Satisfying neighbor preferences on a circle

Danny Krizanc[1], Manuel Lafond[2], Lata Narayanan[3], Jaroslav Opatrny[3], and Sunil Shende[4]

[1] Dept. of Math. and Computer Science, Wesleyan University, Middletown CT, USA, USA dkrizanc@wesleyan.edu
[2] Dept. of Mathematics and Statistics, University of Ottawa, Ottawa, Canada mlafond2@uOttawa.ca.com
[3] Dept. of Comp. Sc. and Soft. Eng., Concordia University, Montreal, QC, Canada lata@cs.concordia.ca, opatrny@cs.concordia.ca
[4] Department of Computer Science, Rutgers University, Camden, USA sunil.shende@rutgers.edu

Abstract. We study the problem of satisfying seating preferences on a circle. We assume we are given a collection of n agents to be arranged on a circle. Each agent is colored either blue or red, and there are exactly b blue agents and r red agents. The w-neighborhood of an agent A is the sequence of $2w+1$ agents at distance $\leq w$ from A in the clockwise circular ordering. Agents have preferences for the colors of other agents in their w-neighborhood. We consider three ways in which agents can express their preferences: each agent can specify (1) a preference list: the *sequence* of colors of agents in the neighborhood, (2) a preference type: the *exact* number of neighbors of its own color in its neighborhood, or (3) a preference threshold: the *minimum* number of agents of its own color in its neighborhood. Our main result is that satisfying seating preferences is fixed-parameter tractable (FPT) with respect to parameter w for preference types and thresholds, while it can be solved in $O(n)$ time for preference lists. For some cases of preference types and thresholds, we give $O(n)$ algorithms whose running time is independent of w.

Keywords: Seating arrangement, Linear algorithm, FPT algorithm

1 Introduction

Alice has invited a large group of people to a dinner party, just before a most contentious election between the Red and Blue political parties. The guests are to be seated at a large circular table. She has now started receiving urgent requests from her guests about the seating arrangements. Some guests want to be assured that they won't be seated close to anyone from the other party. Other guests are more tolerant; they would be happy so long as a majority of their neighbors belong to their own party. There are also argumentative guests who insist on being seated near people mostly from the other party; they want to have a chance to argue with their neighbors all evening. Can Alice satisfy all the guests' preferences? How can she figure out a seating arrangement acceptable to all her guests?

In this paper, we study the problem of satisfying seating preferences on a circle. We assume we are given a collection of n *agents* to be arranged on a circle. Each agent is colored either blue (B) or red (R), and there are exactly b blue agents and $r = n - b$ red agents in all. Any specific clockwise ordering of these n colored agents around the circle is called a *configuration* or n-*configuration*; we are interested in exploring questions about the existence of configurations that, for each agent, satisfy a given constraint on the colors of agents in its neighborhood. For a fixed value w, called *half-window size*, $1 \leq w \leq (n-1)/2$, we define the neighborhood of an agent as the w agents preceding it, the agent itself, and the w agents following it, in the clockwise ordering. The agents preceding and following it are called the *proper neighbors* of the agent.

Agents can express **preferences** for the colors of agents in their neighborhood in several possible ways. We consider three ways in which color specifications can be expressed. For every agent, we specify its:

Preference list: the exact sequence of colors desired in its neighborhood. Each preference list is a string over $\{B, R\}$ of length $2w + 1$.

Preference type: the desired *number* of proper neighbors with the *same color* as the agent. Furthermore, the number of red (resp. blue) nodes with preference type i is denoted r_i (resp. b_i).

Preference threshold: the *minimum number* of proper neighbors with the *same color* as the agent. Furthermore, the number of red (resp. blue) nodes with threshold i is denoted ρ_i (resp. β_i).

We assume that *all agents specify their preferences in the same manner*, e.g., they all specify (possibly different) preference thresholds. For any one of the three preference specification methods, a given configuration of agents is said to be **valid** if and only if all the agents have their preferences satisfied in their respective neighborhoods within the configuration. For example, the 10-configuration in Figure 1 specified by the string $RRBRRRBRRR$, is valid for the following preference specifications with half-window size $w = 2$:

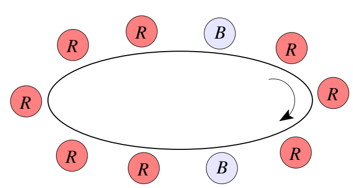

Fig. 1. An example of a seating configuration of agents

- Preference lists: Two blue agents with lists $RRBRR$; two red agents with lists $RBRRR$; two red agents with lists $RRRBR$; one red agent for lists $BRRRB$, $BRRRR$, $RRRRR$, and $RRRRB$.

- Preference types: Two blue agents with types 0, one red agent with type 2, six

red agents with type 3 and one red agent with type 4.
- Preference thresholds: Two blue agents with threshold 0; two red agents with threshold 2 and six red agents with threshold 3. Note that this configuration also satisfies many other preference thresholds, for example, two blue agents with threshold 0, and eight red agents with threshold 1.

We are interested in the following problem: *For a given preference specification with n agents with a given half-window size w, construct a valid n-configuration, or determine that it does not exist.*

In the sequel, we consider the above problem for each of the three kinds of preference specifications. We consider the cases when all nodes have the same preference (homogeneous preferences), when the nodes of the same color have the same preference (homogeneous within type), and the general (heterogeneous preference) case.

1.1 Related work

Our problem was largely motivated by the influential work of Schelling [19, 20] on how the preferences of individuals can potentially lead to the undesirable effect of global segregation. In the Schelling's model, we imagine people of two different types (Red and Blue in our example) who are placed on a line (or later on a grid). Each person would like at least a fraction τ of their size w neighborhood to be of their type. Consider a process whereby a randomly chosen pair of people (of different type) who are both unhappy in their current location are allowed to switch positions. Schelling showed via simulations that this would eventually lead to segregation by type even in the case where $\tau \leq \frac{1}{2}$. His observations led to a great deal of research in both the sociological and mathematics literature in attempts to verify his conclusions [2, 3, 8, 10, 16, 21, 22]. Recently, this analysis has been taken up in the theoretical computer science literature [4, 11]. In that work, the model is analyzed as a random process and it is shown that the expected size of the resulting segregated neighborhoods is polynomial in w on the line [4] and exponential in w on the grid [11] but that in both cases it is independent of the overall number of participants.

Our work may be thought of as a first attempt to analyze this model from a worst-case deterministic perspective rather than on average. Given the preferences of the individuals, we ask whether there is a configuration under which they may all be satisfied. As far as we can determine, we are the first to consider this question. However, as might be expected, related problems have been studied in the graph coloring literature. In particular, a *perfect 2-coloring* of a graph is a 2-coloring of the graph where nodes of a given color are required to have a given number of neighbors of the same color. Parshina [17] characterized the perfect 2-colorings of infinite circulant graphs with a continuous set of distances which corresponds to the infinite line with each node connected to all of its neighbors within a fixed distance. We rely upon this characterization in order to solve the version of our problem when all nodes of the same color have the same preference type (see Section 4.1). Somewhat less related but of a similar flavor are (m, k)-*defective colorings* whereby the nodes of a graph are colored by

m colors in such a way that each vertex is adjacent to at most k vertices of the same color. This concept was introduced in [1, 5] and continues to be studied [6, 7, 13]. Since we are interested in colorings with at least some number of vertices of the same color in each vertex's neighborhood, these results do not appear to be directly applicable to our problems.

1.2 Our results

We assume that the input for a seating arrangement problem is a sequence of preference specifications for all n agents, all of the same kind.

1. For preference lists, we give in Section 3 an $O(n)$ algorithm to construct a valid configuration.
2. In Section 4.1 we give an algorithm to construct a valid configuration for homogeneous preference types in $O(n)$ time.
3. For heterogeneous preference types and for preference thresholds we show in Section 4.2 and Section 5 that satisfying seating preferences is FPT for parameter w. In particular, for each of the two cases, we give an algorithm to construct a valid configuration that has time complexity $O(n + 2^{2^{2w+1}} w(w2^{2w+1} + \log n)2^{(2w+1)2.5\cdot 2^{2w+1}} \log n)$.

2 Notation

We use strings to represent configurations of agents, taken circularly. Following standard terminology, for any string u, we denote by $|u|$ the length of u and by u^k the string u repeated k times. We denote by u^+ the infinite set of strings consisting of *one or more repetitions* of the string u, and $\overline{u}$ denotes the *complementary* string obtained by replacing each color symbol in u by its complementary color, i.e., replacing B with R and vice-versa. An *n-configuration* is a specific clockwise ordering of n agents around the circle, i.e., a *cyclic string* $s[0]\ s[1]\ \dots\ s[n-1]$, denoted by $\mathbf{s[0:n]}$, where the i^{th} agent has color $s[i] \in \{B, R\}$. For $i < j$, let $s[i:\ j]$ denote the contiguous substring of agents $s[i]\ s[i+1]\ \dots\ s[j-1]$ in the configuration. Thus $s[i:\ i+1] = s[i]$. We follow the convention that sequence indices: (a) always start from reference position 0; (b) are interpreted modulo n; and (c) increase in *clockwise order* around the circle. Thus given a half-window size w, the neighborhood of agent $s[i]$ consists of $s[i-w:\ i]$, its *left* neighbors, $s[i]$ itself, and $s[i+1:\ i+w+1]$, its *right* neighbors. Since we assume that $n \geq 2w+1$, each agent has $2w$ distinct neighbors. A *maximal* contiguous sequence of agents of the same color is called a *run* of that color and is denoted as a string, e.g. B^k is a blue run of length k.

3 Preference lists

When given the preference lists of agents, the feasibility problem is solvable in polynomial time: this can be done by constructing an appropriate directed graph representation of the preferences, and finding an Euler tour in the graph.

The technique was originally proposed for DNA fragment assembly in [18]. Let $S = \{s_0, s_1, \ldots, s_{n-1}\}$ be the set of preference lists given by the n agents (recall that each s_i is a string of length $2w+1$ over the color alphabet $\{B, R\}$). Define S' to be the set of $2w$-length strings obtained by dropping either the *first* or the *last* symbol from a preference list in S. That is, $s' \in S'$ if and only if there exists a preference list $s \in S$ so that either $s' = s[0 : 2w]$ or $s' = s[1 : 2w+1]$.

We now construct a digraph $G = (V, E)$ as follows. We set $V = S'$, so that each vertex is uniquely labeled with a string in S'. Since the i^{th} agent has preference list s_i, we create a directed edge labeled i from the node labeled $s_i[0 : 2w]$ to the node labeled $s_i[1 : 2w+1]$. As an example, let $w = 2$, and suppose that the preference list for the i^{th} agent is $s_i = RRBBR$. Then i is the *edge-label* of an edge from the node labeled $RRBB$ to the node labeled $RBBR$. Notice that if there are two agents with the same preference list, then there will be two edges between the same two vertices in the graph G.

Suppose i and j are two consecutive edge-labels along a path in G. Then, notice that agent j can be placed to the right of agent i in a configuration. In the example above, the path

$$RRBB \xrightarrow{i} RBBR \xrightarrow{j} BBRR$$

indicates that agent j with preference list $s_j = RBBRR$ can follow agent i in a configuration. Clearly, there is an Euler tour in the graph G iff there is a valid n-configuration of the agents. Since the graph G has at most $2n$ nodes and exactly n edges, we get:

Theorem 1. *For a preference list specification, a valid configuration, if one exists, can be found in time $O(n)$.*

4 Preference types

4.1 Homogeneous preference types

We first consider the situation where all the blue agents have *homogeneous* preferences for exactly i blue neighbors each and likewise, all the red agents have homogeneous preferences for exactly j red neighbors each. Such configurations were studied by Parshina [17] where some characterizations were provided for *infinite* valid strings. In particular, the paper does not consider the values of n, r and b, the actual number of red and blue agents present.

In what follows, we will consider that w, i and j are fixed. A string s over alphabet $\{R, B\}$ is *valid* if each blue agent in s has i blue neighbors, and each red agent has j red neighbors.

The analysis of homogeneous preference types is divided into two parts. We first paraphrase the results from Parshina [17] that establish various conditions under which valid strings exists, regardless of the values of r and b. Then we use these results to decide, for given values of r and b, if a valid configuration exists.

The following result, stated without proof, is paraphrased from the paper by Parshina [17]. It characterizes the valid strings for a given w, i (the blue preference type) and j (the red preference type).

Lemma 1 ([17]). *Let s be a string over alphabet $\{R, B\}$. Then s is valid if and only if one of the following statements holds:*

1. $i + j = 2w - 2$ *and*
   - *both i and j are even and $s \in u^+$ for some string u of length $w + 1$ containing $1 + i/2$ Bs and $1 + j/2$ Rs, or*
   - *$i = j$ and $s \in (u\overline{u})^+$ for some string u of length w.*
2. *$i + j = 2w - 1$ and $s \in u^+$ for some string u of length $2w + 1$ containing $i + 1$ Bs and $j + 1$ Rs.*
3. $i + j = 2w$ *and*
   - *both i and j are even and $s \in u^+$ for some string u of length w containing $i/2$ Bs and $j/2$ Rs, or*
   - *$i = j$ and $s \in (u\overline{u})^+$ for some string of length $w + 1$.*

Lemma 1 immediately allows us to construct valid configurations for some values of r, b, n, but not for all values for which a valid configuration exists. In the remainder of this section, we find necessary and sufficient conditions for the existence of all valid configurations. To do this, we need to identify appropriate periods of the valid strings described in Lemma 1.

Characterizing the periods of valid strings. For a string s, we call p a *period* of s if $s \in p^+$. We say that s is *aperiodic* if the only period of s is itself. A period p of s is *minimal* if p itself is aperiodic. We start with the next simple result that allows us to search for "good" minimal periods of valid strings.

Proposition 1. *There exists a valid configuration $s[0 : n]$ for given r and b and $n = r + b$ if and only if there exists a valid string $s' \in p^+$ with minimal period p such that $n = k \cdot |p|$ for some integer k, and p has r/k red colors and b/k blue colors.*

Proof. If $s[0 : n]$ is valid for r and b, then $s' = s$ along with any minimal period p of s' must satisfy the statement. Conversely, it is clear that p^k as described is a valid configuration for r and b. □

For any integer k, we define $t(k)$ to be the *largest power of two that divides k* (with $t(k) = 1$ if k is odd). We first need the two following intermediate results.

Lemma 2. *Suppose that $i + j = 2w$. Then there exists a valid configuration $s \in (u\overline{u})^+$ such that $|u| = w + 1$ and $u\overline{u}$ has a period p if and only if $i = j$, $|p|$ divides $2w + 2$, $|p|$ is a multiple of $t(2w + 2)$ and $p = q\overline{q}$ for some string q.*

Proof. ($\Rightarrow$) The necessity of $i = j$ is due to Lemma 1(c). The fact that $|p|$ divides $2w+2$ is immediate, since $|u\overline{u}| = 2w+2$ and p is a period of $u\overline{u}$. Let k be the integer such that $u\overline{u} = p^k$. We claim that $|p|$ does not divide $w+1$. Assume on the contrary that $|p|$ also divides $w+1$, say $|p| = (w+1)/h$ for some integer h. Then $|p^h p^h| = 2w+2 = |p^k|$, implying $u\overline{u} = p^h p^h$, which is a contradiction since it implies $u = \overline{u}$.

Given the above claims, suppose now that $|p|$ is not a multiple of $t(2w+2)$. Then $t(|p|) < t(2w+2)$. As $|p^k| = 2w+2$, we have $t(k|p|) = t(2w+2)$, and so $t(k) > 1$ and k must be even, say $k = 2k'$. But $2k'|p| = 2w+2$ implies that $k'|p| = w+1$, a contradiction since $|p|$ does not divide $w+1$.

Finally to see that $p = q\overline{q}$ for some q, first note that this trivially holds if $p = u\overline{u}$. Otherwise, $|p| < 2w+2$. Write $p = p_1 p_2$, where $|p_1| = |p_2| = |p|/2$ (note that given the above, $|p|$ must be even). Since $(u\overline{u})^+ = (p_1 p_2)^k$, we either have $u = (p_1 p_2)^{k/2}$ or $u = (p_1 p_2)^{(k-1)/2} p_1$. The former is not possible, as it would imply $u = \overline{u}$. The latter implies $\overline{u} = p_2 (p_1 p_2)^{(k-1)/2}$, which in turn implies $p_1 = \overline{p_2}$.

($\Leftarrow$) Let k be such that $|p| = (2w+2)/k$. Because $|p|$ contains all the even prime factors of $2w+2$ and $|p|$ divides $2w+2$, k must be a product of odd factors and hence must be odd. Write $u = (q\overline{q})^{(k-1)/2} q$ and $\overline{u} = \overline{q}(q\overline{q})^{(k-1)/2}$. One can check that $u\overline{u} = p^k$ and, by Lemma 1, $s \in u^+$ is a valid configuration. $\square$

We now give the analogous statement for the case $i + j = 2w - 2$. Since its proof is essentially identical to the above, it is omitted.

Lemma 3. *Suppose that $i + j = 2w - 2$. Then there exists a valid configuration $s \in (u\overline{u})^+$ such that $|u| = w$ and $u\overline{u}$ has period p if and only if $i = j$, p divides $2w$, $|p|$ is a multiple of $t(2w)$ and $p = q\overline{q}$ for some string q.*

We are now ready to give the characterization of minimal periods of s.

Theorem 2. *There exists a valid configuration $s \in p^+$ with minimal period p if and only if one of the following conditions holds:*

1. *$i + j = 2w - 2$, $|p| = (w+1)/k$ for some integer k and p has $(i+2)/(2k)$ B's and $(j+2)/(2k)$ R's;*
2. *$i + j = 2w - 1$, $|p| = (2w+1)/k$ for some integer k and p has $(i+1)/k$ B's and $(j+1)/k$ R's;*
3. *$i + j = 2w$, $|p| = w/k$ for some integer k, and p has $i/(2k)$ B's and $j/(2k)$ R's;*
4. *$i = j = w$, $|p|$ divides $2w+2$, $|p|$ is a multiple of $t(2w+2)$ and $p = q\overline{q}$ for some string q.*
5. *$i = j = w - 1$, $|p|$ divides $2w$, $|p|$ is a multiple of $t(2w)$ and $p = q\overline{q}$ for some string q.*

Proof. Let us first show that any of the conditions of statement is sufficient. For (1) and (3), $s \in p^+$ is valid by Lemma 1 parts 1 and 3, respectively. For (2), $s \in p^+$ is valid by Lemma 1 part 2. Condition (4) is sufficient by Lemma 2 and (5) by Lemma 3.

We now show that one of these conditions must hold for $s \in p^+$ to be valid. Suppose first that $i + j = 2w - 2$. By Lemma 1, either $s \in u^+$ for some u with $|u| = w + 1$, or $s \in (u\overline{u})^+$ for some u with $|u| = w$. If $s \in u^+$ with $|u| = w + 1$, let p be the minimal period of u, with $u = p^k$. Then u must have $i/2 + 1$ B's and $j/2 + 1$ R's, from which it follows that (1) holds. If instead $s \in (u\overline{u})^+$ with $|u| = w$, then by Lemma 3, we have that (5) holds.

Suppose that $i + j = 2w$. Again, we handle the two possible cases prescribed by Lemma 1. If $s \in u^+$ with $|u| = w$, let p be the minimal period of u, with $u = p^k$. Then (3) must hold. If $s \in (u\overline{u})^+$ with $|u| = w + 1$, by Lemma 2, it is (4) that holds.

Finally suppose that $i + j = 2w - 1$. By Lemma 1, if $s \in u^+$ and p is a period of u, $u = p^k$, then p must have $(i + 1)/k$ blue characters and $(j + 1)/k$ red characters, and (2) holds. $\square$

The problem of constructing a configuration therefore amounts to building a string p such that $|p|$ divides $r + b$ and that satisfies one of the above conditions. Proposition 1 combined with Theorem 2 can be used to derive when such a string p can be constructed. It is not hard to check that each case enumerated in the following corollary corresponds to one of the cases of Theorem 2.

Corollary 1. *For given w, i, j, r and b, there exists a valid configuration if and only if one of the following conditions holds:*

1. *$i+j = 2w-2$, $w+1$ and $r+b$ have a common divisor d such that $(2w+2)/d$ divides both $i + 2$ and $j + 2$;*
2. *$i+j = 2w-1$, $2w+1$ and $r+b$ have a common divisor d such that $(2w+1)/d$ divides both $i + 1$ and $j + 1$;*
3. *$i + j = 2w$, w and $r + b$ have a common divisor d such that $2w/d$ divides both i and j;*
4. *$i = j = w$, $2w + 2$ and $r + b$ have a common divisor d such that d is a multiple of $t(2w + 2)$;*
5. *$i = j = w - 1$, $2w$ and $r + b$ have a common divisor d such that d is a multiple of $t(2w)$.*

Theorem 3. *If all red agents have preference type i and all blue agents have preference type j, a valid configuration, if one exists, can be constructed in $O(n)$ time.*

Proof. To determine the existence of a valid configuration, it suffices to find a divisor d that satisfies one of the cases in Corollary 1. As d must be a divisor of one of $\{w, w+1, 2w, 2w+1, 2w+2\}$, one can simply try every integer between 2 and $2w+2$ and verify whether it meets one of the above requirements. Assuming that division can be done in constant time, this procedure takes time at most $O(w)$. As for the construction problem, once a suitable d is found, is it easy to construct a minimal period p satisfying Theorem 2. The main bottleneck here is to construct the output, which can be done in time $O(n)$. $\square$

4.2 Heterogeneous preference types

We begin with a straightforward result about the case $w = 1$.

Theorem 4. *For half-window size $w = 1$, given the number of agents of each preference type, a valid configuration, if one exists, can be found in $\Theta(n)$ time.*

Proof. We will develop necessary and sufficient conditions for a valid configuration to exist for a given collection of agents with given preference types. In any configuration of agents, the agents must occur in an even number of *runs*, with each run consisting of some number of agents of the same color: we will refer to them as blue or red runs. If there are k runs, then starting from a fixed point at the boundary of two runs we can number the runs around the circle as S_0, S_1, $\ldots$, S_{k-1} with the odd-numbered runs being blue and the even numbered ones being red.

Consider a valid configuration. We note that that a blue run in the configuration consists of a single agent if and only if it is of type 0. Consider a blue run containing $k \geq 2$ agents: the $(k-2)$ agents in the interior of the run must all be of type 2 and the two agents at the run boundary are or type 1. Thus,

$$b_2 > 0 \implies b_1 \geq 2 \text{ and } r_2 > 0 \implies r_1 \geq 2 \tag{1}$$

and since the agents of type 1 occur in pairs of the same color, b_1 and r_1 are even and the following necessary conditions hold:

$$b_0 + \frac{b_1}{2} = \frac{K}{2} = r_0 + \frac{r_1}{2} \tag{2}$$

Together, these conditions are necessary to guarantee a valid configuration. In fact, they are also sufficient: given a collection of agents that satisfies the above constraints, we can construct a configuration of *interleaved* blue and red runs as follows. Starting at a reference point on the circle, the first b_0 blue runs (resp. r_0 red runs) consist of singleton blue agents of type 0 (resp. red agents of type 0). The first non-singleton blue (or red run) is special: it has two blue agents of type 1 (resp. red agents of type 1) at the ends of the run and *all* the blue agents of type 2 (resp. red agents of type 2) in its interior. Now, the remaining blue (or red) runs are each configured to have exactly two blue agents of type 1(resp. two red agents of type 1). □

Next we consider the case when agents have heterogeneous preference types for arbitrary window sizes. Recall that b_i (resp. r_i) is the number of blue (resp. red) agents that have preference i. The neighborhood of a blue agent with preference i must be a string of length $2w + 1$ with the $(w + 1)^{th}$ symbol being B and with exactly $i + 1$ occurrences of B (that is i occurrences other than the middle symbol). Similarly, the neighborhood of a red agent with preference i is a string of length $2w + 1$ with the $(w + 1)^{th}$ symbol being R and with exactly $i + 1$ occurrences of R.

Given a valid configuration $\mathcal{C}$, that is, one that satisfies all the agents' preferences, let S be the set of all substrings (interpreted circularly as usual) of $\mathcal{C}$

of length $2w+1$. Furthermore, let $P_i \subseteq S$ and $Q_i \subseteq S$ denote the set of strings denoting neighborhoods of blue and red agents (resp.) of preference type i in the given configuration. Finally, for every $y \in S$, let X_y be the number of occurrences of y in $\mathcal{C}$, that is, the number of agents in the configuration whose neighborhood is represented by the string y.

We now present a set of inequalities and equalities concerning the quantities X_y. First, by definition,

$$X_y > 0 \text{ for every } y \in S. \tag{3}$$

Next, observe that since $\mathcal{C}$ is a valid configuration, for every $i = 0, 1, \ldots 2w$ with $b_i \neq 0$,

$$\sum_{y \in P_i} X_y = b_i. \tag{4}$$

Similarly for every $i = 0, 1, \ldots 2w$ with $r_i \neq 0$

$$\sum_{y \in Q_i} X_y = r_i. \tag{5}$$

Let T be the set of all substrings of length $2w$ of $\mathcal{C}$ (substrings interpreted circularly). Equivalently, T is the set of the strings obtained by deleting either the first or the last symbol of a string in S. Consider $z \in T$. Since the string z can be preceded and succeeded by an R or a B, it follows that for each $z \in T$

$$X_{Rz} + X_{Bz} = X_{zR} + X_{zB}. \tag{6}$$

Note that if any of the strings Rz, Bz, zR, zB do not exist in S, we simply remove the corresponding X quantity from the above equality, thereby preserving (3).

We define a De Bruijn multi-digraph G_S with respect to the set S, as described below. For every string z in T, we have a corresponding node in G. For every string $y \in S$, we create X_y directed edges in G_S all labeled y, from the node s to the node t where s and t are the strings obtained by dropping the first and the last symbols of y respectively. Now observe that the out-degree of a node z is $X_{zR} + X_{zB}$ (if either zR or zB is not in S, simply replace the corresponding term by 0), and its in-degree is $X_{Rz} + X_{Bz}$. By Equations 6, for every node z, its indegree equals its outdegree. In fact, it is easy to see that the configuration $\mathcal{C}$ corresponds to an Euler tour in the graph G_S.

Thus we have shown the following lemma:

Lemma 4. *If $\mathcal{C}$ is a valid configuration, and S is the set of its substrings of length $2w+1$, then Equations 3, 4, 5, and 6 hold, and furthermore, $\mathcal{C}$ corresponds to an Euler tour in the associated De Bruijn multigraph.*

We now describe our algorithm to construct a valid configuration, if it exists. We repeat the following steps for every subset S of strings of length $2w+1$, until a valid configuration is found.

Step 1: Fix a subset S of strings of length $2w+1$. Set up and solve the ILP described by Equations 3, 4, 5, and 6, with respect to the set S.

Step 2: If the ILP has a feasible solution, construct the De Bruijn multi-digraph G_S, using the values of the variables X_y in the solution of the ILP.

Step 3: If the digraph G_S is connected, find an Euler tour in G_S, and build a configuration by traversing the Euler tour and appending the middle symbol of each arc's label.

To analyze the complexity of our algorithm, we use the following result of Lokshtanov[15] derived from Lenstra [14] with improvements by Kannan [12] and Frank and Tardos [9] (see [15], Theorem 2.8.2):

Theorem 5. *[15] A solution to an ILP may be found in time $O(p^{2.5p}L\log(N))$ where p is the number of variables in the ILP, L is a bound on the number of bits required to describe the ILP and N is the maximum of the absolute values any variable can take.*

We are ready to prove the main result of this section:

Theorem 6. *For agents with heterogeneous preference types, a valid configuration can be found in $O(n + 2^{2^{2w+1}} w(w2^{2w+1} + \log n)2^{(2w+1)2.5\cdot 2^{2w+1}} \log n)$ time, if one exists. Therefore the problem is FPT for parameter w.*

Proof. The correctness of the algorithm above can be seen as follows. Suppose for a particular subset S of strings, the corresponding ILP has a feasible solution, and the multi-digraph G_S is connected. Since Equations 6 are satisfied, for every node in the graph G_S, its indegree equals its outdegree. Therefore H_S admits an Euler tour; clearly the Euler tour corresponds to a valid configuration.

If for every set S, either the graph G_S is not connected, or there is no feasible solution to the ILP, then it follows from Lemma 4 that there is no feasible configuration. To see this, it is enough to observe that if $\mathcal{C}$ is a feasible configuration, then for S the set of all substrings of $\mathcal{C}$ of length $2w+1$, G_S is connected and the number of occurrences of each $y \in S$ in the configuration $\mathcal{C}$ form a feasible solution to the ILP.

To analyze the running time, observe that the main cost is in solving the ILP generated for each of the possible subsets of strings of length $2w+1$, i.e., solving $2^{2^{2w+1}}$ ILPs. The ILPs have at most 2^{2w+1} variables. They contain 2^{2w+1} equations of type 3 each requiring $O(w)$ bits to describe. There are $2(2w+1)$ equations of type 4 and 5 each requiring $O(w2^{2w+1} + \log n)$ bits and there are 2^{2w} equations of type 6 each requiring $O(w)$ bits. The total length of an ILP is then $O(w(w2^{2w+1} + \log n))$ and thus by Theorem 5, it can be decided in $O(w(w2^{2w+1} + \log n)2^{(2w+1)2.5\cdot 2^{2w+1}} \log n)$. Constructing the graph and deciding if it is connected takes only an additional $O(2^{2w})$ time so that the overall runtime is $O(2^{2^{2w+1}} w(w2^{2w+1} + \log n)2^{(2w+1)2.5\cdot 2^{2w+1}} \log n)$. $\square$

5 Preference Thresholds

Recall that β_i and ρ_i are the number of blue and red agents respectively with threshold i. We start with a simple result about the case when all agents have the same threshold.

Theorem 7. *Suppose all nodes have the same threshold t. Then there exists a valid configuration if and only if one of the following is true:*

1. $r = 0$ *or* $b = 0$.
2. $t \leq w$ *and* $r, b \geq t + 1$

Therefore, a valid configuration, if one exists, can be found in $O(n)$ time.

Proof. If $r = 0$ (resp. $b = 0$), then clearly any permutation of blue (resp. red) agents satisfies all their preferences. Therefore we assume $r, b > 0$ (there are both red and blue agents). Since each agent requires t or more neighbors of its own color, it is necessary that $r, b \geq t + 1$. Suppose $t > w$ and there is a valid configuration. There must be at least one run of each color. Consider a pair of adjacent agents A and B such that A is red and B is blue. Since A has at least t other red neighbors, at most one of which is not a neighbor of B, and A is an additional red neighbor of B, it follows that B has at least t red neighbors. But then B can have at most $2w - t < w < t$ blue neighbors, a contradiction. We conclude that $t \leq w$. It is easy to see that a single run each of red and blue agents provides a valid configuration. □

Next we consider the special case of heterogeneous thresholds with $w = 1$. This means only three different values of thresholds are possible: 0, 1, 2.

Theorem 8. *For $w = 1$, and heterogeneous preference threshold specifications, a valid configuration can be constructed, if one exists, in $\Theta(n)$ time.*

Proof. If either $b = 0$ or $r = 0$, clearly any permutation of all blue agents satisfies all their preferences. So assume $r, b > 0$. Then we must have at least one run of each color. The endpoints of each run must be agents of threshold either 0 or 1. That is, we need:

$$\rho_0 + \rho_1 \geq 2 \text{ and } \beta_0 + \beta_1 \geq 2$$

If this condition is satisfied, the following configuration is valid: create one run of each color, the endpoints of the runs are agents with threshold 0 or 1, the other agents are placed inside the runs in an arbitrary fashion. □

Finally, we consider the general case when agents have heterogeneous preference thresholds, for arbitrary values of w. Any blue agent with preference threshold $B_{\geq i}$ (resp. any red agent with preference threshold $R_{\geq i}$), for some $i : \; 0 \leq i \leq 2w,$ requires *at least* i blue (resp. red) agents in its neighborhood. Any such blue (resp. red) agent has its neighborhood constrained to be a string in P_k (respectively Q_k) where $i \leq k \leq 2w$ (recall that P_k and Q_k respectively denote all possible neighborhoods of blue and red agents containing exactly k

proper neighbors of their own color in a given configuration). This suggests modifying the ILP above by replacing the equalities in (4) with inequalities, one for each $0 \leq i \leq 2w$:

$$\sum_{y \in \cup_{i \leq k \leq 2w} P_k} X_y \geq \sum_{i \leq k \leq 2w} \beta_k \tag{4-i}$$

and replacing (5) likewise with:

$$\sum_{y \in \cup_{i \leq k \leq 2w} Q_k} X_y \geq \sum_{i \leq k \leq 2w} \rho_k \tag{5-i}$$

with Equations (4-0) and (5-0) above being satisfied *with equality*. It is straightforward to see that the complexity of solving the problem is exactly the same as for heterogeneous preference types, as it involves solving a different ILP with the same number of variables and constraints. The theorem below follows:

Theorem 9. *For agents with heterogeneous threshold types, a valid configuration can be found in $O(n + 2^{2^{2w+1}} w(w2^{2w+1} + \log n)2^{(2w+1)2.5 \cdot 2^{2w+1}} \log n)$ time, if one exists. Therefore it is FPT for parameter w.*

6 Discussion

We considered three ways of specifying seating preferences and for each of these three kinds of specifications, we gave algorithms to construct such an arrangement when possible. Our main result is that satisfying seating preferences is fixed-parameter tractable (FPT) with respect to half-window size parameter w for preference types and thresholds, while it can be solved in linear time for preference lists. We also gave linear time algorithms for some special cases. We remark that if the input is given simply as a set of distinct preference specifications, and the number of agents that desire them, the existence of a valid configuration can be determined in $O(w)$ time for homogeneous preference types and in $O(2^{2^{2w+1}} w(w2^{2w+1} + \log n)2^{(2w+1)2.5 \cdot 2^{2w+1}} \log n)$ time for heterogeneous preference types and thresholds.

The existence of a polynomial algorithm in both w and n for heterogeneous preference types and thresholds remains open, as does the construction of valid configurations for grids. It would be interesting to solve the general case when each agent independently decides the manner of expressing its preference. Another direction of interest would be to start with a given configuration, and *move* the agents in an efficient manner to final positions satisfying their preferences.

References

1. James A. Andrews and Michael S. Jacobson. On a generalization of chromatic number. *Congressus Numerantium*, 47:33–48, 1985.
2. Stephen Benard and Robb Willer. A wealth and status-based model of residential segregation. *Mathematical Sociology*, 31(2):149–174, 2007.

3. Itzhak Benenson, Erez Hatna, and Ehud Or. From Schelling to spatially explicit modeling of urban ethnic and economic residential dynamics. *Sociological Methods & Research*, 37(4):463–497, 2009.
4. Christina Brandt, Nicole Immorlica, Gautam Kamath, and Robert Kleinberg. An analysis of one-dimensional Schelling segregation. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, pages 789–804. ACM, 2012.
5. Lenore J. Cowen, Robert H. Cowen, and Douglas R. Woodall. Defective colorings of graphs in surfaces: partitions into subgraphs of bounded valency. *Journal of Graph Theory*, 10(2):187–195, 1986.
6. Lenore J. Cowen, Wayne Goddard, and C. Esther Jesurum. Coloring with defect. In *SODA*, pages 548–557, 1997.
7. Lenore J. Cowen, Wayne Goddard, and C. Esther Jesurum. Defective coloring revisited. *Journal of Graph Theory*, 24(3):205–219, 1997.
8. Luca Dall'Asta, Claudio Castellano, and Matteo Marsili. Statistical physics of the Schelling model of segregation. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(07):L07002, 2008.
9. András Frank and Éva Tardos. An application of simultaneous Diophantine approximation in combinatorial optimization. *Combinatorica*, 7(1):49–65, 1987.
10. Adam Douglas Henry, Pawel Pralat, and Cun-Quan Zhang. Emergence of segregation in evolving social networks. *Proceedings of the National Academy of Sciences*, 108(21):8605–8610, 2011.
11. Nicole Immorlica, Robert Kleinberg, Brendan Lucier, and Morteza Zadomighaddam. Exponential segregation in a two-dimensional schelling model with tolerant individuals. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 984–993. SIAM, 2017.
12. Ravi Kannan. Minkowski's convex body theorem and integer programming. *Mathematics of operations research*, 12(3):415–440, 1987.
13. Fabian Kuhn. Weak graph colorings: distributed algorithms and applications. In *Proceedings of the twenty-first annual symposium on Parallelism in algorithms and architectures*, pages 138–144. ACM, 2009.
14. Hendrik W. Lenstra Jr. Integer programming with a fixed number of variables. *Mathematics of operations research*, 8(4):538–548, 1983.
15. Daniel Lokshtanov. New methods in parameterized algorithms and complexity. *Ph.D. Thesis, University of Bergen, Norway*, 2009.
16. Romans Pancs and Nicolaas J. Vriend. Schelling's spatial proximity model of segregation revisited. *Journal of Public Economics*, 91(1):1–24, 2007.
17. Olga G. Parshina. Perfect 2-colorings of infinite circulant graphs with continuous set of distances. *J. Applied and Industrial Math.*, 8(3):357–361, 2014.
18. Pavel A. Pevzner, Haixu Tang, and Michael S. Waterman. An Eulerian path approach to DNA fragment assembly. *Proceedings of the National Academy of Sciences*, 98(17):9748–9753, 2001.
19. Thomas C. Schelling. Models of segregation. *The American Economic Review*, 59(2):488–493, 1969.
20. Thomas C. Schelling. Dynamic models of segregation. *Journal of mathematical sociology*, 1(2):143–186, 1971.
21. H. Peyton Young. *Individual strategy and social structure: An evolutionary theory of institutions*. Princeton University Press, 2001.
22. Junfu Zhang. A dynamic model of residential segregation. *Journal of Mathematical Sociology*, 28(3):147–170, 2004.