Dance Dance Solution: An IP Approach to Scheduling Dance Rehearsals at CMU

Ayush Mediratta, Judy Li, Rick Sun, Feliks Ma

December 9, 2024

1 Abstract

Scheduling rehearsals for dance groups at CMU with differing room constraints, practice requirements, member availability, and fast-approaching performance deadlines is a complex logistical problem. This paper presents an integer programming model to optimize rehearsal schedules by balancing dancer and room availability, eliminating overbooking, and meeting rehearsal frequency requirements for each group. Key constraints ensure that dancers are only scheduled when available, room capacity is not exceeded, and dancers are given breaks between rehearsals. Our model allows for the designation of subgroups that must meet custom rehearsal goals and ensures that all rehearsals are assigned to plausible rooms and time slots. Applied to real data from an active CMU dance group, the model produces an optimal rehearsal schedule that minimizes conflicts, maximizes efficiency, and demonstrates a flexible framework for complex scheduling needs.

2 Assumptions

We will make a few key assumptions that will allow us to precisely formulate the scheduling problem.

1. There are choreographers and lead dancers whose attendance is important for productive rehearsals.
2. Each dancer can be limited to a certain number of rehearsals per day, and only at times that match their availability. This prevents scheduling conflicts and allows dancers to have sufficient rest between rehearsals.
3. Rehearsals are scheduled based on the availability and capacity of rooms. If the room capacity is sufficient, multiple groups can rehearse in the same room at the same time.
4. There may be subsets of dancers that are required to rehearse together. This assumption acknowledges that some groups within the dance production need to coordinate their choreography by attending the same rehearsals.

3 IP Formulation

To optimize the rehearsal scheduling process, a Linear Integer Programming (LIP) model is used, which considers the availability of the dancers, the room capacities, the rehearsal times and the coordination requirements of the subgroups.

3.1 Variables

Let $\mathcal{N}$ be the set of all dancers $\{1, 2, \ldots, N\}$, with $N = |\mathcal{N}|$. There are exactly M rehearsals that must be scheduled and T time slots which rehearsals can be scheduled at. For brevity, let $\mathcal{M} = \{1, 2, \ldots, M\}$ and $\mathcal{T} = \{1, 2, \ldots, T\}$. Let R be the number of dance rooms available for use, and let $\mathcal{R} = \{1, 2, \ldots, R\}$.

The binary IP variables:

- x_{ijkr}: Indicates if person i attends rehearsal j at time slot k in room r.
- y_{jl}: Indicates if everyone in group S_l attends rehearsal j, where S_l is defined below.
- z_{jkr}: indicates if rehearsal j takes place at time slot k in room r.

The data variables:

- a_{ik}: A binary indicator of whether dancer i is available to attend a rehearsal at time slot k.
- D: A fixed period of time slots representing a day of possible rehearsal times(ex: if rehearsals are one hour each and there are three hours in each day when rehearsals can be scheduled, then $D = 3$).
- F: The maximum number of rehearsals a dancer may attend in D time slots.
- $S_1, S_2, \ldots, S_n \subseteq \mathcal{N}$: n groups of dancers (possibly overlapping), and let $\mathcal{S} = \{1, 2, \ldots, n\}$.
- p_l: The number of practices everyone in group S_l must attend together.
- c_{ij}: A measure of how important dancer i is at rehearsal j (ex: if dancer i is a choreographer or lead dancer, they will have a higher importance at certain rehearsals).
- f_{rk}: A binary indicator of whether room r is available for use at time slot k.
- g_r: The number of dancers that can fit in room r at once.

3.2 Constraints

$$\sum_{r\in\mathcal{R}} x_{ijkr} \leq a_{ik} \qquad \forall i\in\mathcal{N}, j\in\mathcal{M}, k\in\mathcal{T} \tag{1}$$

$$\sum_{\substack{j\in\mathcal{M}\\ r\in\mathcal{R}}} x_{ijkr} \leq 1 \qquad \forall i\in\mathcal{N}, k\in\mathcal{T} \tag{2}$$

$$p_l \leq \sum_{j\in\mathcal{M}} y_{jl} \qquad \forall l\in\mathcal{S} \tag{3}$$

$$y_{jl} \leq \sum_{\substack{k\in\mathcal{T}\\ r\in\mathcal{R}}} x_{ijkr} \qquad \forall l\in\mathcal{S}, i\in S_l, j\in\mathcal{M} \tag{4}$$

$$\left(\sum_{\substack{i\in S_l\\ k\in\mathcal{T}\\ r\in\mathcal{R}}} x_{ijkr}\right) - |S_l| + 1 \leq y_{jl} \qquad \forall l\in\mathcal{S}, j\in\mathcal{M} \tag{5}$$

$$\sum_{\substack{k=k'\\ j\in\mathcal{M}\\ r\in\mathcal{R}}}^{k'+D} x_{ijkr} \leq F \qquad \forall i\in\mathcal{N}, k'\in\{1,2,\ldots,T-D\} \tag{6}$$

$$x_{ijkr} \leq z_{jkr} \qquad \forall i\in\mathcal{N}, j\in\mathcal{M}, k\in\mathcal{T}, r\in\mathcal{R} \tag{7}$$

$$\sum_{\substack{r\in\mathcal{R}\\ k\in\mathcal{T}}} z_{jkr} = 1 \qquad \forall j\in\mathcal{M} \tag{8}$$

$$z_{jkr} \leq f_{rk} \qquad \forall j\in\mathcal{M}, r\in\mathcal{R}, k\in\mathcal{T} \tag{9}$$

$$\sum_{\substack{i\in\mathcal{N}\\ j\in\mathcal{M}}} x_{ijkr} \leq g_r \qquad \forall k\in\mathcal{T}, r\in\mathcal{R} \tag{10}$$

$$x_{ijkr}, y_{jl}, z_{jkr} \in \{0,1\}$$

Here is a contextual interpretation of each constraint:

- Constraint (1): Ensures that dancers are only scheduled when they are available.
- Constraint (2): Ensures the dancer cannot be assigned to overlapping rehearsals.
- Constraint (3): Ensures that each group S_l meets its required number of rehearsals.
- Constraints (4) and (5): Ensures consistency between group attendance y_{jl} and individual dancer attendance x_{ijkr}. Specifically, y_{jl} is set to 1 only if all dancers in group S_l are present for rehearsal j.

- Constraint (6): Ensures that the number of rehearsals a dancer can attend in a fixed period D is F.
- Constraint (7): Ensures that x_{ijkr} (dancer room time assignment) is consistent with z_{jkr} (room time assignment).
- Constraint (8): Ensures each rehearsal has a unique room and time slot.
- Constraint (9): Ensures that rehearsals are only assigned to rooms available during the scheduled time slot.
- Constraint (10): Ensures that the number of dancers assigned to a room does not exceed its capacity g_r.

3.3 Objective Function

The objective is to maximize the weighted attendance of the dancers at all rehearsals, where the weights reflect the importance of specific dancers, such as choreographers or lead performers. The objective function is given as:

$$\text{maximize} \quad \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{M}} \sum_{k \in \mathcal{T}} \sum_{r \in \mathcal{R}} c_{ij} x_{ijkr}$$

4 Data Collection

We interviewed an anonymous dance group at CMU to obtain data for the current semester that we could use to test our model. We found that there were 30 dancers (N) who needed to schedule 20 rehearsals (M) across 10 weeks with each day having 3 available time slots (D). We asked dancers in the group to provide us with the time slots and rooms they were able to use throughout the semester ($T = 210, R = 3$).

We also requested each dancer's expected availability at every time slot (a_{ik}). We requested information on subgroups of their piece to find out which dancers needed to practice with each other to work on combined choreography (S_l; $n = 3$) and asked the choreographers to determine how many practices subgroups needed together (p_l) as well as an upper limit for the number of practices a dancer can attend in a day ($F = 1$). We also asked choreographers to identify the lead dancers in their pieces. Using this info, the choreographers and lead dancers were given different weights from the other dancers in each piece (c_{ij}). Specifically, choreographers were given an importance of 5 for rehearsals they choreograph, while lead dancers were given an importance of 3 for rehearsals they are leads in. All dancers that were not a choreographer or lead for a certain rehearsal were assigned an importance of 1 for that rehearsal.

Room availability and occupancy data was obtained from CMU's 25Live room reservation website (f_{rk}, g_r).

After obtaining this initial dataset, we used it to synthetically create a larger dataset with 100 dancers, 70 rehearsals to schedule, and 6 rooms in order to test the model for performance in larger cases.

5 Finding an Optimal Schedule

To solve our integer program, we encoded it in gurobipy, a Python wrapper for the Gurobi optimization program. Gurobi uses a variety of methods to solve integer programs, including but not limited to LP relaxations, applying Branch and Bound, cutting planes, and heuristics. It starts by solving a relaxed version of the problem to provide bounds, then systematically explores feasible solutions using Branch and Bound, pruning branches based on infeasibility or suboptimality. Cutting planes tighten the relaxation to speed up convergence, while heuristics generate quick feasible solutions. These techniques are supported by presolve routines that simplify the problem, reducing complexity and improving performance.

After entering our collected data, the number of variables the program was required to optimize over was

$$\#x_{ijkr} + \#y_{jl} + \#z_{jkr} = (30 \times 20 \times 210 \times 3) + (20 \times 3) + (20 \times 210 \times 3) = 390660$$

Gurobi was able to find an optimal assignment. From the variable assignment, we extracted the total schedule, including the location and the dancers assigned to each rehearsal. We also extracted each dancer's individual schedule, sorted by timeslot. Examples of both are shown here:

```
Rehearsal 0 (Timeslot 6, Room 0) has dancers:
 [0, 1, 2, 3, 5, 6, 8, 9, 11, 13, 14, 15, 16, 17,19, 21, 22, 23, 24, 25, 27, 28]
Rehearsal 1 (Timeslot 19, Room 0) has dancers:
 [0, 1, 2, 3, 4, 5, 7, 8, 9, 11, 14, 15, 16, 17, 18, 19, 20, 21, 22, 24, 26, 28, 29]
Rehearsal 2 (Timeslot 23, Room 0) has dancers:
 [0, 2, 3, 4, 6, 7, 8, 9, 12, 13, 14, 15, 16, 18, 19, 20, 21, 22, 23, 24, 27, 28]
Rehearsal 3 (Timeslot 35, Room 0) has dancers:
 [0, 1, 2, 3, 4, 5, 6, 7, 9, 10, 11, 12, 13, 16, 17, 19, 20, 21, 23, 24, 25, 26, 29]
```

```
Dancer 0's Schedule:
  Timeslot 6 - Room 0 (Rehearsal 0)
  Timeslot 19 - Room 0 (Rehearsal 1)
  Timeslot 23 - Room 0 (Rehearsal 2)
  Timeslot 35 - Room 0 (Rehearsal 3)
  Timeslot 54 - Room 0 (Rehearsal 4)
  Timeslot 69 - Room 0 (Rehearsal 5)
  Timeslot 91 - Room 0 (Rehearsal 7)
  Timeslot 110 - Room 0 (Rehearsal 8)
  Timeslot 117 - Room 0 (Rehearsal 9)
  Timeslot 125 - Room 0 (Rehearsal 10)
  Timeslot 130 - Room 0 (Rehearsal 11)
  Timeslot 149 - Room 0 (Rehearsal 14)
  Timeslot 155 - Room 0 (Rehearsal 15)
  Timeslot 171 - Room 0 (Rehearsal 16)
  Timeslot 200 - Room 0 (Rehearsal 18)
  Timeslot 205 - Room 0 (Rehearsal 19)
```

Examining the optimal solution yields some notable details. One significant fact is that only the largest room (with capacity 23) is used for all rehearsals, despite its lack of availability during some time slots, in order to maximize attendance. This optimization of attendance is also demonstrated by the fact that each rehearsal is filled with nearly the maximum number of dancers possible.

Also, the assigned choreographers and lead dancers are present in nearly all rehearsals they are important in, while many of the other dancers present are chosen to satisfy the subgroup constraints. In the example above, dancer 0 is a choreographer for rehearsals 0-9 and is assigned to attend 9 of these 10.

After obtaining this solution, we attempted to optimize a schedule of 70 rehearsals for our synthetic dataset of 100 dancers, with 6 available rooms and 7 subgroups. However, this scaled the problem to an intractable degree, as the number of variables to be optimized became

$$\#x_{ijkr} + \#y_{jl} + \#z_{jkr} = 8908690$$

Gurobi was not able to make significant progress toward a solution for this larger dataset.

6 Conclusion and Future Improvements

6.1 Summary of Results

We have successfully developed and implemented an Integer Programming model to optimize rehearsal schedules for dance groups. The model effectively balances dancer availability, room constraints, and subgroup requirements to produce conflict-free schedules while prioritizing key performers, such as choreographers and lead dancers. By applying the model to real-world data, we demonstrated its capability to optimize attendance and resource utilization, achieving an efficient and practical schedule for small dance groups.

Despite its success, the model has notable limitations. First, its reliance on the exact optimization technique of integer programming makes it computationally intensive for large datasets with numerous dancers, rehearsals, and time slots. Second, the current formulation assumes static dancer and room availability, which may not fully reflect the dynamic nature of real-world scheduling, where last-minute changes are common. Lastly, the rigidity of predefined subgroups limits flexibility in cases where dancer assignments evolve over time due to changing choreography or production needs.

6.2 Future Improvements

To address these limitations and improve the model's practicality, several avenues for future work are proposed:

1. **Development of Heuristic-Based Methods:** Designing heuristic approaches, such as genetic algorithms or simulated annealing, could approximate optimal solutions more efficiently, making the model scalable to larger datasets.

2. **Dynamic Scheduling Capabilities:** Extending the model to account for dynamic changes in dancer and room availability would enhance its adaptability. Incorporating real-time updates could allow for re-optimization when unexpected conflicts or cancellations arise.

3. **Incorporating Flexible Deadlines:** Allowing for multiple deadlines for groups or subgroups would better reflect staggered performance needs and rehearsal timelines, making the model more realistic for diverse production schedules.

4. **Alternative Problem Formulations:** Exploring alternative formulations, such as network flow models or constraint satisfaction frameworks, might uncover new optimization techniques that simplify or accelerate the scheduling process.

5. **Flexible Subgroup Assignments:** Enhancing the model to support dynamic subgroup changes would accommodate evolving choreography needs and improve workload distribution across dancers.