User-Driven Occupant Satisfaction and Space Utilization Tool for the New Hybrid Office Work Paradigm

Nicolas Lomas

A thesis

submitted in partial fulfillment of the

requirements for the degree of

Master of Science

University of Washington

2023

Committee:

Christopher Meek

Narjes Abbasabadi

Program Authorized to Offer Degree:

Architecture

University of Washington

Abstract

User-Driven Occupant Satisfaction and Space Utilization Tool for the New Hybrid Office Work Paradigm

Nicolas Lomas

Chair of the Supervisory Committee:

Christopher Meek

Department of Architecture

Hybrid-style work has become the default model of office operation in many industries. Estimates of up to 74% of US companies are switching to or practicing an at-home and in-person hybrid work style (Kugler et.al., 2022). Hybrid work provides organizations and workers with considerable savings on time, transportation, maintenance, and energy costs. With hybrid work becoming a multi-industry standard, the utilization of a workspace optimization and scheduling tool will allow companies to organize and improve their coordination and productivity; while cutting down on their office's carbon footprint, size, and cost. Relatedly, "hoteling" is an office scheduling strategy that has workers change desks and spaces on a daily basis, creating a more flexible office that allows workers access to a range of amenities and spaces. The thesis

proposes a system that offers an open-source algorithm for improved office space planning, utilization, and continuous worker feedback to guide facility planning and operations decisions. Multiple sorting agents will be developed in order to prioritize different desired outcomes (productivity or coordination) or to provide recommendations to appropriately size space while improving worker experience. These agents will be tested in a range of office staffing, scheduling, and future planned utilization scenarios. The program will also provide insight to building managers, company owners, and architects on the use and preferences of different office spaces and amenities.

Table Of Contents:

1. Introduction

1.1 The Rise of Telework / Research Context:

Subsequent to COVID-19, remote work became the primary office work strategy across many industries. Allowing people to continue to work through a dangerous pandemic from the comfort of their own homes, remote work became many people's preferred method of work. Recent studies following the pandemic have shown many benefits to remote work. These benefits include savings for office workers in the form of reduced transportation costs and reduced time traveling. The benefits of comfort, time-saving, and reduction of monetary costs, lead to greater worker satisfaction. Improved worker satisfaction has been linked to improved worker productivity and health. While workers received many benefits, employers also received benefits of their own. Employers were able to save on energy costs and with improved worker satisfaction companies can realize a greater employee retention rate and growth in productivity (US General Service Administration, 2006). Entirely remote work also poses organizational challenges, however, with pitfalls being lack of equipment needed for work, lack of computer literacy, lack of stable connection, and lack of outreach/isolation from coworkers (Ferreira et.al., 2021) . This isolation can lead to poor coordination and integration of new workers creating a poor work environment. Additionally, operating an entirely remote office requires a great amount of effort and money must also be invested in employee equipment and tech support. However, following the widespread availability of COVID-19 vaccines in-person work has become less hazardous once again.

With a safe return to the office, many have still sought to maintain a remote style of work. In a survey by the Building Owners and Managers Association (BOMA) conducted in 2022,

nearly two-thirds of office workers and office managers would prefer hybrid-style work (Kugler et.al., 2022). The introduction of a hybrid style of work; a style of work that has office workers come in a specified portion of the week into the office while also getting a portion of the week remote, has provided a compromise that preserves some of the benefits of remote work. Hybrid work has also become a great solution to the shortcomings of remote work. With Hybrid style work conserving an office work culture it alleviates the worker isolation felt by many making it easier to integrate new workers and facilitate collaboration. Hybrid work has become an industry standard with estimates of up to 74% of US companies stating they are switching to or currently practicing an at-home and in-person hybrid work style (Kugler et.al., 2022). This transition allows workers to work from home saving large portions of transportation costs and time.

With a reduction in the number of workers in transit and in office, energy use has decreased as well. However, even with fewer employees in the office, large office buildings begin to create more wasteful energy loads and even more unused office space. With expensive rent and energy costs, wasting space can be pricey for any given company. Unused space, while being more costly to a business, also creates greater carbon emissions and removes the space's opportunity for utility by being reserved for tasks never needed. 78% of office tenants say they are likely to reassess their space needs or are unsure if they have the correct office sizing (Kugler et.al., 2022). Adding complications, hybrid style work has allowed for shared desk spaces which would allow tenants to reduce the need for large office sizes even further, in order to reduce rent cost and energy use. With a lack of guidance on how to facilitate office use and begin sizing for hybrid work, tenants are often left without tools to evaluate current and predict

future space needs. Many companies have begun to reassess office spaces and cancel lease renewals pulling out of or reducing size in many urban contexts (Kugler et.al., 2022). This adversely affects local economies that have relied on worker patronage spelling the end for many local tertiary businesses (Basnet N, 2023). To better assess space usage dynamics, this thesis proposes a workspace utilization framework and tool that allows building owners, designers, and decision-makers to optimize space utilization for productivity, cost, and improve worker satisfaction.

1.2 Proposed Methodology: (Table 1)

To help better organize daily work operations and provide guidance for hybrid office sizing, three algorithms that will be referred to as agents, will be developed and tested for viability. These three agents include a best fit decreasing random sort algorithm (Random Agent), a top three best fit decreasing sort algorithm with worker-defined space metrics (Averaged Sort Agent), and finally an additional top three best fit decreasing sort algorithm with measured space metrics (Measured Sort Agent). All agents are named after their ways of thinking and will be fully explained in Methodology. This thesis will additionally create an occupancy model to simulate worker behaviors and feedback in order to test the efficiency of the three space allocation agents. This feedback will be used to gather and share space use information that in a real-world context will be shared with a designer, architect, or consultant to tackle identified issues to improve space use and optimization.

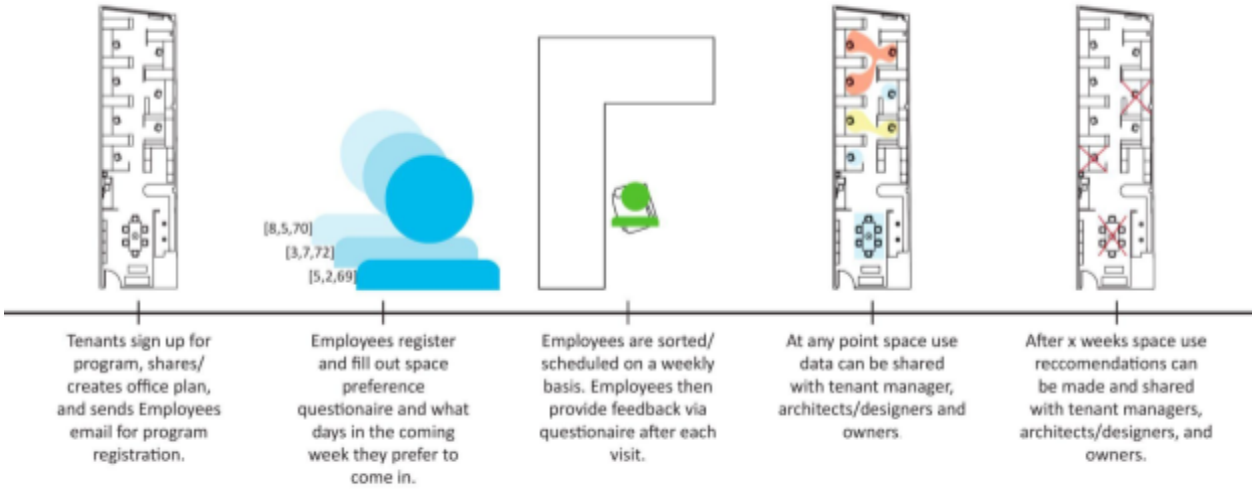
When training and using an AI (artificial intelligence), data is always a pivotal and vital component. For the purpose of this thesis building worker scenario data is generated on a set of odds defined by hand to create various office contexts. The data set created will also have a

shorter list of features (also known as data metrics) to investigate the algorithm's framework and potential to be integrated with real-world data at a later date. Using this stochastic method (explained further in Literature Review: Occupancy Modeling) for each worker feature the program will be able to more closely model human behavior as opposed to the alternative deterministic method (explained further in Literature Review:Occupancy Modeling). In a real-world application, tenant managers/owners will share all employees' emails. From there a questionnaire will be sent out to all employees to ascertain their space quality preference and their in-person workday preferences for that week.

While employees fill out a questionnaire, tenant managers/owners will share office plans, space types, and quantities. From here an office space can be measured to ascertain its thermal, visual, and acoustic space metrics. For this thesis, a single value will be ascribed to define each office workstation's thermal, acoustic, and lighting quality. These values will be used in the worker feedback simulation and the Measured Sort Agent's (further explained in Methodology:Sort Agents) decision-making. For the office plan, this thesis will work with a real-world office in Seattle as an example and a potential future case study. In some contexts, a measurement of the office space qualities cannot be afforded so a process that uses worker feedback to ascertain space qualities will be explored. In this context the Averaged Sorting Agent (further explained in Methodology:Sort Agents) will make office sorting decisions based on the worker's perceived office space qualities. These agents focus on making sort decisions based on space qualities so that the agents may help maximize worker satisfaction. In the future various metrics that seek various outcomes can follow this framework to achieve their goal, but for this thesis, the focus is mainly on worker comfort and worker satisfaction. In either case, the

agents will use space qualities to match workers to their preferred spaces. When workers share a common interest in a workstation the space will be given to the one with the highest priority value. This priority hierarchy is defined by the individual who was the most dissatisfied worker in the previous week. This priority strategy is referred to as the “squeaky wheel” algorithm (Tanimoto, 2007). Using worker preference, and space quality preference, to sort into the workstations the Sort Agent will receive feedback on its decisions and will be updated its understanding of the space (in the case of the Average Agent) and soon a pattern will be created for designers to interpret.

Table 1: Proposed Tool Timeline



2. Literature Review

2.1 Data Considerations:

In real-world applications, it is generally not a wise decision to rely on AI for social context decisions (Hao et.al., 2020). Real-world data can have an intrinsic bias due to societal bias. For example during COVID when schools were purely online, parents dealt with housework, child care, and career work. With more responsibilities at home than a single individual, these parents were predisposed to be less productive at home compared to their single coworkers due to at-home online public schooling (Dunatchik et.al., 2021). With additional responsibilities, data was recorded that showed parents had been more productive at work than at home. In this context, allowing an AI agent to decide in-person work days would require parents, especially single parents, to be in the office more often than a single individual. This agent would completely ignore the inequality it was perpetuating and would be ignorant of the fact that students had returned back to school relieving parents of a major responsibility.

In this thesis, a worker's contracted in-person work days will be randomly created through a weighted odds set but in real life, such matters should never be decided by an algorithm. This thesis' sorting agents for this reason will not propose a method of sorting that adjusts the amount of in-person work days a worker has outside of a random assignment. The fact that data and worker contexts are randomly generated and not adjustable also means that results from this thesis should be considered exploratory, as it is a reflection of human-decided weighted odds instead of a reflection of real-world data. Rather than looking at empirical results as proof, this thesis will show a greater reason to research the algorithm's framework with real-world data. The goal of this thesis is not to produce a specific result but to explore a

potential process that may help facilitate office use as well as identify a pattern of use that could be used in space sizing and design revisions for space optimization.

2.2 Comfort Metrics:

Worker satisfaction leads to a plethora of benefits (Al Horr et.at., 2016). When workers are satisfied with their careers and work, companies see improved employee retention, improved worker health, and improved productivity (Al Horr et.at., 2016). Indoor Environmental Quality (IEQ) is strongly related to occupant satisfaction (Dehnavi et.al., 2020). Three important aspects of IEQ that greatly affect a person's comfort are an indoor environment's visual, thermal, and acoustic qualities (Dehnavi et.al., 2020). A high IEQ and a healthier environment not only improve worker productivity and satisfaction but decreases sick time leave (Al Horr et.at., 2016). On the opposite end, a space with a low IEQ can have a negative effect on workers' satisfaction, performance, and productivity (Al Horr et.at., 2016). Physical comfort that is described by IEQ can be broken up more narrowly than just the three qualities of visual, thermal, and acoustic comfort. Visual comfort can be affected and defined by glare, illuminance, views of nature, distance from windows, and even color quality or newly defined Equivalent Melanopic Lux (EML). While thermal comfort is also affected by many factors like air velocity, temperature, and even clothing. Even acoustic comfort is affected by a multitude of sounds and its potential quality, including a space's reverberation times, speech clarity transmission, or even the volume of coworkers in the space. These aspects of comfort also greatly influence each other and defining a space's IEQ becomes a major undertaking. Understanding what metrics to use and how to ascertain a worker's preference within these

metrics requires a great amount of effort and time, especially when a worker and office are non-static in nature. Simulations to ascertain IEQ could be used to characterize individual workstations; however, these considerations are not within this thesis' scope since evaluating the appropriate dimensionality (number of variables a computer uses to evaluate decisions) of metrics is not a part of the scope of this thesis. Within this scope, however, this thesis will be comparing the results of the proposed Averaged Sort Agent with a worker based understanding of the space and the Measured Sort Agent that uses measured space qualities (which in the thesis will be ascribed by the operator). These space qualities include a single value for visual, acoustic, and thermal comfort. For visual and acoustic comfort a scale of 1-10 is used; 1 indicates a dim environment or a quiet environment while a 10 indicates a bright or loud environment. Thermal comfort will be defined by Fahrenheit temperature values.

Outside of physical comfort, nonphysical aspects of work affect worker satisfaction. A worker's ability to have some level of control over their schedule, their environment, and project autonomy can greatly sway worker satisfaction (Kwon et.al., 2019). Working in a hybrid style office workers have the increased opportunity to have greater control over their work schedule, either working in the office or remotely. This office sorting system will allow workers to have some control over their schedule and greater participation in their work environment via feedback and schedule requests. To optimize worker satisfaction and get the most out of hybrid work benefits the sorting agents will only consider worker preference for certain space qualities and schedule requests.

2.3 Occupancy Modeling:

This thesis uses simulated worker behavior data as a platform to test approaches to scheduling and locating building occupants within a hybrid-style office. While occupancy models simulate workers in a space there are two focuses of occupancy modeling (Coenen et.al., 2014). One focus includes occupancy of a space or equipment; meaning whether a space/piece of equipment is occupied or not. The other focus includes occupancy behaviors within a space; meaning what actions occur in a space and at what frequency. Each focus uses a different strategy of modeling to simulate occupant behavior, preferences, and their effects on their environment.

Using a deterministic model, occupancy can be simulated using a time schedule that follows the percentage of the maximum capacity of a space or equipment being used (Chong et.al., 2021). An example of deterministic modeling is to simulate the amount of power being used by a building's light fixtures, a schedule of occupancy is used to define what percentage of the maximum lighting load of a building is being used at any given hour. This strategy of modeling is helpful when trying to understand how much of the space is being used. However, this strategy of modeling is backed by a large data set of occupancy to define what percent of a given metric is being used at any given time. This data set recording occupancy however was gathered before COVID-19 and before the rise of hybrid-style work. Using deterministic modeling in this context would not yield very accurate or effective results, however, to evaluate how much of the space is being occupied in this thesis the algorithm will simply need to return the number of spaces that are scheduled to be occupied by a worker. If a space is not being scheduled as often as others this can be from an algorithmic inefficiency or because the sorting

agents have identified a space to be commonly unsatisfactory to workers in the space. In a real office context feedback received for each space will confirm whether a space had been occupied or not, helping bolster the occupancy modeling in the future. In addition to improving deterministic models, occupancy feedback can help identify problematic spaces. If a workstation or space has workers calling out a majority of its assignments then there is a likely chance this space is unsatisfactory to many. This model proposed by this thesis will not simulate sick leave, however, will simulate vacation days to test the sorting agents' capacity to optimize the work schedule.

To simulate these behaviors this thesis instead will use a stochastic occupancy model as this model is better suited to simulate occupancy behavior (Chong et.al., 2021). Stochastic methods use the odds of a behavior occurring and run these odds a specified number of times to generate worker behaviors. This thesis uses stochastic methods to generate an office's context/worker features (behaviours). These features include "In-Office Work Days", "Group Assignments", meeting requests (labeled as "Conference Room" requests), worker space quality preferences, in-office day requests(labeled as "Day In" requests), remote day requests (labeled "Day Off" requests), and "Vacation Day" requests. For each feature, a weighted odds set is given to the algorithm to test a range of work contexts. For example, some offices may have many groups trying to meet in a conference space and an increase in conference room spaces may need to be suggested. To simulate this, the algorithm would simply need a set of odds that would have a higher chance of assigning a worker in an office to be a part of a group and have many meeting requests. Using this strategy creating various office contexts for sorting will be streamlined.

2.4 Space Use Analysis:

In their paper "Ontology for Representing Building Users' Activities in Space-Use Analysis (SUA)," Tae Wan Kim et al (Kim et.al., 2014) propose a system describing all user activities in various contexts (Table 2). Under the proposed ontology, user activities are either a typical activity or an atypical activity where typical activities are conducted on a regular basis and atypical activities are rare events that are not considered under space utilization but still need to be accounted for in the design. User activity can be further described by a user, action, constraint, preference, the ratio of people involved in a task, and frequency of how often this activity is done if a typical behavior. Since this thesis begins at post occupancy many assumptions will be established using this ontology. Under this ontology, however, the goal is to share how a given space is being used, while under hybrid work the tasks required can be assumed to be possible in any space due to its portable nature. Rather than trying to evaluate and map all spaces and possible uses for design or permanent worker allocation this thesis will use this ontology to define how the space is being used before exploring how to organize it.

Under this ontology, users can be defined as either an individual or a group entity and are broken up into two classes: "ordinary users" and "key users." A program proposed by Kim to manage office space uses this classification to give access to spaces for key workers based on spatial preference, and ordinary workers spaces based on the minimum requirements for the task (Kim et.al., 2014). To democratize the office and give all employees an equal opportunity to be satisfied with their workspace, this thesis' proposed framework only considers a hierarchy between a group and an individual. This hierarchy is used to ensure that important meetings

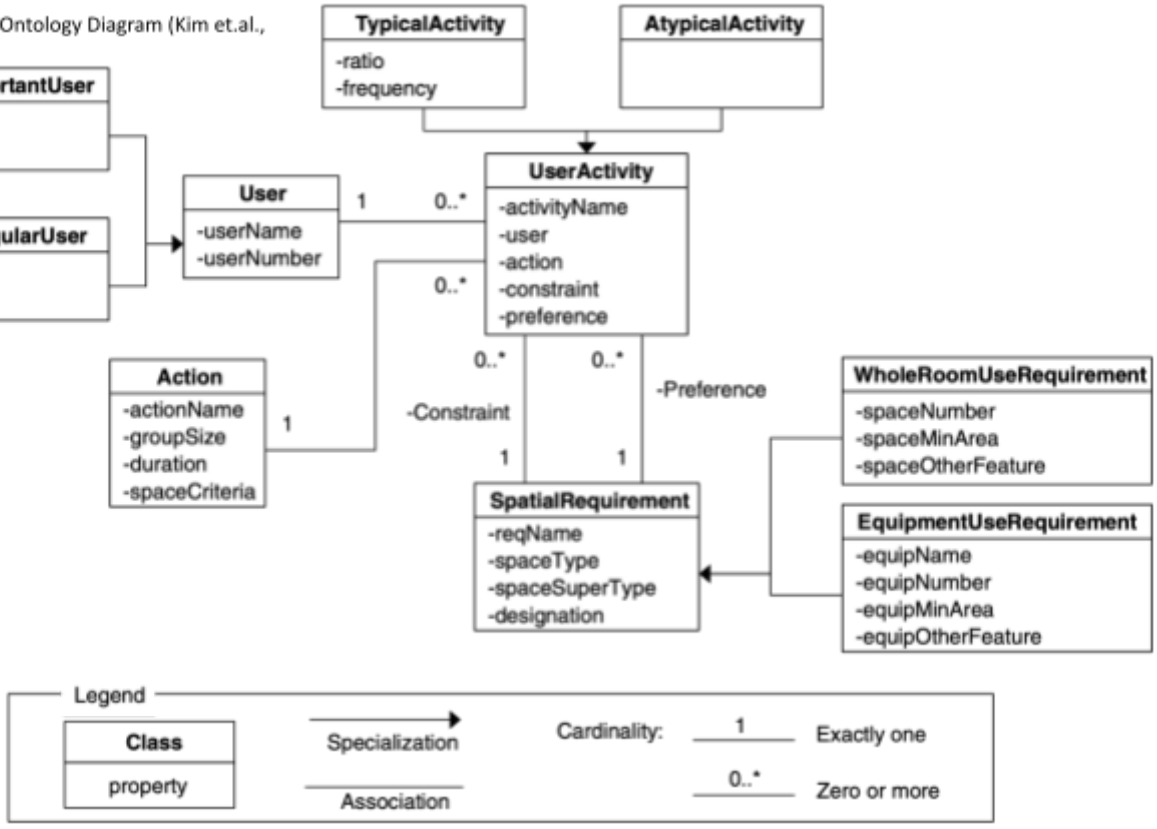
and work coordination can be still facilitated within the open office plans being used as the case study.

"Action" is a description of what is being performed by the worker and is assumed to be a single action without a workflow. "Actions" are broken down into name, group size, duration, and action space size requirements. Where action space size is the minimum space needed for the action per person times the group size. This thesis works with hybrid work in an open office plan which requires work to be flexible, transportable, and for all workers to be well equipped to function. For that reason this framework considers all actions and workflows to be possible at any individual workstation outside of work meetings and coordination, which are handled within conference room spaces.

"Constraints" under SUA are defined as the minimum space requirement of the worker activity. Defining whether it is a task that requires an entire room or a portion of the room where a specific type of equipment is held. Constraints are broken down into a requirement name, space type, space supertype, and designation. "Whole Room" requirements additionally contain a space number, space minimum area, and space other features, while equipment constraints contain its name, number, minimum area, and feature. Space type is the label the space holds, and space supertype is the representation of the space. For this thesis, the space type is either an individual desk space or a conference room marked as 0 and 1 respectively. While the space supertype is the unique number associated with the space and would be its list index number. Within these open office plan contexts, constraints are assumed to be uniform for all tasks and satisfied by all individual spaces.

Preference is described as the spatial requirement for better performance of an activity. This spatial requirement is categorized as having satisfied all needs and classifications of the constraint category. With activity constraints being satisfied by the two different space types this thesis will consider better performance to be what workers' preferred space quality is based on both their feedback and the questionnaire. Both the ratio of people involved in a task and the frequency of a task will not be considered or used within this framework's sort agent, however, it provides crucial information in creating an agent for space recommendation. Overall this framework is great at describing the use of space and creating a framework for both a sort agent and space recommender.

Table 2: Ontology Diagram (Kim et.al., 2014)



The above table describes the ontology presented by Kim et.al. and is a defining framework for establishing the Office's context and Sort Agents' roles. Again however, this thesis begins at post occupancy and this framework is only used in establishing assumptions of the space used within a hybrid office. Despite the ontologies goal being to share how a given space is being occupied, this ontology when applied to hybrid work fails to account for the portability of work that has arisen from the rise of laptops and portability of the internet. For that reason the tasks required can be assumed to be possible in any space due to its portable nature and the aforementioned assumptions to be used in this thesis.

2.5 Computer Decision-making and Previous Space Allocation Algorithms:

The Space Allocation Problem (labeled SAP) can be described as another form of the common computational problem known as the Bin Packing Problem (Rui et.al., 2010). The Bin Packing Problem is a space optimization problem that has different volumes and sizes of spheres that need to be put into the lowest number of bins possible without overflowing the bins. One solution to the problem is to order the spheres by their weights and place them into the first open bin that can fit them until all spheres are sorted. This solution is known as the Best Fit Decreasing (GeeksforGeeks, 2023) and will be used in some form in the Sort Agents' logic.

Using this solution the Best Fit Random Sort Agent will use a linear process that gives no forward thought. To develop a more thoughtful sorting algorithm it's important to understand first how computers make decisions. At any given moment between enacting an office schedule decision an individual and a computer must evaluate the possible outcomes. An individual

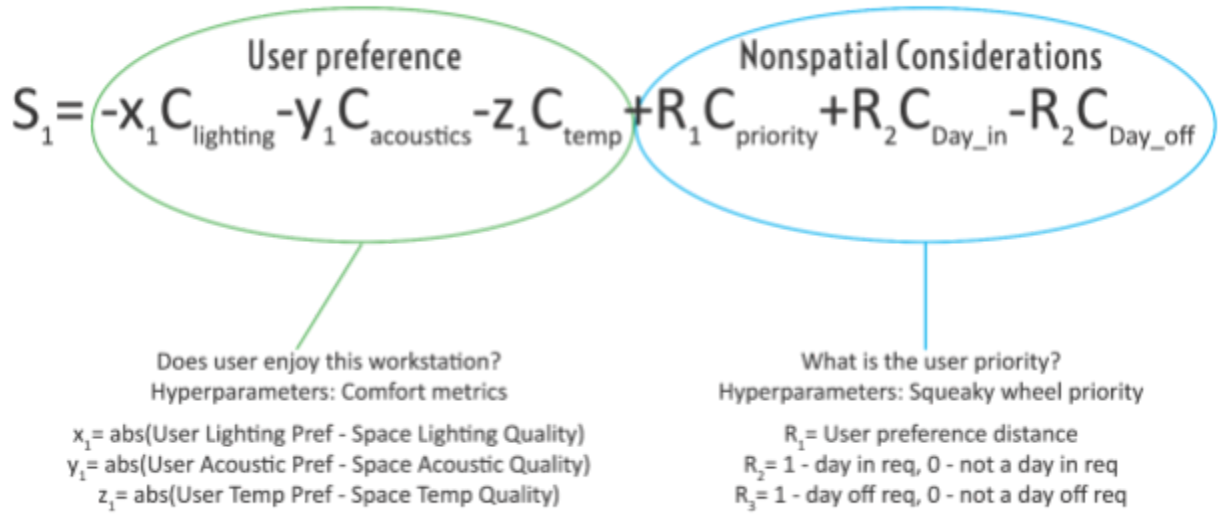
would consider many things in this context; what days workers have requested in, what days workers have requested off, what spaces are available, etc. A computer, however, does not have an arbitrary understanding of the value of these decisions. Instead, a computer will need to know what moves are possible and what the desired outcomes are. This is communicated in the form of hard and soft constraints. A hard constraint is any rule that must be followed for an action to be considered possible. While a soft constraint is a desired outcome that is not required for a state to be considered legal but will be rewarded or penalized depending on the desired outcome. A hard constraint for example would be that workers cannot be scheduled to be present in the office on a vacation day. A soft constraint would be that we want to see workers satisfied visually, thermally, and acoustically. All hard and soft constraints used in this thesis will be described further in Methodology: Sort Agent. To evaluate the quality of a possible decision a computer uses an equation that rewards and penalizes the soft constraints while only evaluating decisions that follow the hard constraints. This equation or static evaluation is defined by the creator and gives the computer a method of scoring the value of each possible decision. A Static Evaluation (Table 3) will include a variable signifying a metric or consideration to make a decision on and a coefficient that weighs the importance of that decision. The larger the coefficient the more important that consideration or variable is to making a decision. The computer will take all these variables times their coefficients and sum them together to get a score for a possible scheduling decision. If a tenant manager wants to make decisions that have workers more likely to be sorted into the office on the days they have requested in, the program would assign a higher value coefficient to the “Day In” variable. Meaning that when the computer finds a decision where a worker can be sorted on a day they have requested in versus

a day they have not, the day they have requested it will be a higher value decision comparatively. Conversely, if a tenant manager wants to avoid sorting workers on days they have requested off then the program would enact a penalty in the equation which would be a negative coefficient.

The Sorting Agents' static evaluation (Table 3) will include 2 reward variables; one for user priority and another for fulfilling a "Day In" request. User priority is a heuristic (a method of estimation) that describes the total dissatisfaction felt by a worker from the previous week's office schedule. How this heuristic is measured will be described further in the Methodology: Feedback Agent section below. However, using this priority method, also referred to as the Squeaky Wheel Algorithm (Tanimoto, 2007), the agents will be able to democratize and share the more desirable workstations. "Day In" requests will be a binary variable either being fulfilled and receiving a variable value of 1 or will not be fulfilled and receiving a variable value of 0. Using a binary the Sorting Agents will either access the coefficient and its reward or will miss it and not be rewarded for its decision. These Sort Agents' static evaluations will also include 4 penalties: one for scheduling a worker on their day off, and the other three being the distance from the worker's preference a workstation is for lighting, acoustics, and temperature. "Day Off" requests will follow the same binary used for the "Day In" variable, while "User Preference Distance" is the numeric difference between a worker's spatial preference and the workstation's spatial quality. This "User Preference Distance" heuristic is further described in the

Methodology: Feedback Agent section below.

Table 3: Proposed Static Evaluation break down



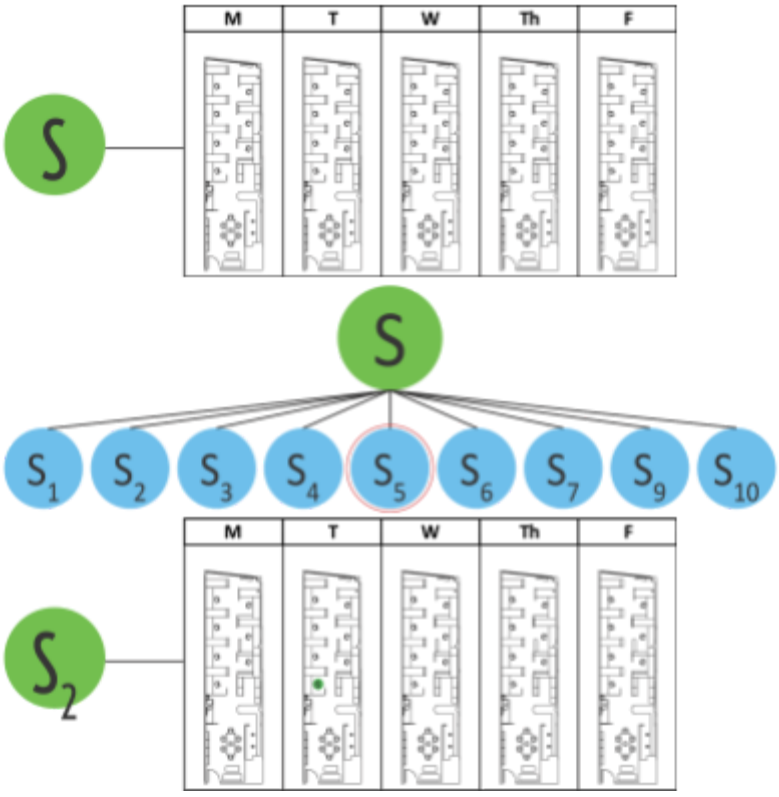
Writing an equation in different manners will yield different results as the computer will be lead to prioritize different decisions. For this reason, many methods have been developed to ascertain the most optimal weights or level of importance of a given static evaluation's variables. Methods such as Genetic Algorithms (GA) which are a metaheuristic that creates a population of agents with various weights. After testing their efficiency the GA takes the top-performing agents and mixes and matches their weights to create offspring in a similar fashion to animals. After multiple iterations or generations, the top-performing agent is selected. GA has been used in the past to optimize space allocation algorithms (Rui et.al., 2010). These algorithms however have not yet been applied to hybrid-style office spaces.

While this thesis does not use a method for optimizing the weights of the Sort Agents it does give consideration to it in Methodology: Sort Agents and Future Research. This is because while weights are important the agents value all three spatial needs equally since there isn't a real-world data set that would point to one being more important than another to worker

satisfaction/comfort. The Agents, however, do want to prioritize comfort over “Day In”/”Day Off” requests and the reward/penalty is valued at a lower weight. Instead of weight optimization, this thesis will test whether the Sort Agents can outperform the Random Agent. Since there isn’t a data set linked to real office data, beating the Random Agent will show the efficiency of using a forward-searching method and comfort vs a space optimization problem solution method.

The benefit of using an algorithm to schedule an office is beyond just having an efficient quantitative means of decision-making. A computer can also evaluate more decisions within a shorter amount of time than a person. Consider the start point of an empty office schedule to be a state, meaning no one has been scheduled into any workspace on any day yet. All possible first decisions evaluated would be its subsequent possible states while the possible state that is ultimately selected is the next state to evaluate possibilities from (Table 4).

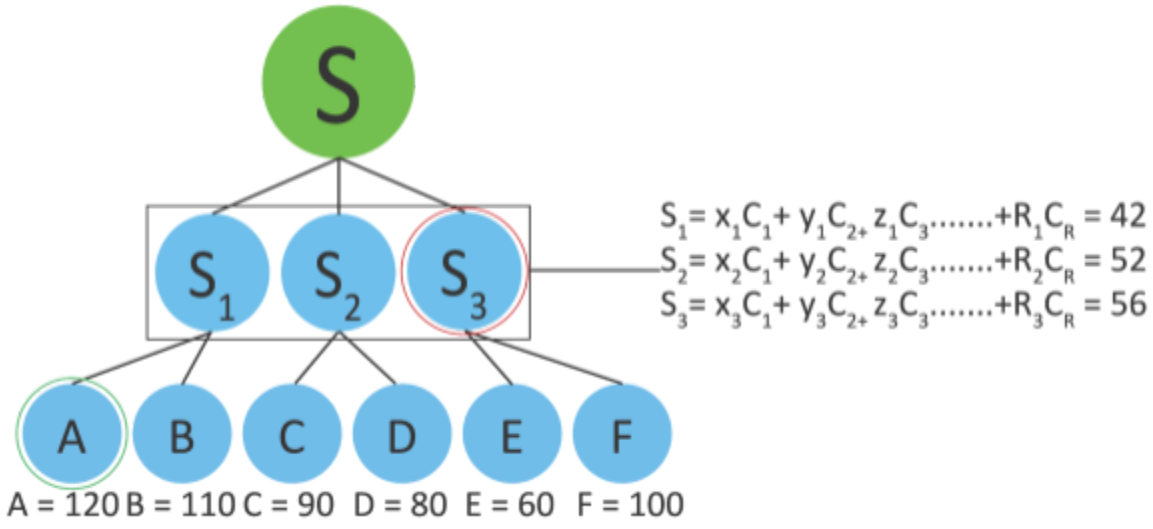
Table 4: Breadth-first Search



In this context scheduling one person into a single space on a single day would be considered the next state. For every person, workday, and workstation available the amount of possible states goes up immensely. Given that every workstation, on any given work day, can have any office worker the amount of possibilities is enormous. If there were 20 workers in a single office with 10 spaces the number of possible states for a single decision would be 20 (workers) X 10 (workstations) X 5 (days in a work week) = 1,000 possible decisions/states. An algorithm can search through these 1,000 possible states and choose the best one or the highest-valued decision much faster than any human. Checking all the first possible moves is known as a Breadth-first search and is a state search method commonly known in computer science. Evaluating all first possible moves (Breadth-first) and selecting the best solution is

known as a Greedy Algorithm (Table 5). This process of decision-making however can lead to nonoptimal solutions as the highest-scoring move might not lead down a decision tree that leads to the most optimal solutions (Table 5).

Table 5: Greedy Algorithm



To illustrate this point further consider a game of chess where a player's queen is in danger of being taken the next turn by an enemy pawn but our pawn can take the enemy's bishop. A Greedy Algorithm would take the bishop because it did not consider the next move where the player loses their queen, rather than the first move where the best choice is to take the bishop. This would lead to the player losing their queen and missing the optimal path for victory (Table 6).

table 6: Chess Greedy Algorithm Example (White's Move)

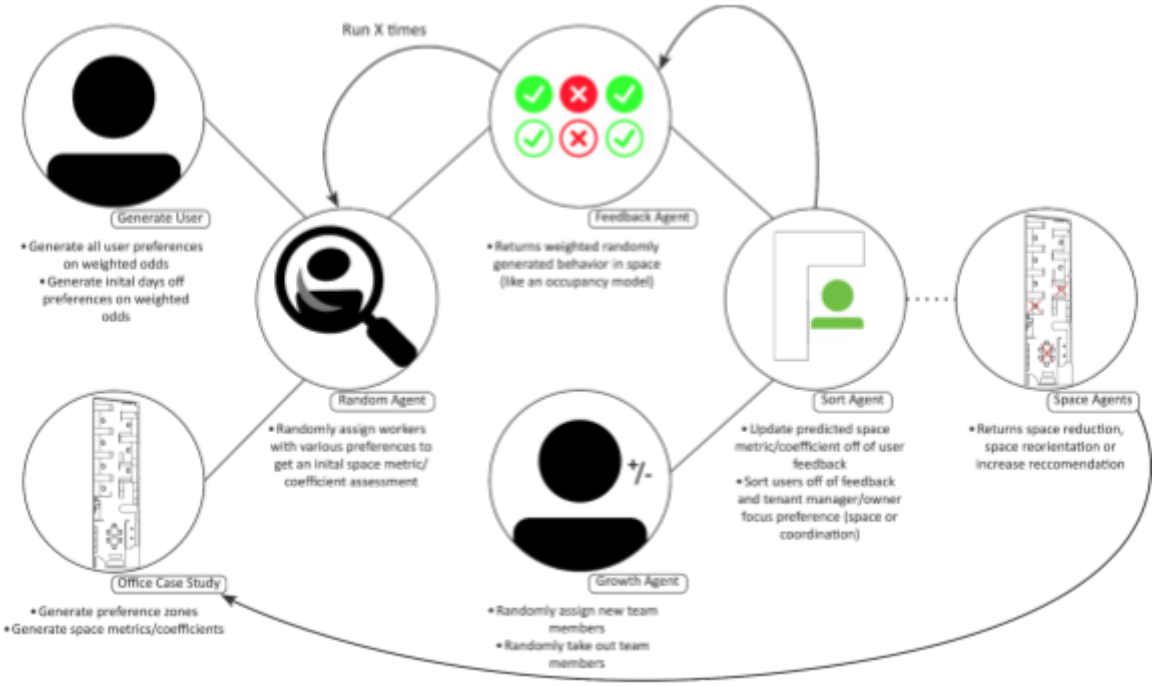
(Black's Move)

Now consider an algorithm that knows every possible move and always selects the ones where it is the winner at the end of the game. This search method of exploring all the way down to the end and returning to search for the next possibility all the way down is known as a depth-first search (Table 7). This method while very strong takes an immense amount of computation to successfully complete. This expensive cost is extremely inefficient and a new method of state exploration that uses a mix of these two methods is needed. In this thesis, the Sort Agents will combine the practice of the Best Fit Decreasing solution with a deep and wide state search method. This is discussed more in-depth in Methodology: Sort Agents.

3. Methodology

3.0 Computational Framework: (Table 7)

Table 7: Proposed Methodology



To run the algorithm developed in this thesis the program will need 3 sets of inputs. The first input is the number of weeks the agents will sort for, the second input is the number of workers to create, and the last input is the weighted odds set of each worker feature in the form of a Python dictionary. This dictionary is composed of a key and a value where the key is the outcome and the value is the odds of the outcome occurring. With these three inputs, the algorithm first runs the Worker Generator algorithm to create the office workers and their preferences. Afterward, the data created is sent to the Random Agent along with the office case study. From there the Random Agent will sort the workers using a method described in Method: Random Agent. After creating the office schedule the Random Agent sends the office schedule,

worker data, and true office data (operator ascribed spatial data of the office that represents the real world values of the space) to the Feedback Agent for worker feedback to be generated. The Feedback Agent will then read the worker's preference and compare it to the workstation's true spatial quality and return whether they enjoyed the space and what aspects they were satisfied or dissatisfied with. Afterward, the Feedback Agent will update the algorithm's understanding of the workstations' spatial quality and the Random Agent will be its next sort. After running this loop between the Random Agent and the Feedback Agent for the specified number of weeks the Feedback Agent will forward the spatial metrics of the office it has ascertained to the Averaged Sort Agent. The Averaged Sort Agent will then follow the same loop as the Feedback Agent that the Random Agent followed for the specified number of weeks. All sortings and the resulting feedback will be stored and visualized at the end of all sortings and shared with architects, tenant managers, tenant owners, designers, or consultants who will make a design revision for the space and the process will begin again. If the office invests more in its space and installs sensors for exact spatial measurements then a sort agent can use this exact measurement to make better decisions than a sort agent that uses spatial estimations from the Feedback Agent. For this reason, this methodology proposes two agents the Averaged Sort Agent, which uses spatial metrics returned from the Feedback Agent that uses a process described in Methodology: Feedback Agent to understand the space, and the Measured Sort Agent which simulates the use of sensors to ascertain the spatial qualities of the space and uses the true office spatial values. In this alternate context, the computational process this thesis follows is the same but the Measured Sort Agent replaces the Averaged Sort Agent.

3.1 Worker Generator

To begin this computational process, a worker data set is created using the Worker Generator Algorithm (WG). Worker data sets are stored as a Pandas Dataframe (2-dimensional labeled data structure with columns) where a behavior is a column, a row is an individual worker, and the cell that they intersect at is where an individual's behavior/feature is stored. This Algorithm is given the number of desired workers to generate, as well as a dictionary for each feature/behavior that contains the possible outcomes and their individual likelihood of occurring. For example, the number of days a worker is contracted for is assigned by the Dictionary*: {1: 20, 2: 20, 3: 20, 4: 20, 5: 20}*, where each worker has a ⅕ chance of being contracted for anywhere between 1-5 in-person work days, with the remaining work week being remote. These probabilities can be adjusted to generate a preferred office context with different worker behaviors or preferences. As another example, putting a higher probability in assigning a worker to a group should correlate to more workers needing conference rooms and will allow us to see how the future sorting agents make a recommendation for this type of collaborative office. After receiving this dictionary of behavioral odds WG will take the first possible outcome and store its probability of occurring in a list. From there WG will take the next possible outcome's probability and sum the previous outcome's probability and add it to the list. This creates a range for each outcome and once WG has gone through all outcomes a random number is selected from 1 to the total sum of all outcomes. With this number, WG will find the range that it falls into on the ordered list and finally, with this position, the correlated outcome from the dictionary is selected and stored in the proper cell. Using the "In Office Days"

example again, Table 8 visualizes the process of random selection using Dictionary: *{1: 15, 2: 20, 3: 40, 4: 20, 5: 5}*.

Table 8: Worker Feature Assignment

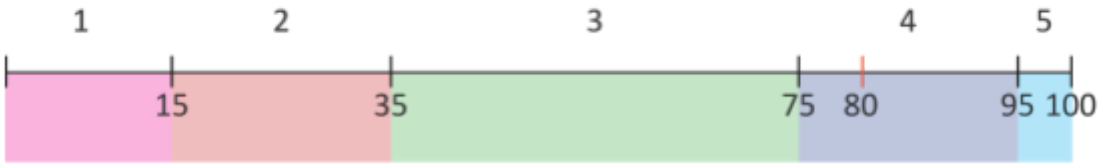
contracted_days:

{1: 15, 2: 20, 3: 40, 4: 20, 5: 5}

random_number selected: 80

[15, 35, 75, 95, 100]

outcome = 4



WG generates two types of behaviours Weekly and Non-weekly behaviours. Weekly behaviors include “Conference Room” requests, “Days In” requests, “Days Off” requests, and “Vacation Days”. Non-weekly behaviours include “In Office Days”, “Group Assignments”, and all spatial preferences. A static behavior that is worker specific and will remain static is “Picky/Stoic” which is stored as a number from 1-3 that defines how picky a worker is to having their spatial preference and will be described more in Methodology: Feedback Agent. “In Office Days” assigns a number 1-5 for the number of days in a workweek an employee comes in and tells future agents how many days a worker should be sorted into the office. “Group Assignment” assigns a group number 0-X where X is the total number of groups an operator decides to give by the dictionary input they create. A nonzero number tells the algorithm what

group a worker is in and if a worker has a group number of 0 this individual has no group assignment and should never be assigned to a conference workspace. This creates groups of coworkers who either need to coordinate in a meeting or can be more frequently placed near each other in individual workspaces. If an employee is placed into a group they must have 1 conference room request. This requirement ensures that all individuals in a group will have at least 1 meeting a week. The number of “Conference Room” requests is equivalent to the number of times that person must be in a group meeting. A group can still meet without all members, so each member in a group may have a different number of “Conference Room” requests. However, these “Conference Room” requests cannot exceed the number of “In Office Days”. To create variation in the number of “Conference Room” requests between workers, a binary check is run for each day except for 1, an employee is contracted in the office. This minus 1 is to account for the required 1 meeting a week criteria already established. A binary check (Table 9) will use the same practice of generation as the other worker behaviors/features, however, will generate an outcome of 1 or 0 instead. If the binary check is passed meaning a 1 is selected instead of a 0, then an additional 1 is added to the “Conference Room” requests. If a 0 is randomly selected then nothing is added to the “Conference Room” requests. This “Conference Room Request” behavior, therefore, is generated first by whether or not an employee is in a group (based on Group Assignment), then by running the binary checks for the established amount of time, and finally storing the final number of requests into the worker’s Pandas cell.

Next, the WG randomly generates a worker's preferences assigning numbers for “Lighting Preference”, “Volume Preference”, and “Temperature Preference”. Both “Lighting

Preference" and "Volume Preference" are generated by randomly assigning a number from 1-10 with 1 representing a desire for lower volume spaces or lower lighting levels and 10 being a willingness to be in loud or bright spaces. Since this project uses Fahrenheit, "Temperature Preference" is generated in a range of 68-75. All use the same random odds structure in assigning workers' preferences.

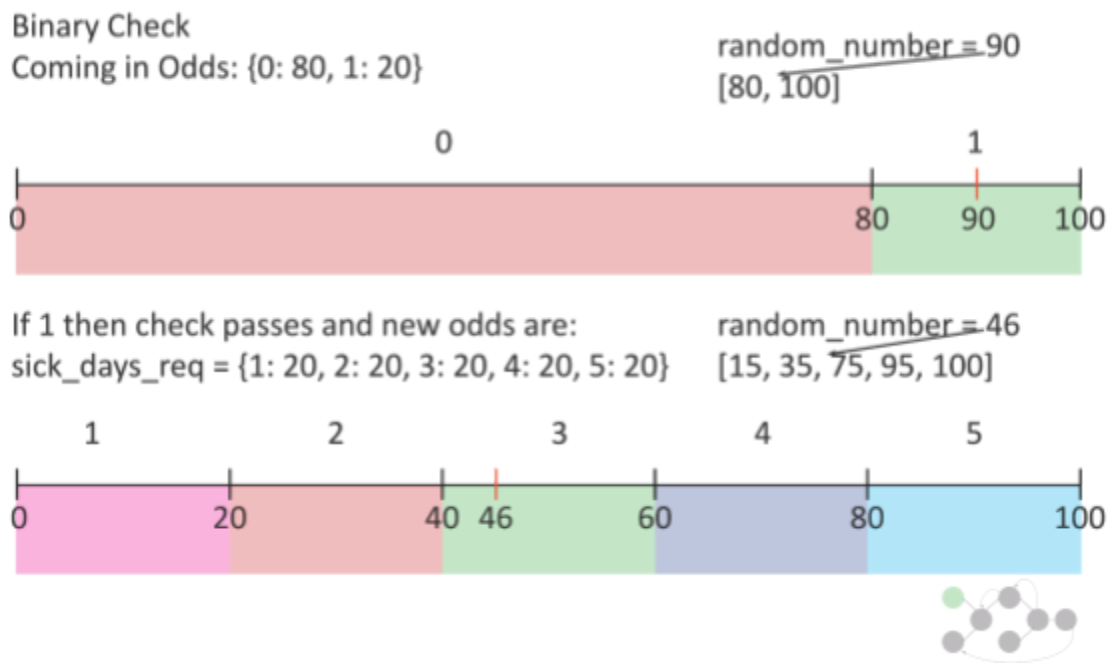
Finally, worker workweek preferences are generated randomly by running a binary check into randomly assigning a number of days that behavior exists, and finally randomly selecting exact days for that behavior. An example using "Days off" requests, a request check is run by randomly choosing a number between 1 and 10 and if that number is higher than the associated weight (9 in this example) then another number is selected between 1-X (X being the number of contracted in person days this worker has) to decide how many in-office days are being requested off. Finally, for that many days, a number between 1-5 representing Monday through Friday (M-F) is selected and placed in a Python set to avoid repeating days. Returning this set a binary is stored for each day. Days requested off will store a 0 in the correlating worker day request cell. For example, having a 0 for "M_Off_Request" means that Monday was not requested off, and having a 1 for "Th_Off_Request" means Thursday has been requested off. The same process is run for "Vacation Days" and a binary like in "Days Off" request is created from this set for future recall. "Days In Requests" runs the same process as "Days Off" requests however with a lower threshold to pass meaning the range for having a day in request is wider than a day off request. The "Days In" requests set is subtracted by the "Days Off" requests to avoid having workers request the same day in and off. As an example for the "Vacation Days" request process, consider a binary check with odds of *{0: 80, 1: 20}*, and odds for selecting any

given day as *{1: 20, 2: 20, 3: 20, 4: 20, 5: 20}*. After passing the binary check the number of days called out is selected randomly from 1-5 and the process of selecting a day randomly is run for the selected amount of days (table 10).

3.2 Office Case Study

In order for the agents to make sorting decisions the office space is defined using

Table 10: Worker Feature Assignment With Binary Check



another Pandas Dataframe. This dataframe has a row representing each workspace (individual and conference) and a column for each detail that represents that space. These details include the space type (individual workstation or conference room), the day from M-F (with each day having its own column), and the spatial quality metrics one for each metric (visual, acoustic, and thermal). “Space Type” is defined by a 0 or a 1, with 0 being an individual desk space, and 1 being a conference room. For each conference room space, an additional row is made for each hour of the work day. In this thesis for simplicity's sake, this office space will only allow 2 rows for each conference room space so that two groups can be scheduled in one space on a single

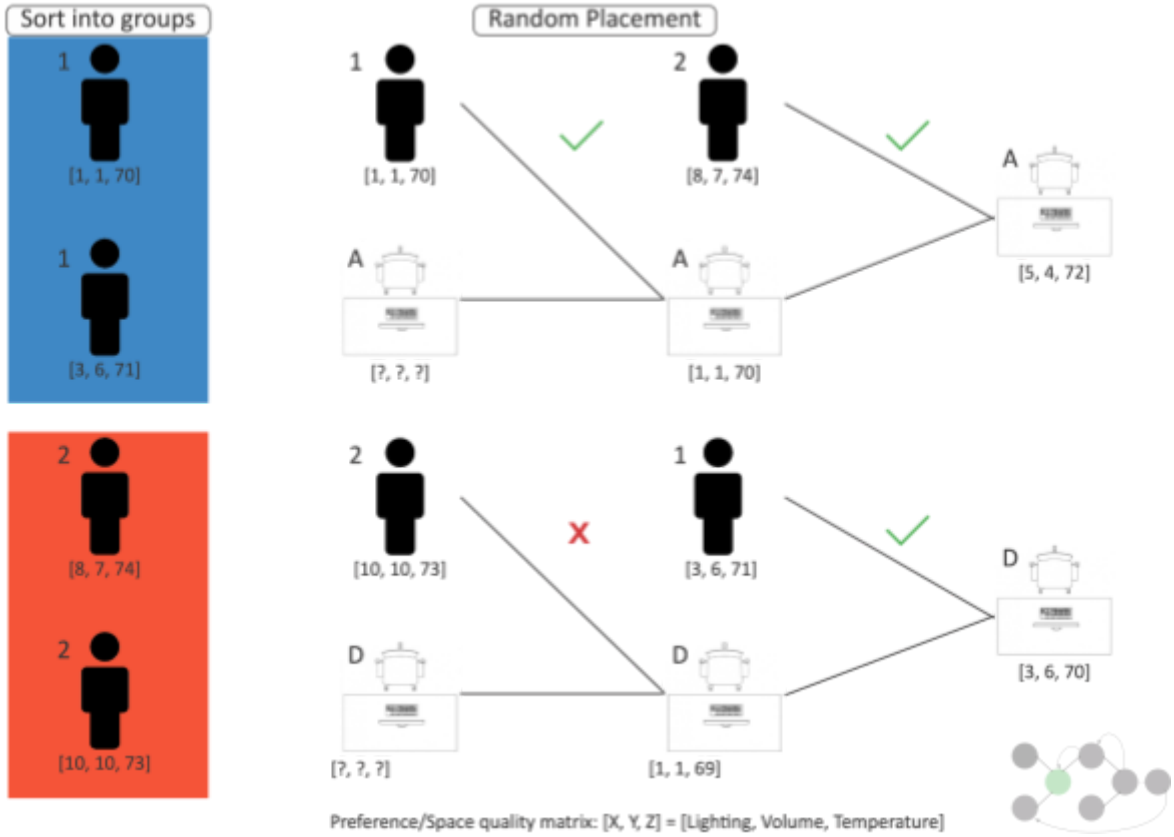
given day. In the future, an individual conference room space can be given a row for each hour of the work day and workers can request how long a meeting should be scheduled for and be placed that amount of times into the conference room rows. Each row/space has a column for lighting, acoustics, and temperature to be updated and called in the future. When a future agent makes a decision on what worker is placed into what space, they will assign an index associated with that worker into a given office cell that corresponds to a space (row) and day (column named after day). When an agent updates the space's quality metrics of a space it will perform the operation on the row representing that space in the column noted for that quality.

3.3 Random Agent

The Random Agent is the first agent to make a sorted schedule and sorts workers in a way that attempts to pinpoint a workstation's spatial qualities. These sorting decisions start first by calling forward the stored office Pandas dataframe that represents the office schedule and spatial qualities. If a dataframe is not given and instead the office case study start command is received (start = True) then the office dataframe used will be the initial start dataframe stored in the algorithm. The returned office dataframe uses the same Seattle office layout with the correct number of rows for workspaces and cells for the days they are available, however, this initial dataframe has no spatial qualities stored in its columns and instead uses a 0 for all spatial qualities across all workspaces. This indicates that no one has ever been assigned to these spaces and no space quality metric has been yet discovered for this space since this is the first run for that office. To create a better understanding of the space the Random Agent when making individual sortings will start by randomly choosing a space quality to order workers on using their correlating preference (table 11). Once ordered two groups are formed, a group on

the higher end of the preference scale and a group on the lower end. Splitting these two groups at the midpoint (5 for acoustics and visual metrics and 71 for thermal) the Random Agent then sorts an individual from one of these groups using criteria discussed in “Individual Sorting” and in the next available day for this workstation, if one is available, sorts a worker from the opposite group. This process gives the Feedback Agent a greater chance at returning at least 1 satisfied worker in a given space and aids in the process of creating an understanding of the spatial quality. With this satisfied worker, the Feedback Agent will average out the spatial preferences they were satisfied with, along with all others who have enjoyed this space in order to define the workstation's spatial qualities. If a worker is dissatisfied with a workstation's spatial quality then it can be assumed that the workstation’s spatial quality is on the opposite end of the scale and the spatial understanding can be updated with a slight skew to account for this dissatisfaction. Before any individual is sorted however the Random Agent will start first by scheduling group meetings.

Table 11: Random Sort Sorting Process For Space Quality Search



Group Sorting:

For this code section, there are two situations where the group sorting search should end. The first is when there is no longer anyone left to be scheduled into a meeting with another person in their group and the second is when there are no more conference room spaces for a meeting to take place. If either of these two conditions are not met there is still space and people to be sorted and the section will continue to run. If one condition is met however, there is no way to sort a group into a space and a check is run to see how many people missed a meeting request and how many meeting requests were missed. This will then skip any further group sorting and share the numbers of these two results. If the sorting ends because it ran out of individuals to sort then neither variable will be stored as they will be 0.

To find if either condition is met, we separate individuals based on their group assignment and check their group length. In code, this is accomplished simply using a Pandas mask that filters workers with a specified worker feature outcome. Using this mask the Random Agent pulls out any worker that has a given group assignment, “In Office Days” above 0, and has a conference room request above 0. After each sorting, an “In Office Days” requirement, and a “Conference Room” request count will be taken off. Eventually, the workers will be removed by the mask and the filtered group dataframe will eventually reach a size of 1 or 0. This length of 1 can exist when a single individual has more requests than anyone else but since they have no one to meet with this request will go unfulfilled. Due to the random nature of the data set in this thesis some groups may have an individual who has requested more meetings than anyone else. For this reason, the group size condition to be met for termination will be, so long as any group is greater than one individual the group sorting will continue. After passing the group size

check the section creates an office dataframe mask that calls the rows that have a space type of 1 (for conference room spaces) and a 0 (an open space) in one of its column days.

Passing the size checks each group that has a size larger than 1 is placed into a list using the “group_rand_choice” section of code also known as a definition. From there “group_rand_choice” randomly selects a group number from the list of existing office groups still unsorted and the chosen group is given an assignment in a conference room space using the “group_sort” definition. If there is only one group to be sorted this will ensure that only they are selected to avoid the program looping on a group that is already sorted. This also ensures that no one group is favored over another.

With this randomly selected group, the group_sort definition sums the “Day In” requests from each group member and orders them from greatest to least. Taking the most common day-in request name the Random Agent assigns it a day index with “M_Day_in” request being 1 and “F_Days_in” being 5 and the most common day is saved as a number variable. Next, the “office_space_mask” that pulls out all conference room spaces with an open day is searched one by one to find if there is an open space on the most commonly requested day. If there is no open space the next most commonly requested day is used. If a day is found then a mask removes group members who have requested that day off and a few checks are run to ensure various possible situations are accounted for. The first check is to ensure that the group with the now removed members is not smaller than 2. From there a check is run to see if a group has not already been assigned that day. If group 3 for example is already assigned on Tuesday, the next most popular day is selected and starts from the beginning of the flag checks again. Next, the individual spaces are checked on that day to ensure that after the meeting there is a space for

everyone scheduled for that meeting that day. If there are not enough spaces for each worker the next most common day is checked from the start. Once all checks are passed the conference room space and individual workspaces are assigned to the group and individuals who are available on that day. Finally, a “Day In” and “Conference Room” request count is removed and the masks for the condition checks are refreshed to remove anyone who has met their weekly meeting requirements. Afterward, the group size and conference room space checks are rerun and the process starts over from random group selection until the previously mentioned group sorting search termination conditions are met. If the section runs through all days with this group and can’t find a space to place them then a count for how many people were not sorted and how many meetings were missed is saved. The amount of meetings missed for this group is counted by taking the 2nd most conference room requested individuals and using their request count. This is to avoid meetings that cannot be made since the most requested person can have a higher count that no one would be able to meet with them on. Once saved all conference room request counts for this group are set to 0 and the group mask is refreshed to empty the group size and meet one of the conditions to end the group sorting.

Individual Sorting:

Once all group sortings are complete individual workers will be sorted into a workspace. For an individual sorting to end similar conditions must be met to end the processes. The first condition is that there are no more individuals who need to be sorted into the space and the second condition is that there are no more spaces in the office for individuals to be sorted into. If either of these conditions are met then the section will terminate. If the termination condition met is that the office ran out of space then a variable is saved for the total number of

individuals who missed a day and the total number of days they missed. To find the open spaces and amount of individuals unsorted a mask for the individual workspaces and individuals not fully sorted is called using definitions “indi_work_space_mask” and “unsorted_individuals”. If either of these conditions has not been met yet however then the sorting request is made.

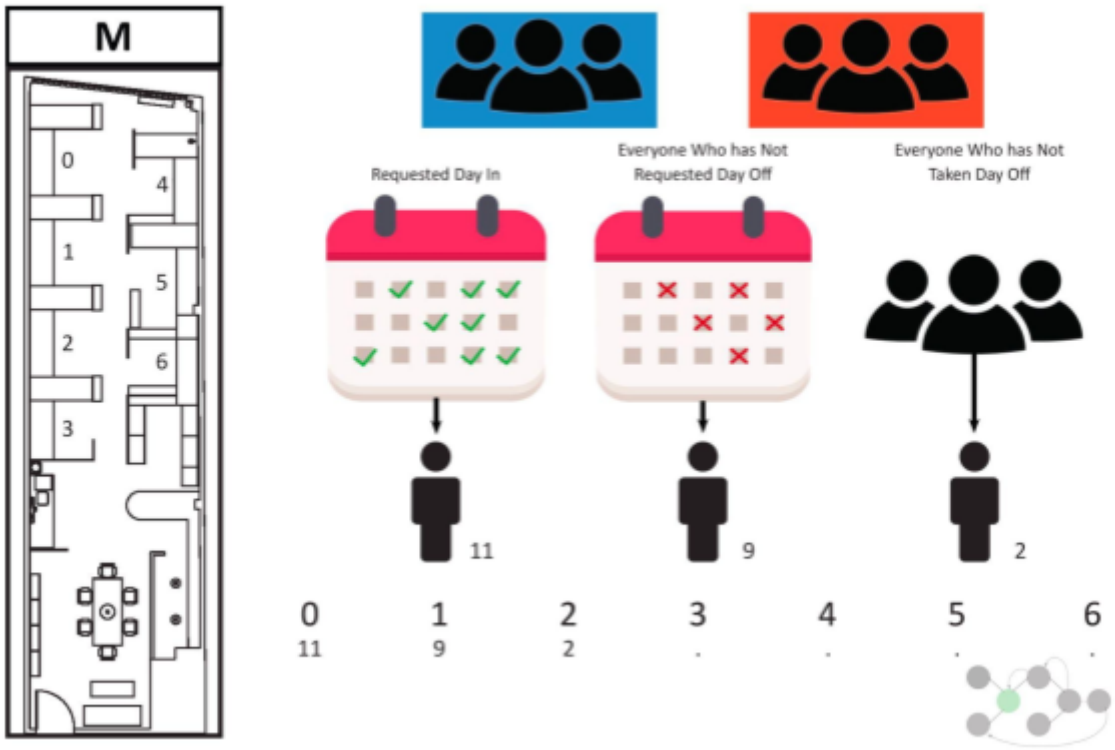
This section begins by separating individuals into two different groups per space quality metric and is separated based on their sensitivity to space quality preferences. The sensitive grouping includes individuals with low acoustic, low temperature, or highlighting preferences. On the opposite end, non-sensitive people include high acoustic, high temperature, or low lighting preferences. Temperature is a space quality metric that is hard to determine sensitivity to because unlike lighting and acoustics being too high or too low might not affect a nonsensitive person, however, preferring a low or high temperature and not receiving it will always affect the individual. For the purpose of this thesis, however, that is not an issue as the groupings are created to use two polar groupings for space-quality exploration only. For this reason, we only need to separate individuals with opposite ends of a preference.

After these groups are made, a space quality to make a sort decision off of is selected randomly. Using a list with an index of each space quality, a number is selected randomly then all groups are refreshed through a filter that takes out those who have been fully sorted before going into the sort definition with the quality variable. The “indi_sort” agent is given the space quality variable and enters the section of code associated with the variable. Calling the open individual workspace mask the program runs through each day looking for the first day the first workspace in the mask is open. The program makes an index variable for the sensitivity group call with an odd number selecting the sensitive group and an even number selecting the

nonsensitive group. The index begins at 0 and adds 1 to each run to alternate between sensitive and nonsensitive calling. When a space quality, open space, empty day, and sensitive grouping are selected the Random Agent checks to ensure the first sensitive group is larger than 0 and has at least one person who is able to be sorted. If it is not, a 1 is added to the “sense_index” and moves on to check the non-sensitive group. If it has then a mask calling the individuals in the sensitive quality group who have requested that “Day In” is created using the “day_in_mask”. This mask also accounts for people who have already been scheduled for that day and removes their index from the mask and consideration. This is accomplished by getting all individuals that have been scheduled that day and all individuals in x group and placing them into separate sets. These sets are subtracted from each other and only the individuals who have not already been scheduled into that space and are a part of x group (x being the sensitive or nonsensitive group following the randomly chosen space quality group) remain. If this returns an empty mask then an additional mask is run to remove all individuals who have requested that day off from the original x group. If no one who has not requested this day off is left on the given day then all workers are considered and one is randomly selected (table 12). This order of priority follows a sudo Best Fit Decreasing method with the highest priority being given to individuals who have requested a specific day in and the lowest individuals who have requested

a specified day off.

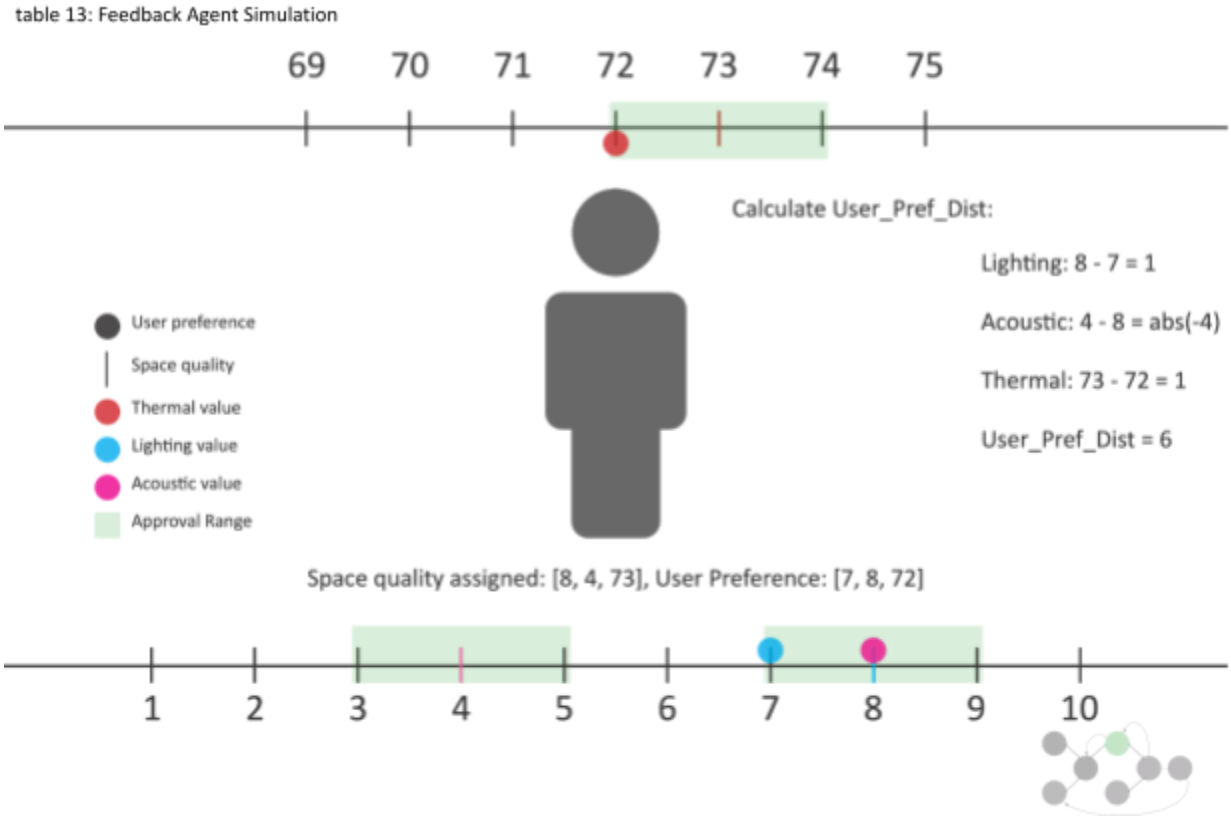
table 12: Best Fit Decreasing Under Random Agent



The selected individual is then sorted into that space on the given day. This is accomplished by replacing 0 with the number representing the worker's index + 1 since Python stores and counts the first individual as 0 and the empty Pandas cell is also stored as 0. Without this addition of 1 the worker at 0 will always be overwritten in the schedule by others since the program seeks out 0 as an empty space to filter by. After this process has been completed then the masks and office space are refreshed and checked for the two previously discussed conditions and the process repeats until they are met. Once a condition has been met the agent returns a dataframe with the office schedule and the feedback agent begins to simulate worker behavior.

3.4 Feedback Agent (table 13)

table 13: Feedback Agent Simulation



To understand the space and review worker satisfaction, the Feedback Agent is used to simulate whether a worker has enjoyed a workspace or not. This agent also helps the Random Agent in creating an understanding of the space by averaging out the quality preferences of all workers who enjoyed a given space. In a real-world application, this feedback is received through a questionnaire that comes up after each visit to the office. In this framework simulating a worker's enjoyment of a workspace is accomplished by first retrieving a true space value. This true value will be a static space quality value for each space and its three qualities on the first prototype. In the future, this value can be rotated on a seasonal, monthly, or even hourly basis. In a real-world application, this value is not needed as it is used to simulate worker satisfaction and workers can simply share that information with us via a questionnaire.

However, this value can replace the previously discussed Feedback Agent if space recorders can be used to have a more exact spatial value each day. This initial static value is written by the code operator and stored in a JSON file within the framework's folder. Using the true space quality value of a given office and its spaces, the agent then iterates through the office schedule and finds the workers placed in a given space on a given day. Once the agent has the space and worker, it compares the true workspace values to the worker's preference values. Taking the difference between the worker preference values and the true space values the algorithm creates on the initial run, a column in the worker data dataframe for the total distance a worker's preference has been from the space's quality. If this column already exists the algorithm instead updates the "User Preference Distance" column by totaling the absolute value of the distance between the true and worker preference values. "User Preference Distance" at the end is the total distance from the user's preference and the true value across all space qualities and all scheduled seating assignments. This variable is used in the future to understand which workers were the most satisfied and will incentive future agents to schedule them after less satisfied workers to create a rotation of user priority.

If the worker is within a range defined by the value +x or - x where x is the "Picky/Stoic" behaviour value then they have approved the space's quality and a binary is also returned. With this, the worker binary is added to a variable totaling the number of workers who enjoyed that space. At the same time, the agent totals all those who enjoyed the spaces' preference values together. These values are then stored in a separate JSON file for future access. Using the totals an average for each quality is returned for the space and is now the space's new quality metric. For a given workspace with index 2, the true values of the workspace's spatial quality metrics

are [8, 7, 72]. A worker with a stoic value of 2 is placed in that space has a preference of [6, 5, 70] and is within a range of 2 for lighting, 3 for acoustics, and 2 for thermal comfort. This would mean that the worker is added to a total of all workers who enjoyed the lighting and temperature but not to the acoustic total count. The worker's preference values will also be added to the total of all lighting and temperature preference values for people who enjoyed the space. These two numbers will then be divided by each other and return the average.

A similar process takes place for the conference rooms however, everyone in the meetings is checked and averaged rather than just a single individual per day. For conference rooms, the agent finds what group is sorted into a conference space and checks who in that group was assigned a seat on that day. Since workers have to be scheduled in an individual space to be in a meeting and some people can be in a group but not scheduled for that meeting, this process ensures that only workers who are in that group and assigned to that meeting on that day are sampled for feedback. Once the workers of the group and space are found then the same process of averaging out the workers who enjoyed that space begins. This simple agent will then return the new associated space quality metrics for the Averaged Sort Agent to make decisions on.

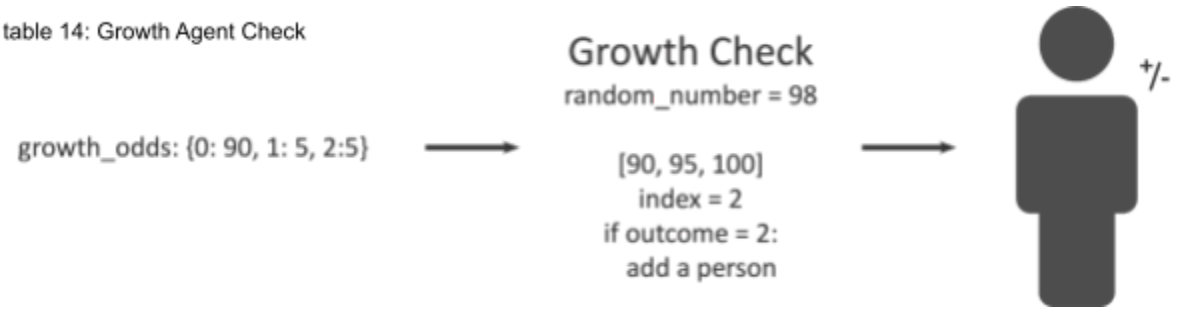
After all worker feedback is generated the Feedback Agent must reassign weekly behaviours. This process is the same as the UG algorithm however replaces what behaviour was established in the previous week. Non-weekly behaviours are not meant to be purely static as these behaviours are still subject to change but are only reassigned on occasion. To reassign Non-weekly behaviours the Feedback Agent runs a loop once for each worker in which a number is randomly selected between 1-10, if the randomly selected number is greater than 8

then for the individual worker the Non-weekly behaviours are reassigned. Once worker data has been refreshed the Feedback Agent returns the office dataframe, worker dataframe, and stored satisfaction variables.

3.5 Growth Agent (table 14)

The Growth Agent is used to simulate office growth or decline. In real-world applications this Agent can be called upon to help reduce or grow the office footprint either slowly removing the number of days and amount of workers required to be in the office or slowly increasing till the office reaches an upper limit. The process of simulating office growth is similar to generating worker features. Using a dictionary containing the odds for growth the Growth Agent randomly selects a 0, 1, or 2. If the Growth Agent randomly selects 0 then no growth or decline is simulated. If the Growth Agent randomly selects 1 then an individual is removed from the office roster. If the Growth Agent randomly selects 2 then an individual is added to the office roster.

table 14: Growth Agent Check



3.6 Sort Agents

After establishing an understanding of the space using the Random Agent and Feedback Agent the Averaged Sort Agent can begin sorting. While there are two Agents developed in this thesis they receive different office space quality measurements and provide different outcomes. Still, their decision-making processes are the exact same and are interchangeable in this section.

Much like the Random Agent the Sort Agents starts by sorting office groups and their conferences then moves onto the individual worker sortings.

Group Sorting:

When sorting groups the Sort Agent will continue to search until one of two conditions are met. The two conditions are the same as the Random Agent with the first condition being that there must be a group with more than one individual in it to be sorted or there must be an open conference room and enough individual workspaces for a group to be sorted. The difference between the Random Agent and the Sort Agents however are their drastically different approach to the making of sorting decisions.

While the Random Agent randomly selects a group to be placed on a given day the Sort Agents compare potential decisions before selecting one. This process of decision making utilizes a static evaluation with soft constraints focused on the size of the group, the number of group members who have requested this day in, and satisfying workers spatial quality preferences. This static evaluation has two rewards one for the size of the group and the other for the number of group members who have requested the potential sorted day in. The static evaluation also has a penalty for each spatial quality taking the difference between the worker's preference and the conference room's spatial quality (either measured or ascertained by worker feedback depending on the agent being used) and multiplying it by the penalty. If the worker's preference matches the conference room's spatial quality then the resulting 0 ensures no penalty is enacted and in this way the lower the penalty the closer to a satisfied sorting the Sort Agents will be.

Before the Sort Agents can use the static evaluator to evaluate any possible state the Sort Agents need to know what are the hard constraints. For group sortings the Sort Agents cannot assign individuals into conference rooms if they are not a part of the group, or if they do not have any more meeting requests. The Sort Agent cannot sort a group into a meeting room if they have already been sorted on that day. The Sort Agents also cannot consider a state if the group meeting does not have more than 1 person involved. Finally the Sort Agents cannot schedule a meeting for a group if there are not enough open individual workspaces for the workers to move to before and after their meeting.

To improve the depth and number of possible states the Sorting Agents can iterate through to find a solution, the Sorting Agents runs a depth-first search on the first possible group sorting move. From here the Sorting Agents store the top 3 states and search for the best solution that follows any one of these decisions. In this way the Sorting Agents can improve upon the process of a Greedy Algorithm by searching deeper while avoiding extreme computational cost that would come from searching all first possible states deeper. After the best solution following the top 3 states is found the Sorting Agent returns to the parent state that it stemmed from and sorts that group into the conference room evaluated. To decide what worker gets what individual work space a “Best Fit Decreasing” solution is run with the highest priority individual to sort being the one with the highest “User Preference Distance”.

Individual Sorting:

The individual sorting process uses a static evaluation that has soft constraints for the user priority, “Day In” requests, “Day Off” requests, and comfort metrics. The static evaluation utilizes 2 rewards one for user priority and another for fulfilling “Day In” requests. The static

evaluation also utilizes 4 penalties 1 for each spatial quality, and one for sorting in a worker who has requested the specified day off. The Sort Agents use the same process of calculating the spatial quality penalty as the group section using the difference between the worker's preference and workspace spatial quality distance. The hard constraints used for individual sortings require that the Sort Agents does not include states that have a worker come in on a vacation day, has a worker sorted into the office twice on a single day, or has a worker coming in more days than they are contracted for. Table 15 shares all hard and soft constraints for both the group and individual sortings.

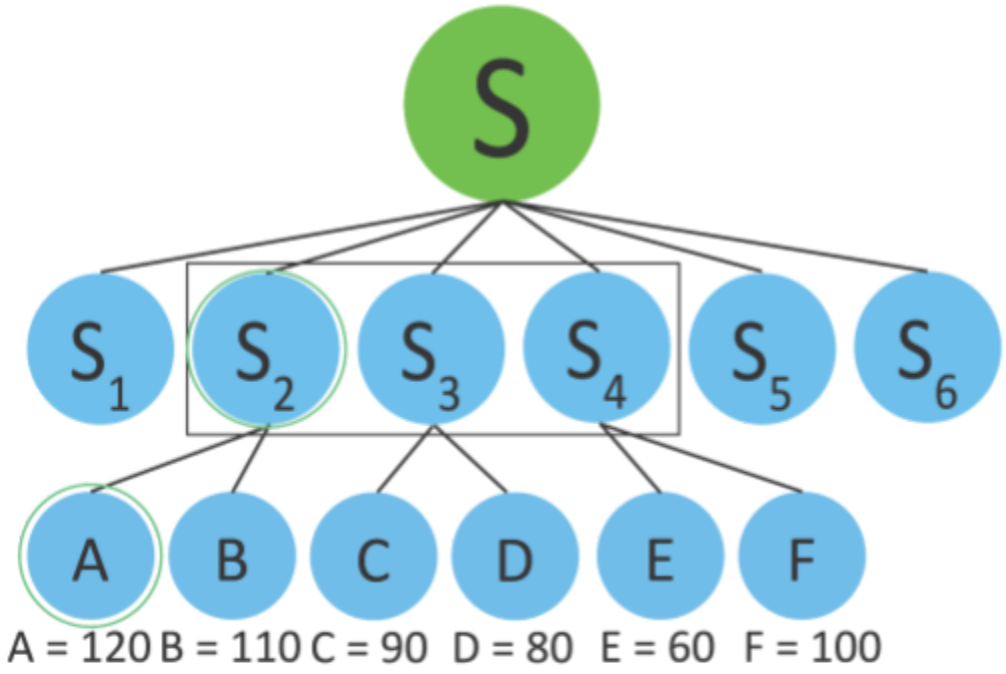
table 15: Hard & Soft Constraints For Sort Agents

Hard Constraint Rules to follow	Soft Constraint Desired outcomes
• Worker can't be assigned a day requested off • Worker can't be assigned more days than contracted • Worker can't be assigned a meeting room if not in a group • Worker can't be assigned to a meeting that isn't part of their group • Worker can't be assigned to a meeting if out of conference room requests	• Place a worker into a workstation that provides visual comfort. • Place a worker into a workstation that provides thermal comfort. • Place a worker into a workstation that provides acoustic comfort. • Prioritize workers who enjoyed the space the least last week.

Using the "User Preference Distance" value we can prioritize sorting individuals who were less satisfied with their space assignment in the last week's sorting. For the first sorting using the Sort Agent the least satisfied worker will be based on the last results of the Random Agent's feedback. Again using the Best Fit Decreasing solution the Sort Agents order the

workers based on their “User Preference Distance” however instead of selecting the highest priority worker the Sort Agents will select the top 3 highest priority workers. This is to help widen the state search for a more optimal solution without requiring the Agents to use up too much computation. With the top 3 priority workers the Sort Agents perform a breadth-first search selecting the top 3 states and searching all the possible outcomes that stem from these top 3 states. From there the best state is found and the parent state it stems from is selected as the decision (table 16).

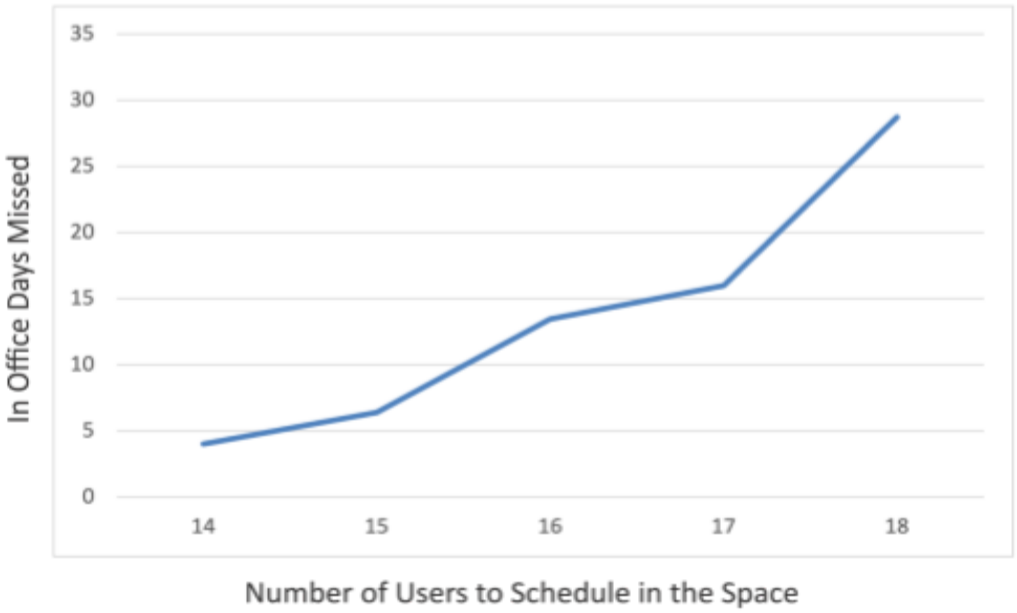
table 16: Top Three Best Fit Decreasing Sort Agent



4. Results

A major component to the efficiency of any office scheduling system is the number of workers the Agent is required to accommodate and how many “In Office Days” these workers are required to fulfill. Too many individual workers with too many “In Office Days” could result in not enough workspaces being available for everyone and many workers may miss in-office work days. Too few workers could result in under-utilized office spaces and would require very little need for an office scheduling system. For these reasons when testing the proposed office sorting process the number of workers and “In Office Days'' created should not be too few or too many. The number of workers and “In Office Days” that would be the most effective at showing the efficiency of the Sort Agents, is one in which after using the Random Sort agent (who has no forward thought) still has workers miss work days while the office is not completely utilized. In this context the missed days can be from poor scheduling decisions rather than an over scheduled office. This will be the perfect opportunity to see how the forward thinking agents perform. After running various office sizes the selected office size was found to be 15 workers (table 17) with a “In Office Days”odds dictionary of *{1: 15, 2: 20, 3: 40, 4: 20, 5: 5}* making it likely that workers will need to come in for 2-4 office work days a week.

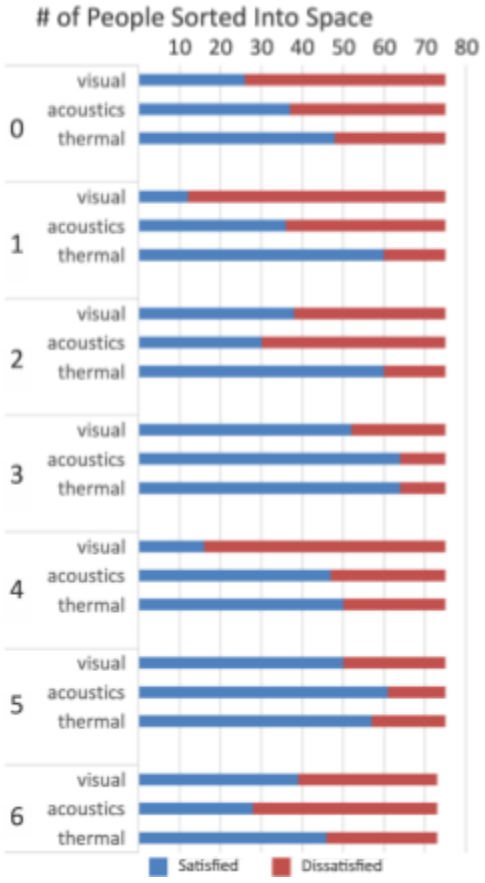
table 17: Missed Days to Number of Workers Under Random Agent



The Agents' efficiency will be evaluated by how often it filled out the workspace to capacity, how many days its workers missed, how many workers had days they missed, and what was the total dissatisfaction felt per week. These Sort Agents were ran for 15 weeks at 5 days a week making the maximum capacity of a given space in this context 75 individual sortings (15 weeks * 5 days). At the same time the following dictionary is used to generate worker space preference: Thermal - *{68: 2, 69: 15, 70: 15, 71: 18, 72: 25, 73: 12, 74: 6, 75: 5, 76: 2},* Lighting - *{1: 2, 2: 5, 3: 5, 4: 20, 5: 10, 6: 10, 7: 25, 8: 5, 9: 5, 10: 13}, Acoustic - {1: 10, 2: 5, 3: 5, 4: 20, 5: 10, 6: 10, 7: 25, 8: 5, 9: 5, 10: 5}.* Using this dictionary we would expect spaces that have temperatures from 71-73, spaces with a lighting value from 4-7, and spaces with an acoustic range from 4-7 to be preferred by most workers. When evaluating space use we would expect these spaces to have a high degree of satisfaction if the Sort Agents are working efficiently and recognizing the established pattern.

The Random Agent had many spaces fully utilized with only one space not being fully scheduled (table 18). Using the Random Agent the office schedule was able to avoid many contexts where workers required an office space but could not be scheduled into the only available spaces. This issue can potentially occur when a Sort Agent does not make proper decisions and is left with workers who can't be scheduled either due to a vacation day, or because they had already been sorted that day. With this promising space utilization the Random Agent was also able to fulfill 70% of sorting requests on average per week. To understand the dissatisfaction felt the "User Preference Distance" is utilized again and the sum of all worker "User Preference Distance" is stored per week. With 15 workers needing to share a single office space the Random Agent provided on average a total of 307.21 "User Preference Distance" per week. As shown in table 17 the Random Agent had an average of 7 missed "In Office Days'' on average. Of the 7 days missed 5 workers were responsible for them. Meaning that 5 workers had at least 1 or 2 missed "In Office Days''. The Random Agent was able to utilize most of the space having sorted most space to their max capacity. While many spaces were being utilized under the Random Sort Agents scheduling the percent of satisfaction was low per space per spatial quality (table 18). Of the 21 spatial qualities (3 per space with 7 spaces) 7 of them had less than a 50% satisfaction rate with many others barely achieving a percent above 50%. The Random Agent also mis-placed workers in spaces 1 and 2, both of which would fall into a majority of the workers' preferences. The office context either did not lend to the generated workers preference or the Random Agent who does not consider spatial preference did not make viable decisions.

table 18: Random Agent Satisfaction by Space Results



Space Quality

Visual: 9
Acoustics: 9
Thermal: 74

Visual: 7
Acoustics: 7
Thermal: 73

Visual: 6
Acoustics: 5
Thermal: 72

Visual: 4
Acoustics: 8
Thermal: 72

Visual: 8
Acoustics: 5
Thermal: 74

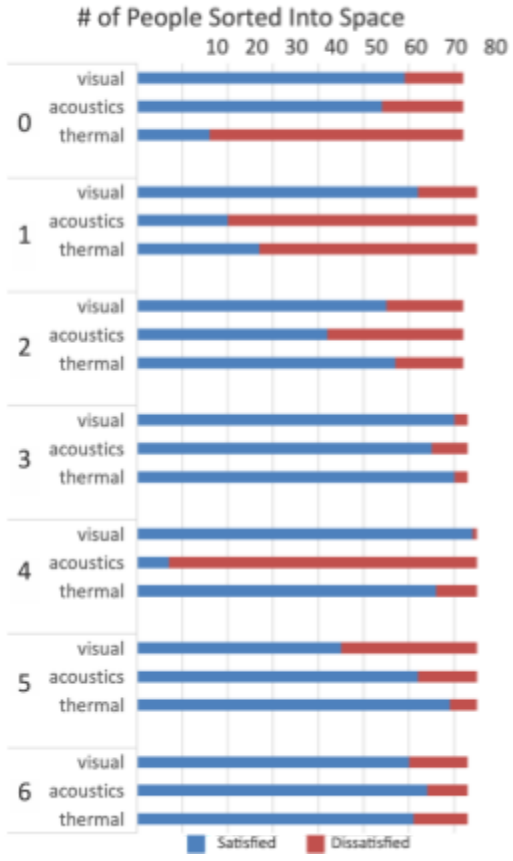
Visual: 6
Acoustics: 4
Thermal: 73

Visual: 5
Acoustics: 3
Thermal: 72

Using the same office context along with the 15 week with 5 day set up the Average Sort Agent had 8 ”In Office Days”missed on average with 4 workers missing out on at least one or three in office workdays. As a product of the additional missed “In Office Days”more spaces did not reach full utilization with spaces 0, 2, and 3 not reaching full utilization by a few missed days each. The Averaged Sort Agent also had an average weekly “User Preference Distance” total of 329.06 having 21.85 more dissatisfaction under the “User Preference Distance” metric than the Random Agent. The Average Sort Agent however satisfied more “Day In” requests with an average of 81% of all “Day In” requests being fulfilled. With a majority of the 21 spatial qualities showing a high level of satisfaction the Average Sort Agent showed a higher number of satisfied workers over the Random Agent despite its higher worker dissatisfaction (table 19). In spaces

that have a majority of the workers spatial preference the Average Sort Agent only failed to hit a high degree of satisfaction in a single spatial quality. This would indicate that the generated workers did not fit into the majority of all spatial quality preferences but rather fit into 2 of the 3 and the reward from these 2 qualities out weighed the penalty from the single non majority preference

table 19: Average Sort Agent Satisfaction by Space Results



Space Quality

Visual: 9
Acoustics: 9
Thermal: 74

Visual: 7
Acoustics: 7
Thermal: 73

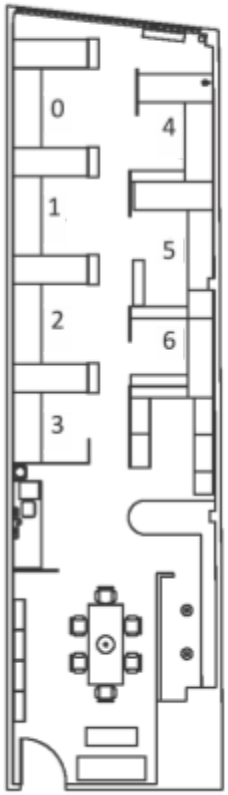
Visual: 6
Acoustics: 5
Thermal: 72

Visual: 4
Acoustics: 8
Thermal: 72

Visual: 8
Acoustics: 5
Thermal: 74

Visual: 6
Acoustics: 4
Thermal: 73

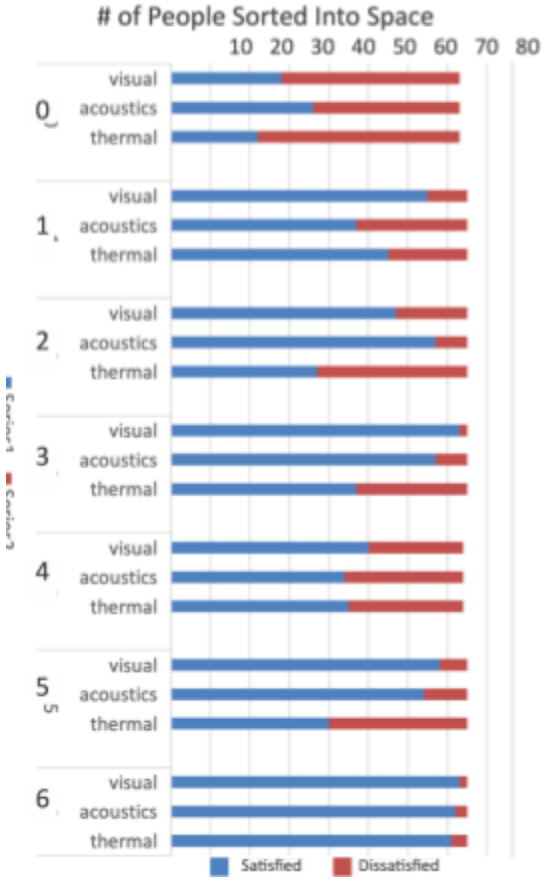
Visual: 5
Acoustics: 3
Thermal: 72



Replacing the Averaged Sort Agent with the Measured Sort Agent and using the same office context the Measured Sort Agent had 9 ”In Office Days” missed on average with 5 workers missing out on at least one or three in office workdays. With more missed “In Office Days” a few spaces did not reach full utilization with spaces 0, and 1 not reaching full utilization by a few days each. The Measured Sort Agent also had an average weekly “User Preference

Distance” total of 276.32 having 30.89 less dissatisfaction under the “User Preference Distance” metric than the Random Agent. The Measured Sort Agent was able to satisfy slightly more “Day In” requests with an average of 82% of all “Day In” requests being fulfilled. With a majority of the 21 spatial qualities showing a high level of satisfaction the Measured Sort Agent showed a more consistent sorting than the Averaged Sort Agent and a higher number of satisfied workers than the Random Agent (table 20).

table 20: Measured Sort Agent Satisfaction by Space Results



Space Quality

Visual: 9
Acoustics: 9
Thermal: 74

Visual: 7
Acoustics: 7
Thermal: 73

Visual: 6
Acoustics: 5
Thermal: 72

Visual: 4
Acoustics: 8
Thermal: 72

Visual: 8
Acoustics: 5
Thermal: 74

Visual: 6
Acoustics: 4
Thermal: 73

Visual: 5
Acoustics: 3
Thermal: 72

5. Conclusions

The purpose of this thesis is to propose and explore an office sorting algorithm that will streamline the labor time it takes to organize a hybrid office's operations while providing a greater worker experience based on comfort to improve worker satisfaction in a hybrid office. With the use of this algorithm an understanding of the space's use and overall opinion shared by the workers can be utilized by a design team or tenant manager to make design changes to improve the workspace or to replicate well received design decisions across other hybrid workspaces. The agents developed were tested using a simulation across 15 work weeks composed of 5 work days each.

The Random Agent performed the best of the three agents in utilizing the space provided and had a single office space that was not fully utilized. The Random Agent however did not produce a large amount of satisfied workers in these decisions. The "User Preference Distance" the Random Agent creates was not the worst performing Agent with the Averaged Sort Agent underperforming the Random Agent.

The Average Sort Agent also underperformed in utilizing the office having more spaces missing full utilization and an additional missed day on average over the Random Agent. The additional day paired with fewer number of workers missing these days show the Averaged Sort Agent might be over prioritizing the high priority workers and leaving many low priority workers left out. The Average sort Agent also saw a higher percentage of satisfaction despite the higher levels of dissatisfaction. This conflict in metrics can be explained by the way worker satisfaction is simulated; because the "User Preference Distance" uses the sum total of the distance from a perfect spatial preference sorting, the dissatisfied workers can be further away under the

Averaged Sort Agent than the Random Agent. So, while the Averaged Sort Agent had more satisfied workers the dissatisfaction felt from the workers who were not satisfied was much more polarizing than in the Random Agent.

Finally the Measured Sort Agent which uses the same methods as the Average Sort but with the true spatial values used for worker feedback to simulate the use of sensors to ascertain spatial metrics. The Measured Sort Agent provided the highest degree of satisfaction for both the percentage of satisfied workers and the least amount of dissatisfaction felt. The Measured Sort agent also utilized the space very well with a majority of the spaces being completely utilized or nearly fully utilized. While the Measured Agent performs well across all defined metrics of efficiency the Random and Average Sort Agents outperform the Measured Sort Agent in very specific contexts. The Random Agent performed the best in space utilization only missing full utilization from 1 workspace. With the Average Sort Agent's use of spatial quality feedback the understanding developed has a degree of error that makes office sortings slightly less efficient, though making feedback more insightful by identifying more specific issues. In many runs of the program the Average Sort Agent had great satisfaction across 2 of the 3 spatial qualities however struggled with the final metric. This struggle helps pinpoint a more exact issue with a space that the Measured Agent might miss due to its less polarizing nature. For designers and tenant managers using the Sort Agent might yield a better form of trouble shooting different spaces despite being less efficient. Still the Measured Agents broad success provides a good argument for more specific metric use within a static evaluation and provides greater reason to explore the efficiency of state search algorithms to tackle space utilization problems.

6. Opportunities for further research:

While this thesis touches on many broad topics to create a product to tackle a specific purpose, more computational methods could be integrated into the computational framework used to make a more robust system. Integrating more specific spatial metrics in the future could provide a better insight into worker satisfaction and increase the number of variables or dimensionality involved in a static evaluation. Additionally the use of coefficient optimization methods could greatly improve a high dimensional (many variables) static evaluation. A method to reduce the high dimensionality while also performing a coefficient optimization known as a linear regression would allow designers to pinpoint the exact space quality metrics that correlate more closely to defining worker satisfaction/comfort. This method would require a real life data set but will improve the proposed algorithm's static evaluation by properly weighing the importance of each metric in defining satisfaction and reducing the number of variables required. Outside of coefficient optimizations future research can improve the computation's state search methods. Using multiprocessing methods in python could save computation time and create a more efficient algorithm. The application of multiprocessing would not only save time but improve the number of states an algorithm can search to find the optimal solution. To improve state search more individuals ordered by the Best Fit Decreasing Solution could be selected than just the top 3 if time allows it. Running multiple weekly schedules with a simulation aspect slows down the amount of time we have for sorting decisions and in a real world application agents can search even deeper and wider. The application of real world data would be a very fruitful method of improving and exploring an algorithm like this further.

7. Bibliography

Adewumi, A.O., and M.M. Ali. “A Multi-Level Genetic Algorithm for a Multi-Stage Space Allocation Problem.” Mathematical and Computer Modelling 51, no. 1–2 (2010): 109–26. https://doi.org/10.1016/j.mcm.2009.09.004.

Al Horr, Y., Arif, M., Kaushik, A., Mazroei, A., Katafygiotou, M., & Elsarrag, E. (2016). Occupant productivity and office indoor environment quality: A review of the literature. Building and Environment, 105, 369-389.

Amasuomo, Tamaraukuro, and Japo Amasuomo. “Perceived Thermal Discomfort and Stress Behaviours Affecting Students’ Learning in Lecture Theatres in the Humid Tropics.” Buildings 6, no. 2 (2016): 18. https://doi.org/10.3390/buildings6020018.

Asadi, Somayeh, Ehsan Mostavi, Djamel Boussaa, and Madhavi Indraganti. “Building Energy Model Calibration Using Automated Optimization-Based Algorithm.” Energy and Buildings 198 (2019): 106–14. https://doi.org/10.1016/j.enbuild.2019.06.001.

Basnet, N. (2023, January 8). *More than 2,000 businesses left downtown Seattle during pandemic, USPS says*. Bizjournals.com. https://www.bizjournals.com/seattle/news/2023/01/08/downtown-seattle-lost-thousands-of-businesses.html#:~:text=According%20to%20USPS%2C%2015%2C919%20businesses,making%20address%20change%20requests%20citywide.

Boubekri, Mohamed, Jaewook Lee, Piers MacNaughton, May Woo, Lauren Schuyler, Brandon Tinianov, and Usha Satish. “The Impact of Optimized Daylight and Views on the Sleep Duration and Cognitive Performance of Office Workers.” International Journal of Environmental Research and Public Health 17, no. 9 (2020): 3219. https://doi.org/10.3390/ijerph17093219.

Chen, Xing Bin, and Tae Wan Kim. “Automated Mapping of User Activities onto Flexible Space in Space-Use Analysis.” Journal of Construction Engineering and Management 143, no. 8 (2017). https://doi.org/10.1061/(asce)co.1943-7862.0001328.

Chong, Adrian, Godfried Augenbroe, and Da Yan. “Occupancy Data at Different Spatial Resolutions: Building Energy Performance and Model Calibration.” Applied Energy 286 (2021): 116492. https://doi.org/10.1016/j.apenergy.2021.116492.

Coenen, Marja, and Robert A.W. Kok. “Workplace Flexibility and New Product Development Performance: The Role of Telework and Flexible Work Schedules.” European Management Journal 32, no. 4 (2014): 564–76. https://doi.org/10.1016/j.emj.2013.12.003.

Dario Landa-Silva, Edmund K. Burke, (2007) Asynchronous Cooperative Local Search for the Office-Space-Allocation Problem. INFORMS Journal on Computing 19(4):575-587. https://doi.org/10.1287/ijoc.1060.0200

Dehnavi, Soltani Elham. Meek, Christopher. Johnson, Brian. 2019. “X-Maps: A Computational Method for Space Planning Using Multi-Variate Occupant Comfort.” Masters diss., University of Washington

Dunatchik, A., Gerson, K., Glass, J., Jacobs, J. A., & Stritzel, H. (2021). Gender, parenting, and the rise of remote work during the pandemic: Implications for domestic inequality in the United States. *Gender & Society*, *35*(2), 194–205. https://doi.org/10.1177/08912432211001301

E. G. Coffman, M. R. Garey, and D. S. Johnson, “Approximation algorithms for bin packing: A survey,” pp. 46–93, 1997.

Ferreira, R.; Pereira, R.; Bianchi, I.S.; da Silva, M.M. Decision Factors for Remote Work Adoption: Advantages, Disadvantages, Driving Forces and Challenges. J. Open Innov. Technol. Mark. Complex. 2021, 7, 70. https://doi.org/10.3390/joitmc 7010070

Galanti, Teresa, Gloria Guidetti, Elisabetta Mazzei, Salvatore Zappalà, and Ferdinando Toscano. “Work from Home during the COVID-19 Outbreak.” Journal of Occupational & Environmental Medicine 63, no. 7 (2021). https://doi.org/10.1097/jom.0000000000002236.

GeeksforGeeks. (2023, April 25). *Bin packing problem (minimize number of used bins)*. GeeksforGeeks. https://www.geeksforgeeks.org/bin-packing-problem-minimize-number-of-used-bins/

Glean, Aldo A., Stanley D. Gatland, and Ihab Elzeyadi. “Visualization of Acoustic Comfort in an Open-Plan, High-Performance Glass Building.” Buildings 12, no. 3 (2022): 338. https://doi.org/10.3390/buildings12030338.

Hao, K. (2020, April 2). *Ai is sending people to jail-and getting it wrong*. MIT Technology Review. https://www.technologyreview.com/2019/01/21/137783/algorithms-criminal-justice-ai/

Hockey, G. Robert, and Fiona Earle. “Control over the Scheduling of Simulated Office Work Reduces the Impact of Workload on Mental Fatigue and Task Performance.” Journal of Experimental Psychology: Applied 12, no. 1 (2006): 50–65. https://doi.org/10.1037/1076-898x.12.1.50.

Kastner, Patrick & Dogan, Timur. (2020). Predicting space usage by multi-objective assessment of outdoor thermal comfort around a university campus.

Kim, Tae Wan, and Martin Fischer. “Automated Generation of User Activity–Space Pairs in Space-Use Analysis.” Journal of Construction Engineering and Management 140, no. 5 (2014). https://doi.org/10.1061/(asce)co.1943-7862.0000839.

Kim, Tae Wan, and Martin Fischer. “Ontology for Representing Building Users’ Activities in Space-Use Analysis.” Journal of Construction Engineering and Management 140, no. 8 (2014). https://doi.org/10.1061/(asce)co.1943-7862.0000881.

Kim, Tae Wan, Youngchul Kim, Seung Hyun Cha, and Martin Fischer. “Automated Updating of Space Design Requirements Connecting User Activities and Space Types.” Automation in Construction 50 (2015): 102–10. https://doi.org/10.1016/j.autcon.2014.12.010.

Kong, Xiaoqiang, Amy Zhang, Xiao Xiao, Subasish Das, and Yunlong Zhang. “Work from Home in the Post-COVID World.” Case Studies on Transport Policy 10, no. 2 (2022): 1118–31. https://doi.org/10.1016/j.cstp.2022.04.002

Kugler, Thomas. “Boma International-Building Owners and Managers Association International.” Building Owners and Managers Association (BOMA) International, June 2022. https://www.boma.org/covidimpact.

Kwon, M., Remøy, H., van den Dobbelsteen, A., & Knaack, U. (2019). Personal control and environmental user satisfaction in office buildings: Results of case studies in the Netherlands. *Building and Environment*, *149*, 428–435. https://doi.org/10.1016/j.buildenv.2018.12.021

Ma, Guofeng, Xue Song, and Shanshan Shang. “BIM-Based Space Management System for Operation and Maintenance Phase in Educational Office Building.” JOURNAL OF CIVIL ENGINEERING AND MANAGEMENT 26, no. 1 (2019): 29–42. https://doi.org/10.3846/jcem.2019.11565.

Pereira, Rui & Cummiskey, Kevin & Kincaid, Rex. (2010). Office space allocation optimization. 2010 IEEE Systems and Information Engineering Design Symposium, SIEDS10. 112 - 117. 10.1109/SIEDS.2010.5469670.

Policy, United States. General Services Administration. Office of Governmentwide. *Innovative Workplaces*. 2006. https://www.gsa.gov/system/files/Innovative_Workplaces-508_R2OD26_0Z5RDZ-i34K-pR.pdf.

Reinhart, Christoph F., John Mardaljevic, and Zack Rogers. "Dynamic Daylight Performance Metrics for Sustainable Building Design." LEUKOS 3, no. 1 (2006): 7–31. https://doi.org/10.1582/leukos.2006.03.01.001.

Rodrigues, Eugénio, Adélio Rodrigues Gaspar, and Álvaro Gomes. "An Approach to the Multi-Level Space Allocation Problem in Architecture Using a Hybrid Evolutionary Technique." Automation in Construction 35 (2013): 482–98. https://doi.org/10.1016/j.autcon.2013.06.005.

Roman, Morean Gisela. "Evaluating Progressive Corporate Workplaces" Brenau University. 2021

Sun, Kaiyu, Da Yan, Tianzhen Hong, and Siyue Guo. "Stochastic Modeling of Overtime Occupancy and Its Application in Building Energy Simulation and Calibration." Building and Environment 79 (2014): 1–12. https://doi.org/10.1016/j.buildenv.2014.04.030.

Tanimoto, Steven. (2007). The Squeaky Wheel Algorithm: Automatic Grouping of Students for Collaborative Projects.

Ulker, Ozge & Landa-Silva, Dario. (2012). Evolutionary local search for solving the office space allocation problem. 2012 IEEE Congress on Evolutionary Computation, CEC 2012. 1-8. 10.1109/CEC.2012.6253009.

Xiong, Jie, Athanasios Tzempelikos, Ilias Bilionis, Nimish M. Awalgaonkar, Seungjae Lee, Iason Konstantzos, Seyed Amir Sadeghi, and Panagiota Karava. "Inferring Personalized Visual Satisfaction Profiles in Daylit Offices from Comparative Preferences Using a Bayesian Approach." Building and Environment 138 (2018): 74–88. https://doi.org/10.1016/j.buildenv.2018.04.022.