Single-Winner Voting on Matchings

Niclas Boehmer
Hasso Plattner Institute
University of Potsdam
Potsdam, Germany
niclas.boehmer@hpi.de

Jessica Dierking
Hasso Plattner Institute
University of Potsdam
Potsdam, Germany
jessica.dierking@hpi.de

ABSTRACT

We introduce a single-winner perspective on *voting on matchings*, in which voters have preferences over possible matchings in a graph, and the goal is to select a single collectively desirable matching. Unlike in classical matching problems, voters in our model are not part of the graph; instead, they have preferences over the entire matching. In the resulting election, the candidate space consists of all feasible matchings, whose exponential size renders standard algorithms for identifying socially desirable outcomes computationally infeasible. We study whether the computational tractability of finding such outcomes can be regained by exploiting the matching structure of the candidate space. Specifically, we provide a complete complexity landscape for questions concerning the maximization of social welfare, the construction and verification of Pareto optimal outcomes, and the existence and verification of Condorcet winners under one affine and two approval-based utility models. Our results consist of a mix of algorithmic and intractability results, revealing sharp boundaries between tractable and intractable cases, with complexity jumps arising from subtle changes in the utility model or solution concept.

KEYWORDS

Matchings; Single-Winner Voting; Algorithmic Analysis; Condorcet winner; Pareto optimality

ACM Reference Format:
Niclas Boehmer and Jessica Dierking. 2026. Single-Winner Voting on Matchings. In *Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), Paphos, Cyprus, May 25 – 29, 2026*, IFAAMAS, 9 pages. https://doi.org/10.65109/RKNE1743

1 INTRODUCTION

Alice is the head of HR at Condorcet Consulting. Each year, Condorcet Consulting offers several different internship positions and receives a large number of applications. The task of the HR team is to decide which applicant should be assigned to which internship position, with some applicants only being eligible for a subset of the positions. This task becomes all the more difficult because the CEO, CTO, and other members of the management board of the company would like to implement different assignments of the applicants to positions. Therefore, Alice and her team must find a way to aggregate these conflicting opinions into a single assignment. She quickly realizes that even deciding which principles she should apply when choosing a desirable assignment is not straightforward. What does it mean for one assignment to be better than another? And can one even determine that a proposed assignment has no clearly superior alternative?

The situation reminds her of voting scenarios, in which voters have preferences over candidates and a single compromise candidate must be selected: In her problem, each possible assignment of applicants to positions constitutes a candidate, and the members of the management board act as voters who express preferences over these candidates. Alice describes her problem to her friend Bob, who is an expert in voting. He notices that, while similar in spirit, the problem differs in some ways fundamentally from standard voting scenarios, where the candidate set is typically small and efficiently enumerable. In contrast, the number of possible assignments in Alice's problem grows exponentially with the number of applicants and positions, rendering most classical voting algorithms computationally intractable, as it becomes infeasible to iterate over all candidates. The two are thus left wondering how to aggregate the different preferences and how to do so in a computationally efficient way.

We model Alice's problem as *voting on matchings*: given an underlying graph G, select a single matching based on the voters' preferences over possible matchings. The motivating example corresponds to a special case in which G is bipartite, with internship positions on one side and applicants on the other.[1] We put forward a new perspective on decision-making in matching under preferences and resource allocation: In the extensive literature on matching under preferences [16, 20] and on resource allocation [7], agents are typically part of the underlying graph and have preferences over their matched partner or allocated resource. In other words, each agent cares only about the part of the outcome that directly affects them. In contrast, we adopt a centralized perspective where voters represent entities that express preferences over the entire outcome. As a result, our model can capture situations in which multiple central stakeholders—like the members of the management board of a company or teachers in a class—have preferences over the allocation of resources to individuals or the matching of people into pairs.

In this paper, we focus on the problem of selecting a single winning solution in a voting on matchings instance under three classic notions of social desirability.

Social Welfare Optimization: Selecting candidates that maximize collective utility, evaluated through utilitarian or egalitarian welfare.

Pareto Optimality: Constructing and verifying candidates for which no other candidate is preferred by all voters.

[1] In general, we do not assume the underlying graph to be bipartite, although almost all of our hardness and algorithmic results also hold in this case.

Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), C. Amato, L. Dennis, V. Mascardi, J. Thangarajah (eds.), May 25 – 29, 2026, Paphos, Cyprus. © 2026 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). https://doi.org/10.65109/RKNE1743

Condorcet Criterion: Proving existence or verifying *Condorcet winners*, i.e., candidates that defeat all others in pairwise majority comparisons.

1.1 Our Contributions

We analyze the computational complexity of identifying desirable outcomes when voting on matchings in a given graph. The exponential number of possible matchings renders standard voting algorithms inapplicable. Underlying our study is the central question of whether the combinatorial structure of matchings—the feasibility constraint defining the candidate space—suffices to restore tractability for classical notions of collective optimality.

We provide a complete complexity landscape for three key solution concepts—social welfare, Pareto optimality, and Condorcet winners—under three utility models (see Table 1).[2] Each voter specifies a preferred matching, which we lift to utilities under three models, inspired by standard scoring vectors from social choice theory: In the affine utility model, the utility a voter derives from a matching is an affine function of the overlap with the voter's preferred matching, generalizing, for instance, Borda-style scoring. The other two models are approval-based, i.e., a matching provides either zero or one utility: In the one-edge approval model, a voter approves a matching whenever its overlap with the preferred matching is nonempty. In the κ-missing approval model, parameterized by κ, a voter approves a matching whenever at most κ edges from their preferred matching are missing.

Overall, examining Table 1, we find that most problems are computationally intractable, with Pareto optimality emerging as the "most tractable" of the three solution concepts. Still, we find several cases where we can use the problem's underlying structure to derive polynomial-time results, thereby identifying the boundaries of tractability. However, the discovered islands of tractability are highly fragile: even minor modifications, such as increasing the parameter κ or strengthening the solution concept from weak to strong Condorcet winners, can lead to a sharp jump in complexity. Returning to our central question, our findings indicate that, in the case of matchings, the combinatorial structure of the candidate space alone does not suffice to restore the tractability of classic notions of collective optimality in the face of an exponential candidate space. We next discuss our results in more detail. Results whose proofs are (partly) deferred to the full version [6] are marked with (★).

In Section 3, we study utilitarian and egalitarian welfare. We show that maximizing utilitarian welfare is polynomial-time solvable under affine utilities but NP-complete for the approval-based utility models. For egalitarian welfare, we show NP-completeness under affine and one-edge approval utilities. For κ-missing approval, we show that maximizing egalitarian welfare is closely related to Satisfiability, allowing us to mirror the complexity jump between 2-Sat and 3-Sat to show the same complexity jump between 1-missing approval, which we show to be polynomial-time solvable, and 2-missing approval, which we show to be NP-hard. In Section 4, we discuss (weak and strong) Pareto optimality. We

[2] Our polynomial-time results hold for general graphs. We do not explicitly seek to show NP-hardness for specific graph classes; nonetheless, almost all of our hardness reductions hold on bipartite graphs and simple graph structures such as collections of paths and stars.

Table 1: Overview of the results. See Section 2 for definitions. Results for "affine" hold for all affine utility functions and results for "κ-missing" for any $\kappa \in \mathbb{N}_0$.

<table>
<tr><th></th><th>Affine</th><th>One-Edge</th><th>κ-missing ($\kappa > 1$)</th><th>κ-missing ($0 \leq \kappa \leq 1$)</th></tr>
<tr><td>Utilitarian Welfare</td><td>P [Theorem 3.1]</td><td colspan="3">NP-complete [Theorem 3.5]</td></tr>
<tr><td>Egalitarian Welfare</td><td colspan="3">NP-complete [Theorems 3.2 and 3.4]</td><td>P [Theorem 3.3]</td></tr>
<tr><td>wPO-Construction</td><td colspan="4">P [Observation 4.4]</td></tr>
<tr><td>sPO-Construction</td><td>P [Proposition 4.5]</td><td colspan="2">NP-hard [Proposition 4.7]</td><td>P [Proposition 4.6]</td></tr>
<tr><td>wPO-Verification</td><td colspan="3" rowspan="2">coNP-complete [Theorem 4.2]</td><td rowspan="2">P [Theorem 4.3]</td></tr>
<tr><td>sPO-Verification</td></tr>
<tr><td>wCW-Existence</td><td rowspan="2">NP-hard [Theorem 5.3]</td><td colspan="3">P [Observation 5.4]</td></tr>
<tr><td>sCW-Existence</td><td colspan="3">NP-hard [Theorem 5.5]</td></tr>
<tr><td>wCW-Verification</td><td colspan="4" rowspan="2">coNP-complete [Theorem 5.2]</td></tr>
<tr><td>sCW-Verification</td></tr>
</table>

show that constructing Pareto optimal candidates is generally easier than verifying whether a given candidate is Pareto optimal. In the approval-based settings, we connect the verification of Pareto optimal candidates to the problem of maximizing egalitarian welfare, yielding the same complexity jump as before. In the affine case, we can utilize the feasibility of maximizing utilitarian welfare to construct strongly Pareto optimal candidates, while verifying Pareto optimality turns out to be coNP-complete, which we show via a reduction from the corresponding problem in approval-based multiwinner voting. Section 5 deals with Condorcet winners. We show that deciding the existence of Condorcet winners is NP-hard across all our utility models (except for the trivial existence of weak Condorcet winners under approval-based utilities). The verification of weak and strong Condorcet winners is coNP-complete under all considered utility functions.

In Section 6, we examine a variant of our problem in which we restrict the candidate space to the set of *maximal matchings*, asking whether this restriction is sufficient to regain tractability. This investigation is motivated by two considerations. First, implementing a non-maximal matching is arguably unreasonable, as such matchings are clearly inefficient. Second, natural hardness proofs for many of our problems rely on voters specifying non-maximal matchings—intuitively, because this allows voters to constrain only a small part of the decision. To obtain more robust hardness results and rule out that hardness relies on the presence of such voters, we consider the restriction to maximal matchings.[3]

Utilizing this restriction, we are able to show that for κ-missing utilities, the size of the candidate space is effectively rendered polynomial for a fixed κ (see Theorem 6.1), which yields XP algorithms for all considered computational problems. For the remaining problems, except for the verification of weakly Pareto optimal candidates in the one-edge approval model, which becomes trivial, we show that reducing the candidate space to maximal matchings does not regain tractability.

[3] In this spirit, most hardness proofs directly establish hardness for instances restricted to maximal matchings, thereby implying hardness for the general case. All polynomial-time algorithms for the general case naturally apply to the restricted setting as well.

1.2 Related Work

The problem of selecting a matching based on agents' preferences has been extensively studied in algorithmic game theory and computational social choice. Foundational work on stable and popular matchings includes [8, 14, 20]. More recently, adopting a multiwinner voting perspective, Boehmer et al. [5] examined the problem of selecting k matchings that proportionally reflect agents' preferences. In contrast to most of this literature, in which agents are embedded in the graph and have preferences over the partner(s) to whom they are matched, our approach assumes that voters have preferences over the entire matching. We model the problem as a single-winner voting scenario with an exponentially large candidate space. In the following, we discuss two previously studied voting scenarios that deal with exponentially large candidate spaces and have been extensively studied in the literature: multiwinner voting and voting in combinatorial domains.

In *multiwinner voting*, given a set of candidates, the goal is to select a winning committee, i.e., a subset of the candidates, of a given size [10, 17]. *Voting in combinatorial domains* considers settings in which possible outcomes are defined as assignments to multiple issues, such as choosing a common menu or deciding on multiple referenda at the same time [18].

Most work on both models treats outcomes as decomposable collections of individual candidates or decisions, typically assuming additive utilities, where the utility of an outcome is the sum of the utilities of its components. Within this framework, various fairness concepts, including different notions of proportionality [2, 3, 21, 23, 24], have been analyzed. In contrast, our work adopts a *single-winner perspective*, treating each outcome as a structured, indivisible object, and analyzing its properties compared to all other feasible outcomes. We assume that each voter casts their top choice and derives utility based on a distance notion to that top choice. Freeman et al. [13] study a similar approach in a participatory budgeting setting. Our perspective motivates the study of classical notions like Pareto optimality [19] and the Condorcet criterion [12], which were originally studied in single-winner contexts. We build on results of Aziz and Monnot [4] and Darmann [9], who study the complexity of testing these properties for multiwinner voting. Our work extends these ideas by investigating these criteria for more complex feasibility constraints (beyond simple cardinality bounds) and a broader range of utility models (see the full version of this paper [6] for a more detailed discussion).

Closely related to our feasibility constraint, which requires candidates to form matchings in an underlying graph, are recent generalizations of multiwinner voting that impose structural constraints on feasible committees, such as matroid, packing, or matching constraints [11, 21, 22]. Our model for voting over matchings can be viewed as a variant of matching-constrained multiwinner voting as studied by Fain et al. [11], where candidates are edges of a graph and feasible committees are required to form a matching. Our work differs from these approaches in the considered preference models and solution concepts. While Fain et al. [11] adopt a multiwinner perspective with additive utilities over edges, our model allows for richer utility functions and focuses on classical single-winner criteria such as Pareto optimality, Condorcet winners, and social welfare optimization. In this sense, we extend their framework and related studies, which are primarily centered on proportionality, toward a more general analysis of preference aggregation under structural feasibility constraints.

2 PRELIMINARIES

We will now formally define the voting on matchings model. Let $G = (W, E)$ be a graph. We denote by C the set of all matchings in G, i.e., the subsets of E in which each node is incident to at most one edge. In the following, we refer to matchings as *candidates*. We define a set of *voters* V, where each voter $v \in V$ has a preferred matching $M^*(v) \in C$.[4] We study the problem of selecting a single matching from C based on the voters' preferences.

2.1 Utility Functions

The utility that a voter $v \in V$ derives from a candidate $M \in C$ is defined by a *utility function* $u : V \times C \to \mathbb{R}$. We assume that voters prefer candidates from which they derive higher utilities and are indifferent between candidates from which they derive the same utility. We assume throughout that a voter's utility for a matching depends only on the matching's overlap with the voter's preferred matching. Our utility functions are inspired by standard scoring vectors from single-winner voting. We define three different utility models.

First, we define affine utility functions. Herein, a voter gains strictly more utility whenever a candidate has a bigger overlap with their ideal candidate. Two candidates provide the same utility if and only if they have the same overlap size. Formally:

Definition 2.1 (Affine Utility). Let M be a candidate and $v \in V$ be a voter. Let $\alpha_v \in \mathbb{R}_{>0}$ and $\beta_v \in \mathbb{R}$. An *affine utility function* for v is a utility function of the form $u(v, M) = \alpha_v |M^*(v) \cap M| + \beta_v$.

The coefficients of the utility functions may differ across voters and are treated as part of the voter's representation. That is, a voter $v \in V$ is represented by their preferred matching $M^*(v)$ together with parameters $\alpha_v \in \mathbb{R}_{>0}$ and $\beta_v \in \mathbb{R}$.

Remark: As an extension, one could consider the utility model in which agents assign an individual utility to each edge, and the utility for a matching is the sum of the individual edge utilities. This model is similar to the matching-constrained multiwinner voting studied by Fain et al. [11]. In our case, moving to this extension does not change the complexity results, as hardness carries over directly, and our polynomial-time results can be adapted.

As a second model, we consider two approval-based settings where voters either approve or disapprove a candidate based on a similarity threshold. A voter derives utility one if they approve a candidate and utility zero otherwise.

We start by defining the *one-edge approval model*, in which a voter approves a candidate whenever the overlap with their preferred matching is non-empty, i.e., there is at least one common edge between the candidate and the preferred matching.[5]

[4] We assume without loss of generality that each edge in G is in the preferred matching of at least one voter, as otherwise we can simply delete the edge.

[5] Fix some threshold $\kappa \in \mathbb{N}_0$. When considering one-edge approval parameterized by κ, where voters only approve candidates with an overlap of at least κ edges, all our NP-hardness results still hold: There is a simple reduction from one-edge approval, as we can add $\kappa - 1$ edges that are approved by all voters. As for most of our considered problems, NP-hardness already holds for the one-edge case, we decided to omit the parameterized case.

Definition 2.2 (One-Edge Approval Utility). Let M be a candidate and $v \in V$ be a voter. The utility that v derives from M is

$$u(v, M) := \begin{cases} 1 & \text{if } M^*(v) \cap M \neq \emptyset \\ 0 & \text{otherwise.} \end{cases}$$

We next define the *κ-missing approval model*, parameterized by a threshold $\kappa \in \mathbb{N}_0$, which specifies how much a candidate may deviate from a voter's preferred matching: A voter approves a candidate if at most κ edges from their preferred matching are missing.

Definition 2.3 (κ-Missing Approval Utility). Let M be a candidate and $v \in V$ be a voter. The utility that v derives from M is

$$u(v, M) := \begin{cases} 1 & \text{if } |M^*(v) \setminus M| \leq \kappa \\ 0 & \text{otherwise.} \end{cases}$$

2.2 Solution Concepts

We are interested in a variety of solution concepts.

Social Welfare. We consider two different classical approaches to social welfare. The *utilitarian* social welfare defines social welfare as the sum of the individual utilities across all voters[6], giving rise to the following computational problem:

Definition 2.4 (Utilitarian Welfare). Given candidates C, voters V, and a number $k \in \mathbb{R}$, we ask whether there exists a candidate $c \in C$ whose utilitarian welfare is at least k, i.e., does $\max_{c \in C} \sum_{v \in V} u(v, c) \geq k$ hold?

In contrast, the *egalitarian* social welfare defines social welfare as the minimum utility across all voters.

Definition 2.5 (Egalitarian Welfare). Given candidates C, voters V, and a number $k \in \mathbb{R}$, we ask whether there exists a candidate $c \in C$ whose egalitarian welfare is at least k, i.e., does $\max_{c \in C} \min_{v \in V} u(v, c) \geq k$ hold?

Pareto Optimality. A Pareto improvement for a candidate c formalizes the idea that there exists a candidate c' such that no voter would object to switching from c to c'. A candidate is called Pareto efficient or Pareto optimal if no such improvement exists [19]. We distinguish two types of Pareto improvements, leading to two notions of Pareto optimality. A candidate $c' \in C$ is a *strong Pareto improvement* for a candidate $c \in C$ if all voters derive a strictly higher utility from c' than from c. A candidate for which no such improvement exists is called *weakly Pareto optimal.* A candidate $c' \in C$ is a *weak Pareto improvement* for a candidate $c \in C$ if at least one voter derives strictly higher utility from c' than from c, while no voter derives a strictly smaller utility from c' than from c. A candidate for which no such improvement exists is called *strongly Pareto optimal.* Note that every strong improvement is also a weak improvement, and thus strong Pareto optimality implies weak Pareto optimality. By definition, weakly and strongly Pareto optimal candidates always exist, but are not necessarily unique. We define the problems of finding and verifying Pareto optimal candidates:

Definition 2.6 (w(s)PO-Construction). Given candidates C, and voters V, find a candidate $c \in C$ that is weakly (strongly) Pareto optimal.

Definition 2.7 (w(s)PO-Verification). Given candidates C, voters V, and a candidate $c \in C$, we ask whether c is weakly (strongly) Pareto optimal.

Condorcet Winners. A *Condorcet winner* is a candidate that defeats every other candidate in a pairwise majority comparison. In such a comparison, we count the number of voters who strictly prefer each candidate, ignoring those who are indifferent: Formally, a candidate $c \in C$ wins a pairwise comparison against another candidate $c' \in C$ if more voters prefer c to c' than the other way around, and ties if the numbers are equal. We distinguish between *weak Condorcet winners*, who win or tie the pairwise comparison against every other candidate, and *strong Condorcet winners*, who strictly win all comparisons [12]. Unlike Pareto optimal candidates, Condorcet winners may not exist. We define the problems of deciding whether a Condorcet winner exists and verifying whether a given candidate is a Condorcet winner.

Definition 2.8 (w(s)CW-Existence). Given candidates C and voters V, we ask whether a weak (strong) Condorcet winner exists.

Definition 2.9 (w(s)CW-Verification). Given candidates C, voters V, and a candidate $c \in C$, we ask whether c is a weak (strong) Condorcet winner.

3 SOCIAL WELFARE

In this section, we study the problems Utilitarian Welfare and Egalitarian Welfare. We demonstrate that their computational complexity decisively depends on the underlying utility model. Specifically, we provide polynomial-time algorithms for Utilitarian Welfare under affine utilities and for Egalitarian Welfare under 0-missing and 1-missing approval. For all remaining cases, we establish NP-completeness.

3.1 Affine Utility Functions

In the affine setting, the utility that a voter v gains from a single edge is either α_v if the edge is part of the voter's preferred matching or zero otherwise. Thus, the contribution of an edge to a candidate's utilitarian welfare depends solely on the voters' preferred matchings and their α values, but not on the remaining selected edges. This property allows us to maximize utilitarian welfare in polynomial time by finding a maximum-weight matching.

Theorem 3.1 (★). *Utilitarian Welfare under affine utility functions is solvable in polynomial time.*

Proof sketch. We define a weight function that assigns an edge a weight equal to the utility it contributes to utilitarian welfare. A maximum-weight matching under this weight function maximizes utilitarian welfare and is constructable in polynomial time. ◇

In contrast, the utility an edge contributes to a matching's egalitarian welfare depends on the other selected edges, as the other edges influence whether a voter's utility for a matching exceeds k. It turns out that this lack of "decomposability" renders the problem NP-hard:

[6] When considering affine utilities with $\alpha_v = 1$ and $\beta_v = 0$, maximizing the utilitarian welfare directly corresponds to computing a Borda winner, assuming that a voter gives candidates in the same equivalence class the same score. Maximizing utilitarian welfare in the 0-missing approval setting corresponds to finding a Plurality winner.

Theorem 3.2 (★). *Egalitarian Welfare under affine utility functions is NP-complete.*

Proof sketch. To establish this result, we reduce from 3-Sat by constructing a graph composed of gadgets encoding the variables of the 3-Sat instance, with matchings corresponding to truth assignments. The voters represent the clauses of the instance, where each voter derives positive utility from a matching if and only if the corresponding assignment satisfies the clause. Hence, a candidate with positive egalitarian welfare corresponds directly to a fulfilling assignment of the 3-Sat instance. ◇

3.2 Approval-based Utility Functions

In the approval-based setting, the interpretation of utilitarian and egalitarian welfare changes slightly, since the only possible utilities a matching can provide for a voter are zero and one. Thus, a candidate's utilitarian welfare is simply the number of voters that approve the candidate, while the egalitarian welfare is one if and only if *all* voters approve the candidate. We show that Egalitarian Welfare can be solved in polynomial time under 0-missing approval and 1-missing approval, but remains NP-complete for all other utility models. The hardness of Utilitarian Welfare follows directly when Egalitarian Welfare is hard. We show that Utilitarian Welfare is NP-complete in the other cases as well.

Finding a candidate with positive egalitarian welfare corresponds to finding a candidate approved by all voters. Using a similar idea as in Theorem 3.2, we connect Egalitarian Welfare under κ-missing approval to $\kappa + 1$-Sat. Mirroring the complexity jump between 2-Sat and 3-Sat, we show that Egalitarian Welfare is solvable in polynomial time for $\kappa \in \{0, 1\}$, but becomes NP-complete for $\kappa > 1$. We start by establishing the positive result:

Theorem 3.3 (★). *Egalitarian Welfare under κ-missing approval for $\kappa \leq 1$ is solvable in polynomial time.*

Proof ($\kappa = 1$). Let $G = (W, E)$ be the input graph and V the set of voters. We construct a corresponding 2-Sat instance with $|E|$ variables and $O(|V| \cdot |E|^2)$ clauses. For each edge $e \in E$, we introduce a variable x_e. We add the following types of clauses:

- For every edge $e \in E$ and each edge $\hat{e}$ adjacent to e, we add the clause $(\overline{x_e} \vee \overline{x_{\hat{e}}})$, ensuring the matching constraint. Formally, let $C_M := \{(\overline{x_e} \vee \overline{x_{\hat{e}}}) \mid e \in E, \hat{e} \in N(e)\}$, where $N(e)$ are the edges that share an endpoint with e.
- For each voter $v \in V$, we add a clause for every pair of distinct edges in their preferred matching. That is, for $e \neq \hat{e} \in M^*(v)$, we add clause $(x_e \vee x_{\hat{e}})$. Formally, let $C_v := \{(x_e \vee x_{\hat{e}}) \mid e \neq \hat{e} \in M^*(v)\}$ and $C_V := \bigcup_{v \in V} C_v$.

The complete set of clauses for the 2-Sat formula is $C := C_M \cup C_V$. We solve this 2-Sat instance using a linear-time algorithm, such as the one by Aspvall et al. [1]. If the 2-Sat instance is satisfiable, we return Yes; otherwise, No.

We now show that a candidate with egalitarian welfare one corresponds to a satisfying assignment, and vice versa.

[⇒] Assume there exists a candidate M with an egalitarian welfare of one. That is, M satisfies the matching constraint, and for every voter $v \in V$, at most one edge of $M^*(v)$ is missing from M. Let $x_e = \mathit{true}$ if and only if $e \in M$. Then:

- Each clause $(\overline{x_e} \vee \overline{x_{\hat{e}}}) \in C_M$ is satisfied because M contains at most one of any pair of adjacent edges.
- For each voter v, every clause $(x_e \vee x_{\hat{e}}) \in C_v$ is satisfied because at most one edge from $M^*(v)$ is excluded from M.

[⇐] Assume the 2-Sat instance is satisfiable. Let $M := \{e \mid x_e \text{ is assigned } \mathit{true}\}$. Then:

- M satisfies the matching constraint, since every pair of adjacent edges e and $\hat{e}$ is not simultaneously included, as otherwise $(\overline{x_e} \vee \overline{x_{\hat{e}}})$ would not be satisfied.
- For each voter $v \in V$, the clauses C_v ensure that at most one edge from $M^*(v)$ is excluded from M: If there were two edges $e \neq \hat{e} \in M^*(v) \setminus M$, the clause $(x_e \vee x_{\hat{e}})$ would not be satisfied. Therefore, v approves M, and the candidate has egalitarian welfare of one.

□

In contrast, for κ-missing approval with $\kappa > 1$ and for one-edge approval, we can establish hardness using a similar idea as for Egalitarian Welfare under affine utilities.

Theorem 3.4 (★). *Egalitarian Welfare under one-edge approval and under κ-missing approval with $\kappa > 1$ is NP-complete.*

Proof sketch. The reduction idea is similar to Theorem 3.2. For κ-missing approval, we reduce from $\kappa + 1$-Sat, and for one-edge approval, from 3-Sat. We construct gadgets corresponding to the variables of the Sat instance and voters corresponding to the clauses. A voter approves a candidate if at least one of the edges of the respective clause is selected. Therefore, a matching with egalitarian welfare of one directly translates to a fulfilling assignment of the Sat instance. ◇

For Utilitarian Welfare, NP-hardness for one-edge and κ-missing approval with $\kappa > 1$ follows directly from the NP-hardness of Egalitarian Welfare, which we established in Theorem 3.4: In the approval-based setting, a candidate has an egalitarian welfare of one if and only if it has a utilitarian welfare of $|V|$. In contrast, for 0-missing approval and 1-missing approval, we cannot extend the polynomial-time solvability of Egalitarian Welfare from Theorem 3.3 to Utilitarian Welfare.[7] We show NP-hardness via a reduction from Independent Set.

Theorem 3.5 (★). *For any $\kappa \in \mathbb{N}_0$, Utilitarian Welfare is NP-complete under κ-missing approval and under one-edge approval.*

Proof sketch. For 0-missing approval, we reduce from Independent Set. We construct an instance such that a candidate approved by k voters corresponds to an independent set of size k. Specifically, we add a voter for each node in the input graph $\tilde{G}$. For each edge in $\tilde{G}$, we add a gadget consisting of two connected edges, each corresponding to one endpoint of the original edge in $\tilde{G}$. The gadgets are not connected to each other and pairwise disjoint. A voter corresponding to node i approves all edges that correspond to i. Under 0-missing approval, a voter approves a matching if and only if all of their corresponding edges are included. For the correctness, observe that two voters corresponding to adjacent nodes

[7] The 2-Sat algorithm from before would become a variant of Max 2-Sat which is NP-complete [15].

in $\tilde{G}$ cannot both approve the same matching, since two edges from their preferred matchings share an endpoint.

For 1-missing approval, we add a separate control gadget and additional voters such that any candidate including a specific "blocking" edge cannot reach the required utility threshold. This enforces behavior equivalent to the $\kappa = 0$ case. ◇

4 PARETO OPTIMALITY

We now turn to Pareto optimality. Here, only voters' ordinal preferences over candidates matter. We show that for affine utility functions, the ordinal preference relation is independent of the concrete coefficients of the utility function.

Lemma 4.1 (★). *All affine utility functions agree on whether a voter $v \in V$ prefers candidate M over M'.*

This property holds as voters prefer matchings with larger overlap with their preferred matching under all affine utility functions, and are indifferent between two matchings with the same overlap size. Lemma 4.1 allows us to disregard the specific coefficients of the utility function in the affine setting. Throughout this section, we assume for each voter $v \in V$ that $\alpha_v = 1$ and $\beta_v = 0$.

4.1 Verification

Recall that our goal is to determine whether a given candidate is Pareto optimal. It turns out that verifying weak and strong Pareto optimality is solvable in polynomial time for 0-missing and 1-missing approval, but coNP-complete for all other utility models.

To show hardness, for affine utilities, we can reduce from verifying the Pareto optimality of a given committee in an approval-based multiwinner election, which was shown to be NP-hard by Aziz et al. [4]. In the approval-based settings, the question of whether a candidate is Pareto optimal is closely related to the question of whether there is a matching that is approved by every voter, i.e., that has positive egalitarian welfare. Utilizing this connection, we show coNP-completeness of w/sPO-Verification for one-edge approval and κ-missing approval with $\kappa > 1$. Due to technical reasons, we reduce from $\kappa + 1$-Sat for sPO-Verification under κ-missing approval and from the corresponding Egalitarian Welfare problem in the other cases:

Theorem 4.2 (★). *wPO-Verification and sPO-Verification under affine utility functions, one-edge approval and κ-missing approval with $\kappa > 1$ are coNP-complete.*

For κ-missing approval with $\kappa \leq 1$, we use the polynomial-time algorithms from Theorem 3.3 for Egalitarian Welfare to search for Pareto improvements by checking if there are candidates that are approved by all currently approving and at least one additional voter. This tractability stands in contrast with the NP-hardness under the other utility models, further highlighting the jump in complexity between 1-missing approval and 2-missing approval.

Theorem 4.3. *wPO-Verification and sPO-Verification under κ-missing approval with $\kappa \leq 1$ are solvable in polynomial time.*

Proof. For sPO-Verification, given a graph G, a set of voters V, and a candidate M, let V_M be the set of voters that approve M. For each voter $v \in V \setminus V_M$ we can test if there is a matching in graph G that is approved by all voters in $V_M \cup \{v\}$ by checking if there is a matching with positive egalitarian welfare for graph G and voters $V_M \cup \{v\}$ using the polynomial-time algorithms from Theorem 3.3. If such a matching exists for some voter $v \in V \setminus V_M$, we return No; otherwise, we return Yes.

For wPO-Verification, let G be a graph, V a set of voters, and M a candidate. If M is approved by at least one voter, then M is trivially weakly Pareto optimal, as that voter cannot improve. Otherwise, finding an improving matching is equivalent to finding a matching that is approved by all voters, i.e., a matching with an egalitarian welfare of one, for which we can again use the algorithm described in Theorem 3.3 to test if such a matching exists. □

4.2 Construction

We now turn to the construction of Pareto optimal candidates. We show that finding Pareto optimal candidates is often easier than verifying whether a given candidate is Pareto optimal. This phenomenon is not uncommon, as Pareto optimal candidates are generally not unique, and in some settings, they can be identified efficiently by exploiting simple structural properties. It is trivial to find a weakly Pareto optimal candidate: We can simply select a candidate that provides maximum utility to some voter. Since this voter cannot strictly prefer any other candidate, the selected candidate is weakly Pareto optimal.

Observation 4.4. *wPO-Construction is solvable in linear time for all considered utility functions.*

Finding strongly Pareto optimal candidates is less straightforward, as a dominating candidate only requires one voter to strictly improve. Depending on the utility model, the problem remains solvable in polynomial time or becomes NP-hard. We first discuss the polynomial cases: affine utility functions and κ-missing approval with $\kappa \leq 1$.

Proposition 4.5. *sPO-Construction under affine utility functions is solvable in polynomial time.*

Proof. Under affine utility functions, any candidate with maximum utilitarian welfare is strongly Pareto optimal, since a dominating candidate would have strictly higher utilitarian welfare. Therefore, a candidate with maximum utilitarian welfare, which can be found in polynomial time (see Theorem 3.1), is strongly Pareto optimal. □

In the approval-based setting, a Pareto improvement requires at least one additional voter to approve the candidate, while preserving all existing approvals. The construction of strongly Pareto optimal candidates for 0-missing and 1-missing approval utilizes the polynomial-time solvability of sPO-Verification (Theorem 4.3) by iteratively finding Pareto improvements until the current candidate is Pareto optimal.

Proposition 4.6 (★). *sPO-Construction under κ-missing approval with $\kappa \leq 1$ is solvable in polynomial time.*

Proof sketch. Using the algorithm described in Theorem 4.3, we iteratively find weak improvements to our current candidate until the candidate is strongly Pareto optimal. ◇

Similar to the hardness of sPO-Verification for one-edge approval and κ-missing approval with $\kappa > 1$, we use the corresponding hardness results for Egalitarian Welfare (Theorem 3.4) to show the hardness of sPO-Construction.

Proposition 4.7 (★). *sPO-Construction under one-edge approval and under κ-missing approval with $\kappa > 1$ is NP-hard.*

Proof sketch. The idea is to use sPO-Construction as a subroutine to solve Egalitarian Welfare. In our reduction, if the candidate returned by sPO-Construction is approved by all voters, it has an egalitarian welfare of one. Otherwise, no candidate with positive egalitarian welfare exists, as such a candidate would Pareto dominate the returned candidate. ◇

5 CONDORCET WINNERS

We now investigate the problem of identifying Condorcet winners, i.e., candidates that win (or tie) the pairwise comparison against every other candidate. As with Pareto optimality, only the ordinal preferences of voters over candidates matter when determining Condorcet winners. Thus, by Lemma 4.1, for affine utilities we can disregard the specific coefficients of the utility function and again assume $\alpha_v = 1$ and $\beta_v = 0$ for each voter $v \in V$.

In the approval-based setting, we observe the following useful relationship between Condorcet winners and candidates with maximum utilitarian welfare:

Lemma 5.1 (★ folklore). *In the approval-based setting, a candidate M is a weak Condorcet winner if and only if it has maximum utilitarian welfare. If no other candidate achieves the same utilitarian welfare, then M is also a strong Condorcet winner.*

5.1 Verification

Verifying whether a given candidate is a Condorcet winner is coNP-complete under all considered utility models. The proofs use reductions from 3-Sat and the respective Utilitarian Welfare problems in the approval-based settings, and from wPO-Verification in the affine setting.

Theorem 5.2 (★). *wCW-Verification and sCW-Verification are coNP-complete under affine utilities, one-edge approval and κ-missing approval for any $\kappa \in \mathbb{N}_0$.*

Proof sketch. For the affine setting, we reduce from wPO-Verification. The idea is to add extra voters to the given instance who strictly prefer the given candidate M over any other candidate. These voters receive strictly lower utility from any other candidate, ensuring that another candidate M' can only win or tie the pairwise comparison against M if all original voters strictly prefer M' over M, i.e., if M' constitutes a strong Pareto improvement for M.

In the approval-based setting, we leverage Lemma 5.1 to construct reductions from Utilitarian Welfare and 3-Sat. For κ-missing approval we reduce from Utilitarian Welfare. We modify the instance by inserting a candidate M approved by additional voters who approve no other candidate and ask whether M is a Condorcet winner. The instance is constructed such that if M is not a Condorcet winner, then there exists a candidate with utilitarian welfare of k in the original instance. In the one-edge approval setting, we reduce from 3-Sat, where we use a similar idea as before: We construct gadgets for the variables and voters for the clauses such that a voter approves a candidate if and only if the corresponding clause is satisfied. Using additional control voters, we construct a candidate that is a Condorcet winner unless there exists a fulfilling assignment for the 3-Sat instance. ◇

5.2 Existence

In contrast to Pareto optimal candidates, Condorcet winners are not guaranteed to exist. We therefore study the computational complexity of deciding whether a Condorcet winner exists in a given instance. We first show that wCW-Existence and sCW-Existence are coNP-hard in the affine setting. Then, we turn to the approval-based setting, where only sCW-Existence remains coNP-hard, while wCW-Existence becomes trivial.

To show that wCW-Existence and sCW-Existence under affine utilities are coNP-hard, we can use an idea very similar to the reduction in Theorem 5.2. We reduce from wPO-Verification and transform the instance such that the input candidate M is the Condorcet winner or no Condorcet winner exists.

Theorem 5.3 (★). *wCW-Existence and sCW-Existence under affine utility functions are coNP-hard.*

In the approval-based setting, a weak Condorcet winner is guaranteed to exist: As shown in Lemma 5.1, each candidate with maximum utilitarian welfare is a weak Condorcet winner, and such candidates are naturally guaranteed to exist.

Observation 5.4. *wCW-Existence is solvable in constant time for both one-edge approval and κ-missing approval for any $\kappa \in \mathbb{N}_0$.*

In contrast, the existence of strong Condorcet winners remains coNP-hard; as established in Lemma 5.1, their existence requires the candidate with maximum utilitarian welfare to be unique. To show hardness, we can use a construction similar to the reductions in Theorem 5.2. We construct a graph in which either the candidate with maximum utilitarian welfare or the solution of the Sat-instance is the Condorcet winner, or no Condorcet winner exists.

Theorem 5.5 (★). *sCW-Existence under one-edge approval and κ-missing approval for any $\kappa \in \mathbb{N}_0$ is coNP-hard.*

6 MAXIMAL MATCHINGS

In this section, we analyze a variant of voting on matchings in which we restrict the candidate space to maximal matchings. A matching is *maximal* if no additional edge can be added without violating the matching property. We ask whether this restriction suffices to regain tractability for some of our solution concepts. Our investigation is motivated by the fact that this restriction prunes the candidate space by eliminating "inefficient" and thus suboptimal candidates.

While we believe the maximal matching assumption is generally plausible, we also note that restricting to maximal matchings is not always natural or intuitive. For instance, assume we are in a resource allocation setting with three agents $\{A, B, C\}$ and three resources $\{1, 2, 3\}$, and a voter v wants agent A to get resource 1 and agent B to get resource 2. If the voter does not care at all about agent C, forcing them to include edge $(C, 3)$ in their preferred matching feels counterintuitive, as our model would treat the satisfaction v

derives from $(C, 3)$ the same as for $(A, 1)$ or $(B, 2)$. Especially in the approval-based setting, where the inclusion of a single additional edge can decide between approval and disapproval, this restriction might be too strong.

As this setting is a special case of our general setting, polynomial-time algorithms apply directly. In the following, we will discuss two questions that become feasible. For all other questions, our NP-hardness reductions, which can be found in the full version [6], also hold in the case of maximal matchings. An overview of results for the restricted case can be found in Table 2.

We present the two cases in which tractability is regained. We first show that under κ-missing approval, restricting to maximal matchings makes the number of *relevant* candidates, i.e., candidates that are approved by at least one voter, polynomial for fixed κ.[8] Consequently, standard algorithms that iterate over all relevant candidates yield XP algorithms parameterized by κ for all studied problems.

Theorem 6.1. *For any $\kappa \in \mathbb{N}_0$, under κ-missing approval on maximal matchings, the number of maximal matchings that are approved by at least one voter is in $O\left(|V||E|^{3\kappa}\right)$.*

Proof. Consider a graph G, a set of voters V, and a fixed $\kappa \in \mathbb{N}_0$. For an arbitrary voter $v \in V$, we want to bound the number of matchings that are approved by v. A matching M is approved by v if at most κ edges from $M^*(v)$ are not in M. Let $k \leq \kappa$ and let M be a maximal matching with $|M^*(v) \setminus M| = k$. We aim to bound $|M \setminus M^*(v)|$. Since $M^*(v)$ is maximal, any edge in $M \setminus M^*(v)$ must be adjacent to one of the k edges from $M^*(v) \setminus M$. Each such edge has two endpoints, so $|M \setminus M^*(v)| \leq 2k$. There are $\binom{|M^*(v)|}{k}$ ways to choose $M^*(v) \setminus M$. For each of the $2k$ endpoints of the edges in $M^*(v) \setminus M$, we can choose from at most $|E|$ edges, yielding an upper bound of $|E|^{2k}$ possibilities for the edges in $M \setminus M^*(v)$. Thus, the total number of matchings that are approved by v can be upper bounded by $\sum_{i=0}^{\kappa} \binom{|M^*(v)|}{i} |E|^{2i}$, which is in $O(|E|^{3\kappa})$. Since there are $|V|$ voters, the total number of matchings approved by any voter is in $O(|V||E|^{3\kappa})$. ◻

Further, in contrast to the general case, wPO-Verification under one-edge approval utilities becomes trivial when restricting the candidate space to maximal matchings.[9] As (apart from the trivial case where G is empty) every maximal matching contains at least one edge, it is approved by at least one voter and therefore is weakly Pareto optimal.

Observation 6.2. *wPO-Verification on maximal matchings under one-edge approval is solvable in constant time.*

7 CONCLUSION

We proposed and analyzed a model for single-winner voting on matchings. Unlike prior work, our model captures a broad class of collective decision-making problems in which multiple stakeholders hold global preferences over entire matchings, rather than local preferences concerning their own assignments.

Our comprehensive complexity taxonomy reveals sharp boundaries between tractable and intractable cases. While some variants, such as Utilitarian Welfare under affine utilities or Egalitarian Welfare under 0-missing approval and 1-missing approval, remain computationally feasible, most others quickly become intractable. This demonstrates that the structure imposed by the matching constraint does not compensate for the exponential size of the outcome space, highlighting a fundamental limitation of applying classical single-winner aggregation principles in domains with complex candidate constraints. Further restricting the candidate space, for instance, to maximal matchings, does not generally restore tractability. Designing tractable yet meaningful solution concepts for single-winner voting with complex, exponentially-sized candidate spaces remains an important direction for future research.

Further, it would be interesting to identify natural restrictions on the candidate space or on voter preferences that yield additional tractable cases. In particular, the parameterized complexity with respect to the number of voters remains an open question. For instance, can Egalitarian Welfare under affine utilities be solved in polynomial time for a constant number of voters?

While our analysis focuses on notions of social desirability, it would also be promising to study the complexity of winner determination for specific voting rules in the context of voting on matchings. Assuming that voters provide weak rankings of matchings by their overlap with their preferred matching, our results already answer several such questions:

For finding a Borda winner, polynomial-time solvability follows from the polynomial-time solvability of Utilitarian Welfare under affine utilities (see Footnote 6). For Plurality, winner determination corresponds to Utilitarian Welfare under 0-missing approval, implying NP-hardness in the general case and polynomial-time solvability under maximal matchings. For Condorcet-consistent voting rules, the exact complexity remains open, but our hardness results for Condorcet existence and verification under affine utilities suggest computational intractability.

Table 2: Overview of the results for maximal matchings, with differences to the general case marked in green and yellow.

	Affine	One-Edge	κ-missing
Utilitarian Welfare	P [Theorem 3.1]	NP-complete [Theorem 3.5]	
Egalitarian Welfare	NP-complete [Theorems 3.2 and 3.4]		
wPO-Construction	P [Observation 4.4]		P (for constant κ) [Theorem 6.1]
sPO-Construction	P [Proposition 4.5]	NP-hard [Proposition 4.7]	
wPO-Verification	coNP-complete [Theorem 4.2]	P [Observation 6.2]	
sPO-Verification		coNP-complete [Theorem 4.2]	
wCW-Existence	NP-hard [Theorem 5.3]	P [Observation 5.4]	
sCW-Existence		NP-hard [Theorem 5.5]	
wCW-Verification	coNP-complete [Theorem 5.2]		
sCW-Verification			

[8] In the general setting, no such bound is possible, since a voter's preferred matching may contain few edges; in that case there can even be exponentially many matchings that contain a preferred matching.

[9] This result only holds since we assume that each edge in G is approved by at least one voter. Otherwise, the problem would remain coNP-hard as we could reduce from Egalitarian Welfare similar to the unrestricted case.

REFERENCES

[1] Bengt Aspvall, Michael F Plass, and Robert Endre Tarjan. 1979. A linear-time algorithm for testing the truth of certain quantified boolean formulas. *Information processing letters* 8, 3 (1979), 121–123.

[2] Haris Aziz, Markus Brill, Vincent Conitzer, Edith Elkind, Rupert Freeman, and Toby Walsh. 2017. Justified representation in approval-based committee voting. *Social Choice and Welfare* 48, 2 (2017), 461–485.

[3] Haris Aziz, Edith Elkind, Shenwei Huang, Martin Lackner, Luis Sánchez-Fernández, and Piotr Skowron. 2018. On the complexity of extended and proportional justified representation. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 902–909.

[4] Haris Aziz and Jérôme Monnot. 2020. Computing and testing Pareto optimal committees. *Autonomous Agents and Multi-Agent Systems* 34 (2020), 1–20.

[5] Niclas Boehmer, Markus Brill, and Ulrike Schmidt-Kraepelin. 2025. Proportional representation in matching markets: selecting multiple matchings under dichotomous preferences. *Social Choice and Welfare* 64, 1 (2025), 179–220.

[6] Niclas Boehmer and Jessica Dierking. 2026. Single-Winner Voting on Matchings. arXiv:2601.19653 [cs.GT] https://arxiv.org/abs/2601.19653

[7] Sylvain Bouveret, Yann Chevaleyre, and Nicolas Maudet. 2016. Fair Allocation of Indivisible Goods. In *Handbook of Computational Social Choice*, Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, and Ariel D. Procaccia (Eds.). Cambridge University Press, 284–310.

[8] Ágnes Cseh. 2017. Popular matchings. *Trends in computational social choice* 105, 3 (2017).

[9] Andreas Darmann. 2013. How hard is it to tell which is a Condorcet committee? *Mathematical Social Sciences* 66, 3 (2013), 282–292.

[10] Edith Elkind, Piotr Faliszewski, Piotr Skowron, and Arkadii Slinko. 2017. Properties of multiwinner voting rules. *Social Choice and Welfare* 48 (2017), 599–632.

[11] Brandon Fain, Kamesh Munagala, and Nisarg Shah. 2018. Fair allocation of indivisible public goods. In *Proceedings of the 2018 ACM Conference on Economics and Computation*. 575–592.

[12] Peter C. Fishburn. 1977. Condorcet Social Choice Functions. *SIAM J. Appl. Math.* 33, 3 (1977), 469–489.

[13] Rupert Freeman, David M Pennock, Dominik Peters, and Jennifer Wortman Vaughan. 2019. Truthful aggregation of budget proposals. In *Proceedings of the 2019 ACM Conference on Economics and Computation*. 751–752.

[14] David Gale and Lloyd S Shapley. 1962. College admissions and the stability of marriage. *The American mathematical monthly* 69, 1 (1962), 9–15.

[15] Michael R Garey, David S Johnson, and Larry Stockmeyer. 1974. Some simplified NP-complete problems. In *Proceedings of the sixth annual ACM symposium on Theory of computing*. 47–63.

[16] Bettina Klaus, David F Manlove, and Francesca Rossi. 2016. *Matching under preferences*. Cambridge University Press.

[17] Martin Lackner and Piotr Skowron. 2023. *Multi-winner voting with approval preferences*. Springer Nature.

[18] Jérôme Lang and Lirong Xia. 2016. Voting in Combinatorial Domains. In *Handbook of Computational Social Choice*, Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, and Ariel D. Procaccia (Eds.). Cambridge University Press, 197–222.

[19] Dinh The Luc. 2008. Pareto optimality. In *Pareto optimality, game theory and equilibria*. Springer, 481–515.

[20] David Manlove. 2013. *Algorithmics of matching under preferences*. Vol. 2. World Scientific.

[21] Tomáš Masařík, Grzegorz Pierczyński, and Piotr Skowron. 2024. A generalised theory of proportionality in collective decision making. In *Proceedings of the 25th ACM Conference on Economics and Computation*. 734–754.

[22] Ivan-Aleksandar Mavrov, Kamesh Munagala, and Yiheng Shen. 2023. Fair multiwinner elections with allocation constraints. In *Proceedings of the 24th ACM Conference on Economics and Computation*. 964–990.

[23] Dominik Peters, Grzegorz Pierczyński, and Piotr Skowron. 2021. Proportional participatory budgeting with additive utilities. *Advances in Neural Information Processing Systems* 34 (2021), 12726–12737.

[24] Luis Sánchez-Fernández, Edith Elkind, Martin Lackner, Norberto Fernández, Jesús Fisteus, Pablo Basanta Val, and Piotr Skowron. 2017. Proportional justified representation. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 670–676.