

Counting Timetables with Overlapping Performer Memberships and Fairness in Performance Order

Formulation, Exact Counting Methods, and Preliminary Observations on Membership
Structure

Daichi Satoh

from Sansu-Sugaku no Atelier
(Atelier of Arithmetic and Mathematics)

October 24, 2026

Version 1; unrefereed research note
English translation of the Japanese version

Abstract

We formulate the problem of counting timetables satisfying constraints on performance order and break placement when some performers belong to more than one performing group. Membership is represented by a binary matrix. The model simultaneously incorporates prescribed first and last groups, breaks after blocks of three to five groups, and a prohibition on consecutive appearances by the same person without an intervening break. We further define a fairness measure based on each person’s cumulative deviation in performance positions, and give exact recurrences for counting timetables that first minimize the number of breaks and then minimize the maximum deviation. We establish elementary propositions concerning constraint graphs, performance-block decompositions, inclusion–exclusion, membership-pattern compression, and interactions between constraint relaxations. Finally, we present reproducible checks using fully synthetic data. The principal components of this note are based on existing combinatorics and graph theory; establishing mathematical novelty remains a task for future investigation.

Keywords: timetables; exact counting; overlapping memberships; combinatorics; graph theory; dynamic programming; fairness; break placement.

Contents

1	Introduction	4
1.1	Motivation	4
1.2	Questions addressed in this note	4
1.3	Related work and the scope of this note	4
1.4	Development of the problem setting	5
1.5	Organization	5
2	A basic model using sets and a membership matrix	5
2.1	Variable group sizes	6
3	Timetables and breaks	7

3.1	The possible number of breaks	7
3.2	A necessary condition from an individual's appearance count	8
4	A general counting formula	9
4.1	The minimum number of breaks	10
5	Fairness in individuals' performance positions	10
5.1	A single measure of fairness	10
5.2	Integer-scaled deviations and state compression	10
5.3	A simple lower bound on fairness	11
5.4	Lexicographic optimization	11
6	Exact computation using subset states	11
6.1	Counting at a fixed deviation threshold	11
6.2	Transitions for fixed break positions	12
6.3	States handling all length patterns simultaneously	12
6.4	Computational complexity	13
7	Decomposition into performance blocks	13
7.1	Counting orders within a block	13
7.2	Symmetry of internal blocks	14
7.3	Compression to unordered set partitions	14
7.4	A block recurrence incorporating fairness	15
8	Inclusion–exclusion and the structure of forbidden edges	15
8.1	Events associated with forbidden edges	15
8.2	Counting forced adjacencies from linear-forest components	16
8.3	Eliminating zero terms and computational limitations	16
9	Compression of membership patterns	16
9.1	People with identical membership patterns	16
9.2	Containment of membership sets and redundancy	16
9.3	A minimum constraint representation preserving forbidden pairs	17
10	Effects and interactions of constraint relaxations	17
10.1	The effect of one person's constraint	17
10.2	Relaxing constraints for several people	17
10.3	A generating function for individual violations	18
11	Counting formulas for special membership structures	18
11.1	Only one person has multiple memberships	18
11.2	Several people have multiple memberships	19
12	Checks using small computations	19
12.1	Validation strategy	19

12.2	An eight-group inclusion–exclusion check	20
12.3	An eight-group fairness check with explicit endpoints	20
12.4	A synthetic example with unequal group sizes	20
12.5	Why operational data are not published	21
13	Constraint structure and timetable counts	21
13.1	Why membership counts are not enough	21
13.2	Constraint impact	22
13.3	Directions for reducing computation	22
13.4	The role of individual sensitivity analysis	22
14	Limitations and future work	22
14.1	What has been completed	22
14.2	What remains incomplete	23
14.3	Open questions	23
14.4	Public data and privacy	23
14.5	Presenting the results	23
15	Conclusion	24
A	Notation	24
B	A component formula for forced adjacencies	25
C	Procedure for exact counting	25
D	Scope of publication as a research note	26

1 Introduction

1.1 Motivation

At an event featuring several performing groups, a performer may belong to more than one group. Drawing lots for the performance order, or arranging the groups randomly, may then cause the same person to perform consecutively without a break. Adjusting the order is particularly difficult when there are several dozen groups and some people belong to five or more of them.

Consecutive appearances are not the only issue in practice. It may also be undesirable for one person's appearances to be concentrated near the beginning of a long event. The first and last groups are often prescribed, and breaks may have to be inserted at regular intervals. This research began with the question of how many orders satisfy all these conditions, rather than merely how to find one such order.

1.2 Questions addressed in this note

Suppose the following information is given:

1. The number of performing groups and the set of members of each group.
2. The groups that perform first and last.
3. The minimum and maximum number of groups in each performance block separated by breaks.
4. A rule prohibiting consecutive appearances by the same person without an intervening break.
5. A fairness measure for assessing bias in performance positions.

We seek the total number of timetables with the minimum feasible number of breaks and, among those timetables, optimal fairness. Groups of four to seven people are the main practical motivation, but most of the theory is stated for more general finite set systems.

1.3 Related work and the scope of this note

Automated timetabling is related to combinatorial optimization, constraint satisfaction, and search algorithms. See Schaerf's overview [5] and the survey of educational timetabling by Ceschia et al. [6]. Their settings and constraints do not coincide exactly with ours, but they illustrate the diversity of timetabling problems and the importance of validation data.

A closer analogue to performers appearing in multiple pieces is the *talent scheduling problem* in film production. Cheng, Diamond, and Lin [1] considered talent holding costs associated with scene order, and Garcia de la Banda, Stuckey, and Chu [2] studied a dynamic programming solution. For music rehearsals, Sakulsom and Tharmmaphornphilas [3] investigated scheduling with musicians' assignments, piece lengths, and waiting times. The classical dynamic programming study by Adelson, Norman, and Laporte [4] is also related.

Our concern is the exact number of timetables that lexicographically minimize the break count and maximum appearance deviation under fixed endpoints, block-length bounds, and restrictions on consecutive appearances. It is not simply to find one order minimizing waiting costs. This distinction does not imply that the problem is wholly separate from existing optimization problems. The novelty of the counting formulation, and the completeness of our coverage of the closest related work, remain unsettled.

Counting Hamiltonian paths in a graph of allowable adjacencies is a classical problem. In view of methods such as Bax’s inclusion–exclusion algorithms [7, 8], the inclusion–exclusion formulas and block decompositions here should be regarded as specializations of known frameworks to the present setting. We do not claim that these theorems are the first of their kind; we organize the material as a research note that can be checked.

1.4 Development of the problem setting

The setting was developed step by step from difficulties in arranging an actual concert. Initially, we considered a multi-day event and sought to avoid concentrating a particular person’s appearances on the first day. To clarify the mathematical structure, we first restricted attention to a one-day event. We then considered break intervals for about thirty groups and adopted blocks of three to five consecutive groups, rather than treating performance durations as independent variables.

We subsequently clarified that groups need not all have six members but may have four to seven; that a person’s number of memberships may vary, roughly from one to seven; that the first and last groups should be fixed; and that a person may appear in consecutive groups when a break separates them. These are distinct conditions. Confusing them changes the objects being counted.

Finally, we adopted the priority of first minimizing the number of breaks and then reducing the deviation of the person whose appearances are most unevenly distributed. Multiple days, varying performance durations, equipment changeover times, preferred performance positions, and individual recovery times are not part of the standard problem in this note. They are left as possible extensions.

1.5 Organization

Section 2 introduces the membership matrix, Section 3 the break constraints, Section 4 the general counting formula, and Section 5 fairness. Sections 6 and 7 give dynamic programming and block decompositions, Section 8 inclusion–exclusion, Section 9 membership-pattern compression, and Section 10 interactions between constraint relaxations. Section 11 gives formulas for special structures. Section 12 validates the methods on synthetic examples suitable for publication, and Section 13 discusses the relationship between structure and counts. Section 14 describes limitations and future work.

2 A basic model using sets and a membership matrix

Definition 2.1 (People and groups). Let $P = \{1, \dots, m\}$ be the set of people and $V = \{1, \dots, n\}$ the set of performing groups. Define the membership matrix by

$$X = (x_{pi}) \in \{0, 1\}^{m \times n}, \quad x_{pi} = \begin{cases} 1 & \text{if person } p \text{ belongs to group } i, \\ 0 & \text{otherwise.} \end{cases}$$

Definition 2.2 (Group size and membership count). Write the size of group i and the number of groups to which person p belongs as

$$\ell_i = \sum_{p=1}^m x_{pi}, \quad d_p = \sum_{i=1}^n x_{pi},$$

respectively. Practical applications often assume $4 \leq \ell_i \leq 7$, but there is no requirement that all groups have six members. The values d_p may vary from person to person.

Proposition 2.3 (Double counting of person–group memberships).

$$\sum_{i=1}^n \ell_i = \sum_{p=1}^m d_p.$$

Proof. Both sides equal $\sum_{i=1}^n \sum_{p=1}^m x_{pi}$. □

This identity provides a cross-check between the distribution of group sizes and the distribution of individual membership counts. For example, thirty groups of six members give 180 person–group memberships, although there need not be 180 distinct people.

Definition 2.4 (Membership sets). Define the groups containing person p and the members of group i by

$$A_p = \{i \in V : x_{pi} = 1\}, \quad B_i = \{p \in P : x_{pi} = 1\}.$$

Thus $d_p = |A_p|$ and $\ell_i = |B_i|$.

Definition 2.5 (Overlap and adjacency compatibility). For distinct groups i, j , define their number of shared members by

$$w_{ij} = \sum_{p=1}^m x_{pi}x_{pj} = |B_i \cap B_j|.$$

Define their compatibility for consecutive performance by

$$c_{ij} = \mathbf{1}(w_{ij} = 0) \quad (i \neq j).$$

Thus $c_{ij} = 1$ if the groups have no member in common, and $c_{ij} = 0$ otherwise.

Definition 2.6 (Overlap and compatibility graphs). The *overlap graph* $G_X = (V, E_X)$ has edge set

$$E_X = \{\{i, j\} : 1 \leq i < j \leq n, w_{ij} > 0\}.$$

The *compatibility graph* H_X has an edge between distinct i, j precisely when $c_{ij} = 1$. It is the complement of G_X .

Remark 2.7. The overlap graph G_X determines the prohibition on consecutive appearances. In general, however, calculating fairness also requires each person’s membership set A_p . These two descriptions therefore have different theoretical roles.

Example 2.8 (Membership counts alone are insufficient). Suppose two people each belong to three groups. If they belong to the same three groups, the structure of forbidden pairs differs from the case in which their triples are different. Consequently, n , m , and the distribution of membership counts do not, in general, uniquely determine the timetable count.

2.1 Variable group sizes

When the group sizes ℓ_i vary between four and seven, the membership-matrix formulation needs no modification. The existence of a common member, rather than the value of ℓ_i itself, directly determines adjacency compatibility.

Proposition 2.9 (Adding an exclusive member preserves compatibility). *If a new person is added to one group and belongs to no other group, the overlap graph G_X does not change. If fairness is evaluated only for people belonging to at least two groups, the fairness function defined in this note also does not change.*

Proof. The new person belongs to only one group and hence creates no new overlap between distinct groups. The person is also excluded from the fairness evaluation. Both assertions follow. $\square$

Remark 2.10. Any imposed upper bound on group size must still hold after the addition. If fairness is defined over all people, the second assertion requires separate consideration.

3 Timetables and breaks

Definition 3.1 (Performance order). Represent the order of all groups by a permutation

$$\pi = (\pi_1, \pi_2, \dots, \pi_n) \in S_n.$$

When the opening group a and closing group z are prescribed, assume $a \neq z$ and require $\pi_1 = a$ and $\pi_n = z$.

Definition 3.2 (Break positions). Let $D \subseteq \{1, \dots, n-1\}$ be the set of break positions. A value $j \in D$ means that a break is inserted between the groups in positions j and $j+1$. The number of breaks is $k = |D|$. A person may perform in both groups adjacent to a break.

Definition 3.3 (Performance blocks). A *performance block* is a consecutive sequence of groups containing no break, obtained by splitting the performance order at the breaks. With k breaks there are $b = k + 1$ blocks. Write their lengths as $\lambda = (\lambda_1, \dots, \lambda_b)$.

The standard conditions in this note are

$$3 \leq \lambda_j \leq 5 \quad (1 \leq j \leq b), \quad \sum_{j=1}^b \lambda_j = n.$$

Performance durations themselves are not variables. Slots of approximately five to ten minutes motivate the setting, but the mathematical model imposes no duration constraints.

Definition 3.4 (No consecutive appearances without a break). A timetable (π, D) satisfies the prohibition on consecutive appearances if

$$\prod_{\substack{1 \leq j < n \\ j \notin D}} c_{\pi_j, \pi_{j+1}} = 1.$$

No condition on $c_{\pi_j, \pi_{j+1}}$ is imposed at a boundary $j \in D$.

3.1 The possible number of breaks

Proposition 3.5 (Break-count range from block lengths). *If every block has length between three and five, the number of breaks satisfies*

$$\left\lceil \frac{n}{5} \right\rceil - 1 \leq k \leq \left\lfloor \frac{n}{3} \right\rfloor - 1.$$

Conversely, for every integer k satisfying this inequality, at least one block-length sequence can be constructed.

Proof. For $b = k + 1$ blocks, it is necessary that $3b \leq n \leq 5b$. Conversely, when these inequalities hold, start each block at length three and distribute the remaining $n - 3b \leq 2b$ groups with at most two additional groups per block. $\square$

Example 3.6 (Thirty groups). For thirty groups, the block-length constraints allow five to nine breaks. With five breaks, every block has length five. Overlap constraints may nevertheless make a timetable with five breaks infeasible.

Example 3.7 (Thirty-one groups). With thirty-one groups and six breaks, the seven block lengths must lie between three and five. Their possible multisets are

$$(4^4, 5^3), \quad (3, 4^2, 5^4), \quad (3^2, 5^5),$$

where an exponent denotes the multiplicity of a length. Thus the number of break placements is

$$\binom{7}{3} + \frac{7!}{1!2!4!} + \binom{7}{2} = 35 + 105 + 21 = 161.$$

Only if one additionally requires any two block lengths to differ by at most one do just the 35 permutations of $(4^4, 5^3)$ remain.

Proposition 3.8 (Generating function for break placements). *The number of ways to place k breaks so that each block has length between three and five is*

$$R(n, k) = [x^n](x^3 + x^4 + x^5)^{k+1},$$

where $[x^n]$ denotes the coefficient of x^n .

Proof. Choosing a block length corresponds to choosing one term of $x^3 + x^4 + x^5$. The coefficient counts selections for $k + 1$ blocks whose lengths sum to n . $\square$

Remark 3.9 (Two interpretations of evenly spaced breaks). Requiring breaks after every three to five groups differs from requiring any two block lengths to differ by at most one. We use the former as standard. The latter can be imposed by restricting the allowed length sequences. In that case, put $q = \lfloor n/(k+1) \rfloor$ and $r = n \bmod (k+1)$. There are

$$\binom{k+1}{r}$$

length sequences, provided all lengths that occur are allowed: $3 \leq q \leq 5$ when $r = 0$, and $3 \leq q \leq 4$ when $r > 0$.¹

3.2 A necessary condition from an individual's appearance count

Proposition 3.10 (A break-count lower bound from one person). *If person p belongs to d_p groups, avoiding consecutive appearances by that person without a break requires*

$$k \geq \max\{0, 2d_p - n - 1\}.$$

Proof. There are $d_p - 1$ intervals between the d_p appearances of person p . Each interval must contain at least one group not involving p , or a break. There are only $n - d_p$ such groups and k breaks, so $d_p - 1 \leq (n - d_p) + k$ is necessary. The case $d_p = 0$ gives only the trivial bound $k \geq 0$. $\square$

Remark 3.11. This condition is necessary, not sufficient. When several people's memberships interact, the inequalities may hold individually while no globally feasible timetable exists.

¹The condition $q + 1 \leq 5$ is needed only when a block of length $q + 1$ actually occurs.

4 A general counting formula

Definition 4.1 (Length patterns). Let

$$\mathcal{L}_{n,k} = \left\{ (\lambda_1, \dots, \lambda_{k+1}) \in \{3, 4, 5\}^{k+1} : \sum_{j=1}^{k+1} \lambda_j = n \right\}.$$

A length pattern λ uniquely determines a break set $D(\lambda)$ through its proper partial sums.

Definition 4.2 (Feasible timetables). Define the set of feasible timetables with k breaks by

$$\begin{aligned} \mathcal{T}_k(X; a, z) = \{ (\pi, D(\lambda)) : \pi \in S_n, \pi_1 = a, \pi_n = z, \lambda \in \mathcal{L}_{n,k}, \\ c_{\pi_j, \pi_{j+1}} = 1 \text{ for every } j \notin D(\lambda), 1 \leq j < n \}. \end{aligned}$$

Write $T_k(X; a, z) = |\mathcal{T}_k(X; a, z)|$.

Definition 4.3 (Counting with a fixed length pattern). For a fixed $\lambda \in \mathcal{L}_{n,k}$, put

$$\mathcal{T}_\lambda(X; a, z) = \{ (\pi, D(\lambda)) \in \mathcal{T}_k(X; a, z) \}, \quad T_\lambda(X; a, z) = |\mathcal{T}_\lambda(X; a, z)|.$$

Thus $T_k(X; a, z) = \sum_{\lambda \in \mathcal{L}_{n,k}} T_\lambda(X; a, z)$. When indices or endpoints are omitted, the membership structure and fixed endpoints are understood from the context.

Theorem 4.4 (General counting formula). *For any membership matrix X and distinct fixed endpoints a, z ,*

$$T_k(X; a, z) = \sum_{\lambda \in \mathcal{L}_{n,k}} \sum_{\substack{\pi \in S_n \\ \pi_1 = a, \pi_n = z}} \prod_{\substack{1 \leq j < n \\ j \notin D(\lambda)}} c_{\pi_j, \pi_{j+1}}.$$

Proof. The product is one precisely when every adjacency without a break is allowed, and zero otherwise. Each pair (π, D) therefore contributes exactly one if it satisfies the conditions. $\square$

Remark 4.5. This formula gives an exact specification, but directly evaluating it requires enumerating permutations and is extremely expensive. It does not imply the existence of a closed-form evaluation. Moreover, m, n , and the values d_p alone do not generally determine the count; the matrix X , or equivalent relational information, is needed.

Corollary 4.6 (No overlapping memberships). *If distinct groups have no members in common, then*

$$T_k(X; a, z) = (n-2)! [x^n] (x^3 + x^4 + x^5)^{k+1}.$$

Proof. Every c_{ij} equals one, so all $(n-2)!$ endpoint-fixed permutations are allowed for each length pattern. Apply Proposition 3.8. $\square$

Corollary 4.7 (General upper bound). *For every membership matrix,*

$$0 \leq T_k(X; a, z) \leq (n-2)! R(n, k).$$

Proof. The right-hand side counts timetables after removing all overlap prohibitions. Adding constraints cannot increase their number. $\square$

4.1 The minimum number of breaks

Definition 4.8 (Minimum break count). If there is a k with $\mathcal{T}_k(X; a, z) \neq \emptyset$, define

$$k^*(X; a, z) = \min\{k : T_k(X; a, z) > 0\}.$$

If no feasible k exists, k^* is undefined.

Minimizing the break count does not mean that additional breaks are always acceptable. With block lengths restricted to three through five, feasibility for one value of k need not imply feasibility for a larger value, because the partition may violate the length constraints. Feasibility should therefore be checked in increasing order of k .

5 Fairness in individuals' performance positions

5.1 A single measure of fairness

We use cumulative deviations in appearance counts to measure uneven distributions, such as a person's appearances being concentrated early with none late in the event. This is a modeling choice, not a complete representation of physical burden or the difficulty of moving equipment.

Definition 5.1 (Cumulative appearance count). For an order π , let

$$s_p(j; \pi) = \sum_{h=1}^j x_{p, \pi_h} \quad (0 \leq j \leq n).$$

This counts person p 's appearances among the first j groups. In particular, $s_p(0; \pi) = 0$ and $s_p(n; \pi) = d_p$.

Definition 5.2 (Individual cumulative deviation). Take jd_p/n as the ideal cumulative count when person p 's appearances are evenly distributed throughout the event. Define

$$E_p(\pi) = \max_{0 \leq j \leq n} \left| s_p(j; \pi) - \frac{jd_p}{n} \right|.$$

Definition 5.3 (Maximum fairness deviation). Let $P_2 = \{p : d_p \geq 2\}$ be the people appearing in at least two groups. When $P_2 \neq \emptyset$, define

$$F_X(\pi) = \max_{p \in P_2} E_p(\pi).$$

When $P_2 = \emptyset$, set $F_X(\pi) = 0$.

Remark 5.4. The position of a person appearing in only one group is not optimized here. This definition makes the practical priority explicit; it is not a uniquely universal notion of fairness. A different model including all people can be defined if needed.

5.2 Integer-scaled deviations and state compression

Definition 5.5 (Deviation of a completed set). For $S \subseteq V$, define $c_p(S) = \sum_{i \in S} x_{pi}$ and

$$\Delta_X(S) = \max_{p \in P_2} |nc_p(S) - |S|d_p|.$$

Set this value to zero if $P_2 = \emptyset$.

Proposition 5.6 (The current deviation depends only on the completed set). *For any partial timetable whose set of completed groups is S , the integer-scaled cumulative deviation at that point is $\Delta_X(S)$. In particular, this value does not depend on the order of the groups within S .*

Proof. After $j = |S|$ groups, the appearance count is $c_p(S)$, which is unchanged by rearranging the groups within S . Substitute into the definition. $\square$

Thus the quantity to be optimized can be written as

$$nF_X(\pi) = \max_{0 \leq j \leq n} \Delta_X(\{\pi_1, \dots, \pi_j\}).$$

Removing the denominator n allows an implementation using integer arithmetic only.

5.3 A simple lower bound on fairness

Proposition 5.7 (Deviation lower bound from fixed endpoints). *If the opening group is a and the closing group is z , every permitted order π satisfies*

$$nF_X(\pi) \geq \max_{p \in P_2} \max\{|nx_{pa} - d_p|, |nx_{pz} - d_p|\}.$$

When P_2 is empty, the right-hand side is interpreted as zero.

Proof. At $j = 1$, we have $s_p(1; \pi) = x_{pa}$, so the integer-scaled deviation is $|nx_{pa} - d_p|$. At $j = n - 1$, we have $s_p(n - 1; \pi) = d_p - x_{pz}$, giving deviation $|d_p - nx_{pz}|$. Neither value can exceed the maximum over all positions. $\square$

Remark 5.8. Proposition 5.7 can be evaluated without examining any intermediate part of the order. It provides a basic measure for investigating the effect of prescribed opening and closing groups on optimal fairness.

5.4 Lexicographic optimization

Definition 5.9 (Optimal fairness and number of optimal solutions). When k^* exists, define

$$F^*(X; a, z) = \min_{(\pi, D) \in \mathcal{T}_{k^*}(X; a, z)} F_X(\pi)$$

and

$$N^*(X; a, z) = |\{(\pi, D) \in \mathcal{T}_{k^*}(X; a, z) : F_X(\pi) = F^*\}|.$$

The optimization has two stages: first minimize the number of breaks, and then minimize the maximum deviation subject to that break count. Only timetables satisfying both optima contribute to N^* . The count T_{k^*} of all feasible timetables with the minimum number of breaks must be distinguished from the count N^* of those that also achieve optimal fairness.

6 Exact computation using subset states

6.1 Counting at a fixed deviation threshold

Let $Q \in \mathbb{Z}_{\geq 0}$ be a threshold for the integer-scaled fairness deviation. For break count k and threshold Q , define

$$H_X(k, Q) = \#\{(\pi, D) \in \mathcal{T}_k(X; a, z) : nF_X(\pi) \leq Q\}.$$

Write $H_X(k, \infty) = T_k(X; a, z)$.

Proposition 6.1 (Threshold characterization of optimal counts). *If a feasible timetable exists, then*

$$\begin{aligned} k^* &= \min\{k : H_X(k, \infty) > 0\}, \\ Q^* &= \min\{Q \in \mathbb{Z}_{\geq 0} : H_X(k^*, Q) > 0\}, \\ N^* &= H_X(k^*, Q^*), \quad F^* = Q^*/n. \end{aligned}$$

Proof. The first equality is the definition of k^* . Since $nF_X(\pi)$ is an integer, it has a minimum feasible value Q^* . Every timetable with deviation at most Q^* attains exactly that optimum. The remaining equalities follow. $\square$

6.2 Transitions for fixed break positions

Fix a break set D . For $S \subseteq V$, $i \in S$, and $\ell \in \{1, \dots, 5\}$, let $D_Q(S, i, \ell)$ count partial timetables in which the completed groups are S , the last group is i , the current block length is ℓ , and every prefix set U encountered so far satisfies $\Delta_X(U) \leq Q$.

The initial condition for a fixed opening group is

$$D_Q(\{a\}, a, 1) = \mathbf{1}(\Delta_X(\{a\}) \leq Q).$$

When adding a group $j \notin S$ next:

- If $|S| \notin D$, add it without a break only if $c_{ij} = 1$.
- If $|S| \in D$, insert a break and add it regardless of c_{ij} .
- Always require $\Delta_X(S \cup \{j\}) \leq Q$ after the addition.
- Add the closing group z only in the final position.

Thus, with $\ell' = \ell + 1$ when there is no break and $\ell' = 1$ when there is a break, update

$$D_Q(S \cup \{j\}, j, \ell') += D_Q(S, i, \ell).$$

Here $+=$ denotes adding the right-hand side to the value already stored on the left. Given D , the allowed block length at each position is determined. Sum the completed states, then sum over all length patterns to obtain $H_X(k, Q)$.

6.3 States handling all length patterns simultaneously

If break positions are not fixed, add the current number of breaks h to the state, giving $D_Q(S, i, \ell, h)$. The initial state is

$$D_Q(\{a\}, a, 1, 0) = \mathbf{1}(\Delta_X(\{a\}) \leq Q),$$

with all other states initially zero. For $j \notin S$, the transition without a break is allowed when $\ell < 5$, $c_{ij} = 1$, and the deviation condition holds after the addition:

$$D_Q(S \cup \{j\}, j, \ell + 1, h) += D_Q(S, i, \ell, h).$$

A transition with a break is allowed when $3 \leq \ell \leq 5$, $h < k$, and the same post-addition deviation condition holds:

$$D_Q(S \cup \{j\}, j, 1, h + 1) += D_Q(S, i, \ell, h).$$

After all groups have been placed, require $h = k$ and $3 \leq \ell \leq 5$ for the last block. The closing group z must not be used before the final position.

Theorem 6.2 (Correctness of the fixed-threshold DP). *The sum of the final-state counts obtained from the initialization, transitions, and terminal conditions above equals $H_X(k, Q)$; explicitly,*

$$H_X(k, Q) = \sum_{\ell=3}^5 D_Q(V, z, \ell, k).$$

Proof. Every feasible timetable has a unique construction sequence obtained by adding one group at a time from the beginning. The transitions check all restrictions on consecutive appearances, block lengths, fixed endpoints, and cumulative deviations, so no invalid construction sequence is counted. Conversely, every prefix of a timetable satisfying the conditions corresponds to one of the stated transitions. Each feasible timetable is therefore counted exactly once. $\square$

Remark 6.3 (The danger of retaining only the best prefix value). For each state (S, i, ℓ, h) , retaining only paths with the smallest maximum deviation so far can be useful for computing the optimal value, but may undercount optimal solutions. A path with a larger intermediate deviation can ultimately have the same maximum deviation as another path because of a common later deviation. Exact counting therefore fixes Q and counts all paths satisfying that threshold.

6.4 Computational complexity

There are 2^n possible subsets S . Including the last group i , block length, and break count makes the complexity grow exponentially in general. Precomputing $\Delta_X(S)$ uses the individual membership sets. For about 24 groups, an implementation using bit sets may be considered; for thirty or more, further work on state reduction, search order, symmetry, and memory management is needed.

Remark 6.4 (Integer sizes). Timetable counts can grow on a factorial scale. They may exceed the range of 32-bit or 64-bit integers, so implementations must use arbitrary-precision integers or another type, such as 128-bit integers, only where its range is demonstrably sufficient. The small synthetic examples fit in 64-bit integers, but sufficient capacity must be guaranteed for general inputs.

7 Decomposition into performance blocks

7.1 Counting orders within a block

Definition 7.1 (Block order counts). For $S \subseteq V$ with $3 \leq |S| \leq 5$, let $h_X(S)$ be the number of orders of all groups in S , each appearing once without a break, such that every adjacent pair is compatible. Let $h_{X,a}(S)$ count such orders starting with a , and $h_{X,z}^{\text{end}}(S)$ those ending with z . For inappropriate sets, define these quantities to be zero.

Since $|S| \leq 5$, one can compute $h_X(S)$ by checking at most $5! = 120$ permutations. Choosing which groups belong to the same block remains a large problem.

Theorem 7.2 (Block decomposition formula). *Fix $k \geq 1$. Summing over ordered partitions $(S_1, \dots, S_{k+1})$ of V into nonempty sets with $3 \leq |S_j| \leq 5$ gives*

$$T_k(X; a, z) = \sum_{\substack{S_1 \sqcup \dots \sqcup S_{k+1} = V \\ 3 \leq |S_j| \leq 5}} h_{X,a}(S_1) \left(\prod_{j=2}^k h_X(S_j) \right) h_{X,z}^{\text{end}}(S_{k+1}).$$

Proof. A person may appear in both groups bordering a break. Thus, for a fixed block partition, the allowed orders within the blocks can be chosen independently. Impose the fixed endpoint conditions only on the first and last blocks, and sum over all ordered partitions. $\square$

Remark 7.3. Different orders of the group sets assigned to the blocks count as different timetables. Counting unordered block partitions therefore requires an appropriate permutation factor.

7.2 Symmetry of internal blocks

The quantity T_k is summed over all allowed length sequences, rather than being restricted to one λ . Permuting internal blocks generally also permutes their lengths.

Theorem 7.4 (Divisibility from internal-block symmetry). *Let $k \geq 1$, with no fairness restriction. The number of feasible timetables with exactly k breaks satisfies*

$$(k-1)! \mid T_k(X; a, z).$$

Proof. The first block contains a and the last contains z . Permuting the $k-1$ intervening blocks as whole blocks preserves within-block compatibility and the length bounds. The blocks have disjoint, nonempty group sets, so distinct permutations of the internal blocks yield distinct timetables. The feasible timetables therefore split into orbits of size $(k-1)!$. $\square$

Remark 7.5. This assertion concerns T_k , without fairness restrictions. Since F_X refers to performance positions, exchanging internal blocks may change its value. It does not follow in general that $(k-1)!$ divides the optimal count N^* .

7.3 Compression to unordered set partitions

For $U \subseteq V$ and $j \in \mathbb{Z}_{\geq 0}$, let $P_j(U)$ be the sum of $\prod h_X(S)$ over partitions of U into j unordered blocks of sizes three to five. Set $P_0(\emptyset) = 1$ and $P_0(U) = 0$ when $U \neq \emptyset$.

Proposition 7.6 (Recurrence for unordered blocks). *For $U \neq \emptyset$, $j \geq 1$, and $u = \min U$,*

$$P_j(U) = \sum_{\substack{S \subseteq U, u \in S \\ 3 \leq |S| \leq 5}} h_X(S) P_{j-1}(U \setminus S).$$

Proof. Each partition has a unique block S containing u . Removing that block leaves a partition into $j-1$ unordered blocks. This correspondence is a bijection. $\square$

Corollary 7.7 (Counting via unordered partitions). *For $k \geq 1$,*

$$T_k = (k-1)! \sum_{\substack{A, C \subseteq V, A \cap C = \emptyset \\ a \in A, z \in C \\ 3 \leq |A|, |C| \leq 5}} h_{X,a}(A) h_{X,z}^{\text{end}}(C) P_{k-1}(V \setminus (A \cup C)).$$

Proof. After selecting the first block A and last block C , partition the remaining groups into unordered internal blocks. There are $(k-1)!$ orders of those blocks, and no compatibility condition is imposed across a break. $\square$

7.4 A block recurrence incorporating fairness

Fix a fairness threshold Q . Let U be the set of groups already completed, and let S , disjoint from U , be the set in the next block. Define $h_{X,Q}(U, S)$ as the number of orders of S that respect the prohibition on consecutive appearances and satisfy $\Delta_X(U \cup W) \leq Q$ for every prefix set W of that order within S . Impose the prescribed endpoint conditions for the first and last blocks as well.

Proposition 7.8 (Block recurrence with fairness). *For fixed block count b and threshold Q , initialize $D_0(\emptyset) = 1$ and update*

$$D_{j+1}(U \cup S) += D_j(U)h_{X,Q}(U, S).$$

Then $D_b(V)$ is the total number of timetables satisfying the corresponding conditions. Fix the endpoints in the first and last blocks, and restrict each block length to three through five.

Proof. Every timetable has a unique block decomposition and a unique order within each block. Since U records all groups already completed, all information required for subsequent fairness checks is available. The updates correspond bijectively to complete construction sequences, so each timetable is counted exactly once. $\square$

8 Inclusion–exclusion and the structure of forbidden edges

8.1 Events associated with forbidden edges

For each edge $e = \{i, j\} \in E_X$, let A_e be the event that i, j are adjacent without a break. Let Ω_λ be all candidates satisfying the fixed endpoints and block-length sequence λ .

Theorem 8.1 (Counting by inclusion–exclusion). *For $F \subseteq E_X$, let $A_\lambda(F)$ count candidates in which every edge $e \in F$ occurs as an adjacency without a break. Then*

$$T_\lambda(X; a, z) = \sum_{F \subseteq E_X} (-1)^{|F|} A_\lambda(F).$$

Proof. This is the inclusion–exclusion principle for finite sets, counting candidates in which none of the forbidden events A_e occurs. $\square$

Lemma 8.2 (A linear-forest condition for nonzero terms). *If $A_\lambda(F) > 0$, every vertex of (V, F) has degree at most two and the graph is acyclic. Each nontrivial connected component is a path on at most five vertices.*

Proof. In a performance order, a group can be adjacent only to its predecessor and successor. Furthermore, all adjacency edges without breaks form a union of paths, one within each block. A subset of those edges cannot branch or contain a cycle, and a connected component cannot exceed the maximum block length of five. $\square$

Remark 8.3. A linear forest with all components of size at most five need not give $A_\lambda(F) > 0$. The opening or closing group may lie in the interior of a path, or the component sizes may not fit the prescribed block lengths. Lemma 8.2 gives only a necessary condition.

8.2 Counting forced adjacencies from linear-forest components

Suppose F is a linear forest, treating isolated vertices as one-vertex components. Let its component sizes be $s_1, \dots, s_c$. If a, z lie in the same component and there is at least one break, then $A_\lambda(F) = 0$. Each of a, z must also be an endpoint of its path component.

When these conditions hold, assign components to the blocks separated by breaks so that their total sizes equal the prescribed λ_j . Assign the component containing a to the first block, and that containing z to the last. Every nontrivial component not fixed by a global endpoint has two orientations; the orientations of the components containing fixed endpoints are determined. Components can then be ordered as contiguous units within each block.

This procedure constructs orders in which all forced edges are adjacent. Thus $A_\lambda(F)$ can be decomposed into a finite sum over component assignments and orientations. Appendix B gives the explicit coefficients.

8.3 Eliminating zero terms and computational limitations

The maximum block length of five restricts the nonzero terms in the inclusion–exclusion formula. However, when there are many forbidden edges, there can still be many subsets forming linear forests. Inclusion–exclusion is therefore not necessarily faster than dynamic programming. Determining the membership structures for which it is advantageous is a topic for further work.

9 Compression of membership patterns

9.1 People with identical membership patterns

Definition 9.1 (Equivalence of membership patterns). Define $p \sim q$ by $A_p = A_q$. Thus people belonging to exactly the same groups form an equivalence class.

Proposition 9.2 (Duplicate membership patterns do not change the counting conditions). *If $A_p = A_q$, removing either person’s membership information from the compatibility test leaves G_X unchanged. The maximum-based fairness measure F_X can also be evaluated after removing duplicate copies of the same membership pattern.*

Proof. Whenever p is a common member of a pair of groups, so is q , so the presence or absence of a forbidden edge is unchanged. For every order and position, their cumulative appearance counts and membership counts agree, hence their deviations agree. Including the same value twice in a maximum does not change it. $\square$

9.2 Containment of membership sets and redundancy

Proposition 9.3 (Redundancy of prohibitions under containment). *If $A_p \subseteq A_q$, the consecutive-appearance prohibition for p is contained in that for q .*

Proof. Every pair $\{i, j\} \subseteq A_p$ involving p is also contained in A_q . The pair therefore remains forbidden under q ’s constraint. $\square$

Remark 9.4. Proposition 9.3 allows omission of constraint information for adjacency checks. It does not generally allow removal of person p ’s membership data from fairness evaluation. Compression of prohibitions and compression of fairness information must be distinguished.

9.3 A minimum constraint representation preserving forbidden pairs

Let the pairs forbidden by person p 's memberships be

$$E_p^{\text{all}} = \{\{i, j\} : i, j \in A_p, i < j\}.$$

Then $E_X = \bigcup_p E_p^{\text{all}}$. A person set $Q \subseteq P$ preserves all prohibitions precisely when

$$\bigcup_{p \in Q} E_p^{\text{all}} = E_X.$$

We may define the size of a minimum constraint representation by

$$\rho(X) = \min \left\{ |Q| : Q \subseteq P, \bigcup_{p \in Q} E_p^{\text{all}} = E_X \right\}.$$

This is a set-cover problem on the forbidden edge set.

Remark 9.5. The quantity $\rho(X)$ is a candidate measure of how much the adjacency constraints can be compressed, but it does not alone determine the timetable count. Giving it a new name does not establish a new invariant of independent interest; its relationship to existing set-cover quantities and its research value require investigation.

10 Effects and interactions of constraint relaxations

10.1 The effect of one person's constraint

Define

$$U_p = \{\{i, j\} : B_i \cap B_j = \{p\}, i < j\}, \quad u_p = |U_p|.$$

Thus u_p counts group pairs that become compatible only after relaxing person p 's consecutive-appearance constraint.

Proposition 10.1 (No uniquely forbidden edges). *If $u_p = 0$, relaxing only person p 's consecutive-appearance constraint does not change $T_k(X; a, z)$.*

Proof. The equality $u_p = 0$ means every group pair forbidden because of p also has a shared member other than p . Removing p 's constraint therefore leaves the compatibility matrix unchanged. $\square$

Remark 10.2. The converse fails in general. Even when $u_p > 0$, the count does not change if fixed endpoints or break conditions prevent every feasible order from using the newly permitted pairs.

10.2 Relaxing constraints for several people

Let $R \subseteq P$ be the people allowed to appear consecutively. Keep the prohibitions for everyone outside R . With k and all other conditions fixed, write $T_k(R)$ for the resulting count.

Definition 10.3 (Set of violating people). For a candidate timetable τ , let $V(\tau) \subseteq P$ be the set of all people belonging to both groups of at least one adjacent pair without a break.

Theorem 10.4 (Decomposition of relaxed counts). *Let $b_k(U)$ count candidate timetables with $V(\tau) = U$. Then, for $R \subseteq P$,*

$$T_k(R) = \sum_{U \subseteq R} b_k(U).$$

Proof. Allowing consecutive appearances only for people in R admits exactly the candidates whose violating-person set is a subset of R . Partition these candidates by $V(\tau)$. $\square$

Corollary 10.5 (Exact interactions via Möbius inversion).

$$b_k(U) = \sum_{R \subseteq U} (-1)^{|U|-|R|} T_k(R).$$

Proof. Apply Möbius inversion on the subset lattice of a finite set. $\square$

Theorem 10.6 (Supermodularity of constraint relaxation). *For $R \subseteq P$ and distinct $p, q \notin R$,*

$$\begin{aligned} T_k(R \cup \{p, q\}) - T_k(R \cup \{p\}) \\ - T_k(R \cup \{q\}) + T_k(R) \geq 0. \end{aligned}$$

Proof. By Theorem 10.4, the left-hand side counts candidates satisfying $V(\tau) \subseteq R \cup \{p, q\}$ and $p, q \in V(\tau)$. It is therefore a nonnegative integer. $\square$

Remark 10.7 (Optimality conditions must be held fixed). This supermodularity statement requires a fixed break count and, if imposed, a fixed fairness threshold. If k^* or F^* is reoptimized after each relaxation, the resulting counts need not satisfy the same difference inequality.

10.3 A generating function for individual violations

Definition 10.8 (Violation generating function). Let $v_p(\tau)$ be the number of adjacent positions at which person p appears consecutively without a break in timetable τ . For a candidate set Ω , define

$$\Phi_X(z_1, \dots, z_m) = \sum_{\tau \in \Omega} \prod_{p=1}^m z_p^{v_p(\tau)}.$$

Setting every $z_p = 0$ gives the number of candidates with no violations, using the usual polynomial-evaluation convention $0^0 = 1$. Setting $z_p = 1$ only for selected people and zero for the others counts candidates in which only the selected people's violations are allowed.

Proposition 10.9 (Degree of the individual-violation polynomial).

$$\deg_{z_p} \Phi_X \leq \max\{0, d_p - 1\}.$$

Proof. Person p appears in d_p groups. Arranged in one sequence, these appearances can form at most $d_p - 1$ adjacent pairs without a break. If $d_p = 0$, there are no such pairs. $\square$

Remark 10.10. The absence of high-degree terms does not imply that the whole representation is small. Both the number of variables and the number of nonzero monomials may grow as the number of people with multiple memberships increases.

11 Counting formulas for special membership structures

11.1 Only one person has multiple memberships

Suppose one person p belongs to r groups and every other person belongs to at most one group. Assume also that the prescribed opening and closing groups a, z do not contain p .

Definition 11.1 (Polynomial for nonadjacent selections). Define

$$P_s(x) = \sum_{j=0}^{\lceil s/2 \rceil} \binom{s-j+1}{j} x^j.$$

The coefficient of x^j counts ways to select j pairwise nonadjacent positions from s consecutive positions.

Theorem 11.2 (Formula for one person with multiple memberships). *Under the preceding assumptions, if there is at least one break and the block-length sequence is $\lambda = (\lambda_1, \dots, \lambda_b)$, then*

$$T_\lambda = r!(n-r-2)! [x^r] \left(P_{\lambda_1-1}(x) \prod_{j=2}^{b-1} P_{\lambda_j}(x) P_{\lambda_b-1}(x) \right).$$

Proof. Because the first and last groups are fixed and do not contain p , the first and last blocks have $\lambda_1 - 1$ and $\lambda_b - 1$ free positions, respectively. An internal block has λ_j free positions. The polynomial P_s counts choices of positions in each block such that groups containing p are never adjacent within that block. The coefficient of x^r in the product counts choices of r positions overall. The r distinct groups containing p can be assigned to them in $r!$ ways; the remaining $n - r - 2$ groups can be assigned to the other free positions in $(n - r - 2)!$ ways. $\square$

Example 11.3 (Ten groups, one person in four groups). Assume the opening and closing groups do not contain the person with multiple memberships, and fix the block lengths to $(5, 5)$. Then

$$P_4(x) = 1 + 4x + 3x^2, \quad [x^4]P_4(x)^2 = 9.$$

Consequently,

$$T_{(5,5)} = 4! 4! \times 9 = 5184.$$

11.2 Several people have multiple memberships

Index the people with multiple memberships by $1, \dots, h$. Assign each group i a type $S_i \subseteq \{1, \dots, h\}$ consisting of the people from this list who belong to it. Groups i, j can be consecutive without a break precisely when $S_i \cap S_j = \emptyset$.

There are at most 2^h membership types. One can construct a transfer matrix with types as states and use a multivariate generating function to count sequences respecting within-block compatibility. When h is large, however, using all 2^h types becomes difficult, so it is preferable to store only types that actually occur.

Remark 11.4. Even if the membership sets of different people are disjoint, their individual placement counts cannot simply be multiplied, because groups containing different people compete for the same performance positions. A generating function or multivariate recurrence must count membership types jointly.

12 Checks using small computations

12.1 Validation strategy

Counting implementations can suffer from integer overflow, incorrectly implemented endpoint conditions, double counting of break placements, or incorrect handling of deviation thresholds at intermediate stages. For small numbers of groups, we therefore compare (i) direct enumeration of all permutations, (ii) subset-state dynamic programming, and (iii) block decomposition where possible.

12.2 An eight-group inclusion–exclusion check

Consider eight groups, with B01 opening and B08 closing. Using group indices one through eight, set the nine forbidden pairs to

$$E_X = \{\{1, 2\}, \{1, 3\}, \{2, 3\}, \{3, 4\}, \{4, 5\}, \\ \{5, 6\}, \{6, 7\}, \{7, 8\}, \{2, 6\}\}.$$

A synthetic membership system realizing exactly these pairs is obtained by introducing one distinct person for each pair, belonging only to those two groups. Fill each group to six members using additional exclusive members. This construction creates no other forbidden pairs. With exactly one break, the counts for the possible length patterns are as follows. This example checks

Table 1: Direct enumeration and inclusion–exclusion for eight groups

Block lengths	Direct enumeration	Inclusion–exclusion
(4, 4)	50	50
(3, 5)	54	54
(5, 3)	59	59
Total	163	163

the operation of the inclusion–exclusion formula on a concrete forbidden graph; the numerical values do not themselves prove a general theorem. Since the forbidden pairs, endpoints, and length patterns are all explicit, the table can be checked by enumerating permutations directly.

12.3 An eight-group fairness check with explicit endpoints

In a different synthetic eight-group example, let membership types A, B, and C have three, three, and two groups, respectively. All groups of a given type share four people who belong only to that type. Add two exclusive members to each group, with no shared people between types. Every group then has six members and there are 28 distinct people.

Label the groups B01–B08 and assign the types A,A,A,B,B,B,C,C in that order. Fix B01 (type A) as the opening group and B08 (type C) as the closing group.

With the length pattern fixed to (4, 4), the total number satisfying the consecutive-appearance prohibition is 108, the optimal integer-scaled fairness deviation is $Q^* = 6$, and the number of optimal solutions is 36. In the standard problem allowing all three patterns (3, 5), (4, 4), and (5, 3), the one-break counts are

$$T_1 = 348, \quad Q^* = 6, \quad N^* = 156.$$

The fairness-optimal counts for the three patterns are 48, 36, and 72, respectively. The values 108 and 36 apply only to the fixed pattern (4, 4), not to the total over all allowed break placements. The membership structure is determined, up to relabeling of people, by four shared members per type and two exclusive members per group, and can be reconstructed from this description.

12.4 A synthetic example with unequal group sizes

As another fully synthetic example, define eight groups and 35 people by Table 2. All identifiers were created for this example and do not correspond to a real event. The group sizes are

$$(6, 5, 7, 6, 4, 6, 7, 5).$$

Table 2: Fully synthetic memberships for eight groups and 35 people

Group	Members (all identifiers are fictional)
B01	P01, P02, P03, P04, P05, P06
B02	P07, P08, P09, P10, P11
B03	P01, P12, P13, P14, P15, P16, P17
B04	P02, P07, P18, P19, P20, P21
B05	P01, P08, P22, P23
B06	P03, P12, P24, P25, P26, P27
B07	P09, P22, P28, P29, P30, P31, P32
B08	P02, P13, P33, P34, P35

Fix B01 first and B08 last, and restrict block lengths to three through five. Include only people belonging to at least two groups in the maximum fairness deviation. Exact counting gives

$$k^* = 1, \quad T_1 = 59, \quad Q^* = 8, \quad N^* = 24.$$

The optimal fairness value is $F^* = 8/8 = 1$. The optimal counts for $(3, 5)$, $(4, 4)$, and $(5, 3)$ are 7, 9, and 8, respectively.

All examples in this section use entirely fictional memberships. Their inputs are specified within the text as forbidden pairs, type-based membership rules, or the complete membership table. For the small examples, direct enumeration and subset-state counting were compared and found to agree.

12.5 Why operational data are not published

The actual performance roster motivating the research contains information about individuals' memberships and numbers of overlapping commitments. Replacing names with numbers may still allow people or events to be inferred by matching group counts, membership distributions, or event dates. Accordingly, this public version does not include the operational membership distribution, individual constraint-impact ratios, detailed aggregates that might enable re-identification, or the full incidence matrix. Internal preliminary analyses of real cases are distinguished from the reproducible numerical examples published here.

13 Constraint structure and timetable counts

13.1 Why membership counts are not enough

If person p belongs to d_p groups, that person can forbid $\binom{d_p}{2}$ group pairs. Pairs forbidden by different people may overlap, so

$$|E_X| \leq \sum_{p=1}^m \binom{d_p}{2}.$$

The right-hand side depends only on the distribution of membership counts, while the left-hand side reflects overlaps between forbidden pairs.

Proposition 13.1 (Upper bound on forbidden pairs).

$$|E_X| \leq \min \left\{ \binom{n}{2}, \sum_{p=1}^m \binom{d_p}{2} \right\}.$$

Proof. The edge set E_X is the union of the pair sets induced by individuals' memberships. The size of a union is at most the sum of their sizes, and also at most the number of all group pairs. $\square$

13.2 Constraint impact

Assume $T_k(\emptyset) > 0$. Define the multiplicative effect of relaxing only person p 's consecutive-appearance constraint by

$$I_p^{(k)} = \frac{T_k(\{p\})}{T_k(\emptyset)}.$$

This value is at least one, but is not determined by d_p alone. Knowing u_p alone is also insufficient, since the impact depends on the global feasible structure.

Understanding this structural dependence requires attention not only to the number of forbidden edges but also to their arrangement, graph components, degree sequences, cliques, independent sets, relations to the fixed endpoints, and decomposability into break-separated blocks.

13.3 Directions for reducing computation

The preceding methods suggest three directions:

1. **Precomputation for small blocks.** Compute the valid internal-order counts $h_X(S)$ for all group subsets of sizes three through five.
2. **Membership-pattern compression.** Consolidate identical membership sets and redundant forbidden edges, while separately retaining the information required for fairness.
3. **Graph-structural decomposition.** For special forbidden-edge structures, simplify the counting formulas or derive faster algorithms.

The essential issue is to demonstrate a reduction in the actual number of states, running time, memory, or theoretical complexity, rather than merely to rewrite a counting formula.

13.4 The role of individual sensitivity analysis

The ratio $I_p^{(k)}$ compares counts before and after relaxing a particular person's constraint. Neither d_p nor the number u_p of uniquely forbidden pairs generally determines the global shape of the feasible space; the pair (d_p, u_p) need not determine $I_p^{(k)}$ either.

Preliminary individual analyses using operational data are excluded from this public version because consent and re-identification risk have not been adequately assessed. Future work should reproduce differences in impact ratios between people with the same (d_p, u_p) using public synthetic examples, and independently check them as explicit counterexamples. Private individual-level findings must not be substituted for a general mathematical proof.

14 Limitations and future work

14.1 What has been completed

This note provides a precise formulation of the basic problem, a general counting formula, basic recurrences for subset dynamic programming and block decomposition, and elementary propositions derived from existing combinatorics. Fixed endpoints, breaks, consecutive-appearance prohibitions, and lexicographic optimization of cumulative deviation are treated in one model.

14.2 What remains incomplete

1. No general algorithm has been obtained that counts timetables in time polynomial in the input size for arbitrary large membership matrices.
2. No closed formula based on only a few statistics, such as the distribution of membership counts, has been obtained for the general count.
3. The mathematical novelty of the elementary propositions relative to existing work has not been established.
4. No empirical study has assessed how well F_X represents performers' perceived burdens.
5. Faster exact counting for larger synthetic data and independent third-party reimplementations remain future tasks.

14.3 Open questions

Problem 14.1 (Counting from constraint structure). For which overlap graphs G_X can $T_k(X; a, z)$, with fixed endpoints and bounded block lengths, be computed exactly with lower complexity than in the general case?

Problem 14.2 (Bounds on fairness). Can the membership matrix X , fixed endpoints a, z , and break conditions yield lower bounds on F^* sharper than Proposition 5.7, together with attainable upper bounds?

Problem 14.3 (Impact ratios and local constraints). Can meaningful upper and lower bounds on $I_p^{(k)}$ be obtained from the uniquely forbidden pair count u_p , membership count d_p , and local structure of forbidden edges?

Problem 14.4 (The number of fair timetables). Which structural conditions permit computation of N^* , the number minimizing maximum cumulative deviation after minimizing breaks, faster than general subset dynamic programming?

Problem 14.5 (Multiple days). Can the one-day model be extended to $d \geq 2$ days to formulate a counting problem simultaneously minimizing imbalances in appearances between days and cumulative deviations within each day?

Problem 14.6 (More practical breaks and appearance intervals). In addition to allowing consecutive appearances when separated by a break, can a preference for an interval of two or three intervening groups between a person's performances be treated as a separate objective rather than a mandatory constraint?

14.4 Public data and privacy

This public version contains no operational membership roster, individual-level distributions, or pseudonymized operational incidence matrix. Its computational examples use only independent synthetic cases that do not identify real people. Removing names does not eliminate re-identification risk; operational data will not be released without appropriate assessment, including the consent of the people concerned.

14.5 Presenting the results

This note does not announce the completion of a new mathematical theory. It presents a counting model and computational methods arising from a practical problem. When discussing the work

on websites or social media, derived formulas, established theory, the extent of independent checking, and unverified conjectures should be distinguished. Submission to a peer-reviewed journal would require a comprehensive comparison with prior work, independent verification, and clarification of either the novelty of the setting or an algorithmic advantage.

15 Conclusion

We have defined a timetable-counting problem for a set of people with overlapping memberships in performing groups, simultaneously incorporating prescribed opening and closing groups, breaks after blocks of three to five groups, a prohibition on consecutive appearances without a break, and fairness in individuals' performance positions.

After expressing the total count by a general formula, we derived a subset dynamic program to lexicographically minimize the number of breaks and the maximum cumulative deviation, and to count optimal timetables exactly. We also organized the structure of the problem using independence of small blocks, internal-block symmetry, inclusion–exclusion, containment of membership sets, and interactions between constraint relaxations.

Fully synthetic examples give exact counts for unequal group sizes and for overlapping memberships organized by types. Each count depends on the forbidden relations induced by the specified membership sets. Detailed aggregates and counts from operational data are excluded from the public version to protect privacy.

Future work should use these formulations to derive simpler formulas or lower-complexity algorithms for particular membership structures. Such results must also be carefully checked against the existing combinatorial literature, with the aim of developing the problem as a topic of basic mathematical research.

A Notation

Symbol	Meaning
P, V	Sets of people and groups
m, n	Numbers of people and groups
$X = (x_{pi})$	Person–group membership matrix
A_p, B_i	Groups containing person p ; members of group i
d_p, ℓ_i	Person's membership count; group size
w_{ij}	Number of shared members of groups i, j
c_{ij}	Compatibility for consecutive performance without a break (zero or one)
G_X, H_X	Overlap graph; compatibility graph
a, z	Prescribed opening and closing groups
π	Permutation describing the performance order
D	Set of gaps at which breaks are inserted
k, b	Number of breaks; number of blocks $b = k + 1$
λ	Sequence of group counts in the performance blocks
T_k	Feasible count with k breaks, before fairness optimization
k^*	Minimum feasible number of breaks
E_p, F_X	Individual maximum cumulative deviation; maximum over the people included in the fairness evaluation
$\Delta_X(S)$	Integer-scaled deviation of completed set S
Q^*, F^*	Minimum integer-scaled deviation; optimal fairness value
N^*	Count attaining both minimum breaks and optimal fairness

Symbol	Meaning
$R(n, k)$	Number of allowed block-length sequences
$h_X(S)$	Number of valid orders of group set S without a break
u_p	Number of uniquely forbidden pairs for person p
$I_p^{(k)}$	Count ratio after relaxing person p 's constraint
$\rho(X)$	Minimum number of people whose constraints preserve all forbidden pairs
Φ_X	Generating function recording each person's consecutive-appearance violations

B A component formula for forced adjacencies

Assume $k \geq 1$ and let F be a linear forest with components $C_1, \dots, C_c$. Put $s_i = |C_i|$, and let C_{i_a}, C_{i_z} be the components containing a, z , respectively. If both global endpoints are in the same component, or either is an interior vertex of its component, then $A_\lambda(F) = 0$.

Otherwise, assume $i_a \neq i_z$ and that the global endpoints are endpoints of their respective path components. Let $\mathcal{F}$ be the set of assignments $f : \{1, \dots, c\} \rightarrow \{1, \dots, b\}$ to the $b = k + 1$ blocks satisfying $f(i_a) = 1$, $f(i_z) = b$, and

$$\sum_{i: f(i)=j} s_i = \lambda_j \quad (1 \leq j \leq b).$$

For each assignment, put $c_j = |f^{-1}(j)|$. Let h be the number of components with at least two vertices, and set $\epsilon_a = \mathbf{1}(s_{i_a} \geq 2)$ and $\epsilon_z = \mathbf{1}(s_{i_z} \geq 2)$. Then

$$A_\lambda(F) = 2^{h-\epsilon_a-\epsilon_z} \sum_{f \in \mathcal{F}} (c_1 - 1)!(c_b - 1)! \prod_{j=2}^{b-1} c_j!.$$

The first and last components must be placed at the outer ends of their respective blocks, and their orientations are fixed. Each other nontrivial path component has two orientations. The components of internal blocks may be freely permuted. This correspondence counts the forced-adjacency events.

C Procedure for exact counting

The methods are defined by finite-set operations and recurrences. The computational procedure is as follows:

1. From X , determine the compatibility values c_{ij} and the cumulative deviation $\Delta_X(S)$ for each subset $S \subseteq V$.
2. Following Proposition 6.1, examine the break counts allowed by the block-length constraints in increasing order. For each k , count $H_X(k, \infty)$ without a fairness bound; the smallest k for which it is positive is k^* .
3. Fix $k = k^*$ and vary the integer threshold Q to find the smallest Q^* with $H_X(k^*, Q) > 0$.
4. At that threshold, count $H_X(k^*, Q^*)$ exactly using the transitions of Theorem 6.2. This is N^* , and the optimal fairness value is $F^* = Q^*/n$.

When storing counts, add the contributions of distinct partial orders reaching the same state. Retaining only a best fairness value and discarding paths can omit timetables that ultimately become optimal. Counting at a fixed threshold is therefore essential for the stated method.

D Scope of publication as a research note

This is an unrefereed research note recording the author’s mathematical formulation and checks. It does not represent results that have undergone journal peer review. The novelty of the elementary propositions, completeness of the literature coverage, and validity of the fairness measure as a representation of actual burdens remain open to further assessment.

All membership examples are independent synthetic cases whose conditions are explicitly given in the text. For the real event motivating the study, no membership tables or individual aggregates that could identify performers or the event are included.

References

- [1] T. C. E. Cheng, J. E. Diamond, and B. M. T. Lin, “Optimal Scheduling in Film Production to Minimize Talent Hold Cost,” *Journal of Optimization Theory and Applications*, 79(3), pp. 479–492, 1993. <https://doi.org/10.1007/BF00940554>
- [2] M. Garcia de la Banda, P. J. Stuckey, and G. Chu, “Solving Talent Scheduling with Dynamic Programming,” *INFORMS Journal on Computing*, 23(1), pp. 120–137, 2011. <https://doi.org/10.1287/ijoc.1090.0378>
- [3] N. Sakulsom and W. Tharmmaphornphilas, “Scheduling a music rehearsal problem with unequal music piece length,” *Computers & Industrial Engineering*, 70, pp. 20–30, 2014. <https://doi.org/10.1016/j.cie.2013.12.017>
- [4] R. M. Adelson, J. M. Norman, and G. Laporte, “A dynamic programming formulation with diverse applications,” *Operational Research Quarterly*, 27, pp. 119–121, 1976.
- [5] A. Schaerf, “A Survey of Automated Timetabling,” *Artificial Intelligence Review*, 13(2), pp. 87–127, 1999. <https://doi.org/10.1023/A:1006576209967>
- [6] S. Ceschia, L. Di Gaspero, and A. Schaerf, “Educational timetabling: Problems, benchmarks, and state-of-the-art results,” *European Journal of Operational Research*, 308(1), pp. 1–18, 2023. <https://doi.org/10.1016/j.ejor.2022.07.011>
- [7] E. T. Bax, “Inclusion and exclusion algorithm for the Hamiltonian path problem,” *Information Processing Letters*, 47(4), pp. 203–207, 1993. [https://doi.org/10.1016/0020-0190\(93\)90033-6](https://doi.org/10.1016/0020-0190(93)90033-6)
- [8] E. T. Bax, “Algorithms to count paths and cycles,” *Information Processing Letters*, 52(5), pp. 249–252, 1994. [https://doi.org/10.1016/0020-0190\(94\)00151-0](https://doi.org/10.1016/0020-0190(94)00151-0)