# An Approval-Based Model for Single-Step Liquid Democracy

Evangelos Markakis and Georgios Papasotiropoulos

Athens University of Economics and Business

**Abstract.** We study a Liquid Democracy framework where voters can express preferences in an approval form, regarding being represented by a subset of voters, casting a ballot themselves, or abstaining from the election. We examine, from a computational perspective, the problems of minimizing (resp. maximizing) the number of dissatisfied (resp. satisfied) voters. We first show that these problems are intractable even when each voter approves only a small subset of other voters. On the positive side, we establish constant factor approximation algorithms for that case, and exact algorithms under bounded treewidth of a convenient graph-theoretic representation, even when certain secondary objectives are also present. The results related to the treewidth are based on the powerful methodology of expressing graph properties via Monadic Second Order logic. We believe that this approach can turn out to be fruitful for other graph related questions that appear in Computational Social Choice.

## 1 Introduction

Liquid Democracy (LD) is a voting paradigm that has emerged as a flexible model for enhancing engagement in decision-making. The main idea in LD models is that a voter can choose either to vote herself or to delegate to another voter that she trusts to be more knowledgeable or reliable on the topic under consideration. A delegation under LD, is a transitive action, meaning that the voter not only transfers her own vote but also the voting power that has been delegated to her by others. Experimentations and real deployments have already taken place using platforms that support decision-making under Liquid Democracy. One of the first such systems that was put to real use was Župa, intended for a student union for the University of Novo Mesto in Slovenia. Another example is LiquidFeedback that was used by the German Pirate party (among others). Other political parties (such as the Flux Party in Australia) or regional organisations have also attempted to use or experiment with LD, leading to a growing practical appeal. Even further examples include the experiment run by Google via Google Votes, as well as Civicracy and, the more recently developed, Sovereign. We refer to [23] for an informative survey on these systems.

The interest generated by these attempts, has also led to theoretical studies on relevant voting models and has enriched the research agenda of the community. The goals of these works have been to provide more rigorous foundations and highlight the advantages and the negative aspects of LD models. Starting with the positive side, LD definitely has the potential to incentivize civic participation, both for expert voters on a certain topic, but also for users who feel less confident and can delegate to some other trusted voter. At the same time, it also forms a flexible means of participation, since there are no restrictions for physical presence, and usually there is also an option of instant recall of a delegation, whenever a voter no longer feels well represented.

Coming to the critique that has been made on LD, an issue that can become worrying is the formation of large delegation paths. Such paths tend to be undesirable since a voter who gets to cast a ballot may have a rather different opinion with the first voters of the path, who are being represented by her [16]. Secondly, LD faces the risk of having users accumulating excessive voting power, if no control action is taken [5]. Furthermore, another undesirable phenomenon is the creation of delegation cycles, which could result to a waste of participation for the involved voters. Despite the criticism, LD is still a young and promising field, for promoting novel methods of participation and decision-making, generating an increasing interest in the community. We therefore feel that several aspects have not yet been thoroughly studied, and new models and ideas are worth further investigation. Such efforts can help both in tackling some of the existing criticism but also in identifying additional inherent problems.

**Contribution.** We focus on a model, where voters can have approval-based preferences on the available actions. Each voter can have a set of approved delegations, and may also approve voting herself or even abstaining. Our main goal is the study of centralized algorithms for optimizing the overall satisfaction of the voters. For this objective, under our model, it turns out that it suffices to focus only on delegations to actual voters (i.e., delegation paths of unit length). Even with this simpler solution space, the problems we study turn out to be computationally hard. In Section 3, we start with the natural problem of minimizing the number of dissatisfied voters, where we establish a connection with classic combinatorial optimization problems, such as SET COVER and DOMINATING SET. we present approximation preserving reductions which allow us to obtain almost tight approximability and hardness results. The main conclusion from these is that one can have a small constant factor approximation when each voter approves a small number of possible representatives. A constant factor approximation can also be obtained for the variant of maximizing the number of satisfied voters, through a different approach of modeling this as a constraint satisfaction problem. Moving on, in Section 4, we consider the design of exact algorithms for the same problems. Our major highlight is the use of a logic-based technique, where it suffices to express properties by formulas in Monadic Second Order logic. In a nutshell, this approach yields an FPT algorithm, whenever the treewidth of an appropriate graph-theoretic representation of our problem is constant. Under the same restriction, polynomial time algorithms also exist when adding certain secondary objectives on top of minimizing (resp. maximizing) dissatisfaction (resp. satisfaction). To our knowledge, this framework has not received much attention in the social choice community and we expect that it could have further applicability for related problems.

### 1.1 Related Work

To position our work with respect to existing literature, we note that the works most related to ours are [15] and [12]. In terms of the model, we are mostly based on [15], which studies centralized algorithms and where voters specify possible delegations in an approval format. Coming to the differences, their model does not allow abstainers (which we do), but more importantly, [15] studies a different objective and no notion of satisfaction needs to be introduced (in Section 4 we also examine a related question). Our main optimization criteria are inspired mostly by [12], which among others, tries to quantify voters' dissatisfaction. Our differences with [12] is that they have voters with rank-based preferences and their optimization is w.r.t. equilibrium profiles and not over all possible delegations (in Section 5, we also provide a game-theoretic direction with some initial findings). We note also that these works, like ours, are agnostic to the final election outcome (preferences are w.r.t. delegations and not on actual votes).

More generally, the LD-related literature within computational social choice concerns *(i)* comparisons with direct democracy models, *(ii)* game-theoretic stability of delegations, *(iii)* axiomatic approaches. Concerning the first topic, local delegation mechanisms, under which every voter independently is making a choice, have been explored in [18,8]. For the second direction, one can view an LD framework as a game in which the voters can make a choice according to some given preference profile. Such games have been considered in [12,13]. At the same time, game-theoretic aspects have also been studied in [4] and, for the case of weighted delegations, in [27]. Concerning the third direction, a range of delegation schemes have been proposed to avoid delegation cycles [19], accumulation of high power in the election procedure [15] and existence of inconsistent outcomes [9]. Related paradigms to LD have also been considered, e.g. in [1,7,10].

## 2 Preliminaries

### 2.1 Approval Single-Step Liquid Model

We denote by $V = \{1, \ldots, n\}$ the set of agents who participate in the election process and we will refer to all members of $V$ as voters (even though some of them may eventually not vote themselves). In the suggested model, which we refer to as Approval Single-Step Liquid model (ASSL), each voter $i \in V$ needs to express her preferences, in an approval-based format, on the options of *(i)* casting a ballot herself, *(ii)* abstaining from the election, *(iii)* delegating her vote to voter $j \in V \setminus \{i\}$. Namely, a voter may approve any combination of

- casting a ballot herself. We let $C$ denote the set of all such voters.
- abstaining from the voting procedure (e.g., because she feels not well-informed on the topic). We let $A$ denote the set of all such voters.
- delegating her vote to some other voter she trusts. For every $v \in V$, we denote by $N(v)$ the set of approved delegatees of $v$.

Note that we place no restriction on whether a voter accepts one or more of the above options (or even none of them). Hence, in a given instance it may be true that $C \cap A \neq \emptyset$ or that $C \cup A = \emptyset$ or that $v \in C$ and at the same time $N(v) \neq \emptyset$, etc. It is often natural and convenient to think of a graph-theoretic representation of the approved delegations. Hence, for every instance, we associate a directed graph $G = (V, E)$, such that $N(v)$ is the set of out-neighbors of $v$, i.e., $deg^+(v) = |N(v)|$, where $deg^+(v)$ is the out-degree of $v$. This will be particularly useful in Section 4.

Let a delegation function $d : V \to V \cup \{\perp\}$ express the final decision for each voter. We say that $d(v) = v$, if voter $v$ votes, $d(v) = \perp$ if she abstains, and $d(v) = u \in N(v)$, if she delegates to voter $u$. Given a delegation function $d(\cdot)$, we refer to a voter who casts a ballot as a *guru*. The guru of a voter $v \in V$, denoted by $gu(v)$, can be found by following the successive delegations, as given by a delegation function $d(\cdot)$, starting from $v$ until reaching a guru (if possible). Formally, $gu(v) = u$ if there exists a sequence of voters $u_1, \ldots, u_\ell$ such that $d(u_k) = u_{k+1}$ for every $k \in \{1, 2, \ldots, \ell - 1\}, u_1 = v, u_\ell = u$ and $d(u) = u$. Obviously, $gu(v) = v$ if $v$ votes. In case the delegation path starting from $v$ ends up in a voter $u$ for which $d(u) = \perp$ then we say that $v$ does not have a guru and we set $gu(v) = \infty$. Additionally, we do the same for the case where the successive delegations starting from some voter $v$ form or end up in a cycle.

We say that a voter $v$ is satisfied with the delegation function $d(\cdot)$ if $v$ approves the outcome regarding her participation or her representation by another guru-voter. This means that either $d(v) = v$ and $v \in C$ or that $d(v) = \perp$ and $v \in A$ or that $d(v) = u$, with $u \neq v$, $u \neq \perp$ and $gu(u) \in N(v)$. In all other cases, the voter is dissatisfied. Our work mainly deals with the problem of finding centralized mechanisms for the following computational problems:

| MINIMUM SOCIAL COST (MIN-SC)/MAXIMUM SOCIAL GOOD (MAX-SG) | |
|---|---|
| **Given:** | An instance of ASSL, i.e., the approval preferences of $n$ voters regarding their intention to vote, abstain and delegate. |
| **Output:** | A delegation function that minimizes the number of dissatisfied voters/maximizes the number of satisfied voters. |

### 2.2 Warm-up Observations

We start with some observations that will help us tackle the algorithmic problems under consideration. Given an instance of ASSL, let $G$ be the corresponding graph with the approved delegations, as described in the previous subsection. A delegation function $d(\cdot)$, induces a subgraph of $G$ that we denote by $G(d)$, so that $(u, v)$ is an edge in $G(d)$ if and only if $d(u) = v$. Clearly, the out-degree of every vertex in $G(d)$ is at most one and thus it can contain isolated vertices, directed trees oriented towards the gurus, but in general it can also contain cycles, the presence of which can only deteriorate the solution. The next claim shows that we can significantly reduce our solution space. Its proof together with any other missing proof are deferred to the full version of the work.

**Claim 1** . *Consider a solution given by a delegation function $d(\cdot)$. There always exists a solution $d'(\cdot)$ which is at least as good (i.e., the number of satisfied voters is at least as high) as $d(\cdot)$, so that $G(d')$ is a collection of disjoint directed (towards the central vertex) stars, and voters that abstain form isolated vertices.*

Claim 1 justifies the name ASSL. One may discern similarities with proxy voting models (see e.g. [2]), under which every voter is being represented by her delegator, since no transitivity of votes is taken into account. Nevertheless, we still like to think of our model as a Liquid Democracy variant, because it is precisely the objectives that we study together with the centralized approach that enforce Claim 1. When discussing decentralized scenarios or game-theoretic questions (as we do in Section 5), longer delegation paths may also appear.

The next claim shows that for certain voters, we can a priori determine their action, when looking for an optimal solution and that we can be sure about the action of any voter who is dissatisfied under a given delegation function.

**Claim 2** . *Consider a solution given by a delegation function $d(\cdot)$. There always exists a solution which is at least as good (i.e., the number of satisfied voters is at least as high) as $d(\cdot)$, in which (i) every voter in $C$ casts a ballot and (ii) if any voter is dissatisfied, it is because she is casting a ballot without approving it.*

Claim 2 takes care of voters in $C$. We cannot state something similar for the rest of the voters, since it might be socially better to dissatisfy a certain voter by asking her to cast a ballot so as to make other people (pointing to her) satisfied. In practice, this can also occur in cases where voting may be costly (in time or effort to become more informed) and one member of a community may need to act in favor of the common good, outweighing her cost.

## 3 Approximation Algorithms and Hardness Results

In this section we will mainly focus on MIN-SC, but we will also examine MAX-SG in Section 3.2 and further related questions in Section 3.3. We pay particular attention to instances where every voter approves only a constant number of other voters, i.e., $\Delta = \max_v |N(v)| = O(1)$. We find this to be a realistic case, as it is rather expected that voters cannot easily trust a big subset of the electorate.

### 3.1 Social Cost Minimization

We start by showing that the problem is intractable even when each voter approves at most 2 other voters. In fact, we show that our problem encodes a directed version of the DOMINATING SET problem, hence, beyond NP-hardness, we also inherit known results concerning hardness of approximation.

**Theorem 1.** *Let $\Delta = \max_{v \in V} |N(v)| \geq 2$. When $\Delta$ is constant, it is NP-hard to approximate* MIN-SC *with a ratio smaller than* $\max\{1.36, \Delta - 1\}$. *For general instances, it is NP-hard to achieve an approximation better than* $\ln n - \Theta(1) \ln \ln n$.

Since hardness results have been established for $\Delta \geq 2$, it is natural to question whether an optimal algorithm could be found for the case of $\Delta \leq 1$. This scenario is far from unexciting. Consider for instance a spatial model where voters are represented by points in some Euclidean space, interpreted as opinions on the outcomes of some issues. If each voter approves for delegation only the nearest located voter to her, we have precisely that $\Delta = 1$. The following theorem provides an affirmative answer in the above stated question (its proof is actually a direct Corollary of Theorem 5 from Section 4).

**Theorem 2.** *When $\Delta \leq 1$,* MIN-SC *can be solved in polynomial time.*

For higher values of $\Delta$, we can only hope for approximation algorithms. As we show next, we complement Theorem 1 with asymptotically tight approximation guarantees by reducing MIN-SC to the SET COVER problem.

**Theorem 3.** *Let $\Delta = \max_{v \in V} |N(v)| \geq 2$. When $\Delta$ is constant there is a polynomial time algorithm for* MIN-SC *with a constant approximation ratio of* $(\Delta{+}1)$*. For general instances, the problem is* $(\ln n - \ln\ln n + \Theta(1))$*-approximable.*

*Proof.* We will present a reduction that preserves approximability to the SET COVER problem. In an unweighted SET COVER instance, we are given a universe $U$ and a collection $\mathcal{F}$ of subsets of $U$, and ask to find a cover of the universe with the minimum possible number of sets from $\mathcal{F}$. From an instance $I$ of MIN-SC we create an instance $I'$ of SET COVER as follows: We create a universe of elements $U$ by adding one element for every voter, except for certain voters for which there is no such need. In particular, $U$ contains one element for every $v \in V \setminus (C \cup A \cup \{u : \exists u' \in N(u) \cap C\})$. This means that in $U$ we have excluded voters who can be satisfied without delegating to someone else as well as voters who can be satisfied by delegating to members of $C$ (observe that because of Claim 2 (part $(i)$), all voters of $C$ will be assigned to vote). Furthermore, to describe the collection $\mathcal{F}$ of sets in $I'$, for every voter $v \in V \setminus C$ we add the set $S_v = U \cap (\{u : v \in N(u)\} \cup \{v\})$. If some $S_v$ turns out to be the empty set, it can be simply disregarded (e.g. for a voter $v$ with $N(v) \cap C \neq \emptyset$).

**Lemma 1.** *Let $OPT(I), OPT(I')$ be the costs of the optimal solutions in the instances $I$ and $I'$ respectively. Then $OPT(I') \leq OPT(I)$.*

*Proof of Lemma 1.* Let there be $k$ dissatisfied voters in the optimal solution of the MIN-SC instance $I$. By making use of Claim 2, we can assume that these are members of $V \setminus C$ who are assigned to cast a ballot. Hence, for a dissatisfied voter $v$ there exists a corresponding set $S_v$ in $I'$. We will argue that by selecting these $k$ sets that correspond to dissatisfied voters, we have a feasible solution for the SET COVER problem. Towards contradiction, assume that there is an element in $I'$ that has not been covered by any of these sets. Because of the definition of $U$, there must exist a voter $v$ in $I$ who corresponds to that element and who only accepts to delegate to some voters who are not in $C$, i.e., each $u \in N(v)$ has a corresponding set $S_u$ in $\mathcal{F}$ since $u \in V \setminus C$. Moreover, $v$ should be satisfied in

$I$, otherwise the set $S_v$ would have been selected in the solution we constructed for $I'$ and $v$ would have been covered. Therefore, at least one of her approved voters, say $u$, is a guru, and the set $S_u$ covers $v$, which is a contradiction. □

**Lemma 2.** *Given a feasible solution with cost $SOL(I')$ of the produced instance $I'$, we can create a feasible solution of $I$, with cost $SOL(I) \leq SOL(I')$.*

*Proof of Lemma 2.* Say that we are given a solution for $I'$ with cost $SOL(I') = k$, which means that by selecting a number of $k$ sets, it is possible to cover every element of $U$. Consider a delegation function $d(\cdot)$ which asks every voter from $V \setminus C$ whose corresponding set has been selected in the cover, to cast a ballot. Following Claim 2, it also asks every voter from $C$ to cast a ballot. From these, only the former $k$ voters are dissatisfied, who vote but do not belong to $C$. We will argue that we can make all the remaining voters satisfied and hence we will have a solution with $k$ dissatisfied voters.

Consider a voter $v \in V \setminus C$, whose set $S_v$ was not included in the SET COVER solution. If $v \in A$, then $v$ is assigned to abstain and she is satisfied. So, suppose that $v \in V \setminus (A \cup C)$ and also that $N(v) \neq \emptyset$ (otherwise, with $N(v) = \emptyset$, then $S_v$ would have been selected in the cover). There are now two cases to consider:

*Case 1:* $N(v) \cap C \neq \emptyset$. Then $v$ can delegate to a member of $C$ and be satisfied.

*Case 2:* $N(v) \cap C = \emptyset$. Then by the construction of the universe $U$, we have that $v \in U$. Since we have selected a cover for $U$, $v$ is covered by some set. Additionally, we have assumed that $S_v$ was not picked in the cover, hence $v$ is covered by some other set, say $S_u$, which means that $u$ is a voter who is assigned to cast a vote and $v \in S_u$. But then $v$ can delegate to $u$ and be satisfied. □

By combining Lemma 1 and Lemma 2, we have that if we run any $\alpha$-approximation algorithm for the SET COVER instance $I'$, we can find a solution for the MIN-SC instance $I$, with the same guarantee since $SOL(I) \leq SOL(I') \leq \alpha OPT(I') \leq \alpha OPT(I)$. Recall that there exists a well known $f$-approximation algorithm for SET COVER, where $f$ is the maximum number of sets that contain any element. Note also that in our construction, each element of $I'$ that corresponds to a voter $v$ of $I$, belongs to at most $|N(v)| + 1$ sets. This directly yields a $(\Delta + 1)$-approximation for our problem. Alternatively, when $\Delta$ is not bounded, we can use the best currently known approximation algorithm for the SET COVER problem, presented in [25], to obtain the desired result. □

### 3.2 Social Good Maximization

In all voting problems that involve a notion of dissatisfaction, one can study either minimization of dissatisfactions or maximization of satisfactions. The minimization version is slightly more popular, see e.g., [12] (also, in approval voting elections, it is more common to minimize the sum of distances from the optimal solution than to maximize the satisfaction score). Clearly, for ASSL, if we can solve optimally MIN-SC, the same holds for MAX-SG. The problems however can differ on their approximability properties.

Looking back on our findings for MIN-SC, we note that the results from Theorem 1 immediately yield NP-hardness for MAX-SG. The hardness of approximation however does not transfer. The result of Theorem 2 also applies.

**Corollary 1.** *Let $\Delta = \max_{v \in V} |N(v)|$. Then* MAX-SG *is NP-hard even when $\Delta = 2$, and it is efficiently solvable when $\Delta \leq 1$.*

Next, we also provide a constant factor approximation for constant $\Delta$, albeit with a worse constant than the results for MIN-SC. The main insight for the next theorem is that we can exploit results from the rich domain of Constraint Satisfaction Problems (CSPs) and model MAX-SG as such.

**Theorem 4.** *Let $\Delta = \max_{v \in V} |N(v)| \geq 2$. When $\Delta$ is constant there is a polynomial time algorithm for* MAX-SG *with an approximation ratio of $\frac{1}{(\Delta+2)^{\Delta+2}}$.*

We leave as an open problem the question of whether there exist better approximations or whether one can establish hardness of approximation results.

### 3.3 Further Implications: Instances with Bounded Social Cost

We conclude this section by discussing some implications that can be derived by the reductions presented in Section 3.1, on relevant questions to MIN-SC. Let us start with the special case where the optimal cost of an instance $I$ is zero, i.e., it is possible to satisfy all voters. Can we have an algorithm that detects this? It would be ideal to compute a delegation function that does not cause any dissatisfactions, and this is indeed possible. If $OPT(I) = 0$, then any approximation algorithm for MIN-SC of finite ratio will necessarily return an optimal solution. If $OPT(I) > 0$, the approximation algorithm will also return a positive cost. Hence, by using Theorem 3 we have the following:

**Corollary 2.** *Given an instance of* ASSL*, there exists a polynomial time algorithm that decides if it is possible to satisfy all voters, in which case it can also construct an optimal delegation function.*

Taking it a step further, suppose now we ask: Given an instance $I$, is it true that $OPT(I) \leq k$, for some positive constant $k$? This time, we can construct a SET COVER instance $I'$ using the reduction presented in the proof of Theorem 3, and then we can enumerate all possible collections of subsets of size at most $k$. If a solution is found, it corresponds to a set of at most $k$ dissatisfied voters. Hence, we can solve the problem in time $n^{O(k)}$. But now we can question whether there is hope for a substantially better running time. To answer this, we exploit the reduction used in Theorem 1 from DIRECTED DOMINATING SET. In particular, it is known by [14] that DOMINATING SET is $W[2]$-hard when parameterized by the solution cost, even in graphs of bounded average degree. Given that the directed version of DOMINATING SET inherits the hardness results of the undirected version in combination with the proof of Theorem 1, we get:

**Corollary 3.** *Unless $W[2] = FPT$,* MIN-SC *cannot be solved in time $f(k)n^{O(1)}$ even for the case where $\Delta$ is constant, where $f(k)$ is a computable function depending only on the minimum possible number $k$ of dissatisfied voters.*

## 4 Exact Algorithms via Monadic Second Order Logic

The goal of this section is to focus on special cases that admit exact polynomial time algorithms. Our major highlight is the use of a logic-based technique for obtaining such algorithms. To our knowledge, this framework has not received much attention (if at all) from the computational social choice community despite its wide applicability on graph-theoretic problems. We therefore expect that this has the potential of further deployments for other related problems.

### 4.1 Optimization under Bounded Treewidth

The general methodology involves the use of an algorithmic meta-theorem (for related surveys see [17] and [21]) to check the satisfiability of a formula that expresses a graph property, defined over an input graph of bounded treewidth. Roughly speaking, the treewidth is a graph parameter that indicates the "tree-likeness" of a graph. It was introduced independently by various authors mainly for undirected graphs (see [24] for an extended exposition of the origin of the notion) but its definition and intuition can be extended to directed graphs as well [3]. In our case, we will require bounded treewidth for the directed graph associated to an instance of ASSL.

The approach presented here was initiated by Courcelle [11], who used Monadic Second Order (MSO) logic to define graph properties. These, typically ask for some set of vertices or edges subject to certain constraints. For expressing a property in MSO, we can make use of variables for edges, vertices as well as for subsets of them. Apart from the variables, we can also have the usual[1] boolean connectives $\neg, \wedge, \vee, \Rightarrow$, quantifiers $\forall, \exists$, and the membership operator $\in$. The resulting running time for deciding properties expressible in MSO turns out to be exponentially dependent on the treewidth and the size of the formula.

After Courcelle's theorem, there have been several works that extend the algorithmic implications of MSO logic. Most importantly, and most relevant to us, the framework of [3] can handle some types of optimization problems. Consider a formula $\phi(X_1, \dots, X_r)$ in MSO, having $X_1, \dots, X_r$ as free set variables, so that a property is true if there exists an assignment to the free variables that make $\phi$ satisfied. Then, we can optimize a weighted sum over elements that belong to any such set variable, subject to the formula $\phi$ being true (one needs to be careful though as the weights are taken in unary form). A representative example presented in [3] (see Theorem 3.6 therein for a wide variety of tractable problems w.r.t. treewidth) is MINIMUM DOMINATING SET in which we want to minimize $|X|$ subject to a formula that enforces the set $X$ to be a dominating set.

We note that the results we use here require to have a representation of the tree decomposition of the input graph. But even if this is not readily available, its computation is in FPT w.r.t. the treewidth [6].

[1] For ease of presentation, we will also use some set operations that although they are not explicitly allowed, they can be easily replaced by equivalent MSO expressions. For instance, $x \notin A \setminus B \equiv \neg((x \in A) \wedge \neg(x \in B))$ and $A \subseteq B \equiv (\forall x \in A \Rightarrow x \in B)$.

Our first result in this section shows that MIN-SC and MAX-SG are tractable when the treewidth of the associated graph is constant.

**Theorem 5.** *Consider an instance of* ASSL*, and let $G$ be its corresponding graph. Then* MIN-SC *and* MAX-SG *are in FPT w.r.t. the treewidth of $G$.*

*Proof.* It suffices to solve MIN-SC since this yields an optimal solution to MAX-SG as well. In order to apply a framework of MSO logic, we first make a small modification to the graph $G$. We add a special vertex denoted by $a$ and we add a directed edge $(v, a)$ for every $v$ for which $v \in A$. In this manner, abstainers will be encoded by "delegating" their vote to $a$. Let $G' = (V', E')$ be the resulting graph, where $V' = V \cup \{a\}$ and $E' = E \cup \{(v, a) : v \in A\}$. We observe that these additions do not affect the boundedness of the treewidth.

**Lemma 3.** *If $G$ has bounded treewidth, so does $G'$.*

We will create an MSO formula $\phi(D, X)$ with 2 free variables, $D$ and $X$, encoding an edge-set and a vertex set respectively. The rationale is that $\phi(D, X)$ becomes true when the edges of $D$ encode a delegation function and $X$ denotes the set of voters who are dissatisfied by the delegations of $D$. To write the formula, we also exploit the fact that the framework of [3] allows the use of a constant number of "distinguished" sets so that we can quantify over them as well (apart from quantification over $V'$ and $E'$). We will use $V$, along with $C$ and $A$ (of voters who approve casting a ballot or abstaining respectively), as these special sets here. To proceed, $\phi(D, X)$ is the following formula:

$$
\begin{aligned}
&D \subseteq E' \wedge X \subseteq V \setminus C \wedge \\
&(\forall v \in V' \ (deg^+_D(v) \leq 1)) \wedge (\forall u, v, w \in V'((u, v) \in D \Rightarrow (v, w) \notin D))\wedge \\
&(\forall v \in C \ (deg^+_D(v) = 0)) \wedge \\
&(\forall v \in V \ (v \in X \Leftrightarrow (deg^+_D(v) = 0 \wedge v \notin C)))
\end{aligned}
$$

The term $deg^+_D(v) = 0$ can be expressed in MSO logic in a similar way to the more general term of $deg^+_D(v) \leq 1$, which we define formally as

$$deg^+_D(v) \leq 1 \equiv (\exists u \in V' \ (v, u) \in D) \Rightarrow \neg(\exists w \in V' \ ((v, w) \in D \wedge w \neq u)).$$

Concerning the construction of $\phi(D, X)$, the second line expresses the fact that $D$ is a union of disjoint directed stars so as to enforce Claim 1. Anyone with out-degree equal to one within $D$ either delegates to some other voter or abstains (i.e. delegates to vertex $a$), whereas those with out-degree equal to zero in $D$ cast a vote themselves. The third line of $\phi(D, X)$ also enforces Claim 2 (part (i)) so that members of $C$ always cast a vote. The fourth line expresses the fact that the vertices of $X$ are dissatisfied voters. By Claim 2 (part (ii)), the only way to make a voter $v$ dissatisfied is by asking her to cast a ballot when $v \notin C$. Indeed, voters who are not asked to cast a ballot, have out-degree equal to one in $D$, so they either abstain or delegate. This means that either $(v, a) \in D$ or $(v, u) \in D$ for some $u \in V$. In the former case, $v$ is satisfied because $v \in A$ (if $v \notin A$ then the edge $(v, a)$ would not exist in $E'$ and could not have been selected in $D$). In the latter case, $v$ approves $u$ (otherwise the edge $(v, u)$ would not exist) and $u$ casts a vote since $D$ contains only stars. Hence $v$ is again satisfied.

The final step is to perform optimization w.r.t. $|X|$ subject to $\phi(D, X)$ being true. To that end, we can assign a weight $w(v)$ to every vertex $v$ such that $w(a) = 0$ and $w(v) = 1, \forall v \in V' \setminus \{a\}$. Hence $\sum_{v \in X} w(v) = |X|$. Using the result of [3], we can find a delegation function $d(\cdot)$, as given by the edges in $D$, that minimizes the number of dissatisfied voters within the feasible solutions. □

### 4.2 Adding Secondary Objectives

We continue with exhibiting that MSO frameworks can be useful for tackling other related problems as well. For the cases when we can solve MIN-SC (and MAX-SG) optimally, we are investigating whether we can find such a solution with additional properties (whenever the optimal is not unique). Motivated by questions studied in [12], [13] and [15] we consider the following problems:

1. Among the optimal solutions to MIN-SC (or MAX-SG), find one in which a given voter $v$ casts a vote, or answer that no such solution exists.
2. Ditto, with having voter $v$ abstain in an optimal solution.
3. Among the optimal solutions to MIN-SC (or MAX-SG), find one that minimizes the number of abstainers.
4. Among the optimal solutions to MIN-SC (or MAX-SG), find one that minimizes the maximum voting power over all gurus, i.e. the number of voters that she represents (or equivalently that minimizes the maximum in-degree).

The fourth problem is quite important in models of LD, given also the critique that often applies on such models that may accumulate excessive power on some voters. Below we start by addressing the first three problems together.

**Theorem 6.** *Consider an instance of* ASSL*, and let $G$ be its corresponding graph. It is in FPT w.r.t. the treewidth of $G$ to find an optimal solution to* MIN-SC *and* MAX-SG*, in which a given voter casts a ballot or abstains (if such a solution exists). The same holds for minimizing the number of abstainers.*

We come now to the fourth problem, which is the most challenging one. For this, we will use yet another enriched version of the MSO framework, which facilitates the addition of further constraints and helps in solving several degree-constrained optimization problems. As these problems are in general more difficult [22], the results of [26] and [20] yield polynomial time algorithms w.r.t. treewidth, but do not place them in FPT.

For the presentation we will stick to the terminology of [20]. Consider a formula $\phi(X_1, \dots, X_r)$ with free variables $X_1, \dots X_r$. The main idea is to add so-called global and local cardinality constraints and ask for an assignment that satisfies both $\phi$ and the constraints. In the simpler version that we will use here, a global cardinality constraint is of the form $\sum_{i \in [r]} a_i |X_i| \le b$ for given rational numbers $a_i, i \in [k]$ and $b$ (some of these numbers can be zero so that we constrain the cardinality of only some of the free variables). On the other hand, a local cardinality constraint for a vertex has to do with limiting the number of its neighbors or incident edges that belong to a set corresponding to a free variable. For example, if $X_1$ is a free variable of $\phi$ that encodes a vertex set, and $X_2$ is a free variable encoding an edge set, we can have constraints of the form "for each vertex $v$ of $G$, the number of vertices in $X_1$ adjacent to $v$ belongs to a set $a(v)$", where $a(v)$ contains the allowed values (e.g., could be an interval). Similarly, we can express that the number of edges of $X_2$ incident with $v$ can take only specific values from some set $a'(v)$. A nice representative illustration for local constraints in [20] is the CAPACITATED DOMINATING SET problem, where one needs to pick a dominating set $D$ respecting capacity constraints for every $v \in D$.

**Theorem 7.** *Consider an instance of* ASSL *where the associated graph $G$ has constant treewidth. Then among the optimal solutions to* MIN-SC*, we can find in polynomial time a solution that minimizes the maximum in-degree of the gurus.*

*Proof Sketch.* Starting from the directed graph $G$, let $G' = (V', E')$ be the graph used in the proof of Theorem 5, derived from $G$. Our first step is to use Theorem 5 and solve MIN-SC optimally so that we know the cost of an optimal solution. Suppose that we have $c$ unsatisfied voters in an optimal solution.

In order to proceed and utilize the extended MSO framework of [20], we need to work with an undirected graph. To this end, we create an undirected graph $H = (V'', E'')$ from the directed graph $G'$ by having each $v \in V$ correspond to 2 vertices, $v_{in}$ and $v_{out}$ in $V''$. In this manner, out-going edges from $v$ will correspond to edges incident with $v_{out}$ in $H$ whereas incoming edges to $v$ will be incident to $v_{in}$. The graph $H$ will also include the node $a$ for the abstentions, so that in total $V'' = V_{in} \cup V_{out} \cup \{a\}$. Given this construction, it is easy to verify that if $G$ has bounded treewidth, so does $H$.

The next step is to produce a formula for the undirected graph $H$, whose satisfying assignments will correspond to valid delegation functions on the original graph $G$. We will denote our formula by $\psi(D, F, X)$, with the free variables $D, F, X$. As in Theorem 5, the set $D$ will be a subset of edges encoding a valid delegation function. The set $F$ will encode the set of voters who cast a ballot themselves. Finally, the set $X$ will encode the dissatisfied voters induced by $D$.

Following the framework of [20], we now add 2 classes of constraints that we want to be satisfied in addition to $\psi(D, F, X)$. The first one is a so-called global cardinality constraint to ensure that the number of dissatisfied voters is no more than the optimal. Since we have already solved MIN-SC and the solution is $c$, and since $X$ expresses the set of dissatisfied voters, the constraint will be $|X| \leq c$.

Finally, we add the so-called local cardinality constraints. We will produce a set of constraints depended on a fixed number $d \in \{0, 1, \ldots, n\}$ such that the constraints will ensure that the maximum degree of every vertex in $F$ is bounded by $d$. By using [20], we can now decide for every $d$ if there is an assignment to the variables $D, F, X$ that satisfies $\psi(D, F, X)$ together with the global and local cardinality constraints. To summarize, the steps of the overall algorithm are:

(1) Use Theorem 5 to find the optimal number of dissatisfied voters.
(2) Transform $G$ to the undirected graph $H$ and construct the formula $\psi(D, F, X)$.
(3) For $d = 0$ to $n$, decide if $\psi(D, F, X)$ is satisfiable subject to the global and local cardinality constraints introduced above. Stop in the first iteration where this is true and create the delegation function from $D$ and $F$. □

## 5 Discussion and Other Directions

We have presented a model that allows voters to express preferences over delegations via an approval set. Our main goal has been to optimize the overall satisfaction of the voters, which implies that it suffices to focus only on direct delegations to actual voters. Even under this simpler solution space, the problems we study are intractable, even when the out-degree is a small constant. On the positive side, we have exhibited constant factor approximation algorithms for graphs of constant maximum out-degree, as well as exact algorithms under the bounded treewidth condition, even when secondary objectives are also present. It is therefore interesting to see if any other parameter can play a crucial role on the problem's complexity.

All our results also hold under the generalized model where a graph $G$ is given so that the out-neighborhood $N(v)$, of voter $v$, expresses the set of feasible delegations which is a (possibly strict) superset of her approved delegations. On the other hand, the case where the approved delegatees of a voter $v$ are not necessarily neighbors seems more complex (e.g., a voter approves some other person but cannot directly delegate to her due to hierarchy constraints). Finally, the results of Section 3 also hold for weighted voters, whereas the results of Section 4 only hold if the weights are polynomially bounded in unary form.

Another worthwhile direction comes from the fact that the MSO framework primarily serves as a theoretical tool for placing a problem in a certain complexity class but yields impractical running times. One could proceed with a theoretical and/or experimental study of tailor-made dynamic programming algorithms for the problems presented in Section 4. Coming to our last result (Theorem 7), an interesting approach for future work would be to provide algorithms with trade-offs between the total dissatisfaction and the maximum voting power (instead of optimizing one objective and keeping the other as a secondary objective).

### 5.1 Towards a Game-Theoretic Analysis

We conclude our work with a preliminary game-theoretic analysis, which can serve as the basis for a more elaborate future study of these models. Motivated by the approach of [12,13], we define the following simple game: Say that in an instance of ASSL each voter $v$ acts as a strategic player, whose strategy space is $N(v) \cup \{v, \perp\}$. The utility that she can earn from an outcome is either 1, if she is satisfied with that outcome, or 0 otherwise. The first relevant question is whether such games admit pure Nash equilibria, i.e., delegation functions under which no voter is able to unilaterally change her strategy and increase her utility.

In contrast to the model of rank-based preferences of [12,13], in our case, Nash equilibria are guaranteed to exist.

**Proposition 1.** *In every instance of* ASSL*, there exists a pure Nash equilibrium, which can be computed in polynomial time.*

In order to evaluate the equilibria of a game (in terms of the derived social good, or similarly in terms of social cost), we can use the Price of Anarchy as a standard metric. This can be defined as the worst possible ratio between the optimal solution for the social good against the number of satisfied voters at a Nash Equilibrium. Unfortunately, we show below that strategic behavior can lead to quite undesirable solutions and we note that this could act as an argument in favor of using a centralized mechanism, as done in the previous sections, to avoid such bad outcomes.

**Proposition 2.** *The Price of Anarchy for the strategic games of the* ASSL *model, can be as bad as $\Omega(n)$, even when $\Delta \leq 1$.*

Finally, note that Proposition 2 raises the question of coming up with richer game-theoretic models of the delegation process (e.g. richer utility functions or repeated games) so as to understand thoroughly the effects of strategic behavior.

*Acknowledgements.* This work has been supported by the Hellenic Foundation for Research and Innovation (H.F.R.I.) under the "First Call for H.F.R.I. Research Projects to support faculty members and researchers and the procurement of high-cost research equipment" grant (Project Number: HFRI-FM17-3512).

## References

1. Abramowitz, B., Mattei, N.: Flexible representative democracy: An introduction with binary issues. In: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, (IJCAI-19). pp. 3–10 (2019)
2. Anshelevich, E., Fitzsimmons, Z., Vaish, R., Xia, L.: Representative proxy voting. arXiv preprint arXiv:2012.06747 (2020)
3. Arnborg, S., Lagergren, J., Seese, D.: Easy problems for tree-decomposable graphs. Journal of Algorithms **12**(2), 308–340 (1991)
4. Bloembergen, D., Grossi, D., Lackner, M.: On rational delegations in liquid democracy. In: Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence, (AAAI-19). pp. 1796–1803 (2019)
5. Blum, C., Zuber, C.I.: Liquid democracy: Potentials, problems, and perspectives. Journal of Political Philosophy **24**(2), 162–182 (2016)
6. Bodlaender, H.L.: A linear-time algorithm for finding tree-decompositions of small treewidth. SIAM Journal on Computing **25**(6), 1305–1317 (1996)
7. Boldi, P., Bonchi, F., Castillo, C., Vigna, S.: Viscous democracy for social networks. Communications of the ACM **54**(6), 129–137 (2011)
8. Caragiannis, I., Micha, E.: A contribution to the critique of liquid democracy. In: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, (IJCAI-19). pp. 116–122 (2019)

9. Christoff, Z., Grossi, D.: Binary voting with delegable proxy: An analysis of liquid democracy. In: Proceedings of the Sixteenth Conference on Theoretical Aspects of Rationality and Knowledge, (TARK-17). pp. 134–150 (2017)
10. Colley, R., Grandi, U., Novaro, A.: Smart voting. In: Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, (IJCAI-20). pp. 1734–1740 (2021)
11. Courcelle, B.: The monadic second-order logic of graphs. I. Recognizable sets of finite graphs. Information and Computation **85**(1), 12–75 (1990)
12. Escoffier, B., Gilbert, H., Pass-Lanneau, A.: The convergence of iterative delegations in liquid democracy in a social network. In: Proceedings of the Twelfth International Symposium on Algorithmic Game Theory, (SAGT-19). pp. 284–297 (2019)
13. Escoffier, B., Gilbert, H., Pass-Lanneau, A.: Iterative delegations in liquid democracy with restricted preferences. In: Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence, (AAAI-20). pp. 1926–1933 (2020)
14. Golovach, P.A., Villanger, Y.: Parameterized complexity for domination problems on degenerate graphs. In: Proceedings of the Thirty-Fourth International Workshop on Graph-Theoretic Concepts in Computer Science. pp. 195–205 (2008)
15. Gölz, P., Kahng, A., Mackenzie, S., Procaccia, A.D.: The fluid mechanics of liquid democracy. In: Proceedings of the Fourteenth International Conference on Web and Internet Economics, (WINE-18). pp. 188–202 (2018)
16. Green-Armytage, J.: Direct voting and proxy voting. Constitutional Political Economy **26**(2), 190–220 (2015)
17. Grohe, M.: Logic, graphs, and algorithms. Logic and Automata **2**, 357–422 (2008)
18. Kahng, A., Mackenzie, S., Procaccia, A.: Liquid democracy: An algorithmic perspective. Journal of Artificial Intelligence Research **70**, 1223–1252 (2021)
19. Kavitha, T., Király, T., Matuschke, J., Schlotter, I., Schmidt-Kraepelin, U.: Popular branchings and their dual certificates. In: Proceedings of the Twenty-First International Conference on Integer Programming and Combinatorial Optimization. pp. 223–237 (2020)
20. Knop, D., Koutecký, M., Masarík, T., Toufar, T.: Simplified algorithmic metatheorems beyond MSO: treewidth and neighborhood diversity. Logical Methods in Computer Science **15**(4), 12:1–12:32 (2019)
21. Kreutzer, S.: Algorithmic meta-theorems. In: International Workshop on Parameterized and Exact Computation. pp. 10–12. Springer (2008)
22. Masařík, T., Toufar, T.: Parameterized complexity of fair deletion problems. Discrete Applied Mathematics **278**, 51–61 (2020)
23. Paulin, A.: An overview of ten years of liquid democracy research. In: The Proceedings of the Twenty-First Annual International Conference on Digital Government Research. pp. 116–121 (2020)
24. Seymour, P.: The origin of the notion of treewidth. Theoretical Computer Science Stack Exchange (2014), https://cstheory.stackexchange.com/q/27317, accessed at 2021-16-05
25. Slavık, P.: A tight analysis of the greedy algorithm for set cover. Journal of Algorithms **25**(2), 237–254 (1997)
26. Szeider, S.: Monadic second order logic on graphs with local cardinality constraints. ACM Transactions on Computational Logic **12**(2), 1–21 (2011)
27. Zhang, Y., Grossi, D.: Tracking truth by weighting proxies in liquid democracy. arXiv preprint arXiv:2103.09081 (2021)