPs, and its effect is propagated via DAC on the domain of y. In step 2, DAC does not modify any preference (since y is the last variable) and the sets of approved values for y are all equal and contain only b. Thus the elected solution with the sequential procedure is $s = (x = a, y = b)$, which has preference 0.5 for P_1 and P_2, 1 for P_3, P_4, and P_5, and 0.9 for P_6 and P_7.

An alternative to this sequential procedure would be to generate the preference orderings for each voter from their FCSPs, and then to aggregate them in one step via Approval. In our example, $(x = a, y = a)$, $(x = a, y = b)$, and $(x = b, y = a)$ each gets 2 votes, $(x = f, y = f)$, $(x = d, y = d)$, $(x = c, y = c)$, and $(x = g, y = g)$ each gets 1 vote, while all other solutions get no vote. Thus the winner (breaking ties lexicographically) is $(x = a, y = a)$.

The variable ordering which is used in the sequential procedure, is assumed to be given in this paper[3]. In practice, the

[2]Ties are broken lexicographically if needed (for example, when using Plurality and there are ties at the top level). Also, Approval is given the sets of optimal solutions.

[3]In general, different variable orderings may lead to different re-

variable ordering is chosen by the chair of the preference aggregation process, for example by using priority arguments (one may vote first on the most important features), or by aggregating the agents's preferences on such orderings (if the number of features is small, this voting process is feasible).

In [9] it is shown that if the sequential procedure is Condorcet consistent, or anonymous, or neutral, or (strongly) monotonic, or consistent, or efficient, or participative, then all the local rules are so. On the other hand, the opposite holds only for anonymity, (strongly) monotonicity, and consistency. In the following sections we will presents results about other important properties: IIA, non-dictatorship, and strategy proofness.

3 Theoretical results

3.1 Independence to Irrelevant Alternatives

The sequential procedure may be not IIA, even starting from IIA local rules.

Theorem 1 *If all the local rules are IIA, the sequential procedure may be non-IIA.*

Proof: Consider the fuzzy profile (V, D, P) in Fig. 1, with the variable ordering $\langle x, y \rangle$ and where r_i = Approval (which is IIA) for each i. In such a scenario the elected solution is $s = (x = a, y = b)$.

Consider now a different profile, where the winner is $s' = (x = b, y = a)$, which differs from the current profile only on the relationship between assignments other than s'. The new soft profile is (V, D, P') where P' is obtained from P by swapping the preferences of $(x = a, y = a)$ and $(x = b, y = b)$ in P_1 and P_2. If we run the sequential election on P', the sets of approved values for x are: $\{b\}$ for the first two voters and as in P for the other voters. Thus, b is chosen for variable x and, given this, the sets of approved values for y are: $\{b\}$ for the first two voters and $\{a\}$ for the remaining five voters. Thus the winning solution is $s' = (x = b, y = a)$. Thus, despite the fact of using IIA local rules on all variables, the sequential procedure is not IIA. □

On the other hand, if the sequential procedure is IIA, then all local rules are IIA.

Theorem 2 *If one of the local rules is not IIA, then the sequential procedure is not IIA as well.*

Before proving this theorem, we introduce a notion, that will be useful in this proof and in other proofs in this paper, to lift at the sequential procedure level a property that belongs to one of the voting rules used for a variable. In the context of a set of variables V with domain D, and a set of m voters, given a profile p_i over a variable v_i, we define a soft profile over V, say $Ext(p_i) = (V, D, P)$, such that for any sequential procedure $Seq_{O,R}$, where r_i is the i-th component of R, $Seq_{O,R}(Ext(p_i)) \downarrow v_i = r_i(p_i)$, and, in each SCSP in P, the preference of any solution coincides with the preference of its projection on v_i. To achieve this, P consists of SCSPs with only unary constraints: the constraint over v_i respects the ordering in p_i, all other unary constraints are the same for all voters and associate preference 1 to exactly one value per variable and 0 to all other values. Also, all voting rules are assumed to be unanimous. Intuitively, $Ext(p_i)$ extends the orderings over variable v_i given by p_i to a soft profile over all variables.

Proof: Assume a local rule, say r_i, is not IIA. This means that there is a profile over the values of variable v_i, say p_i, a value for v_i, say $d \neq r_i(p_i)$, and another profile p'_i on values of v_i, such that $d = r_i(p'_i)$ and the relationship between $r_i(p_i)$ and d is the same in p_i and p'_i. Let us now consider the two soft profiles $Ext(p_i) = (V, D, P)$ and $Ext(p'_i) = (V, D, P')$. By definition, we have $Seq_{O,R}(Ext(p_i)) \downarrow v_i = r_i(p_i)$ and $Seq_{O,R}(Ext(p'_i)) \downarrow v_i = r_i(p'_i) = d$ and thus $Seq_{O,R}(Ext(p_i)) \neq Seq_{O,R}(Ext(p'_i))$. Moreover, since the preference for any assignment coincides with the preference of its projection over variable v_i, the relationship between $Seq_{O,R}(Ext(p_i))$ and $Seq_{O,R}(Ext(p'_i))$ in each SCSP in P is the same as the one in each corresponding SCSP in P'. Thus the sequential approach is not IIA. □

3.2 Non-dictatorship

It is sufficient for one of the local rules to be non-dictatorial in order for the sequential approach to be non-dictatorial as well, and viceversa.

Theorem 3 *If a local rule is non-dictatorial, then the sequential procedure is non-dictatorial.*

Proof: Assume rule r_i is non-dictatorial. Thus, for each voter j there exists a profile over variable v_i, say p_{ij}, such that $r_i(p_{ij})$ is not a top element for j. Let us now consider the soft profile $Ext(p_{ij})$. Since $Seq_{O,R}(Ext(p_{ij})) \downarrow v_i = r_i(p_{ij})$, it is not one of j's top candidates. □

Theorem 4 *If the sequential procedure is non-dictatorial, then at least one of the local rules is non-dictatorial.*

Proof: If the sequential procedure is non-dictatorial, then, for each agent j, there exists a soft profile (V, D, P_j) such that $Seq_{O,R}(V, D, P_j)$ is a not an optimal solution of the SCSP of j. If $Seq_{O,R}(V, D, P_j) \downarrow v_1$ is not a top element for j on the profile p_1 on values of v_1 obtained applying DAC, then r_1 is non-dictatorial. Otherwise, if $Seq_{O,R}(V, D, P_j) \downarrow v_1$ is a top element for j on the profile p_1 on values of v_1 obtained applying DAC, then we repeat the reasoning for the next local rule. Since $Seq_{O,R}(V, D, P_j)$ is not optimal for j, there must exist a variable v_k such that $Seq_{O,R}(V, D, P_j) \downarrow v_k$ is not a top element for j on the profile p_1 on values of v_k obtained applying DAC. Thus r_k is non-dictatorial. □

3.3 Strategy proofness

The presence of a local rule that is manipulable jeopardizes the strategy-proofness of the sequential approach. Moreover, the strategy-proofness of all local rules is not sufficient for ensuring the strategy-proofness of the sequential procedure.

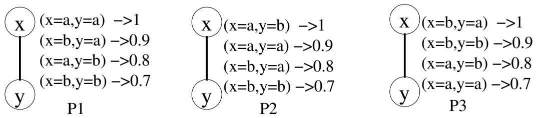
Theorem 5 *If one of the local rules is not strategy proof, the sequential procedure is not strategy proof.*

sults. However, some orderings will produce the same result. Intuitively, these are orderings that differ just for the relative position of variables that are independent on each other, according to all agents (we omit details for lack of space).

Proof: Assume a voting rule r_i is not strategy proof. Thus, there is an agent j and two profiles over variable v_i, say p_i and p'_i, that differ only because, in p'_i, j misreports his preferences which are $r_i(p'_i)$ is preferred to $r_i(p_i)$. Consider now the two soft profiles $Ext(p_i)$ and $Ext(p'_i)$. Since $Seq_{O,R}(Ext(p_i)) \downarrow v_i = r_i(p_i)$ and $Seq_{O,R}(Ext(p'_i)) \downarrow v_i = r_i(p'_i)$, agent j prefers $Seq_{O,R}(Ext(p'_i))$ to $Seq_{O,R}(Ext(p_i))$ and has thus succeeded in manipulating the sequential procedure. □

Theorem 6 *If all the local rules are strategy proof, the sequential procedure may be non-strategy proof.*

Proof: Consider the fuzzy profile with $V = \{x, y\}$, $D = \{a, b\}$, and $P = \{P_1, P_2, P_3\}$, as shown below.



Consider the sequential procedure where $O = \langle x, y \rangle$ and r_i=Plurality for $i = 1, 2$ (Plurality over two candidates is strategy proof). The sequential winner is $(x = a, y = b)$. However, if the first voter lies by swapping the preferences of $(x = a, y = a)$ and $(x = b, y = a)$ in P_1, then in the first step $x = b$ wins, and the final winner is $(x = b, y = a)$, which is better for him. Thus, the sequential procedure is not strategy proof. □

Turning our attention to the complexity of manipulation, if CCM (coalitional constructive manipulation) is easy for all the local rules, it remains so for the sequential approach. Actually, when obtaining the desired manipulation is easy at the local level, our result gives also a polynomial algorithm to set the preferences of the coalition in order to manipulate at the sequential level. Conversely, if CCM is difficult for at least one of the local rules, then it is so for the sequential procedure.

Theorem 7 *If CCM is in P for all the local rules, then it is in P also for the sequential procedure.*

Proof: Consider a DAC soft profile (V, D, P) with n variables and m voters such that all the SCSPs, except those of agents in the coalition, have been specified. The goal of the coalition C is to make candidate $d = (d_1, d_2, \ldots, d_n)$ win. The sequential procedure $Seq_{O,R}$ used is such that $CCM(d_i, C, r_i, P)$ is in P for every r_i in R.

We consider the rules one at a time, following O. It is possible to determine in polynomial time if d_1 can win the election on the domain of v_1 and, if so, how the coalition should vote on the domain values of v_1 to achieve this. If d_1 cannot win the local election, then candidate d cannot win the sequential election. If instead d_1 can win on v_1, then we add for each agent in the coalition a unary constraint on v_1 to his SCSP simulating the ordering that that agent in the coalition must give in order to manipulate successfully. In order to so, the best elements in the ordering are given preference 1 and the following values are given any preference that respects the ordering. Then, $v_1 = d_1$ is fixed in the SCSPs of all the agents not in the coalition, the result is propagated, and DAC is restored.

Next, in polynomial time the possibility of a constructive manipulation for d_2 given r_2 is checked, and we proceed as above, until all the variables have been considered. After (at most) n polynomial steps, we will either have determined that d cannot win, or we will have defined the SCSPs of the agents in the coalition so that d does win. We notice that such SCSPs have only unary constraints and thus can be solved in polynomial time. □

Theorem 8 *If CCM is NP-complete for one of the local rules, then it is so also for the sequential procedure.*

Proof: We reduce polynomially an instance of CCM for a voting rule r to an instance of CCM for the sequential procedure where one of the local rules is r. An instance of CCM for r consists of a candidate d, a coalition C, and a profile p, written as $CCM(d, C, r, p)$. From such an instance, we construct an instance of CCM for the sequential procedure $Seq_{O,R}$ where R is any finite sequence of voting rules $(r_1, \ldots, r_n)$ including r, say in the i-th position, and O is any ordering of variables $v_1, \ldots, v_n$, all having the same domain, containing all the candidates appearing in p. Also, the soft profile considered for such an instance is $Ext(p)$, the coalition remains C, and the candidate is $E(d) = (d_1, \ldots, d_n)$ where $d_i = d$ and d_j for $j \neq i$ has preference 1 in the SCSPs of $Ext(p)$. We will write it $CCM(E(d), C, Seq_{O,R}, Ext(p))$.

If $CCM(d, C, r, p) = true$, then, in the sequential instance, coalition C can set its preferences in profile $Ext(p)$ to make $E(d)$ win. In fact, such agents can modify their preferences over the domain of v_i in the same way as needed to make d win in the non-sequential instance. Conversely, if $CCM(E(d), C, Seq_{O,R}, Ext(p)) = true$, then we compute the orderings over the domain of v_i when $v_1 = d_1, \ldots v_{i-1} = d_{i-1}$ (via DAC). By using such orderings, the coalition can make d win in the non-sequential instance. □

4 Experimental results

If r is a voting rule, we denote by $seq(r)$ the sequential procedure where r is applied at each step. We will compare r and $seq(r)$ for several voting rules r in terms of computational cost, satisfaction of voters and sensitivity to coalitions.

To perform the experiments, we have randomly generated profiles with tree-shaped FCSPs based on the following parameters: the number of voters m, the number of variables n, the number of domain elements d, and the tightness t (the percentage of tuples with preference 0 in each fuzzy constraint). Given values to such parameters, we generate a profile with m tree-shaped FCSPs defined over n variables, with d elements in the domain of each variable, and where each fuzzy constraint involves two variables and has a number of tuples with preference 0 which is $t\%$ of the maximum number of tuples (that is, d^2). For the other tuples, we randomly generate preference values in $(0, 1]$. In each set of tests, we fix all parameters except one, and let this last parameter vary. When a parameter is fixed, we use the following values: $m = 25$, $n = 5$, $d = 6$ and $t = 20\%$. Also, each result is the average over 50 fuzzy profiles with the same parameters' values. The experiments were conducted on a machine with an Intel Core2 Solo 1.40GHz, with 3GB of RAM. We show detailed results only for Borda, since all other rules we considered (Plurality, Approval, and Copeland) have shown the same trends. We have also run experiments that show that the variable ordering does not influence the computation time nor the satisfaction of the agents. However, such results are omitted due to lack of space.

Computation time. As it can be seen in Fig. 2, the sequential procedure substantially outperforms the non-sequential rule in terms of the time needed to find the winner. An increase in the values of all the parameters, except tightness, causes a non-trivial growth in the computation time of the non-sequential approach, while leaving the time for the sequential approach practically unchanged.

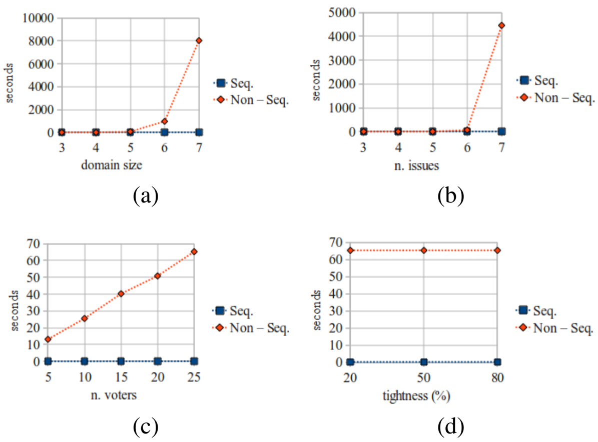


Figure 2: Time for Borda.

Solution quality. We measure the error as the distance, for each voter, of his preference for the winner from the preference of his optimal solutions, averaged over all voters. Quite surprisingly, for all rules, the agents are on average almost equally satisfied with the winner elected by the two approaches (see Figure 3). When the number of values in the domains, or the number of variables, grows, the error grows as well. The reason for this is that an increase in the number of solutions makes it less likely for the winner to have a high preference in each SCSP. The same trend can be observed when the number of voters grows. In this case, however, the higher error is due to an increased amount of disagreement among the larger number of voters. This is supported also by the fact that the preference decreases. When tightness grows, instead, the error decreases. In fact, the less solutions with a non-zero preference each FCSP has, the more likely it is that the winner has a preference close to the optimal one for each voter.

Coalitions. Since each agent's SCSP is generated randomly, the probability that two voters vote equally is very small and thus there is a large amount of disagreement among the agents. To check also instances with some amount of consensus among agents, we considered several partitions of a set of 30 voters into subsets, where all the voters in a subset have exactly the same FCSP. The cardinality of the partitions we have considered are 2, 3, 6, 15, and 30. Our experiments

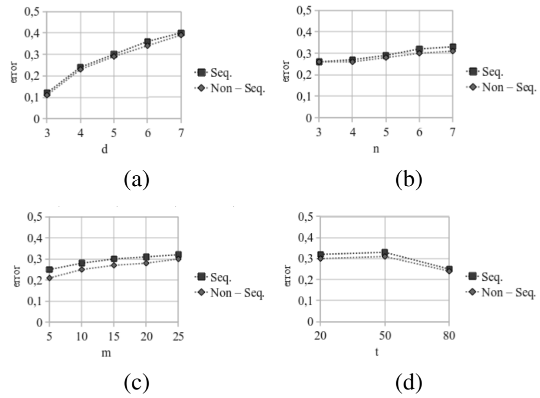


Figure 3: Error for Borda.

(omitted for lack of space) show that the error decreases as consensus increases. Thus the error shown in Fig. 3 is an upper bound w.r.t. to cases with coalitions. This holds for both the sequential and the non-sequential approach. For example, for the Borda rule, the error of the sequential approach ranges from 0.14 with 2 subsets to 0.31 with 30 subsets and for the non-sequential approach from 0.1 with 2 subsets to 0.3 with 30 subsets.

5 Future Work

We plan to consider the complexity of other kinds of manipulation, the presence of feasibility constraints over the decision set, and agents' preferences expressed via a combination of compact preference formalisms.

References

[1] K. J. Arrow and A. K. Sen amd K. Suzumura. *Handbook of Social Choice and Welfare.* North-Holland, 2002.

[2] F. Bacchus and A.J. Grove. Graphical models for preference and utility. In *UAI*, 1995.

[3] C. Boutilier, R. I. Brafman, C. Domshlak, H. H. Hoos, and D. Poole. CP-nets: A tool for representing and reasoning with conditional ceteris paribus preference statements. *JAIR*, 21:135–191, 2004.

[4] V. Conitzer, T. Sandholm, and J. Lang. When are elections with few candidates hard to manipulate. *JACM*, 54(3), 2007.

[5] P. Faliszewski, E. Hemaspaandra, and H. Schnoor. Copeland voting: ties matter. In *AAMAS (2)*, 2008.

[6] C. Gonzales, P. Perny, and S. Queiroz. Preference aggregation with graphical utility models. In *AAAI*, 2008.

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

[8] P. Meseguer, F. Rossi, and T. Schiex. Soft constraints. In P. Van Beek F. Rossi and T. Walsh, editors, *Handbook of Constraint Programming*. Elsevier, 2005.

[9] G. Dalla Pozza, F. Rossi, and B. Venable. Multi-agent soft constraint aggregation:a sequential approach. In *ICAART*, 2011.

[10] A.D. Taylor. *Social Choice and the Mathematics of Manipulation.* Cambridge University Press, 2005.