Resistance to Bribery when Aggregating Soft Constraints (Extended Abstract)

Maria Silvia Pini
University of Padova, Italy
Padova, Italy
pini@dei.unipd.it

Francesca Rossi
University of Padova
Padova, Italy
frossi@math.unipd.it

Kristen Brent Venable
Tulane University & IHMC, USA
kvenabl@tulane.edu

ABSTRACT

We investigate a multi-agent scenario where agents express their preferences over a large set of decisions via soft constraints. We consider sequential procedures (based on Plurality, Approval, and Borda) to aggregate agents' preferences and we study their resistance to bribery attempts to influence the result of the aggregation.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multiagent systems*; F.2 [**Theory of Computation**]: Analysis of Algorithms and Problem Complexity

General Terms

Theory, Algorithms

Keywords

Voting Protocols, Bribery, Soft Constraint Aggregation, Preferences

1. INTRODUCTION

We consider a multi-agent scenario where a collection of agents needs to select a decision from a large set of decisions, over which they express their preferences. This set has a combinatorial structure, i.e., each decision is 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. In this paper we assume that such preferences are modelled by soft constraints. To consider a concrete instance of soft constraints, we focus on fuzzy constraints. The agents' preferences are then aggregated to compute a single "socially optimal" solution via some voting rules. We consider sequential procedures (based on Plurality, Approval, and Borda [1]) to aggregate agents' preferences and we study their resistance to bribery attempts to influence the result of the aggregation. In the bribery problem an external agent, usually called the "briber", has a preferred solution, and tries to get that solution as the result of the voting process, by paying some agents to vote in a certain way, and by doing this while staying within its budget [4]. We measure computational complexity of the bribery problem, thus assuming that a computationally complex bribery problem make the aggregation resistant to bribery. We define several cost schemes for measuring the effort the agent has to make to satisfy the briber's request and we investigate the resistance to bribery of our aggregation procedures for these cost schemes. Bribery when agents vote over a large set of candidates has been considered also in [6, 7], but preferences were modeled via CP-nets and not via soft constraints.

Appears in: *Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2013)*, Ito, Jonker, Gini, and Shehory (eds.), May, 6–10, 2013, Saint Paul, Minnesota, USA.

2. SOFT CONSTRAINTS

A soft constraint [8] involves a set of variables and associates a value from a (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 $a \leq b$ 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, C is a set of soft constraints (each one involving a subset of V), A is the set of preference values. An instance of the SCSP framework is obtained by choosing a specific c-semiring. Choosing $S_{FCSP} = \langle [0,1], max, min, 0, 1\rangle$ means that preferences are in [0,1] and we want to maximize the minimum preference. This is the setting of fuzzy CSPs (FCSPs) that we consider in the paper.

An optimal solution of an SCSP is a complete assignment with an undominated preference. Finding an optimal solution is an NP-hard problem, unless certain restrictions are imposed, such as a tree-shaped constraint graph. Constraint propagation may help the search for an optimal solution. Given a variable ordering o, an 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-shaped constraint graph and the variable ordering is compatible with the father-child relation of the tree [8], since the optimum preference level is the best preference level in the domain of the root variable.

3. VOTING WITH SOFT CONSTRAINTS

Assume to have a set of agents, each one expressing its 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, all the SCSPs have the same set of variables and the same variable domains but they may have different soft constraints, as well as different preferences over the variable domains. In [3] this is the notion of *soft profile*, which is 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. 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. A *fuzzy profile* is a soft profile with fuzzy soft constraints. The idea proposed in [3] to aggregate the preferences in a soft profile in order to compute the winning variable assignment is to sequentially vote on each variable via a voting rule, possibly using a different rule for each variable. Given a soft profile (V,D,P), assume $|V| = n$, and consider an ordering of the variables $O = \langle v_1, \ldots, v_n \rangle$ and a corresponding sequence of local voting rules $R = \langle r_1, \ldots, r_n \rangle$. The sequential procedure is a sequence of n steps, where at each step i, we perform the following tasks. All agents are 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 achieve DAC on their SCSP, considering the ordering O. 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. Finally, the constraint $v_i = d_i$ is added to the preferences of each agent and DAC is achieved to propagate its effect considering the reverse ordering of O. After all n steps have been executed, the winning assignments are collected in the tuple $\langle v_1 = d_1, \ldots, v_n = d_n \rangle$, which is declared the winner of the election. A similar sequential procedure has been considered in [5], when agents' preferences are expressed via CP-nets.

In this paper we employ the sequential procedure described above with Plurality, Approval and Borda. We have an ordering O over the variables and we consider each variable in turn in such an ordering. At each step, each agent provides some information about the considered variable, say X, which depends on the voting rule we use: in *Sequential Plurality* (SP) every agent provides one best value for X, in *Sequential Approval* (SA) all best values for X, while in *Sequential Borda* (SB) a total order (possibly with ties) over the values of X, along with the preference values for each domain element. We then choose one value for the considered variable, as follows: with SP and SA we choose the value voted by the highest number of agents, with SB we select the value with best score, where the score of a value is the sum of its preferences over all the agents. Note that "best" means maximal in the case of fuzzy constraints. Once a value is chosen for a variable, this value is broadcasted to all agents, who fix X to this value in their soft constraints and achieve DAC in the reverse ordering w.r.t. O. We continue with the next variable, and so on until all variables have been handled.

4. BRIBERY PROBLEM AND RESULTS

We now define formally the bribery problem in our scenario, where agents express their preferences via fuzzy soft constraints. We recall that bribery is an attempt to modify the result of the election where there is an outside agent, called the briber, that wants to affect the result of the election by paying some voters to change their votes, while being subject to a limitation of its budget. In defining bribing scenarios in our context, it is thus necessary to decide what the briber can ask an agent to do and how costly it is for the briber to submit a certain request. The cost usually represents the effort the agent has to make to satisfy the briber's request. Notice that the agent can modify the preference values inside its variable domains and/or constraints. We define in several ways the cost of a briber's request, which is to make a certain solution A optimal:

C_{equal}: The cost is fixed (without loss of generality, we will assume it is 1), no matter how many changes are needed to make A optimal;

C_{do}: The cost is the distance from the preference value of A, denoted by $pref(A)$, to the preference value of an optimal solution of the SCSP of the agent, denoted by opt. If we are dealing with fuzzy numbers and we may prefer to have integer costs, the cost is $C_{do} = (opt - pref(A)) * l$, where l is the number of different preference values allowed. For example, if the fuzzy preferences have 2 digits of precision, we have 100 different preferences and we will, thus, have $l = 100$.

C_{don}: The cost is determined by considering both C_{do} and the minimum number of preference values, say t, associated to subparts (aka tuples) of A in the constraints, that must be modified in order to make A optimal. The cost is $C_{don} = ((opt - pref(A)) * l * M) + t$, where M is a large integer which must be greater than $2n-2$ and $1 \leq t \leq 2n-1$, where n is the number of variables. The role of M is to ensure a higher bribery cost for a less preferred candidate.

C_{dow}: The cost is computed similarly to C_{don}, but each preference value to be modified is associated with a cost proportional to the change required on that preference. Let us denote by t_i any tuple of A with preference $\leq opt$. The cost is $C_{dow} = ((opt - pref(A)) * l * M) + \sum_{t_i}(opt - pref(t_i)) * l$, where the role of M is similar to the one in C_{don}, but now its lower bound depends also on the number of preference levels. M must be greater than $l(2n-2)-1$.

We are now ready to define formally our bribery problem: Given a voting rule V and a cost scheme C, (V,C)*-Bribery* is the problem of determining if it is possible to make a preferred candidate win, when voting rule V is used, by bribing agents and by spending less than a certain budget according to cost scheme C. We have studied the computational complexity of this problems for SP, SA, and SB and for the cost schemes defined above. This is an interesting problem since we have shown that winner determination for SP, SA, and SB is computationally easy when agents' preferences are tree-shaped fuzzy CSPs.

Our results are summarized in the table below (NP-c* stands for NP-complete with a restriction on M). We have shown, via reductions from the OPTIMAL LOBBYING (OL) problem [2], that SP, SA, and SB are all resistant to bribery, when the agents express their preferences via tree-shaped fuzzy CSPs and costs are computed according to C_{equal}, C_{do}, C_{don}, or C_{dow}. Results for C_{don}, or C_{dow} require $M > n * m$, where m is the number of voters and n is the number of variables.

	SP	SA	SB
C_{equal}	NP-c	NP-c	NP-c
C_{do}	NP-c	NP-c	NP-c
C_{don}	NP-c*	NP-c*	NP-c*
C_{dow}	NP-c*	NP-c*	NP-c*

We plan to study the resistance to bribery for other voting rules and other bribery cost schemes, as well as the applicability of the bribery results in preference optimization and compilation.

5. REFERENCES

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

[2] R. Christian, M. Fellows, F. Rosamond, and A. Slinko. On complexity of lobbying in multiple referenda. *Review of Economic Design*, 2007.

[3] G. Dalla Pozza, F. Rossi, and K. B. Venable. Multi-agent soft constraint aggregation:a sequential approach. In *Proc. of ICAART'11*, pages 277–282, 2011.

[4] P. Faliszewski, E. Hemaspaandra, and L. Hemaspaandra. How hard is bribery in elections? *JAIR*, 35:485–532, 2009.

[5] J. Lang and L. Xia. Sequential composition of voting rules in multi-issue domains. *Math. Soc. Sciences*, 57(3):304–324, 2009.

[6] N. Mattei, M. S. Pini, K. B. Venable, and F. Rossi. Bribery in voting over combinatorial domains is easy. In *Proc. of AAMAS'12*, 2012.

[7] N. Mattei, M. S. Pini, K. B. Venable, and F. Rossi. Bribery in voting with CP-nets. *Ann. of Math. and Artif. Intell.*, To appear.

[8] F. Rossi, P. V. Beek, and T. Walsh. *Handbook of Constraint Programming*. Elsevier, 2006.