Table 1: Overview of our results. The FPT results with respect to n are trivial, so we omit it. The FPT result marked by (*) is trivial, since both parameters are greater than n. "Para-NP-hard" stands for NP-hardness even with the corresponding parameter being a constant. $\mathcal{M}$ stands for LSum-$\mathcal{R}$ or LMax-$\mathcal{R}$ and $\mathcal{R} \in \{\text{Reg}, \text{Pair}, \text{Balc}, \text{Egal}\}$. Here, $\bar{k} = n - k$, $\bar{t} = \beta - t$, and $\bar{d} = d' - d$, where d' is the minimum $\mathcal{R}$-score, which can be achieved for the LSum(LMax)-$\mathcal{R}$ instance (U, W, L, d').

<table>
<tr><th colspan="2"></th><th>Regret</th><th>Pair</th><th>Balanced</th><th>Egalitarian</th></tr>
<tr><td colspan="2">LSum-$\mathcal{R}$</td><td colspan="2">P (Thm. 4.2)</td><td>?</td><td>P (Thm. 4.2)</td></tr>
<tr><td colspan="2">LMax-$\mathcal{R}$</td><td colspan="2">P (Thm. 4.3)</td><td>NP-hard (Thm. 4.5)</td><td>NP-hard (Thm. 4.4)</td></tr>
<tr><td colspan="2">LPareto-$\mathcal{R}$</td><td colspan="2">P (Thm. 4.6)</td><td colspan="2">co-NP-hard (Thm. 4.7)</td></tr>
<tr><td rowspan="4">(n,t)-$\mathcal{M}$-$\mathcal{R}$</td><td>t</td><td colspan="4">W[1]-hard (Thm. 4.11)</td></tr>
<tr><td>$\bar{t}$</td><td colspan="4">W[2]-hard (Thm. 4.12)</td></tr>
<tr><td>d</td><td colspan="2" rowspan="2">para-NP-hard (Thm. 4.11)</td><td colspan="2" rowspan="2">FPT(*)</td></tr>
<tr><td>$\bar{d}$</td></tr>
<tr><td rowspan="4">(k,β)-$\mathcal{M}$-$\mathcal{R}$</td><td>k</td><td colspan="4">W[1]-hard (Thm. 4.8)</td></tr>
<tr><td>d</td><td colspan="2">para-NP-hard (Thm. 4.8)</td><td colspan="2">W[1]-hard (Thm. 4.8)</td></tr>
<tr><td>$\bar{k}$</td><td colspan="2">W[2]-hard (Thm. 4.9)</td><td colspan="2" rowspan="2">W[1]-hard (Thm. 4.10)</td></tr>
<tr><td>$\bar{d}$</td><td colspan="2">para-NP-hard (Thm. 4.9)</td></tr>
</table>

> ***LSum-$\mathcal{R}$ Matching Problem (LSum-$\mathcal{R}$)***
> **Input**: Two sets of agents U and W of n agents each, a set of preference profiles L, and a positive integer d.
> **Output**: A perfect LSum-$\mathcal{R}$ matching, if exists; otherwise, "No".

We also investigate a generalization of LSum-$\mathcal{R}$ and LMax-$\mathcal{R}$. Given two subsets $U' \subseteq U$ and $W' \subseteq W$, we define the preference profiles collection $L_{U' \cup W'}$ as the preference lists resulting by removing all $x \in \{\{U \setminus U'\} \cup \{W \setminus W'\}\}$ from the lists in L.

Similarly, we only define the problem of finding a (k, t)-LSum-$\mathcal{R}$ matching. (k, t)-LMax-$\mathcal{R}$ is defined analogously.

> ***(k, t)-LSum-$\mathcal{R}$ Matching Problem ((k, t)-LSum-$\mathcal{R}$)***
> **Input**: Two sets of agents U and W of n agents each, a set of preference profiles L with $|L| = \beta$, and three integers d, k, t with $k \leq n$ and $t \leq \beta$.
> **Output**: Two subsets $U' \subseteq U$ and $W' \subseteq W$ with $|U'| = |W'| = k$, and a subset $L' \subseteq L_{U' \cup W'}$ with $|L'| = t$, such that LSum-$\mathcal{R}$ on (U', W', L', d) does not return "No"; otherwise, "No".

2.2 Parameterized Complexity

Parameterized complexity provides a refined exploration of the connection between problem complexity and various problem-specific parameters. A parameterized problem is *fixed-parameter tractable (FPT)* with respect to a parameter k, if there is an $O(f(k) \cdot |I|^{O(1)})$-time algorithm solving the problem, where I denotes the whole input instance and f can be any computable function. Parameterized problems can be classified into many classes with W[1] and W[2] being the basic fixed-parameter intractability classes. For more details on parameterized complexity, we refer to [16, 18, 31]. We study the parameterized complexity of (k, t)-LSum(LMax)-$\mathcal{R}$, and consider the following parameters: $n = |U| = |W|$, k, $\bar{k} = n - k$, t, $\bar{t} = \beta - t$, d, and $\bar{d} = d' - d$, where d' is the minimum $\mathcal{R}$-score, which can be achieved for the LSum(LMax)-$\mathcal{R}$ instance (U, W, L, d').

3 STRUCTURAL PROPERTIES

We first prove some useful structural properties, which are useful for the following complexity study. This section is divided into two parts. We first show a property of instances with only one layer, and then we show that every LSum(LMax)-$\mathcal{R}$ instance with $\mathcal{R}$ being Reg or Pair has an equivalent LSum(LMax)-$\mathcal{R}$ instance with only one layer.

3.1 Special Case $\beta = 1$

The following observation follows from the definitions of LSum-$\mathcal{R}$ and LMax-$\mathcal{R}$.

Observation 3.1. *If there is only one layer, then LSum-$\mathcal{R}$ is equivalent to LMax-$\mathcal{R}$.*

Then we explore the relation between LSum-Egal and LSum-Balc when there is only one layer.

Lemma 3.1. *Given an LSum-Egal instance (U, W, L, d) with $|L| = 1$, we can construct in polynomial time an equivalent LSum-Balc instance $(U \cup P, W \cup Q, L', d')$ with $|L'| = 1$.*

3.2 From Multiple Layers to Single Layer

Next, we prove that with $\mathcal{R}$ being Reg, Pair, Balc or Egal, every LSum-$\mathcal{R}$ instance can be reduced to a one-layer instance.

Lemma 3.2. *Every LSum-$\mathcal{R}$ instance (U, W, L, d) can be reduced in polynomial time to a new equivalent LSum-$\mathcal{R}$ instance with only one layer and $\mathcal{R} \in \{Reg, Pair, Balc, Egal\}$.*

Finally, we prove that with $\mathcal{R}$ being Reg or Pair, every LMax-$\mathcal{R}$ instance can be reduced to a one-layer instance.

Lemma 3.3. *Every LMax-Reg(Pair) instance (U, W, L, d) can be reduced in polynomial time to a new equivalent LMax-Reg(Pair) instance with only one layer.*

4 COMPLEXITY RESULTS

We first present classical complexity results of LSum-$\mathcal{R}$, LMax-$\mathcal{R}$ and LPareto-$\mathcal{R}$. Next, we consider the parameterized complexity of the more general (k, t)-$\mathcal{M}$-$\mathcal{R}$ with $\mathcal{M}$= {LSum, LMax}.

4.1 LSum-$\mathcal{R}$

We first prove that when there is only one layer, LSum-$\mathcal{R}$ with $\mathcal{R}$ being Reg, Pair or Egal can be solved in polynomial time by reducing them to the Minimum weighted perfect matching problem (MWPM). Given a bipartite graph $G = (V_F \cup V_R, E)$ with V_F and V_R being two disjoint vertex sets, each edge $e \in E$ having an integer weight $h(e) \geq 0$, and a positive integer d, MWPM tries to find a perfect matching M with $\sum_{e \in M} h(e) \leq d$. A perfect matching in a graph is a set of disjoint edges saturating all vertices. MWPM can be solved in polynomial time with Hungarian method (also known as the Kuhn–Munkres algorithm) [26, 30].

Theorem 4.1. *LSum-$\mathcal{R}$ with $\mathcal{R}$ being Reg, Pair or Egal is polynomial-time solvable when there is only one layer.*

Proof. Given an LSum-Egal instance (U, W, L, d), we can reduce it to an equivalent MWPM instance $(V_F \cup V_R, E)$. First, for each man $u \in U$, we construct a vertex $v_f \in V_F$, and for each woman a vertex $v_r \in V_R$. We add all possible edges between V_F and V_R. The only difference concerning the constructions of the three LSum-$\mathcal{R}$ instances lies in the weights of the edges. Given a pair $\{u, w\}$ and their corresponding vertices v_f, v_r, we set the weight $h(e)$ of $e = \{v_f, v_r\}$ as follows:

- [For Egal] $h(e) = P_u^l(w) + P_w^l(u)$.
- [For Pair] $h(e) = 0$, if $P_u^l(w) + P_w^l(u) \leq d$; otherwise, $h(e) = \infty$.
- [For Reg] $h(e) = 0$, if $P_u^l(w) \leq d$ and $P_w^l(u) \leq d$; otherwise, $h(e) = \infty$.

Then the construction is complete. The equivalence between the instances of LSum-$\mathcal{R}$ and MWPM is obvious. The construction can be done within $O(2n + n^2) = O(n^2)$ time. Since the Hungarian method needs polynomial time, we can conclude that, when there is only one layer, that is, $\beta = 1$, LSum-$\mathcal{R}$ with $\mathcal{R}$ being Reg, Pair, or Egal is solvable in polynomial-time. □

By Lemmas 3.2 and Theorem 4.1, we can get the following theorem.

Theorem 4.2. *LSum-Reg, LSum-Pair, and LSum-Egal are in P.*

4.2 LMax-$\mathcal{R}$

In analog to Theorem 4.2, Lemma 3.3 and Theorem 4.1 imply the following result.

Theorem 4.3. *LMax-Reg and LMax-Pair are in P.*

Next we show LMax-Egal is NP-hard by reducing the 3sat problem to LMax-Egal. Given a variable set V and a clause set C with each clause containing exactly three literals, 3sat asks whether there exists a satisfying truth assignment that sets at least one literal in each clause to be true.

Theorem 4.4. *LMax-Egal is NP-hard.*

Proof. Given a 3sat instance $(V = \{v_1, \cdots, v_n\}, C = \{c_1, \cdots, c_m\})$, we create for each variable $v_i \in V$, two pairs of agents, namely, $u_i, \bar{u_i} \in U$ and $w_i, \bar{w_i} \in W$. Then, create two sets P and Q of auxillary agents, $P = \{p_1, \cdots, p_{6n+2}\}$ and $Q = \{q_1, \cdots, q_{6n+2}\}$. Then, the LMax-Egal instance has $4n + 12n + 4 = 16n + 4$ agents, where $P \cup U$ forms the man side and $Q \cup W$ the woman side.

Next, we create m layers, one for each clause $c \in C$, where the preference lists of each $x \in P \cup Q$ are the same in all layers. The preference list of each $p_i \in P$ has the following form: $q_i \succ \overrightarrow{Q \setminus \{q_i\}} \succ \overrightarrow{W}$, with $\overrightarrow{S}$ denoting an arbitrary but fixed ordering of a set S. The preference lists of $q_i \in Q$ are set accordingly. For two agents $u_i, \bar{u_i} \in U$ which are created for the same variable v_i, we create $2m$ preference lists of the same form, two lists for each layer l, $\succ_{u_i}^l$ and $\succ_{\bar{u_i}}^l$: $w_i \succ q_1 \succ \bar{w_i} \succ \overrightarrow{Q \setminus \{q_1\}} \succ \overrightarrow{W \setminus \{w_i, \bar{w_i}\}}$, where w_i and $\bar{w_i}$ are also created for $v_i \in V$. The preference lists of w_i and $\bar{w_i}$ then have the form: $u_i \succ \bar{u_i} \succ \overrightarrow{P} \succ \overrightarrow{U \setminus \{u_i, \bar{u_i}\}}$ and each layer has also exactly one such list for each of w_i and $\bar{w_i}$. Next, we make the following modifications to the lists $\succ_{u_i}^l$ and $\succ_{\bar{u_i}}^l$ according to the occurrence of variables in clauses. In each layer l_j, which is according to a clause c_j, we exchange the positions of $\bar{w_i}$ and q_1 in $\succ_{\bar{u_i}}^{l_j}$ for each variable v_i occurring in c_j positively; if v_i occurs negatively in c_j then we exchange the positions of q_1 and $\bar{w_i}$ in $\succ_{u_i}^{l_j}$; if v_i does not occur in c_j, then no change is done to the lists. □

By using a similar technique as in the proof of Lemma 3.1, we can reduce LMax-Egal to LMax-Balc and get the following result.

Theorem 4.5. *LMax-Balc is NP-hard.*

4.3 LPareto-$\mathcal{R}$

In this part we show the computational complexity of LPareto-$\mathcal{R}$. First, we show a basic observation that an LPareto-$\mathcal{R}$ matching exists for every instance.

Observation 4.1. *Given an instance of LPareto-$\mathcal{R}$, an LPareto-$\mathcal{R}$ matching always exists for $\mathcal{R}$ being Reg/Pair/Balc/Egal.*

Note that, by Observation 4.1, the decision version of LPareto-$\mathcal{R}$ is easy to solve for $\mathcal{R} \in$ {Reg, Pair, Balc, Egal }; it returns "Yes" for all instances. However, the constructive version admits different complexity behaviors for Reg, Pair, Balc, and Egal. The constructive version requires to output a Layer Pareto-optimal matching, as defined in this paper. Now, we show LPareto-Reg and LPareto-Pair admit polynomial-time solving strategies.

Theorem 4.6. *LPareto-Reg and LPareto-Pair are in P.*

Proof. The basic idea is that, given an arbitrary M, we search for a matching dominating M. If there is no such matching, then M is returned as an LPareto-Reg(Pair) matching; otherwise, we repeat this process for the dominating matching. If the search for a dominating matching for a given matching is polynomial-time doable, this problem can be solved in polynomial time. The algorithm of finding a dominating matching for M is shown in Algorithm 1. Recall that the $\mathcal{R}$-score of a layer, denoted as $\mathcal{R}(M, l)$ with respect to a matching M and a layer l, equals to the maximum $\mathcal{R}$-score of all agents in this layer, that is, $\mathcal{R}(M, l) = \max_{a \in U \cup W}\{\mathcal{R}(a, M, l)\}$. Given a triple $(n, L, \{d_1, \cdots, d_\beta\})$ with n and d_i being integers, we construct a bipartite graph G=$(U \cup W, E)$ with n pairs of vertices, i.e., $|U| = |W| = n$, and there is an edge between $u_i \in U$ and $w_j \in W$, if both $P^{l_q}_{u_i}(w_j) \leq d_q$ and $P^{l_q}_{w_j}(u_i) \leq d_q$ for all $l_q \in L$ under Reg, or $P^{l_q}_{u_i}(w_j) + P^{l_q}_{w_j}(u_i) \leq d_q$ for all $l_q \in L$ under Pair.

Algorithm 1 Finding a dominating matching for M

Input: Set of preference profiles L, a perfect matching M
Output: M' which dominates M

```
Let d_i = R(M, l_i) with 1 ≤ i ≤ β and R being Reg or Pair
for j = 1 to β do
    For 1 ≤ i ≤ β, let d'_i = d_i
    d'_j = d'_j − 1
    Construct a bipartite graph G with (n, L, {d'_1, ··· , d'_β})
    Find a perfect matching M_P of G
    if M_P ≠ ∅ then
        return M_P
    end if
end for
```

We can use the Hungarian Method to compute a maximum matching of a bipartite graph in polynomial time. Then Algorithm 1 runs in polynomial time. Thus, we can solve LPareto-Reg(Pair) by first finding an arbitrary perfect matching M and then applying Algorithm 1 to improve it. Since each application of Algorithm 1 decreases the Reg(Pair)-score of at least one layer by at least one, and the maximum Reg(Pair)-score of one layer is $n(2n)$. Therefore, the whole progress is in polynomial time, and LPareto-Reg(Pair) is in P. □

Now we investigate the computational complexity of LPareto-Egal and LPareto-Balc. Unfortunately, these problems seem to be at least co-NP-hard, since the LPareto-Egal-Determine problem is co-NP-hard, which given an instance of LPareto-Egal and a matching M_0, decides whether M_0 is a solution of LPareto-Egal, that is, whether there is no other matching M dominating M_0. We define the LPareto-Balc-Determine problem in the similar way.

Theorem 4.7. *LPareto-Egal-Determine and LPareto-Balc-Determine are co-NP-hard.*

Proof. To prove this theorem, we need to prove its complementary problem is NP-hard, that is, given an instance (U, W, L, M_0), deciding whether there is a matching M dominating M_0. We call it LPareto-Egal/Balc-Dominating. We establish the NP-hardness by reducing 3-partition to this problem. Given a set of $3m$ integers $\{a_1, \cdots, a_{3m}\}$ with the total sum of the integers being mB and each a_i satisfying $B/4 < a_i < B/2$, 3-partition decides whether this set of integers can be partitioned into m subsets such that the sum of the numbers in each subset is equal to B and each subset contains exactly three integers. 3-partition is strongly NP-hard, that is, it remains NP-hard even if B can be bounded by a polynomial of m.

We first prove this theorem for the Egalitarian score. Given a 3-partition instance $(A = \{a_1, \cdots, a_{3m}\}, B)$, we create for each integer $a_i \in A$, $m+1$ pairs of agents, namely, $u^j_i \in U$ and $w^j_i \in W$ with $0 \leq j \leq m$. Then, create two sets P and Q of auxillary agents $P = \{p_1, \cdots, p_{3mD-B}\}$ and $Q = \{q_1, \cdots, q_{3mD-B}\}$ with $D = (3mB + m + 2)(m + 1)$. Then the LPareto-Egal-Dominating instance has $3m(m + 1) + 3mD - B$ agents per side, where $P \cup U$ forms the man side and $Q \cup W$ the woman side.

Next, we create $m+1$ layers, $l_1, \cdots, l_{m+1}$, among which the first m layers are created for the m subsets. The preference lists of each $x \in P \cup Q$ are firstly set the same in all layers. For each $p_i \in P$, the preference list has the following form: $q_i \succ \overrightarrow{Q \setminus \{q_i\}} \succ \overrightarrow{W}$, with $\overrightarrow{S}$ denoting an arbitrary but fixed ordering of a set S. The preference lists of $q_i \in Q$ are set accordingly. Then, we switch in $\succ^{l_{m+1}}_{q_i}$, p_i with the agent at the last position for $1 \leq i \leq 3mD - B$. For the agents $u^j_i \in U$ with $0 \leq j \leq m$ which are created for the same integer a_i, we create $(m+1)(m+1)$ preference lists of the same form and add $m+1$ lists to each layer. The preference list of u^j_i has the following form, where $\{w^0_i, \cdots, w^m_i\}$ are also created for $a_i \in A$:

$$\succ^{l_s}_{u^j_i}: w^0_i \succ \cdots \succ w^m_i \succ \overrightarrow{Q} \succ \overrightarrow{W \setminus \{w^0_i, \cdots, w^m_i\}}, \qquad \forall 0 \leq s \leq m$$

The preference lists of $w^j_i \in W$ with $0 \leq j \leq m$ then have the following form and each layer has also exactly $m+1$ such lists:

$$\succ^{l_s}_{w^j_i}: p_1 \succ \cdots \succ p_{3mB} \succ u^0_i \succ \cdots \succ u^m_i \succ p_{3mB+1} \succ \cdots \succ p_{3mD-B}$$
$$\succ \overrightarrow{U \setminus \{u^0_i, \cdots, u^m_i\}}, \qquad \forall 0 \leq s \leq m$$

Next, we make some modifications in the above preference lists. In each layer l_j with $1 \leq j \leq m$, which is according to a subset, we do the following modifications for $\succ^{l_j}_w$ with $w \in W$.

- In $\succ^{l_j}_{w^0_j}$, exchange the positions of u^0_j and p_1, where p_1 is at the first position in $\succ^{l_j}_{w^0_j}$.
- In $\succ^{l_j}_{w^0_i}$ with $1 \leq i \leq 3m$ and $i \neq j$, exchange the positions of u^0_i and $p_{3mB+B-m}$, where $p_{3mB+B-m}$ is the auxillary agent at the $(3mB + 1 + B)$-th position in $\succ^{l_j}_{w^0_i}$.
- In $\succ^{l_j}_{w^j_i}$ with $1 \leq i \leq 3m$, exchange the positions of u^0_i and $p_{3mB+1-a_i}$, where $p_{3mB+1-a_i}$ is the auxillary agent at the $(3mB + 1 - a_i)$-th position in $\succ^{l_j}_{w^j_i}$.

Next, we make modifications for the last layer l_{m+1}. For each $1 \leq i \leq 3m$ and $0 \leq j \leq m$, we switch u^j_i with the agent at the last position in $\succ^{l_{m+1}}_{w^j_i}$, that is, u^j_i is the worst agent that w^j_i can be matched to in layer l_{m+1}. Finally, we set $M_0 = \{\{u^j_i, w^j_i\} | u^j_i \in U, w^j_i \in W\} \cup \{\{p_i, q_i\} | p_i \in P, q_i \in Q\}$. □

4.4 (k,t)-LSum-$\mathcal{R}$ and (k,t)-LMax-$\mathcal{R}$

For a given bound d, there can be instances of LSum-$\mathcal{R}$ and LMax-$\mathcal{R}$, which admit no satisfying matching.[3] In this case, it is desirable to seek for a "maximum" matching, that is, a matching satisfying the score bound with subsets of agents and/or a subset of layers. It turns out that even in the case of taking subsets of agents and keeping the layers unchanged or of taking a subset of layers and keeping the sets of agents unchanged, LSum-$\mathcal{R}$ and LMax-$\mathcal{R}$ become NP-hard for all scoring rules. Thus, we investigate their parameterized complexity and achieve both fixed-parameter tractable and intractable results. The FPT result for (k,β)-$\mathcal{M}$-$\mathcal{R}$ with respect to n is trivial, so we omit it.

Theorem 4.8. *Even with $\beta = 1$, (k,β)-$\mathcal{M}$-$\mathcal{R}$ with $\mathcal{M}$= {LSum, LMax} is W[1]-hard with respect to k under all scoring rules, and para-NP-hard with respect to d under Reg and Pair, and W[1]-hard with respect to d under Balc and Egal.*

Proof. We establish this theorem by a reduction from Clique. Given a graph G=(V,E), Clique asks whether there exists in G a complete subgraph with k' vertices. Clique is W[1]-hard with respect to k' [18]. Given an instance (G,k') of Clique with $G = (V,E)$ and $k' > 1$, we construct a $(k,1)$-$\mathcal{M}$-$\mathcal{R}$ instance $(U,W,\{l\},d)$ as follows. We create, for each vertex $v_i \in V$, one agent u_i in U and one agent w_i in W. In the only layer l, we construct the following preference lists for u_i and w_i.

$$\succ^l_{u_i}:\overrightarrow{W(V\setminus N(v_i))} \succ w_i \succ \overrightarrow{W(N(v_i))}$$
$$\succ^l_{w_i}:u_i \succ \overrightarrow{U\setminus\{u_i\}}$$

Here, $N(v_i)$ denotes the neighbors of v_i in G, and for a subset $V' \subseteq V$, $W(V')$ and $U(V')$ denote the sets of W-agents and U-agents, respectively, which are created according to the vertices in V'. $\overrightarrow{U\setminus\{u_i\}}$ denotes the ordering where the agents in $U\setminus\{u_i\}$ are sorted according to the increasing order of their indices. Set $d = 1, 2, k'$ and $2k'$ under Reg, Pair, Balc, and Egal, respectively, and $k = k'$. □

Theorem 4.9. *Even with $\beta = 1$, (k,β)-$\mathcal{M}$-$\mathcal{R}$ with $\mathcal{M}$= {LSum, LMax} is W[2]-hard with respect to $\bar{k}$, and para-NP-hard with respect to $\bar{d}$ under Reg and Pair.*

Proof. We establish this theorem by a reduction from Dominating set. Given a graph $G = (V,E)$, Dominating set asks whether there is a size-k' subset of V, denoted as D, such that every $v \in V$ is in D or a neighbor of at least one member of D. Dominating set is W[2]-hard with respect to parameter k' [18]. Denote the degree of a vertex v as $deg(v)$, and we may assume that $\forall v \in V$, $deg(v) = r \geq 1$. Let $n = |V|$.

Given a Dominating set instance (G,k') with $G = (V,E)$, we construct a $(k,1)$-$\mathcal{M}$-$\mathcal{R}$ instance $(U \cup P, W \cup Q, \{l\}, d)$ as follows. Let $d = r+1$ for Reg, or $d = 2(r+1)$ for Pair. For each $v_i \in V$, we create $r+1$ man agents u_i^j in U and $r+1$ woman agents w_i^j in W with $0 \leq j \leq r$. Then create two sets of auxillary agents $P = \{p_1,\cdots,p_{n\times(k'+d)}\}$ and $Q = \{q_1,\cdots,q_{n\times(k'+d)}\}$. This means that there are $k'+d$ agents in P and $k'+d$ agents in Q for each $1 \leq i \leq n$. Then, let $P_i = \{p_{(i-1)(k'+d)+1},\cdots,p_{i(k'+d)}\}$ and $Q_i = \{q_{(i-1)(k'+d)+1},\cdots,q_{i(k'+d)}\}$.

Next, we create the preference lists of the agents. Add for each $p_i \in P$, the preference list $\succ^l_{p_i}$: $q_i \succ \overrightarrow{Q\setminus\{q_i\}} \succ \overrightarrow{W}$, and for each $q_i \in Q$, the preference list $\succ^l_{q_i}$: $p_i \succ \overrightarrow{P\setminus\{p_i\}} \succ \overrightarrow{U}$ to the preference profile l, where $\overrightarrow{S}$ denotes an arbitrary but fixed ordering of a set S. For each $1 \leq i \leq n$, we add the following preference lists to l, where $n^i(j)$ is the index of the vertex which is the j-th neighbor of v_i for $1 \leq j \leq r$:

$$\succ^l_{u_i^0}:w_i^0 \succ \overrightarrow{Q_i} \succ \overrightarrow{Q\setminus Q_i} \succ \overrightarrow{W\setminus\{w_i^0\}}$$
$$\succ^l_{u_i^j}:w_i^0 \succ w_i^1 \succ \cdots \succ w_i^{r-1} \succ w^0_{n^i(j)} \succ w_i^r \succ \overrightarrow{Q} \succ \overrightarrow{W\setminus\{w_i^0,\cdots,w_i^r\}\cup\{w^0_{n^i(j)}\}}$$
$$\succ^l_{w_i^0}:u_i^0 \succ \overrightarrow{P_i} \succ \overrightarrow{P\setminus P_i} \succ \overrightarrow{U\setminus\{u_i^0\}}$$
$$\succ^l_{w_i^j}:u_i^0 \succ u_i^1 \succ \cdots \succ u_i^{r-1} \succ u^0_{n^i(j)} \succ u_i^r \succ \overrightarrow{P} \succ \overrightarrow{U\setminus\{u_i^0,\cdots,u_i^r\}\cup\{u^0_{n^i(j)}\}}$$

There are totally $n\times(k'+d) + n\times(r+1)$ pairs of agents, and $2n\times(k'+d)+2n\times(r+1)$ preference lists in the layer l. Finally, we set $k = |U|+|P|-k'$, then $\bar{k} = k'$ and $\bar{d} = (r+2)-(r+1) = 1$ for Reg or $\bar{d} = 2(r+2)-2(r+1) = 2$ for Pair, with $r+2$ (or $2(r+2)$) being the minimum Reg(or Pair)-score before deleting the agents. Clearly, the construction is doable in polynomial time. □

Theorem 4.10. *Even with $\beta = 1$, (k,β)-$\mathcal{M}$-$\mathcal{R}$ with $\mathcal{M}$= {LSum, LMax} is W[1]-hard with respect to $\bar{k}$ and $\bar{d}$ under Egal and Balc.*

Proof. Here, we only prove this theorem for $\mathcal{R}$ being Egal. $\mathcal{R}$ being Balc can be proved in a similar way. We give a reduction from Clique. Given a Clique instance $(G=(V,E),k')$ with $|V| = n$ and $|E| = m$, we construct an $(k,1)$-$\mathcal{M}$-$\mathcal{R}$ instance as follows. We create one pair of agents for each $v_i \in V$, that is, u^{v_i} and w^{v_i}. For each $e_i \in E$, we create two pairs of agents, $u_1^{e_i}, w_1^{e_i}, u_2^{e_i}, w_2^{e_i}$. Create two sets of auxillary agents P, Q with $|P| = |Q| = (d^* + 10k')$, where $d^* = 10(n-k') + 6\frac{k'(1+k')}{2} + 7(m - \frac{k'(1+k')}{2})$. There are totally $2n + 4m + 2(d^*+10k')$ agents.

Now we set the preference lists of the agents. Add for each $p_i \in P$, the preference list $\succ^l_{p_i}$: $q_i \succ \overrightarrow{Q\setminus\{q_i\}} \succ \overrightarrow{W}$, and for each $q_i \in Q$, the preference list $\succ^l_{q_i}$: $p_i \succ \overrightarrow{P\setminus\{p_i\}} \succ \overrightarrow{U}$ to the preference profile l, where $\overrightarrow{S}$ denotes an arbitrary but fixed ordering of a set S. For each $v_i \in V$, we add the following preference lists to l, where $Q_i = \{q_{8(i-1)+1},\cdots,q_{8i}\}$:

$$\succ^l_{u^{v_i}}:\overrightarrow{Q_i} \succ w^{v_i} \succ \overrightarrow{Q\setminus Q_i} \succ \overrightarrow{W\setminus\{w^{v_i}\}}$$
$$\succ^l_{w^{v_i}}:u^{v_i} \succ \overrightarrow{P} \succ \overrightarrow{U\setminus\{u^{v_i}\}}$$

For each edge $e_i = \{v_s, v_t\}$, we add the following preference lists to l.

$$\succ^l_{u_1^{e_i}}: w_1^{e_i} \succ w^{v_s} \succ w^{v_t} \succ w_2^{e_i} \succ \overrightarrow{Q} \succ \overrightarrow{W\setminus\{w_1^{e_i}, w_2^{e_i}, w^{v_s}, w^{v_t}\}}$$

[3]Note that each instance of LPareto-$\mathcal{R}$ has a Layer Pareto-optimal matching with respect to the respective scoring rules.

$\succ^{l}_{u_2^{e_i}}: w_1^{e_i} \succ q_{8n+i} \succ w_2^{e_i} \succ \overrightarrow{Q \setminus \{q_{8n+i}\}} \succ \overrightarrow{W \setminus \{w_1^{e_i}, w_2^{e_i}\}}$

$\succ^{l}_{w_1^{e_i}}: u_1^{e_i} \succ u_2^{e_i} \succ \overrightarrow{P} \succ \overrightarrow{U \setminus \{u_1^{e_i}, u_2^{e_i}\}}$

$\succ^{l}_{w_2^{e_i}}: u_1^{e_i} \succ u_2^{e_i} \succ \overrightarrow{P} \succ \overrightarrow{U \setminus \{u_1^{e_i}, u_2^{e_i}\}}$

Finally, let $k = |U| + |P| - k'$, then $\bar{k} = k'$, and let $d = 3d^* + 10k'$, then $\bar{d} = 10k' + \frac{k(1+k')}{2}$ with $d + \bar{d}$ being the minimum Egal-score of this instance before removing agents. □

In the following we turn to investigate the parameterized complexity of (n,t)-$\mathcal{M}$-$\mathcal{R}$, that is, selecting t out of β layers to form a new instance of LSum-$\mathcal{R}$ or LMax-$\mathcal{R}$ and search for a matching satisfying the $\mathcal{R}$-score. Under such a setting, (n,t)-LSum-$\mathcal{R}$ with $\mathcal{R}$ being Reg/Pair/Egal and (n,t)-LMax-$\mathcal{R}$ with $\mathcal{R}$ being Reg/Pair are FPT with respect to β. That is, by enumerating all subsets $L' \subseteq L$, we can reduce an (n,t)-$\mathcal{M}$-$\mathcal{R}$ instance to an equivalent LSum-$\mathcal{R}$ or LMax-$\mathcal{R}$ instance (U, W, L', d) and apply Theorem 4.2 or 4.3. The time of enumerating all subsets of L is within $O(2^{\beta})$.

Theorem 4.11. *(n,t)-$\mathcal{M}$-$\mathcal{R}$ with $\mathcal{M}$ ∈ {LSum, LMax} is W[1]-hard with respect to t under all four scoring rules, and para-NP-hard with respect to d or $\bar{d}$ under Reg and Pair.*

Proof. We give a reduction from Set packing. Given an universe V and a family C of subsets of V, and an integer k', Set packing seeks for a family $C' \subseteq C$ of k' pairwise disjoint sets. Set packing is W[1]-hard with respect to parameter k' [18].

Given a Set packing instance (V,C,k') with $|V| = n'$ and $|C| = m$, we construct an (n,t)-$\mathcal{M}$-$\mathcal{R}$ instance $(U \cup P, W \cup Q, L, d)$ as follows. For each $v_i \in V$, we create $2m$ pairs of agents, $u_i^j, \bar{u}_i^j \in U$ and $w_i^j, \bar{w}_i^j \in W$ with $1 \le j \le m$. We create two sets of auxillary agents P and Q with $|P| = |Q| = d^*$, with d^* being set as follows:

- [For LMax-$\mathcal{R}$] let $d^* = 2, 4, 3mn', 6mn'$ under Reg, Pair, Balc, and Egalitarian, respectively.
- [For LSum-$\mathcal{R}$] let $d^* = 2n, 4n, 3mn'^2, 6mn'^2$ under Reg, Pair, Balc, and Egalitarian, respectively.

Next, we create m layers, one for each subset $c_j \in C$. The preference lists of each $x \in P \cup Q$ are the same in all layers. For each $p_i \in P$, the preference list has the following form: $q_i \succ \overrightarrow{Q \setminus \{q_i\}} \succ \overrightarrow{W}$, with $\overrightarrow{S}$ denoting an arbitrary but fixed ordering of a set S. The preference lists of $q_i \in Q$ are set accordingly. For agents $u_i^j, \bar{u}_i^j \in U$ and $w_i^j, \bar{w}_i^j \in W$ with $1 \le i \le n'$ and $1 \le j \le m$, which are created for the same element v_i, we create $4m$ preference lists of the same form and add 4 lists to each layer. The preference lists of u_i^j and $\bar{u}_i^j$ have the following form: $w_i^j \succ \bar{w}_i^j \succ \overrightarrow{Q \setminus \{q_{d^*}\}} \succ q_{d^*} \succ \overrightarrow{W \setminus \{w_i^j, \bar{w}_i^j\}}$, where w_i^j and $\bar{w}_i^j$ are also created for $v_i \in V$. The preference lists of w_i^j and $\bar{w}_i^j$ are set analogously. Next, we make modifications according to the occurrence of elements in subsets. In each layer l_j, which is according to a subset c_j, we do the following modifications if v_i occurs in c_j.

- For $j = i$, exchange $\bar{u}_i^j$ with p_{d^*} in $\succ^{l_j}_{w_i^j}$.
- For $j \neq i$, exchange u_i^j with p_{d^*} in $\succ^{l_j}_{w_i^j}$.

Finally, set $d = d^*$ under Reg, Pair, and $d = 2d^*$ under Balc, Egalitarian. Set $t = k'$ under all four scoring rules. □

Theorem 4.12. *(n,t)-$\mathcal{M}$-$\mathcal{R}$ with $\mathcal{M}$ ∈ {LSum, LMax} is W[2]-hard with respect to $\bar{t}$ under all four scoring rules.*

5 CONCLUDING REMARKS

We introduce three models for position-based matching with multimodal preferences under four scoring rules. A collection of polynomial-time tractable and intractable results have been achieved: Under rules of Reg and Pair, all three models admit polynomial-time algorithms. Under rules of Balc and Egal, LSum-Egal is known to be polynomial-time solvable, while there is no polynomial-time algorithm for LMax-$\mathcal{R}$ and LPareto-$\mathcal{R}$ unless P=NP.

The classical complexity of one problem remains open, that is, LSum-Balc. We want to mention that this problem is not equivalent to Two-weighted maximum weighted matching (TMWM), which is the Maximum weighted matching problem with exactly two weights assigned to each edge. The target of TMWM is to find a matching, such that the sum of the first weights of all matching edges and the sum of the second weights of all matching edges are both at most d. The only difference between LSum-Balc and TNWM is that the weights of TMWM are allowed to be exponential in n, the number of vertices. Thus, we can prove TMWM is NP-hard by reducing the Partition problem to it, while the same method does not apply to LSum-Balc.

It might be interesting to examine the parameterized complexity of LMax-$\mathcal{R}$ and LPareto-$\mathcal{R}$ under Balc and Egal with respect to β, where β is the number of layers. Since we only focus on parameterized complexity, it might be interesting to examine the approximation complexity of our models.

We only investigate the position-based models. Actually, more models can be adapted to the corresponding version with multimodal preferences. Besides stable matching which has been studied in [13], other models such as popular matching [5, 15, 25] and Pareto-optimal matching [3, 4, 12] might be suitable candidates.

ACKNOWLEDGMENTS

We thank the AAMAS-22 reviewers for their constructive comments. All authors are supported by the National Natural Science Foundation of China (Grants No.61772314, 61761136017 and 62072275.).

REFERENCES

[1] Atila Abdulkadiroğlu, Parag A. Pathak, and Alvin E. Roth. 2005. The New York city high school match. *American Economic Review* 95, 2 (2005), 364–367.
[2] Atila Abdulkadiroğlu, Parag A. Pathak, Alvin E. Roth, and Tayfun Sönmez. 2005. The Boston public school match. *American Economic Review* 95, 2 (2005), 368–371.
[3] Atila Abdulkadiroğlu and Tayfun Sönmez. 1998. Random serial dictatorship and the core from random endowments in house allocation problems. *Econometrica* 66, 3 (1998), 689–701.
[4] David J. Abraham, Katarína Cechlárová, David F Manlove, and Kurt Mehlhorn. 2004. Pareto optimality in house allocation problems. In *Proceedings of the 15th International Symposium on Algorithms and Computation.* 3–15.
[5] David J. Abraham, Robert W Irving, Telikepalli Kavitha, and Kurt Mehlhorn. 2007. Popular matchings. *SIAM J. Comput.* 37, 4 (2007), 1030–1045.
[6] Haris Aziz, Péter Biró, Serge Gaspers, Ronald de Haan, Nicholas Mattei, and Baharak Rastegari. 2020. Stable matching with uncertain linear preferences. *Algorithmica* 82, 5 (2020), 1410–1433.

[7] Haris Aziz, Jiayin Chen, Serge Gaspers, and Zhaohong Sun. 2018. Stability and pareto optimality in refugee allocation matchings. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*. 964–972.

[8] Haris Aziz, Serge Gaspers, Zhaohong Sun, and Toby Walsh. 2019. From matching with diversity constraints to matching with regional quotas. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*. 377–385.

[9] Dorothea Baumeister and Jörg Rothe. 2016. Preference aggregation by voting. In *Economics and Computation, An Introduction to Algorithmic Game Theory, Computational Social Choice, and Fair Division*. Springer, 197–325.

[10] Walter Bossert and Yves Sprumont. 2014. Strategy-proof preference aggregation: Possibilities and characterizations. *Games and Economic Behavior* 85 (2014), 109–126.

[11] Marta Bottero, Valentina Ferretti, José Rui Figueira, Salvatore Greco, and Bernard Roy. 2018. On the Choquet multiple criteria preference aggregation model: Theoretical and practical insights from a real-world application. *European Journal of Operational Research* 271, 1 (2018), 120–140.

[12] Katarína Cechlárová, Pavlos Eirinakis, Tamás Fleiner, Dimitrios Magos, Ioannis Mourtos, and Eva Potpinková. 2014. Pareto optimality in many-to-many matching problems. *Discrete Optimization* 14 (2014), 160–169.

[13] Jiehua Chen, Rolf Niedermeier, and Piotr Skowron. 2018. Stable marriage with multi-modal preferences. In *Proceedings of the 19th ACM Conference on Economics and Computation*. 269–286.

[14] Frances Cooper and David F. Manlove. 2020. Algorithms for new types of fair stable matchings. In *Proceedings of 18th International Symposium on Experimental Algorithms*. 20:1–20:13.

[15] Ágnes Cseh. 2017. Popular matchings. *Trends in Computational Social Choice* 105, 3 (2017).

[16] Marek Cygan, Fedor Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. 2015. *Parameterized Algorithms*. Springer.

[17] Swapnil Dhamal and Y. Narahari. 2013. Scalable preference aggregation in social networks. In *Proceedings of the First AAAI Conference on Human Computation and Crowdsourcing*.

[18] Rodney Downey and Michael Fellows. 2012. *Parameterized Complexity*. Springer Science & Business Media.

[19] Tomás Feder. 1995. *Stable Networks and Product Graphs*. Stanford University Press.

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

[21] Sushmita Gupta, Sanjukta Roy, Saket Saurabh, and Meirav Zehavi. 2019. Balanced stable marriage: how close is close enough?. In *Proceedings of the 16th International Symposium on Algorithms and Data Structures*. 423–437.

[22] Robert W. Irving, Paul Leather, and Dan Gusfield. 1987. An efficient algorithm for the "optimal" stable marriage. *Journal of the ACM* 34, 3 (1987), 532–543.

[23] Pallavi Jain and Nimrod Talmon. 2020. Committee selection with multimodal preferences. In *Proceedings of the 24th European Conference on Artificial Intelligence*. 123–130.

[24] LeSheng Jin, Radko Mesiar, and Ronald R. Yager. 2020. Parameterized preference aggregation operators with improved adjustability. *International Journal of General Systems* 49, 8 (2020), 843–855.

[25] Naoyuki Kamiyama. 2017. Popular matchings with ties and matroid constraints. *SIAM Journal on Discrete Mathematics* 31, 3 (2017), 1801–1819.

[26] Harold W Kuhn. 1955. The Hungarian method for the assignment problem. *Naval research logistics quarterly* 2, 1-2 (1955), 83–97.

[27] Thomas Lukasiewicz and Enrico Malizia. 2019. Complexity results for preference aggregation over (m)CP-nets: Pareto and majority voting. *Artificial Intelligence* 272 (2019), 101–142.

[28] David F. Manlove and Gregg O'Malley. 2012. Paired and altruistic kidney donation in the UK: algorithms and experimentation. In *Proceedings of 11th International Symposium on Experimental Algorithms*. 271–282.

[29] Shuichi Miyazaki and Kazuya Okamoto. 2019. Jointly stable matchings. *Journal of Combinatorial Optimization* 38, 2 (2019), 646–665.

[30] James Munkres. 1957. Algorithms for the assignment and transportation problems. *J. Soc. Indust. Appl. Math.* 5, 1 (1957), 32–38.

[31] Rolf Niedermeier. 2006. *Invitation to Fixed-Parameter Algorithms*. Oxford University Press.

[32] Maria Silvia Pini, Francesca Rossi, Kristen Brent Venable, and Toby Walsh. 2011. Incompleteness and incomparability in preference aggregation: Complexity results. *Artif. Intell.* 175, 7-8 (2011), 1272–1289.

[33] Ariel D. Procaccia and Nisarg Shah. 2016. Optimal aggregation of uncertain preferences. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*. 608–614.

[34] Antoine Rolland. 2013. Reference-based preferences aggregation procedures in multi-criteria decision making. *European Journal of Operational Research* 225, 3 (2013), 479–486.

[35] Alvin E Roth, Tayfun Sönmez, and M. Utku Ünver. 2005. Pairwise kidney exchange. *Journal of Economic theory* 125, 2 (2005), 151–188.

[36] Alvin E Roth, Tayfun Sönmez, and M. Utku Ünver. 2007. Efficient kidney exchange: Coincidence of wants in markets with compatibility-based preferences. *American Economic Review* 97, 3 (2007), 828–851.

[37] Svetlana O. Smerchinskaya and Nina P. Yashina. 2021. Aggregation of preferences in attitude scales. *International Journal of Modeling, Simulation, and Scientific Computing* 12, 4 (2021), 2150026:1–2150026:15.