Hierarchical Seating Allocation (Extended Abstract)

Anton Ipsen[1], Michael Cashmore[1], Parisa Zehtabi [1], Nicolas Marchesotti[1], Kirsty Fielding[1], Daniele Magazzeni[1], Manuela Veloso[1]

[1]AI Research, J.P. Morgan Chase

Introduction

In today's dynamic work environment, commercial office real estate is one of the most significant fixed costs for companies and large organizations. The recent push by many government departments and organizations for a full return to the office (RTTO) has intensified the need to optimize the usage of commercial real estate. A big part of the RTTO effort is continuous planning and visualizing of potential floor plans under different (re-)planning scenarios. Previous research has focused on solving the Seating Allocation Problem (SAP) that aims to allocate fixed-sized teams to seats by formulating the problem as Mixed-Integer Programming (MIP) Capacitated P-Median Problem (CPMP) (Hales and García 2019).

In this extended abstract, we bridge the gap in the SAP literature by introducing the Hierarchical Seating Allocation Problem (HSAP) which considers allocating teams under the scope of their organizational chart. Therefore, the HSAP is motivated by the necessity for large organizations to ensure that teams with close hierarchical relationships are seated in proximity to one another.

Seating Allocation Problem

The SAP can be solved with the following MIP problem, which solves a single subproblem of the HSAP. We define the binary variable x_{ik} as 1 if seat i is allocated to central seat k, and the binary variable y_{kt} as 1 if seat k is the central seat for team t. Moreover, we denote $D(i,k)$ as the distance between seats i and k and S and T as the sets of seats and teams respectively.

$$\min \sum_{i \in S, j \in S} D(i,k) x_{ik} \tag{1}$$

$$\text{s.t.} \sum_{k \in S} x_{ik} \leq 1 \quad \forall i \in S \tag{2}$$

$$\sum_{t \in T} y_{kt} \leq 1 \quad \forall k \in S \tag{3}$$

$$\sum_{k \in S} y_{kt} = 1 \quad \forall t \in T \tag{4}$$

$$\sum_{i \in S_d} x_{ik} = \sum_{t \in T} d_t y_{kt} \quad \forall k \in S \tag{5}$$

The objective function in Eq. (1) minimizes the total distance between the seats allocated to their central seats. The constraints can be explained as follows:

- Eq. (2) stipulates that all seats are assigned to at most one central seat. The less-or-equal sign is due to scenarios where the total seat supply is higher total seat demand.
- Eq. (3) describes that seat k can only be the central seat of at most one team and Eq. (4) describes that each team should have exactly one central seat.
- Eq. (5) stipulates that if seat k is central for team t, then the required number of desks for team t must be assigned to central seat k.

As seen from Eq. (1) the estimate of $D(i,k)$ is crucial to the solution quality of SAP. Previous research uses the Euclidean distance to compute the pairwise seat distances, which fail to reflect the real-life walking distance due to impassable walls and obstacles (Barry et al. 2021; Hales and García 2019).

It is possible to reduce the complexity of the above formulation by reformulating the problem similarly to a K-means clustering problem: (*location*) selecting the central seats with a heuristic, (*allocation*) allocating seats to teams with a MIP (Hales and García 2019).

Estimating Pairwise Seat Distances

Although precise grid-based path-finding algorithms such as A^* or D^* are available, they are often not computationally feasible due to the extensive search space (due to large floor plans) and the large number of pairwise seat distances in large instances (Wu et al. 2021). Therefore we propose constructing a *probabilistic roadmap* (PRM) using rapidly-exploring random trees (RRT) to construct a fully-connected graph of nodes N, corresponding to seats and intermediate positions, and edges E, representing collision-free connections between nodes. The shortest path between each seat can easily be calculated with Floyd-Warshall algorithm. The PRM is depicted in Algorithm 1.

The PRM is initialized as a single node randomly sampled from any collision-free position on the floor plan (line 3).

Copyright © 2025, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

Algorithm 1: Generate PRM

1: **Input:** Maximum Nodes K location of seats S, casting distance δ_c,connection threshold δ_s
2: **Output:** PRM $\langle N, E \rangle$
3: $x_{init} \leftarrow$ random collision-free position
4: $N, E \leftarrow \{x_{init}\}, \emptyset$
5: **for all** $s \in S$ **do**
6: $\quad connected \leftarrow$ connect_seat(s, N, E, δ_s)
7: $\quad$ **while** $\neg connected$ and $|N| < K + |S|$ **do**
8: $\qquad x_{rand} \leftarrow$ random collision-free position
9: $\qquad n \leftarrow$nearest_neighbour(N, x_{rand})
10: $\qquad x_{new} \leftarrow$ cast(n, x_{rand}, δ_c)
11: $\qquad$ **if** $x_{new} \neq \perp$ **then**
12: $\qquad\quad E \leftarrow E \cup$ new_edges(x_{new}, N)
13: $\qquad\quad N \leftarrow N \cup \{x_{new}\}$
14: $\qquad\quad connected \leftarrow$ connect_seat$(s, \{x_{new}\}, E, \delta_s)$
15: $\qquad$ **end if**
16: $\quad$ **end while**
17: **end for**
18: **return** $\langle N, E \rangle$

The process begins by sampling a random collision-free position x_{rand} within the floor plan (line 8); then selecting its nearest neighbour n in the PRM (line 9). A new candidate node x_{new} is created by casting a distance δ_c from n towards x_{rand} (line 10). If the casting edge is not collision-free, no new node is created. Otherwise, both x_{new} and the casting edge are added to the PRM (lines 12–13). Finally, *connect_seat* is called to check if the new seat is within δ_s of x_{new} and if so, connects it to the PRM (line 14).

Hierarchical Seat Allocation

A solution to an HSAP is a set of allocations $X := \{X_0, \dots X_L\}$, where L is the levels in the organizational hierarchy. Each allocation $X_l \in X$ is a mapping $X_l : S \rightarrow T_l$ from seats to teams at that level l, such that for each team $t \in X_l$ the number of desks assigned to that team meets its requirements. Also, the assignment to a non-root team in level $l + 1$ must not violate that of its parent at level t. That is, teams at level $t + 1$ must be assigned to seats that were previously allocated to their parent at level t. Importantly, a parent team can have many child teams.

The solution quality for HSAP is measured in the same way as SAP (total distance of each seat to its team's central seat). However, for the HSAP we take the sum of this metric at every level in the hierarchy.

Solving the HSAP as a single monolithic instance is intractable for small, medium and large scale instances (i.e. over 50 seats). Therefore, we propose a decomposition of the hierarchical seating problem into a series of SAP subproblems that can be solved independently.

The algorithm is initially called with a set of empty allocations ($X_l = \emptyset, \forall X_l \in X$), root teams T, and level $l = 0$. Seats are allocated by a recursive (depth-first) traversal through the organizational hierarchy (line 3-9). At each step of this traversal, teams are first allocated seats (line 2). This allocation is performed using any solution algorithm to SAP. Then, each team's children are allocated seats through a recursive call (line 7). The children of a team (line 5) can only be assigned seats that were allocated to their parent (line 4).

Algorithm 2: Depth First Hierarchical Seat Allocation (DF-HSA)

1: **Input:** Seating allocation problem solver SAP, seats S, teams T,distance matrix D,previous allocation X,iteration l
2: $X_l \leftarrow X_l \cup SAP(S, T, D)$
3: **for** team $t \in T$ **do**
4: $\quad S' \leftarrow \{s : X_l(s) = t\}$
5: $\quad T' \leftarrow children(t)$
6: $\quad$ **if** $T' \neq \emptyset$ **then**
7: $\qquad$ Call $DF - HSA(SA, S', T', D, X, l + 1)$
8: $\quad$ **end if**
9: **end for**
10: **return** X_l

Conclusion & Future Work

In this extended abstract, we presented the Hierarchical Seating Allocation Problem (HSAP) and proposed a novel approach utilizing a Probabilistic Roadmap (PRM) to estimate pairwise seat distances. For future research, we plan to investigate the applicability of HSAP to open-source datasets and conduct a study on the novelty of the PRM in comparison to existing literature.

Disclaimer. This paper was prepared for informational purposes by the Artificial Intelligence Research group of JPMorgan Chase & Co. and its affiliates ("JP Morgan"), and is not a product of the Research Department of JP Morgan. JP Morgan makes no representation and warranty whatsoever and disclaims all liability, for the completeness, accuracy or reliability of the information contained herein. This document is not intended as investment research or investment advice, or a recommendation, offer or solicitation for the purchase or sale of any security, financial instrument, financial product or service, or to be used in any way for evaluating the merits of participating in any transaction, and shall not constitute a solicitation under any jurisdiction or to any person, if such solicitation under such jurisdiction or to such person would be unlawful.

References

Barry, M.; Gambella, C.; Lorenzi, F.; Sheehan, J.; and Ploennigs, J. 2021. Optimal Seat Allocation Under Social Distancing Constraints. *arXiv preprint arXiv:2105.05017*.

Hales, R. O.; and García, S. 2019. Congress seat allocation using mathematical optimization. *TOP*, 27(3): 426–455.

Wu, Z.; Meng, Z.; Zhao, W.; and Wu, Z. 2021. Fast-RRT: A RRT-Based Optimal Path Finding Method. *Applied Sciences*, 11(24): 11777.