Joint Computational Design of Workspaces and Workplans

YONGQI ZHANG, George Mason University, USA
HAIKUN HUANG, George Mason University, USA
ERION PLAKU, George Mason University, USA
LAP-FAI YU, George Mason University, USA

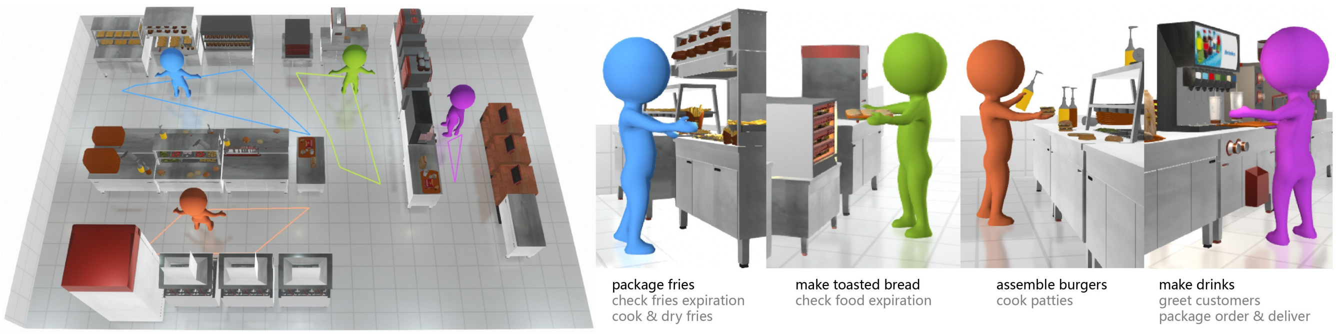


Fig. 1. Given staff properties, a space, and work equipment as input, our approach automatically generates an optimized workspace and workplan.

Humans assume different production roles in a workspace. On one hand, humans design workplans to complete tasks as efficiently as possible in order to improve productivity. On the other hand, a nice workspace is essential to facilitate teamwork. In this way, workspace design and workplan design complement each other. Inspired by such observations, we propose an automatic approach to jointly design a workspace and a workplan. Taking staff properties, a space, and work equipment as input, our approach jointly optimizes a workspace and a workplan, considering performance factors such as time efficiency and congestion avoidance, as well as workload factors such as walk effort, turn effort, and workload balances. To enable exploration of design trade-offs, our approach generates a set of Pareto-optimal design solutions with strengths on different objectives, which can be adopted for different work scenarios. We apply our approach to synthesize workspaces and workplans for different workplaces such as a fast food kitchen and a supermarket. We also extend our approach to incorporate other common work considerations such as dynamic work demands and accommodating staff members with different physical capabilities. Evaluation experiments with simulations validate the efficacy of our approach for synthesizing effective workspaces and workplans.

CCS Concepts: • **Computing methodologies → Graphics systems and interfaces**.

Authors' addresses: Yongqi Zhang, George Mason University, USA, yzhang59@gmu.edu; Haikun Huang, George Mason University, USA, hhuang25@gmu.edu; Erion Plaku, George Mason University, USA, plaku@gmu.edu; Lap-Fai Yu, George Mason University, USA, craigyu@gmu.edu.

0730-0301/2021/12-ART1 $15.00
https://doi.org/10.1145/3478513.3480500

Additional Key Words and Phrases: layout design, computational design

ACM Reference Format:
Yongqi Zhang, Haikun Huang, Erion Plaku, and Lap-Fai Yu. 2021. Joint Computational Design of Workspaces and Workplans. *ACM Trans. Graph.* 40, 228, Article 1 (December 2021), 15 pages. https://doi.org/10.1145/3478513.3480500

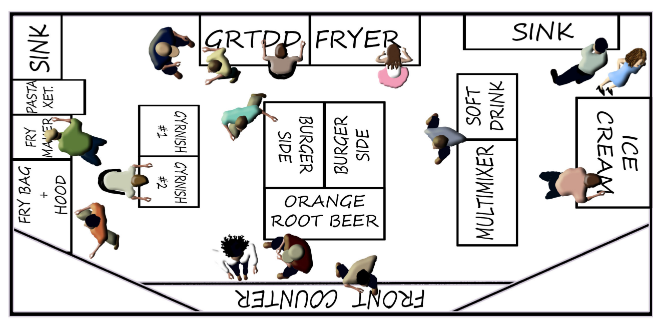


Fig. 2. McDonald brothers' Speedee Service System featured in the film, *The Founder*. On a playground, the staff improvised running a food production pipeline for a fast food kitchen.

1 INTRODUCTION

In the film *The Founder*, the McDonald brothers called on their staff to perform simulations on a playground. Figure 2 depicts this fun experiment: the McDonald brothers envisaged what an optimal food production pipeline for their fast food kitchen should look like and asked their staff to improvise working in the pipeline. Through multiple rounds of trials and errors, they devised the revolutionary "Speedee Service System," which marked the dawn of fast food restaurants. Our work is inspired by this interesting story. Given a space, work equipment, and staff properties, would it be possible to jointly synthesize a workspace and a workplan to optimize work performance and work experience?

The father of skyscrapers, Louis H. Sullivan, coined the phrase "form follows function", meaning that the shape of a building or an object should relate to its intended function. This principle has inspired research on shape modeling [Zhu et al. 2015] and layout design [Fisher et al. 2015; Savva et al. 2016]. Motivated by this principle, we propose a computational design approach to synthesize forms (workspace and workplan) following the functional goals of facilitating teamwork and enhancing overall work performance. Figure 1 illustrates our approach.

Different from residential layout, the ultimate design goal for functional layouts (e.g., workspace) is to support collaborative work. Hence it is crucial for a layout design algorithm to consider human work activities when synthesizing a workspace design, and also to consider how humans can leverage the workspace to collaborate and increase productivity. The arrangement of objects and staff for a functional layout can be challenging since it involves both layout design and logistics considerations. To this end, we formulate their syntheses as an alternating optimization problem: Given an initial workplan, our approach optimizes the layout of the work equipment in the workspace to facilitate collaboration. Given the workspace, our approach optimizes the staff workplan, e.g., their roles and work schedules in the workspace. These two steps are iterated to synthesize an optimized workspace and workplan.

Employee well-being is key to developing workplace resilience. Our approach considers physical wellness of employees at work such as physical endurance and workload balance. Other wellness factors (e.g., work stress) and work considerations (e.g., staff requirements) can be incorporated into our approach in an extension. In this case, our approach adapts the workspace and workplan to cope with the goals. Leveraging the power of optimization and agent-based simulations, our novel approach synthesizes a workspace that is not only visually realistic, but also practical and functional, together with a workplan that informs how humans could leverage the synthesized workspace to produce.

In designing a workspace and a workplan, conflicting objectives (e.g., increasing performance, reducing workload) are common that designers often have to explore trade-offs. To enable design explorations, our alternative optimization approach incorporates Pareto Simulated Annealing (PSA) to generate and keep track of a set of Pareto-optimal design solutions that excel at different objectives. For example, a restaurant may adopt a synthesized design solution that maximizes performance in a peak season, while using another solution that emphasizes workload balance in an off-season. The major contributions of our work include:

- Proposing a novel problem statement of jointly synthesizing a workspace and a workplan to improve collaboration, work performance, and work experience.
- Devising a computational design approach based on alternating optimization to synthesize workspaces and workplans for a variety of common workplaces with practical work considerations.
- Evaluating the efficacy of the synthesized workspaces and workplans through simulations.

2 RELATED WORK

2.1 Computational Layout Design

Layout design is an important area in computer graphics. Researchers have devised generative approaches for synthesizing city layouts [Aliaga et al. 2008; Yang et al. 2013], street layouts [Chen et al. 2008; Peng et al. 2016], architectural layouts [Bao et al. 2013; Wu et al. 2018], etc. We focus on discussing relevant indoor layout synthesis works.

Indoor layout synthesis research has mostly focused on residential layouts. Merrell et al. [2010] used Bayesian networks to encode architectural programs based on residential building layouts generated through an optimization approach. Generative approaches are also devised to synthesize furniture layouts. Yu et al. [2011] and Merrell et al. [2011] optimized furniture arrangement by considering ergonomic factors and interior design guidelines. Recently, researchers leveraged the power of big indoor scene data and deep learning techniques to synthesize furniture layouts. For example, Wang et al. [2018] trained deep convolutional priors for indoor scene synthesis. Ritchie et al. [2019] formulated deep convolutional generative models for fast indoor scene synthesis. Recently, Wang et al. [2019] devised the PlanIT approach for synthesizing indoor scenes using relation graphs and spatial prior networks. Hu et al. [2020] learned floor plan generation using layout graphs. Wu et al. [2019] formulated a data-driven approach to generate floor plans by predicting room and wall locations. While recent research on indoor layout synthesis focuses on the visual realism of furniture arrangement and synthesis efficiency, our approach focuses on synthesizing indoor layouts that support human work activities.

Our work is inspired by the approach of Fu et al. [2017] for synthesizing activity-associated indoor scenes via understanding object relations with activities. Their activity relation graph is learned from potential human positions in the floor plan. In contrast, we consider the work experience and collaboration of workers and simulate work activities via behavior trees. Our approach is also inspired by Feng et al. [2016] for generating mid-scale layout designs (e.g., a shopping mall) considering navigation experience. Their agent activities are predefined and fixed in optimization. In contrast, our approach optimizes agent activities based on their physical fitness and skills in addition to a workplace layout. In summary, our work focuses on generating workspaces that facilitate human work activities and collaboration. In addition, our approach synthesizes a corresponding, compatible workplan that informs people about how to perform tasks in the synthesized workspace.

2.2 Workspace Design

Workspace design is vital to efficient work production. Depending on the workplaces (e.g., offices, warehouses, kitchens, supermarkets), workspace design involves the positioning of machines, instruments, materials, and controls on the production site. In arranging such resources, it is important to consider human factors such as accessibility, comfort, and work ergonomics. Davenport Campbell, an innovative interior design and architecture firm in Sydney, noted that changing the physical workspace would change the way people behave, and that can make all the difference to employee well-being, performance, productivity and organizational

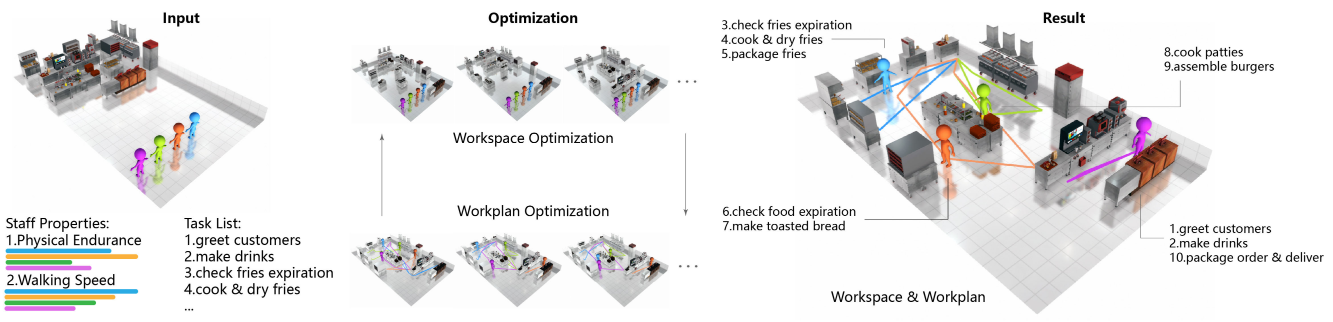


Fig. 3. An overview of our approach.

results [Campbell 2013]. They emphasize the importance of applying human-centered design principles to creating workspaces, such as adapting the workspaces to the individual needs, preferences, and skills of the staff to help them perform. Such principles inspired our computational workspace design approach, which optimizes a workspace by considering individual qualities such as the fitness, work skill, and preferences of the staff.

Human factors and industrial engineering researchers have investigated how workspace design may relate to efficiency and productivity [Brill 1992; Kämpf-Dern and Konkol 2017], workload [Carayon et al. 2003], and workers' health [Broberg 2010; Robertson et al. 2008]. A lean-oriented production layout [Schneider and Ettl 2012] emphasizes a seamless flow of people, material and information, improving workforce morale, efficiency, and production cost effectiveness. Refer to the book by Pizag [2015] for guidelines of creating a thriving workspace. In evaluating the performance of a workspace, our approach also considers common metrics such as time efficiency and obstacle avoidance.

Researchers have also devised computer-aided design (CAD) systems to aid workspace design [Chaffin 2008]. Nagy et al. [2013] developed a system for evaluating work characteristics and providing guidance for designing workstations for an office. Aboulissane et al. [2019] and Shah et al. [2010] developed CAD tools for optimizing the workspaces of parallel robots. Researchers have also applied agent-based simulations for evaluating manufacturing systems [Ruiz et al. 2006] and for predicting workers' behaviors on construction sites [Binhomaid and Hegazy 2020] and warehouses [Pawlewski 2015; Ribino et al. 2018]. Compared to existing tools, our novelty lies on jointly synthesizing a pair of complementary workspace and workplan to optimize different work metrics, informed by agent-based simulations in the loop.

2.3 Task Planning

Graphics and robotics researchers have been working on task planning, which is relevant to workplan design. Bai et al. [2012] introduced a physics-based method for synthesizing concurrent object manipulation tasks for virtual characters. Agrawal and van de Panne [2016] generated task-specific locomotion plans for character animation. Conversely, Ha et al. [2017] jointly optimized the morphology and motion aspects of a robot for a given task. Baykal et al. [2017] optimized the kinematic design of piecewise cylindrical robots to maximize the reachable region in highly constrained settings. Recently, Wang et al. [2020] devised the Scene Mover to perform move planning for scene arrangement using deep reinforcement learning. On the other hand, robotics researchers have tackled task planning using methods such as semantic maps [Galindo et al. 2008] and backward-forward search [Grey et al. 2016]. Researchers have also investigated human-robot interaction in task planning [Alami et al. 2005], and the explicability and predictability of robot task plans [Zhang et al. 2017]. Refer to a survey [Alatartsev et al. 2015] for a review on robot task planning literature. Compared to the previous works, our approach jointly considers the physical capabilities and skills of the staff agents, and the workspace layout, in synthesizing an optimized workplan for task assignments.

Workplan design is also related to production planning [Bitran and Tirupati 1993; Guide Jr 2000; Nishida 1991] and scheduling [Graves 1981] in operations research. The book by Malakooti [2014] discusses the topic in depth. It summarizes the common objectives in design, planning, and control of production systems, which include minimizing costs, risk, use of energy, etc. and maximizing productivity, flexibility, customer satisfaction, employees' job satisfaction, agility, etc. Such discussions informed us of the work performance metrics to incorporate into our workplan design formulation. In addition to jointly synthesizing a workspace and a compatible workplan, our approach also keeps track of a set of Pareto-optimal design solutions with strengths on different objectives (e.g., performance, workload balance), which designers can adopt to cope with the changing demands of a workplace.

3 OVERVIEW

Fast food kitchens are a typical example of functional workspaces in which their indoor layout design involves a substantial amount of individuals' activities and interaction with utilities. By using it as an illustrative example, we explain how our approach synthesizes a workspace and a workplan to facilitate human collaboration to achieve goals such as enhancing work performance. In this example, the equipment objects in the workspace are restored from a McDonald's patent [Ulfig and Venetucci 1997].

Figure 3 illustrates our approach. Given an input space (e.g., a kitchen's space), work equipment (e.g., fryer, drink dispenser, cashier), the staff agents with their properties (e.g., walking speed), and a task list (e.g., make drinks, assemble burgers), our goal is to synthesize an appropriate workspace and a workplan that considers individual work experience and achieve workspace production goals.

Our workplace synthesis is achieved by optimizing a workspace and a workplan against workload costs, performance costs, and user-defined layout costs. The performance costs evaluate the overall work performance associated with the workspace and workplan such as efficiency and collision level. The workload costs evaluate the workload and collaboration experience perceived by the staff agents. The user-specified layout costs encode design priors such as object alignment. We elaborate the cost formulation in Section 6.

The core of our approach is an optimization framework comprising two parts: *workspace optimization* and *workplan optimization* (Section 8). We encode the above considerations as the optimization goals which are evaluated via agent-based simulations of the staff working in the synthesized workspace according to the synthesized workplan. Our approach performs the two optimizations alternatively. In the workspace optimization, our approach modifies the equipment object layout by object arrangement moves. The workspace is iteratively updated until it attains an optimized solution. Then the workplan optimization will take over to optimize the workplan by assigning tasks to the staff. The workspace optimizer then takes the synthesized workplan and further optimizes the workspace. The two optimizations are alternatively applied until convergence.

At each optimization iteration, we perform an agent-based simulation to evaluate the current workspace and workplan. As a typical optimization would require hundreds of iterations, for efficiency, we use a scheduler approach to expedite the agent-based simulations (Section 5 and 7). This approach uses a behavior tree and an A* path-finding technique to execute tasks for agents. At the beginning, our staff agents are assigned with tasks that they need to perform over a period of time. Our scheduler orders and executes tasks in a particular sequence based on the incoming job requests (e.g., customer order requests). The work simulation proceeds efficiently to yield metrics for evaluating the work process.

Through a Pareto simulated annealing process, our approach also keeps track of the Pareto front containing a set of Pareto-optimal solutions with strengths on different work-related criteria, which designers can adopt to set up workspaces and workplans to cope with different work requirements.

4 PROBLEM FORMULATION

4.1 Representation

Figure 3 shows the workspace and workplan arrangement for a fast food kitchen with four staff members, which we use for illustrating our approach. A solution $\phi = \{\mathcal{W}_\mathrm{S}, \mathcal{W}_\mathrm{P}\}$ consists of a workspace $\mathcal{W}_\mathrm{S}$ and a workplan $\mathcal{W}_\mathrm{P}$, as follows:

Workspace. The workspace $\mathcal{W}_\mathrm{S} = \{(\mathbf{p}_i, o_i)\}$ consists of the tuples of the equipment objects, where $\mathbf{p}_i \in \mathbb{R}^2$ refers to the floor position of the center of object i and $o_i \in \{0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}\}$ refers to its orientation. As equipment objects in a workspace are usually regularly oriented, each object is only allowed to rotate by 90 degree for solution search efficiency in the optimization process. Note that not all equipment objects can be accessed from all sides, e.g., the freezer is only accessible at the front.

Workplan. The workplan $\mathcal{W}_\mathrm{P} = \{\tau_i\}$ consists of the sequence τ_i of assigned tasks to each staff member i. Each task is given a task index. For example, $\tau_i = (1, 4, 5)$ means that staff member i is assigned with tasks with indexes 1, 4, and 5, ordered by their priorities (i.e. Task 1's priority is higher than Task 4's). Note that it is possible to assign the same task to multiple staff members.

Table 1. Equipment and tasks of the fast food kitchen restaurant example.

Equipment	**Tasks**
(a) Vegetable fryer	(1) Greet customers
(b) French fry rack	(2) Make drinks
(c) Fries incubator	(3) Check fries expiration
(d) Bun pan rack	(4) Cook & dry fries
(e) Bun grill toaster	(5) Package fries
(f) Freezer	(6) Check food expiration
(g) Grill station	(7) Make toasted bread
(h) Cooked food incubator	(8) Cook patties
(i) Burger-making table	(9) Assemble burgers
(j) Drink & food prep	(10) Package order & deliver
(k) Register	

Table 1 depicts the equipment objects and the tasks of the fast food kitchen. Essentially, each solution ϕ denotes how the equipment objects are arranged in the workspace and the task assignment for each staff member. Based on this specification, an agent-based work simulation is run to evaluate the quality of solution ϕ according to some cost metrics. In the simulation, the staff agents will interact with the equipment objects according to their assigned tasks. For example, when a new customer order comes, a staff member might be in charge of greeting the customer and making drinks, while another staff member might be in charge of cooking and drying fries. Section 7 contains more details. Overall, our approach optimizes the workspace and workplan by updating solution ϕ iteratively.

4.2 Optimization Objective

Our approach aims to optimize a workspace and a workplan to achieve a number of work-related goals. We encode common workplace considerations regarding the performance, workload, and design priors into our optimization framework. Note that our framework can also be extended to encode additional goals and constraints if needed. Our approach synthesizes a solution ϕ by minimizing the following total cost function:

$$C_{\mathrm{Total}}(\phi) = \mathbf{C}_\mathrm{P}\mathbf{w}_\mathrm{P}{}^\mathrm{T} + \mathbf{C}_\mathrm{W}\mathbf{w}_\mathrm{W}{}^\mathrm{T} + \mathbf{C}_\mathrm{L}\mathbf{w}_\mathrm{L}{}^\mathrm{T}, \qquad (1)$$

where $C_\mathrm{P} = [C_{\mathrm{Efficiency}}, C_{\mathrm{Congestion}}, C_{\mathrm{Obstacle}}]$ is a vector of performance costs comprising efficiency, congestion avoidance, and obstacle avoidance cost terms. $\mathbf{w}_\mathrm{P}$ stores the weights of these cost terms. $C_\mathrm{W} = [C_{\mathrm{Walk\ Effort}}, C_{\mathrm{Turn\ Effort}}, C_{\mathrm{Walk\ Balance}}, C_{\mathrm{Turn\ Balance}}]$ is a vector of workload costs comprising effort and workload balance considerations, and $\mathbf{w}_\mathrm{W}$ stores the corresponding weights. $C_\mathrm{L} = [C_{\mathrm{Wall}}, C_{\mathrm{Align}}]$ is a vector of layout costs and $\mathbf{w}_\mathrm{L}$ stores the weights. The layout costs encode the design priors specific to the type of the workspace to be synthesized.

5 STAFF AGENT MODEL

In our agent-based work simulation, we use staff agents to evaluate the object arrangement, individual work experience, and efficiency of production offered by a workspace and a workplan. We use a simple staff agent model to simulate a typical working process in

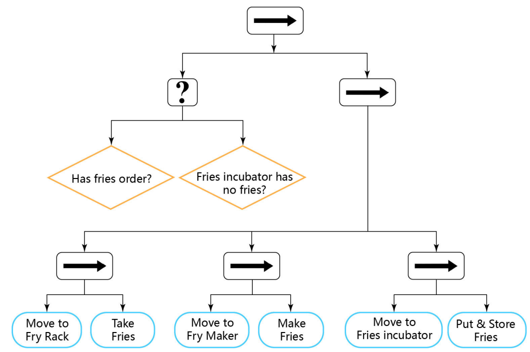


Fig. 4. The behavior tree for the "cook & dry fries" task. Based on the selector node (arrow symbol), a staff agent checks if there is any fries order or if the fries incubator has no fries. If a precondition (in orange) is true, the staff agent executes a sequence of actions (move to fry rack, take fries, etc.) to prepare fries.

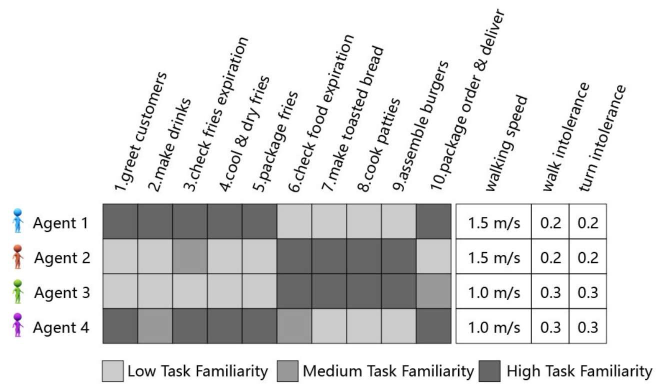


Fig. 5. Properties of the staff agents of the fast food kitchen example.

the workspace. A staff agent represents a human who works at the workspace with the following properties: (a) physical fitness; and (b) work skills (locomotion and task familiarity). These properties help us evaluate individual work experience at the workspace and the overall performance.

Task Execution with Behavior Tree. Each task is associated with a behavior tree, which encodes a series of actions that need to be performed by a staff agent to accomplish the task. Behavior tree is widely used for execution planning, which provides a good extent of flexibility (e.g., executing two different states at once) compared to finite state machine in authoring behaviors for intelligent agents in games. Generally, it comprises selector nodes that describe preconditions and action nodes that denote task destination and execution time. By using control flow nodes, it offers good modularity, scalability, reusability, and flexibility for defining tasks for different workspaces, akin to defining behaviors for non-player characters for computer games or for robots. Refer to [Colledanchise and Ögren 2018] for technical details of behavior trees. The supplementary material contains a general behavior tree structure and several examples with description which can be adopted for modeling different work processes.

Figure 4 shows a behavior tree for the "cook & dry fries" task for the fast food kitchen. To model inter-dependency, we use control flow nodes to create preconditions which are checked before task execution. In this example, two precondition nodes (in orange) are used to determine whether the "cook & dry fries" task can execute. Besides, a staff agent only can execute one task at a time.

5.1 Physical Fitness

Physical Endurance. Physical endurance is a way to measure one's body fitness. As our approach focuses on individual work experience, we define physical endurance from two perspectives: walking and turning. Our approach considers physical endurance in computing the workload costs for the staff. It considers the physical fitness of the staff members in assigning tasks to them. For example, a physically-strong member could be assigned with more physically demanding tasks, which may involve more walking or turning. To model physical endurance, each staff member has a walk intolerance level $\delta^{\text{Walk}} \in [0, 1]$ and a turn intolerance level $\delta^{\text{Turn}} \in [0, 1]$. A high walk (turn) intolerance level means the staff member does not prefer walking (turning) at work.

Locomotion. In general, people's walking speeds depend on their age, gender and height [Crosbie et al. 1997]. Based on walking speed studies [TranSafety 1997], for people whose age is below 65, we set the minimum walking speed as $s_{\text{min}} = 1.0\text{ms}^{-1}$ and the maximum walking speed as $s_{\text{max}} = 1.5\text{ms}^{-1}$. A staff agent's walking speed is:

$$s = s_{\text{min}} + \rho(s_{\text{max}} - s_{\text{min}}), \tag{2}$$

where $\rho \in [0, 1]$ is varied based on the fitness of the person that the staff agent represents. We set ρ to either 0, 0.5 or 1.0 to simulate different walking speeds (slow, normal and fast).

5.2 Work Skill

Task Familiarity. Employers usually assess the skills of their potential staff during the hiring process. Skill development also happens in a workspace after some time of training or working, therefore the staff may possess the same skill set but experienced staff may have higher familiarity for specific tasks which they have worked on for some time. In our running example, our staff agent's familiarity with each task is set to be low, medium or high. Such settings will affect their speed of finishing tasks in the work simulation. Figure 5 shows the physical fitness and task familiarity parameters of the staff agents of the fast food kitchen example.

6 COST TERMS

We discuss the details of the formulation of the costs, namely, performance costs, workload costs, and layout costs, in this section.

6.1 Performance Costs

The performance costs relate to the main work performance offered by a workspace and a workplan. For example, the main work performance of a fast food kitchen is to prepare food to satisfy customers' orders efficiently and to provide a comfortable workspace. The workspace and workplan should be optimized with respect to such performance goals. We consider three performance-related costs: efficiency cost, congestion cost, and obstacle avoidance cost.

$$\begin{aligned}\mathbf{C_P w_P}^T &= w_{\text{Efficiency}} C_{\text{Efficiency}}(\phi) + w_{\text{Congestion}} C_{\text{Congestion}}(\phi) \\ &+ w_{\text{Obstacle}} C_{\text{Obstacle}}(\phi). \end{aligned} \quad (3)$$

Efficiency. We want to optimize the efficiency of serving work orders. Specifically, we formulate a cost to penalize the time needed for completing all work orders:

$$C_{\text{Efficiency}}(\phi) = 1 - \exp(-\frac{\sum_{i=1}^{M} t(i)}{M\sigma_t}), \quad (4)$$

where $t(i)$ is the service time taken to complete work order i computed from our work simulation. A work order is associated with some tasks that need to be accomplished. M denotes the total number of work orders. σ_t refers to the time needed to complete the work order with the maximum anticipated service time. A work order's service time is estimated empirically by summing up the time needed to finish all of its constituent tasks. For the fast food kitchen used as the illustrative example, this cost refers to minimizing the service time for satisfying all the customer meal orders in a simulation. Refer to Section 7 for more details.

Congestion Avoidance. We want to avoid congested locations induced by the workspace and workplan design as congestion results in inconvenience and also safety risks as the staff may bump into each other [Thomas et al. 2006]. As illustrated by Figure 6(a), a location is congested if the paths of multiple staff agents pass through the same location. For example, if multiple staff agents need to pass through the same narrow walkway in a kitchen when executing the workplan, the walkway is regarded as congested. We define a congestion avoidance cost term accordingly:

$$C_{\text{Congestion}}(\phi) = \frac{1}{|L|P} \sum_{l \in L} p(l). \quad (5)$$

In this equation, L refers to the set of locations where congestion is evaluated. In our implementation, we sample these locations in the layout by fitting a grid with a regular interval of 1m. Note that only locations on the free space (i.e. not occupied by an object) are evaluated. For a location $l \in L$, function $p(l)$ returns the number of walking paths that pass through location l's neighborhood (i.e. within 1m of location l) in all work orders. Our approach computes $p(l)$ from the work simulation described in Section 7. P is the maximum number of paths passing through the neighborhood, and is empirically set as the number of agents multiplied by the total number of work orders. This cost term essentially penalizes a workspace and workplan that would result in the staff walking across the same location, hence avoiding congestion.

Obstacle Avoidance. We also want to ensure that the workspace is spacious and uncluttered for smooth navigation and operation. People tend to keep their comfort zones free of obstacles as they walk. For a staff agent in the workspace, we define a staff agent's comfort zone as a circle centered at the agent's position with a radius of 1.219m [Hall 1966].

As the staff agents navigate in the workspace, we want to keep their comfort zones free of obstacles. To achieve this goal, we include a obstacle avoidance cost term, which evaluates the average percentage of comfort zone area occupied by obstacles along the walking paths of the staff agents. A high cost means that the agents encounter a large amount of obstacles in their comfort zones as they walk along their paths to execute tasks. Figure 6(b) illustrates the cost. The cost is defined as follows:

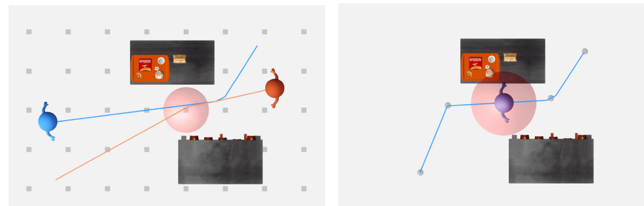


Fig. 6. Illustration of (a) congestion and (b) obstacle avoidance considerations.

$$C_{\text{Obstacle}}(\phi) = \frac{1}{N_{\text{Obstacle}}} \sum_{i} \sum_{x} \theta(\lambda_i(x)). \quad (6)$$

In this equation, λ_i refers to a path of a staff agent i. Each path is sampled regularly with an interval of 1m for evaluation. $\theta(\lambda_i(x))$ computes the percentage of comfort zone area occupied by obstacles as the staff agent is at location $\lambda_i(x)$ on the path, which is obtained from the work simulation described in Section 7. The cost is summed over all paths of all agents. N_{Obstacle} is a normalization constant calculated as the total number of evaluations performed.

6.2 Workload Costs

The workload costs evaluate the work experience encountered by the staff at the workspace. We define workload costs based on the efforts spent by the staff and the balance in workload distribution according to the workspace and workplan design:

$$\begin{aligned}\mathbf{C_W w_W}^T &= w_{\text{Walk Effort}} C_{\text{Walk Effort}}(\phi) + w_{\text{Turn Effort}} C_{\text{Turn Effort}}(\phi) \\ &+ w_{\text{Walk Balance}} C_{\text{Walk Balance}}(\phi) \\ &+ w_{\text{Turn Balance}} C_{\text{Turn Balance}}(\phi). \end{aligned} \quad (7)$$

Ideally, a workspace and a workplan should avoid unnecessary efforts, for example, staff do not need to walk long distances to get things. On the other hand, workload distribution should consider the staff's physical fitness [Dewi and Septiana 2015; MacDonald 2003]. Our workload costs consider such factors as follows.

Walk Effort. It evaluates the walk distances of the staff at work. Each staff agent has a walk intolerance level $\delta^{\text{Walk}} \in [0, 1]$. A high value of δ^{Walk} means a staff agent does not prefer walking. The walk effort cost is defined as:

$$C_{\text{Walk Effort}}(\phi) = 1 - \exp(-\frac{\sum_{i=1}^{N} \delta_i^{\text{Walk}} D_i}{\sum_{i=1}^{N} \delta_i^{\text{Walk}} D_{\max}}), \quad (8)$$

where D_i is the total walk distances for staff agent i in completing all tasks as computed from the work simulation. N refers to the total number of staff agents. δ_i^{Walk} represents the walk intolerance for staff agent i. $D_{\max}$ is a normalization constant calculated as the layout's perimeter length times the maximum number of tasks performed by an agent in a simulation.

Turn Effort. Similarly, to evaluate turning effort, we calculate the amount of body rotation a staff agent makes when it walks at work. Each staff agent has a turn intolerance level $\delta^{\text{Turn}} \in [0, 1]$, where a

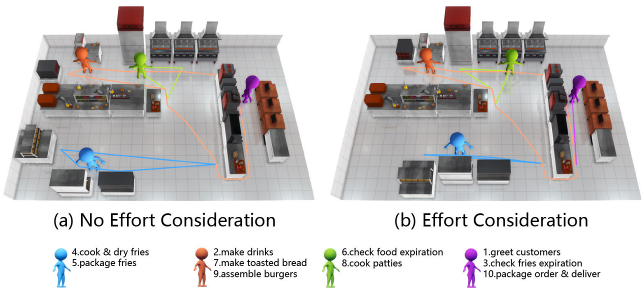


Fig. 7. Given the same workplan for the staff agents, the effort consideration guides the equipment objects of related tasks to stay together to reduce walk and turn efforts.

high turn intolerance means a staff agent does not prefer turning. The turning effort cost is defined as:

$$C_{\text{Turn Effort}}(\phi) = 1 - \exp\left(-\frac{\sum_{i=1}^{N} \delta_i^{\text{Turn}} R_i}{\sum_{i=1}^{N} \delta_i^{\text{Turn}} R_{\max}}\right), \tag{9}$$

where R_i is the total body rotation of agent i in completing all tasks. δ_i^{Turn} denotes the turn intolerance for agent i. $R_{\max}$ is a normalization constant computed as $(e-1)\pi$; e is the maximum number of equipment objects interacted by an agent in a simulation.

Figure 7 shows the effects of effort considerations. Given the same workplan, our optimizer brings the equipment objects of related task closer to reduce walk and turn effort.

In our kitchen examples, two of the four staff agents have higher walk and turn intolerances. Therefore the optimizer tends to assign less physically-demanding tasks (in terms of walk and turn efforts) to these agents. In our experiments, we further demonstrate how the intolerances can be used for modeling teams with a physically-challenged member or a robot assistant.

Walk Balance. Our approach also considers workload balance, in other words, fairness, in distributing workload among the staff as an unbalanced workload assignment may lead to low morale due to unfairness [McBride and Metcalfe 1995]. We introduce a walk balance cost to penalize a biased distribution of walk effort:

$$C_{\text{Walk Balance}}(\phi) = \sqrt{\frac{\sum_{i=1}^{N}(D_i - D_{\text{avg}})^2}{N D_{\max}^2}}, \tag{10}$$

where N is the total number of staff agents in the workspace. D_i is the total walk distances for staff agent i. D_{avg} is the average total walk distances of all staff agents.

Turn Balance. Similarly, we define a turn balance cost to penalize a biased distribution of turn effort:

$$C_{\text{Turn Balance}}(\phi) = \sqrt{\frac{\sum_{i=1}^{N}(R_i - R_{\text{avg}})^2}{N R_{\max}^2}}, \tag{11}$$

where R_i is the total rotation for staff agent i. R_{avg} is the average total rotation of all staff agents.

Figure 8 shows the effects of considering workload balancing. With balancing, the staff agents share the turn and walk efforts more evenly. Please refer to the supplementary material for more details on the ablation study on different cost terms.

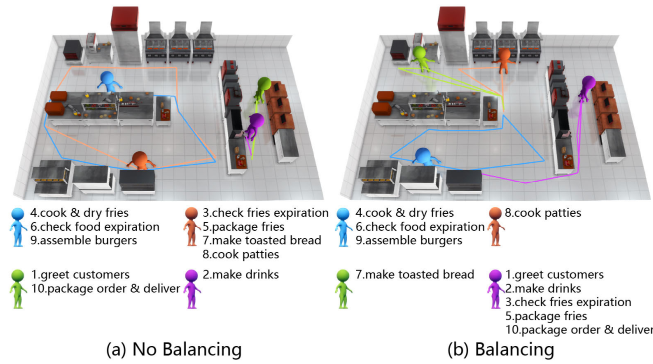


Fig. 8. With workload balancing, the walk and turn efforts are more evenly distributed among the staff agents.

6.3 Layout Costs

Layout costs encode object arrangement styles in the workspace. Inspired by previous work [Merrell et al. 2011], we consider wall proximity and object alignment in the workspace:

$$\mathbf{C_L w_L}^{\mathrm{T}} = w_{\text{Wall}} C_{\text{Wall}}(\phi) + w_{\text{Align}} C_{\text{Align}}(\phi). \tag{12}$$

Wall Proximity. In a workspace design, some equipment (e.g., a grill station) has to stay near a wall due to mechanical and electrical constraints. Therefore, we define a wall cost to evaluate whether an equipment object is close to a wall in the workspace:

$$C_{\text{Wall}}(\phi) = 1 - \exp\left(-\frac{\sum_i W(i)}{\sigma_{\text{Wall}}}\right), \tag{13}$$

where $W(i)$ returns the distance between object i and its nearest wall if the distance is longer than 1m, and zero otherwise. σ_{Wall} is a normalization constant set as the maximum possible distance to the nearest wall.

Object Alignment. We also encourage object alignment for neatness. In our running example, all objects near the wall should be aligned with the object that is closest to the wall. For an object near the center of the layout, it aligns with the nearest larger object. The object alignment cost is defined as:

$$C_{\text{Align}}(\phi) = 1 - \exp\left(-\frac{\sum_i A(i)}{\sigma_{\text{Align}}}\right), \tag{14}$$

where $A(i)$ returns the distances between object i and its nearby target object to achieve the nearest alignment (left, right, or center-alignment). σ_{Align} is empirically set as 0.2 to control the penalty.

7 WORK SIMULATION

At each iteration of the optimization, our approach proposes a solution comprising a workspace and a workplan. Based on this proposed solution, our approach runs a work simulation to compute the performance and workload costs to evaluate the solution.

Before simulation, each staff agent i is assigned with a sequence of tasks τ_i (ordered by their priorities) according to the workplan. During the simulation, a work order is generated around every 40 frames. A work order is associated with some tasks that need to be accomplished. For our kitchen example, a work order refers to a customer meal order (e.g., "ordering fries"), which is associated

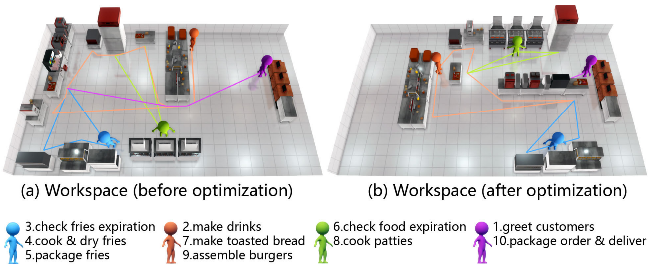


Fig. 9. Workspace optimization. Given a fixed workplan, our approach optimizes the workspace by moving the equipment objects.

with some service tasks (e.g., "(1) greet customers" → "(4) cook & dry fries" → "(5) package fries" → "(10) package order & deliver").

At each frame of the simulation, our approach checks and updates each staff agent's status. If an agent is idle, it picks up an available task it is in charge of according to the workplan. If there are two or more available tasks which the agent is charge of, it will take the task with the highest priority in its task sequence τ_i.

Each task is associated with a behavior tree which encodes the preconditions and a series of actions for executing the task. Figure 4 shows the behavior tree of the "cook & dry fries" task as an example. If the agent is not idle at a simulation frame, meaning that it is in the middle of performing an assigned task, it will continue with performing the next step of the assigned task according to the assigned task's behavior tree.

The simulation also depends on the staff agents' properties. First, the agents have different walking speeds. Second, the speed of finishing an assigned task depends on the agent's familiarity with the task. In our experiment, a high task familiarity refers to a 50% speed-up compared to a medium task familiarity, while a low task familiarity refers to a 50% slowdown.

Our kitchen example refers to a fast food service scenario. In this scenario, three customer meal orders are generated sequentially. We include the details of the customer meal orders in the supplementary material. The simulation ends when all customer orders are completed. During the simulation, we keep track of data such as the agents' walking paths, the obstacles in an agent's comfort zone, etc. for computing the performance and workload costs.

8 OPTIMIZATION

Our goal is to synthesize a solution ϕ comprising a workspace $\mathcal{W}_S$ and a workplan $\mathcal{W}_P$ optimized with respect to the performance, workload, and layout considerations. This is achieved by minimizing the total cost $C_{\text{Total}}(\phi)$ of Equation (1). Our optimization process proceeds in two stages, workspace optimization (Figure 9) and workplan optimization (Figure 10), which are alternately applied. Refer to the supplementary material for their cost values.

Akin to previous approaches [Merrell et al. 2011; Yeh et al. 2012], we apply simulated annealing with a Metropolis-Hastings state-searching step [Chib and Greenberg 1995] to optimize the solution iteratively, for both the workspace optimization stage and the workplan optimization stage. For both optimizations, at each iteration, a move is applied to modify the current solution ϕ to propose a solution ϕ', which is accepted with the following acceptance probability

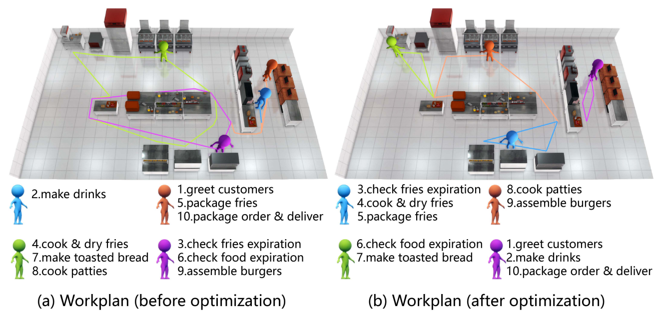


Fig. 10. Workplan optimization. Given a fixed workspace, our approach optimizes the workplan by assigning different tasks to the staff agents.

based on the Metropolis criterion:

$$\mathcal{P}(\phi'|\phi) = \min(1, \frac{f(\phi')}{f(\phi)}), \tag{15}$$

where f(ϕ) is a Boltzmann-like function that encodes the total cost:

$$f(\phi) = \exp(-\frac{1}{t} C_{\text{Total}}(\phi)), \tag{16}$$

and t is the temperature parameter. Initially, a high temperature $t = 1.0$ is set empirically, allowing the optimizer to extensively explore the solution space. Over the iterations, t drops gradually to a low value near zero, making the optimizer more conservative in accepting a proposed solution with a higher cost. The optimization terminates as the change in $C_{\text{Total}}(\phi)$ is smaller than 0.5% over the past 20 iterations. More details are provided below.

Workspace Optimization. In this stage, our approach optimizes the workspace while the workplan is fixed. As Figure 11 shows, each equipment object is associated with a grid of locations that its center can land on. The locations are regularly sampled with an equal interval, taken as the shorter dimension of the object multiplied by a scale factor. At each location, four orientations in multiples of 90° are possible. We define three types of moves for the workspace optimization stage:

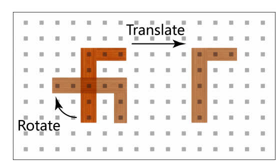


Fig. 11. A grid of locations for placing an object, which can be oriented in multiples of 90°.

- Translation: Randomly select one object. Translate the object in a random direction by a random amount.
- Rotation: Randomly select one object. Rotate the object by 90°, 180°, or 270°.
- Swap: Randomly select two objects. Swap their positions and orientations.

The swap move is helpful for preventing a large object from getting stuck at a corner. At each iteration, a move is randomly selected and applied to propose a solution. The three moves are selected with probabilities 0.4, 0.3, and 0.3, respectively.

Workplan Optimization. In this stage, our approach optimizes the workplan while the workspace is fixed. The sequences of assigned tasks $\{\tau_i\}$ of the staff members are modified to optimize the workplan. We define three types of moves:

Fig. 12. An optimization process example of the fast food kitchen. (a) Result after running the first alternating optimization. (b) Final result.

- Reassignment: Randomly remove 1 to 3 tasks from an agent. Reassign those tasks to another randomly-selected agent.
- Swap Assignments: Randomly select two agents. Select 1 to 3 tasks from each selected agent. Swap their selected tasks.
- Reorder Assignments: Randomly select 2 to 3 tasks of an agent. Randomly reorder those tasks in that agent's sequence. Note that a task closer to the front of the sequence has a higher priority to be taken in the work simulation.

Alternating Optimization. Optimizing the workspace and workplan simultaneously is difficult to keep track of. The optimizer would easily get trapped in a poor local minimum due to the complex optimization landscape. Instead, our approach applies the workspace and workplan optimizations in an alternating optimization fashion. We compare alternative optimization with another baseline approach in the supplementary material.

At initialization, the objects in the workspace are randomly placed and oriented. The staff members are randomly assigned with tasks in the workplan. Then, a workspace optimization is applied, followed by a workplan optimization, hence finishing one round of alternating optimization. Several rounds of alternating optimization are applied until the solution converges.

Moreover, we adopt a coarse-to-fine strategy to help the optimizer locate a solution more efficiently. For the workspace optimization in the first round of alternating optimization, a coarse grid is used with a larger interval between the locations. The purpose is to reduce the search space to facilitate the search of a rough object placement configuration. In the later rounds of the alternating optimization, the workspace optimization uses finer grids with smaller intervals between the locations, allowing the optimizer to refine the object placement configuration. We also set a movement range for the translation move to facilitate object location refinement. Refer to the supplementary material for more details about coarse-to-fine strategy and its ablation study. Figure 12 shows the synthesized workspace and workplan after the first round of alternating optimization and the final result.

Pareto Simulated Annealing. So far we formulate the optimization problem by aggregating the individual cost terms into a weight-ed sum as shown in Equation (1). We obtain one optimal solution with respect to a set of fixed weights. In this section, we discuss how to modify the formulation to employ the Pareto simulated annealing technique (PSA) [Czyżżak and Jaszkiewicz 1998] to solve the multi-objective optimization problem, attaining a set of Pareto-optimal solutions with strengths on different objectives. The set is referred as the Pareto front set. A solution in the set is Pareto-optimal, meaning that at least one objective in the solution is not dominated by that of any other solution in the set. In other words, each solution in the set excels in at least one work-related criterion (e.g., efficiency).

The PSA process starts with a randomly generated sample set of solutions for exploration. Each solution in this sample set is associated with a set of weights, which are randomly initialized. On the other hand, the process also keeps a Pareto front set containing Pareto-optimal solutions discovered so far.

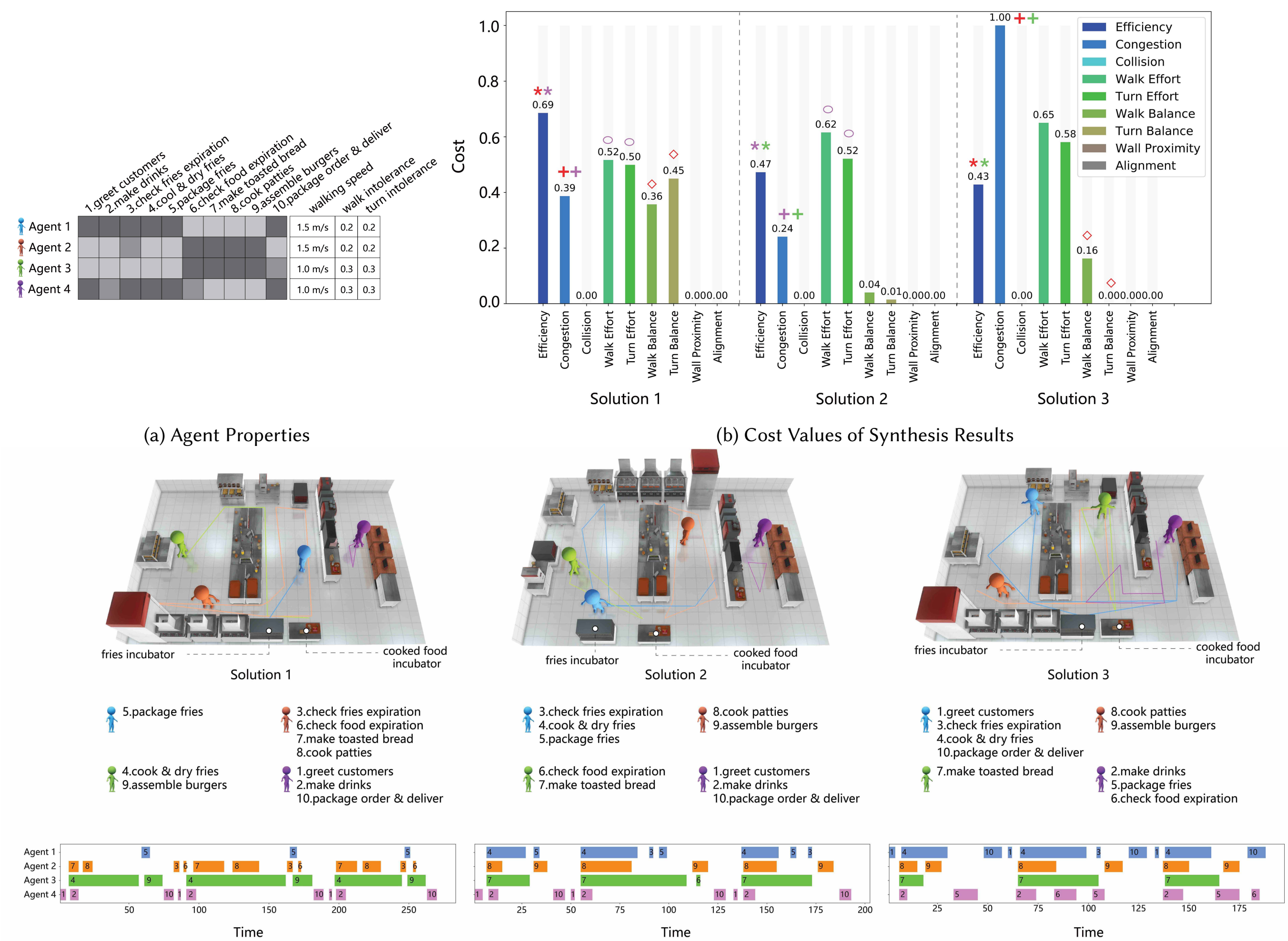
In each iteration of the exploration, a current solution ϕ in the sample set is perturbed by a move to propose a new solution ϕ'. The new solution is compared to solutions in the current Pareto front set. If the new solution dominates any solution in the set, the Pareto front set will be updated accordingly by adding the new solution and removing the dominated solutions.

After updating the Pareto front set, the transition from the current solution ϕ in the sample set to the new solution ϕ' is accepted with the following probability:

$$\mathcal{P}(\phi'|\phi, \mathbf{w}^*) = \min(1, \frac{\hat{f}(\phi', \mathbf{w}^*)}{\hat{f}(\phi, \mathbf{w}^*)}). \tag{17}$$

Let ϕ^* be the nearest solution of the current solution ϕ in the sample set in terms of the Euclidean distance. For notation convenience, we define $\mathbf{w}^* = \{w_i^*\}$ as the associated set of weights of the nearest solution ϕ^*, where w_i^* corresponds to the weight for each cost term $C_i(\phi^*)$. Following the PSA formulation, the weights $\mathbf{w}^*$ of the nearest solution ϕ^* are updated such that the weight w_i^* of each objective worse than the current solution ϕ's objective is scaled up, and the weight w_i^* of each objective better than the current solution ϕ's objective is scaled down. By updating the weights of the nearest solution, a disperse set of solutions can be generated for expanding the pareto set. Refer to [Czyżżak and Jaszkiewicz 1998] for theoretical explanation. $\hat{f}$ is a Boltzmann-like function comprising the weighted sum of costs using the nearest solution's weights $\mathbf{w}^*$:

$$\hat{f}(\phi, \mathbf{w}^*) = \exp(-\frac{1}{t}\sum_i w_i^* C_i(\phi)), \tag{18}$$



(a) Agent Properties (b) Cost Values of Synthesis Results

(c) Synthesized Workspaces and Workplans

Fig. 13. Three Pareto-optimal solutions for fast food kitchen. (a) Input agent properties. (b) Final cost values of the three solutions. (c) The synthesized workspace and workplan of each solution. Overall, solution 1 excels in reducing walk and turn efforts, solution 2 excels in congestion avoidance, and solution 3 excels in efficiency. Refer to the main text for discussion on comparing the synthesis results.

where t is the temperature parameter. Initially, a high temperature $t = 1.0$ is set empirically, allowing the optimizer to explore the solution space extensively. By updating the weights $\mathbf{w}^*$ of the nearest solution ϕ^* as said, a disperse set of solutions are generated for expanding the Pareto front set. The supplementary material shows the pseudocode of the PSA integrated with our optimization.

9 EXPERIMENTS AND RESULTS

We implemented our approach using C# and the Unity game engine. Our optimization approach was run on an Alienware machine equipped with an Intel Core i7-9700 CPU, an NVIDIA GeForce RTX 2070 graphics card, and 32GB of RAM. Our approach synthesized a workspace design in about 150 iterations and a workplan design in about 100 iterations, depending on the input's complexity. The approach typically took about three rounds of alternating optimization to synthesize a workspace and workplan, which took about one hour based on our implementation. For PSA, we started with $3-5$ solutions in the sample set for exploration. The PSA process took about $3-4$ hours to generate a Pareto front set. Speedup of PSA is possible via parallel computing [Banos et al. 2006].

9.1 Different Workplaces

We apply our approach to synthesize workspaces and workplans for different workplaces: *fast food kitchen*, *supermarket*, *restaurant*, and *donation center*. Figure 13 and 14 show the input task lists, agent properties, as well as the synthesized workspaces, workplans, and the schedules of tasks of the agents. The supplementary material contains more details of the inputs and results of the workplaces.

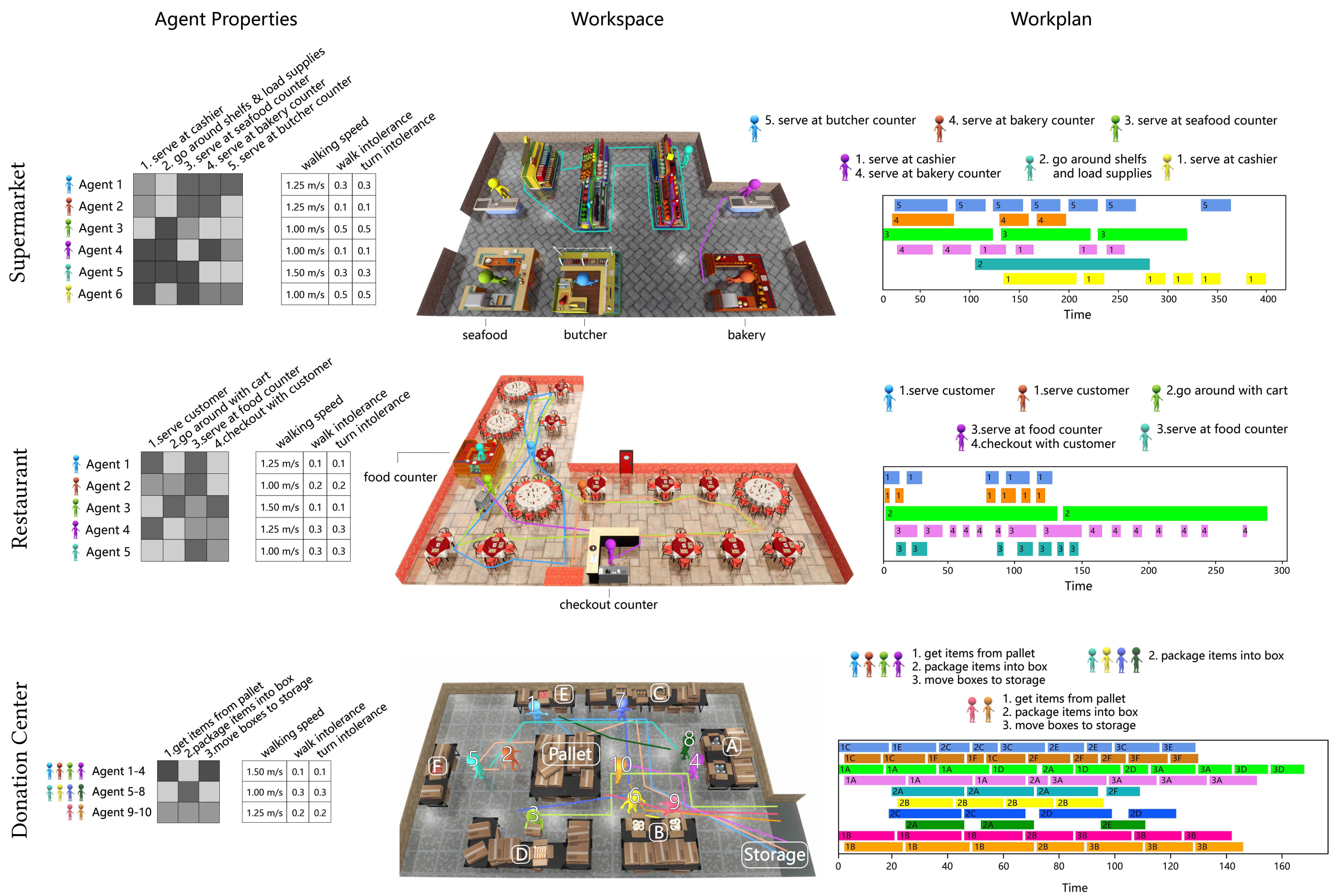


Fig. 14. Results of different workplaces. For the table under agent properties, high gray intensity of a cell refers to high task familiarity.

We show three Pareto-optimal solutions for *fast food kitchen*, and one solution for *supermarket*, *restaurant*, and *donation center* with strengths on efficiency, walk effort, and turn effort.

Fast Food Kitchen. We synthesized a fast food kitchen that consists of four staff agents and ten tasks. Totally, there were 30 solutions in the final Pareto front set. Figure 13 shows the workspaces and workplans of three Pareto-optimal solutions and their final cost values. Overall, solution 1 excels in reducing work and turn efforts; solution 2 excels in congestion avoidance; and solution 3 excels in efficiency. We compare the solutions as follows.

Solution 1 and solution 3 have the same workspace but different workplans. Solution 1 has a higher efficiency cost (see red ⋆) but a lower congestion cost (see red +), comparing to solution 3. This is because, in solution 1, Agent 3 (green) has a slow walking speed and is assigned with two tasks that require walking a longer distance, resulting in a high efficiency cost. On the other hand, since most tasks are assigned to Agent 2 (orange) and 3 (green) in solution 1, there is less overlap among the walk paths and hence less congestion. Moreover, Agent 1 (blue) and Agent 4 (purple) have more idle time in solution 1 than in solution 3, so the walk and turn balance costs of solution 1 are higher than those of solution 3 (see red ◇).

Comparing solution 1 with solution 2, we observe that solution 2 has a lower efficiency cost (see purple ⋆) as staff agents are assigned with familiar tasks in solution 2. Moreover, in solution 2, Agent 2 (orange) and Agent 3 (green) work in separate local spaces, resulting in a lower congestion cost (see purple +). Walk and turn effort costs (see purple ∘) of solution 2 are higher than those of solution 1 as Agent 1 (blue) is assigned with tasks that result in have a long walk path in the workspace in solution 2.

Comparing solution 2 with solution 3, we observe that solution 3 has a lower efficiency cost (green ⋆) but a higher congestion cost (green +). Solution 3 assigns all food-making tasks (4,7,8 & 9) to Agents 1-3 who are good at doing these tasks despite longer walking distances needed. As we can observe in solution 3, their walk paths overlap with each other, thus more congestion happens.

Supermarket. This scenario simulates a typical supermarket scenario which has three food sections (seafood, butcher and bakery) and ten shelves that store different types of items. Customers walk in with a shopping list. They first visit the food sections to buy anything according to their shopping list. Then they check out the shelves to buy any remaining item on their lists. If an item they look for is unavailable on the shelf, they wait for a staff agent to load supplies. Eventually, they finish buying all their goods, go to the closest register to pay, and then leave the supermarket.

The simulation consists of eleven customers. There are six staff agents working on five tasks (four stationary tasks and one mobile task). Stationary tasks require the staff agents to stand at a counter and serve customers. The mobile task requires the staff agents to walk around the shelves and load supplies. Note that a staff agent may be assigned with more than one task and it is up to the optimizer to decide the task assignments.

Figure 14 shows the work skills and physical fitness of the staff agents based on which tasks are assigned as we can observe. The optimizer lets Agent 5 (cyan), who can walk very fast, to walk around the shelves to load supplies. Agent 4 (purple) is assigned to serve at the cashier and help out at the nearby bakery food section occasionally. Besides, the shelves are aligned horizontally and they provide enough space for the customers and staff to access.

Restaurant. This scenario simulates a dim sum restaurant with four large tables and eleven small tables. Five staff agents need to work together to accomplish four tasks: (1) Greet incoming customers and help them find a table to sit; (2) Walk around with a food cart and give customers the food they want; (3) Cook at the food counter and serve customers; (4) Walk to the customer table and check out the customer at the checkout counter. There are three mobile tasks (Task 1, 2 and 4) and one stationary task (Task 3).

In this simulation, groups of customers (each group may have 1 to 12 people) walk and wait for staff agents to lead their way to the table. Once they sit down, one person from that group walks to the food counter to order noodles. The rest waits for the food cart to pass by to order some dim sum. Once they finish eating, they will request a staff agent to help them check out.

As shown in Figure 14, our table arrangement provides enough accessibility space for customers and staff agents to walk around. All staff agents are assigned with tasks they are good at based on their skills and physical fitness. Our workplan optimizer sets Agent 4 (purple) to help out the food counter as customers line up, although its main task is to help customers check out. Our workspace optimizer places the food counter as close to the checkout counter as possible to reduce customers and staff walk distances.

Donation Center. This simulates a sorting scenario that involves ten volunteers working to sort donation items into six categories: canned food (A), toys (B), clothes (C), necessary products (D), pet food (E), and electronic device (F). Among the unsorted items, 31% are canned food, 28% are toys, 16% are clothes, 13% are necessary products, 6% are pet food, and 6% are electronic devices.

There are six tables of different sizes in the scene: two tables with a large capacity, two tables with a medium capacity, and two tables with a small capacity. Each table is assigned to temporarily hold one category of items. When a table is full, the staff needs to package the items on the table into boxes and send the boxes to storage before they could put items on the table again. To enable table category assignment, we add a move type in the workspace optimization to randomly swap item categories between two tables.

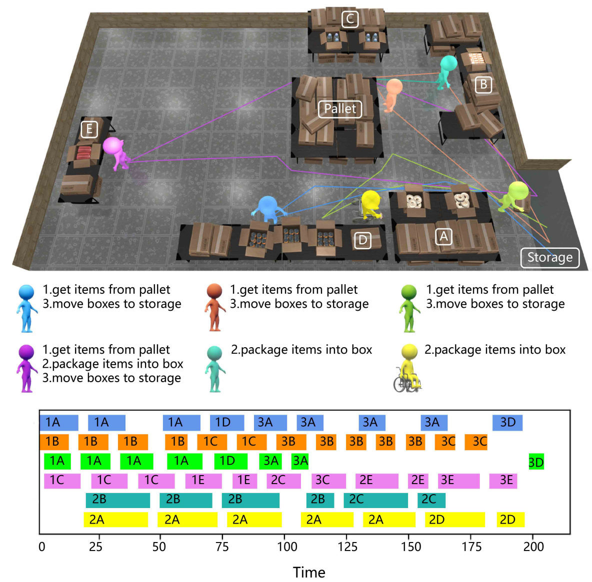


Fig. 15. Result with a wheelchair member with limited mobility (yellow).

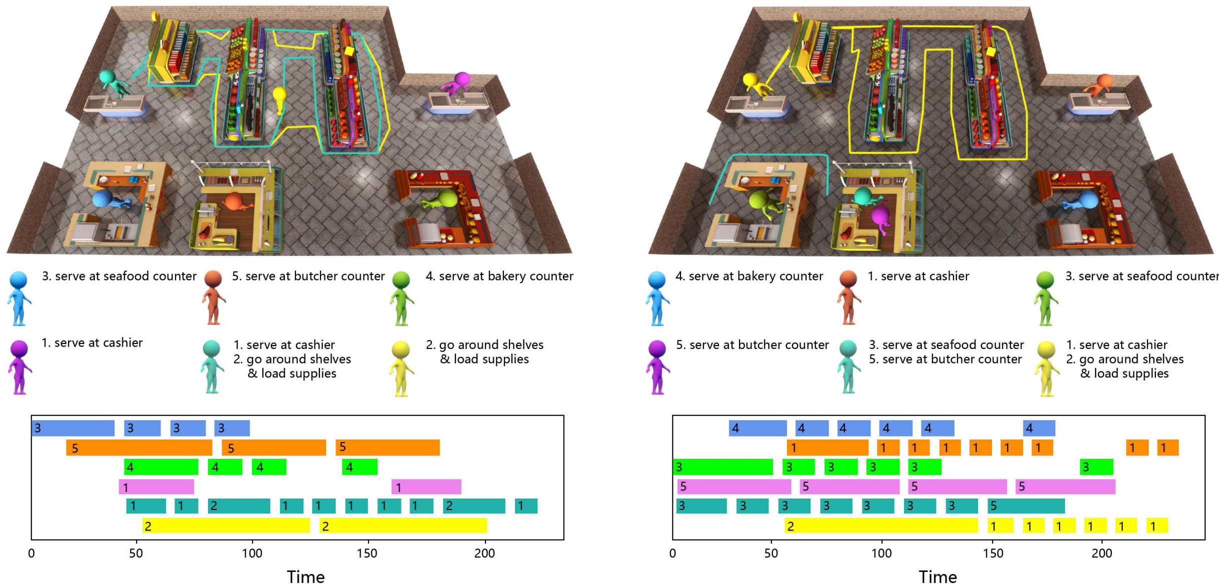
There are three tasks in this scene: (1) get an item from a pallet consisting of items of different categories and bring the item to a table that it belongs to based on its category; (2) package the items on a table into boxes; (3) send boxes to the storage. Each volunteer is assigned with some tasks and item categories it is in charge of.

As shown in Figure 14, our workspace optimizer assigns item categories to the tables based on their capacities. For example, a big table is assigned with the canned food, which takes up 31% of the unsorted items. The pallet is placed at the center with the tables arranged around it to reduce the walk distances. In the workplan arrangement, the six categories of items and the tasks are distributed among the agents according to their properties. As we could observe from workplan, more agents are assigned to sort canned food (A) and toys (B) than pet food (E) and electronic device (F). For Agents 5-8 who have low intolerance in movement and are good at packaging (Task 2), they are all assigned to do Task 2. Since Agents 1-4 have lower intolerance in movement and walk faster than Agents 9-10, they are in charge of more categories.

9.2 Other Scenarios

Wheelchair Member with Limited Mobility. We include a wheelchair volunteer agent in the donation center example. Figure 15 shows the synthesis result. The wheelchair agent (yellow) has high intolerance for movement. Our optimizer assigns this agent to perform packaging tasks at two tables. Note that the agent only moves from one table to another table occasionally as it finishes the packaging task at one table. The two tables where it works are put close to each other to reduce its walk distances. The other agents help with the mobile tasks including getting items from the pallet and moving packages to the storage area.

Dynamic Workplan. Given a synthesized fixed workspace, our approach can be applied to synthesize workplans dynamically based on work demand. Figure 16 shows an example, where two different workplans are synthesized for the morning and afternoon sessions.



(a) Morning Workplan (b) Afternoon Workplan

Fig. 16. Dynamic workplan example. Given the same workspace, our approach synthesizes (a) a morning workplan and (b) an afternoon workplan to satisfy different work demands.

In the morning session, the customers are set to shop shelf items more often than going to the food sections. In the synthesized morning workplan, Agent 5 (cyan) and Agent 6 (yellow) work together to check whether supplies on the shelves are enough. In the afternoon session, as the seafood and meat are on sale, customers tend to go to either of these sections to buy food. In the afternoon workplan, Agent 5 (cyan) serves as a runner to help out both sections instead of serving at the cashier (as it did in the morning session). The supplementary material contains one more example of including a robot assistant in the restaurant. It also contains the details of the input agent properties and tasks of all examples.

10 EVALUATION

10.1 Workspace Design

To validate our approach, we invited 15 participants to design fast food kitchen workspaces, which are compared with our synthesis results. All of the participants have some layout design background (either majored in design, or conducted layout design projects in university or industry). We designed an application which allows a participant to arrange equipment objects in a fast food kitchen space by simple operations to create a workspace.

Design Task. We asked the participants to use our application to design an efficient and realistic workspace for a given fast food kitchen workplan. During the design process, our participants were able to move or rotate the kitchen equipment objects. They could see the properties of different staff agents and their assigned tasks according to the given workplan. They could also run a simulation to see how the staff agents worked in the current workspace. After each simulation, the simulation time and walk distances were displayed for participants to evaluate how the staff agents performed. The participants could rearrange the equipment objects and run simulations until they were satisfied with their results. The supplementary material contains the screenshots of the application and the kitchen workspaces created by the participants.

Result Analysis. To compare the workspaces created by the participants with our synthesis results, we synthesized 15 different workspaces with the same work plan using the workspace optimization. We run simulations on all the workspaces using the Unity game engine to obtain the simulation time, total walk distances, and total body rotation. Moreover, we run simulations on the workspaces with AnyLogic[1], commercial simulation software widely used to simulate traffic, retail operations, supply chains, and logistics for research and business purposes. By using the same parameters (e.g., equipment locations, staff agents' task sequences), we obtained the simulation times in the workspaces via AnyLogic.

Table 2 compares the metrics computed on the workspaces created by the participants and our approach. The workspaces synthesized by our approach attain shorter simulation times, meaning that the given workplan is completed faster on the synthesized workspaces. On average, in our synthesized workspaces, the total walk distance of the agents is shorter and the total body rotation is also smaller, indicating that the agents spend less effort in executing the workplan. We performed t-tests on the two groups of results for each metric. All the p-values are smaller than 0.05, showing that there are significant differences in all the performance metrics.

Besides, on average, it took a participant 144 movements and 43 rotations to create a workspace design, whereas our optimization-based approach synthesized the workspaces fully automatically. Our synthesized results showed a relatively large improvement in terms of the total body rotation, likely because it was unintuitive for the participants to consider the anticipated turn efforts in designing a workspace manually, whereas our optimization approach incorporates this consideration through the turn effort cost term.

[1] https://www.anylogic.com/

10.2 Real-world Workplace Simulation

To evaluate the synthesized workspace and workplan as a whole, we conducted a preliminary real-world user study on workplace simulation. The experiment simulated working in a mini warehouse with many unsorted objects that needed to be recorded on a computer, followed by warehousing. We invited 36 participants to simulate working in the warehouse. The user study consisted of two conditions given in a random order for counterbalancing. In one condition, a pair of participants designed an efficient mini warehouse workspace and workplan, which they followed to do the tasks. In the other condition, the participants performed the tasks following a workspace and workplan synthesized by our approach. The participants first designed the workplan and workspace, then performed both the manually- and automatically-generated scenarios. We performed paired t-test to compare the completion times following the participants' designs and the synthesized design. There was a significant difference in completion times following the participants' designs (414s ± 56) and the synthesized design (351s ± 41); $p < 0.001$, suggesting that the workspace and workplan synthesized by our optimization approach outperforms the participants' designs. Please refer to the supplementary material for comprehensive details about this user study.

11 SUMMARY

Our work sheds light upon the novel research direction of jointly synthesizing a workspace and a workplan. Using our approach, designers can synthesize a workspace and a workplan optimized with respect to common performance-related criteria such as efficiency, congestion avoidance, and obstacle avoidance, and workload-related factors such as walk effort, turn effort, and workload balances.

Limitations and Future Work. In designing a real workspace, there are other considerations such as electrical constraints, mechanical constraints, and safety requirements to satisfy in practice. A full computational design tool should incorporate such constraints into synthesizing a functional work layout. As it stands, our prototype, which focuses on performance and workload criteria, complements rather than replaces existing architectural design software.

Our approach only considers how the equipment layout may affect work performance. There are other design and ergonomic factors that could affect work. For example, the colors, lighting, and furniture layout style of the workspace may affect the mood, creativity, and productivity of the staff. In future work, it would be interesting to incorporate findings from human factors and ergonomics research into a computational design framework for driving a comprehensive workspace optimization.

Our approach uses behavior tree with A^* pathfinding to model simulation because similar approaches have been widely adopted in industry. For future work, other simulation models (e.g., ORCA) which generate collision-free motion simulation can be used to obtain more natural simulation results.

When introducing each parameter, we set the parameters empirically rather than measuring them from real-world experiments. In practice, one could estimate performance parameters via wearable devices (e.g., pedometers) and such data could be used for driving the optimization. It would be interesting to combine our approach with customized agent instructions [Lafreniere et al. 2016] to synthesize workspaces and workplans for a sizeable work team.

Table 2. Evaluation results. In the first and second rows, the numbers shown are the means with the standard deviations in parentheses. The third row shows the p-values of t-tests comparing the results created by the human designers and by our approach.

	AnyLogic Simulation Time (s)	Unity Simulation Time (s)	Total Walk Distances (m)	Total Body Rotation (deg)
Human	89 (5)	106 (6)	61.45 (7.00)	1,202 (156)
Ours	85 (3)	101 (4)	54.60 (5.15)	1,025 (131)
p-value	0.014	0.014	0.016	0.007

For intuitiveness and efficiency, we use simple work simulations specified by behavior trees for evaluating a workspace and workplan solution at each iteration. In reality, work processes could be more complex due to stochastic work events and human behaviors. Therefore, it is important to minimize the gap between simulation and reality. Recent work [Xu et al. 2019] in designing hybrid UAVs controller proposed an error integral block to eliminate the system's steady-state error (induced by flying). Inspired by their error model, we could also model possible errors during simulation to close the gap between simulation and reality. For example, the staff may make mistakes from time to time; equipment objects may stop working. By using a more complex work simulation model, our approach could synthesize a robust workspace and a workplan tolerant to mistakes and flexible to changes.

ACKNOWLEDGMENTS

We are grateful to the anonymous reviewers for their constructive comments. This research is supported by an NSF Graduate Research Fellowship and an NSF CAREER Award (award number: 1942531). The work by E. Plaku is supported by (while serving at) the National Science Foundation. Any opinion, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

Badreddine Aboulissane, Dikra El Haiek, Larbi El Bakkali, and Jalal El Bahaoui. 2019. On the workspace optimization of parallel robots based on CAD approach. *Procedia Manufacturing* 32 (2019), 1085–1092.

Shailen Agrawal and Michiel van de Panne. 2016. Task-based locomotion. *ACM Transactions on Graphics* 35, 4 (2016), 1–11.

Rachid Alami, Aurélie Clodic, Vincent Montreuil, Emrah Akin Sisbot, and Raja Chatila. 2005. Task planning for human-robot interaction. In *Proceedings of the 2005 joint conference on Smart objects and ambient intelligence: innovative context-aware services: usages and technologies*. 81–85.

Sergey Alatartsev, Sebastian Stellmacher, and Frank Ortmeier. 2015. Robotic task sequencing problem: A survey. *Journal of intelligent & robotic systems* 80, 2 (2015), 279–298.

Daniel G Aliaga, Carlos A Vanegas, and Bedrich Benes. 2008. Interactive example-based urban layout synthesis. *ACM Transactions on Graphics* 27, 5 (2008), 1–10.

Yunfei Bai, Kristin Siu, and C Karen Liu. 2012. Synthesis of concurrent object manipulation tasks. *ACM Transactions on Graphics* 31, 6 (2012), 1–9.

R Banos, C Gil, B Paechter, and J Ortega. 2006. Parallelization of population-based multi-objective meta-heuristics: An empirical study. *Applied Mathematical Modelling* 30, 7 (2006), 578–592.

Fan Bao, Dong-Ming Yan, Niloy J Mitra, and Peter Wonka. 2013. Generating and exploring good building layouts. *ACM Transactions on Graphics* 32, 4 (2013), 1–10.

Cenk Baykal and Ron Alterovitz. 2017. Asymptotically Optimal Design of Piecewise Cylindrical Robots using Motion Planning.. In *Robotics: Science and Systems*, Vol. 2017.

Omar Binhomaid and Tarek Hegazy. 2020. Agent-based Simulation of Workers' Behaviors, Productivity, and Safety around Construction Obstacles. *Canadian Journal of*

Civil Engineering ja (2020).

Gabriel R Bitran and Devanath Tirupati. 1993. Hierarchical production planning. *Handbooks in operations research and management science* 4 (1993), 523–568.

Michael Brill. 1992. Workspace design and productivity.. In *The Healthcare Forum Journal*, Vol. 35. 51–53.

Ole Broberg. 2010. Workspace design: a case study applying participatory design principles of healthy workplaces in an industrial setting. *International Journal of Technology Management* 51, 1 (2010), 39–56.

Emma Campbell. 2013. The 7 Principles of Human Design Thinking for Workplace Design. https://www.davenport-campbell.com.au/newsroom/the-7-principles-of-human-design-thinking-for-workplace-design/. Accessed 12-31-2020.

Pascale Carayon, C Alvarado, and Ann Schoofs Hundt. 2003. Reducing workload and increasing patient safety through work and workspace design. *Washington, DC: Institute of Medicine* (2003).

Don B Chaffin. 2008. Digital human modeling for workspace design. *Reviews of human factors and ergonomics* 4, 1 (2008), 41–74.

Guoning Chen, Gregory Esch, Peter Wonka, Pascal Müller, and Eugene Zhang. 2008. Interactive procedural street modeling. In *ACM SIGGRAPH 2008 papers*. 1–10.

Siddhartha Chib and Edward Greenberg. 1995. Understanding the metropolis-hastings algorithm. *The american statistician* 49, 4 (1995), 327–335.

Michele Colledanchise and Petter Ögren. 2018. *Behavior trees in robotics and AI: An introduction*. CRC Press.

Jack Crosbie, Roongtiwa Vachalathiti, and Richard Smith. 1997. Age, gender and speed effects on spinal kinematics during walking. *Gait & Posture* 5, 1 (1997), 13–20.

Piotr Czyżżak and Adrezej Jaszkiewicz. 1998. Pareto simulated annealing—a metaheuristic technique for multiple-objective combinatorial optimization. *Journal of Multi-Criteria Decision Analysis* 7, 1 (1998), 34–47.

Dyah Santhi Dewi and Tyasiliah Septiana. 2015. Workforce scheduling considering physical and mental workload: a case study of domestic freight forwarding. *Procedia Manufacturing* 4 (2015), 445–453.

Tian Feng, Lap-Fai Yu, Sai-Kit Yeung, KangKang Yin, and Kun Zhou. 2016. Crowd-driven Mid-scale Layout Design. *ACM Transactions on Graphics* 35, 4 (2016).

Matthew Fisher, Manolis Savva, Yangyan Li, Pat Hanrahan, and Matthias Nießner. 2015. Activity-centric Scene Synthesis for Functional 3D Scene Modeling. *ACM Transactions on Graphics* 34, 6 (2015).

Qiang Fu, Xiaowu Chen, Xiaotian Wang, Sijia Wen, Bin Zhou, and Hongbo Fu. 2017. Adaptive synthesis of indoor scenes via activity-associated object relation graphs. *ACM Transactions on Graphics* 36, 6 (2017), 1–13.

Cipriano Galindo, Juan-Antonio Fernández-Madrigal, Javier González, and Alessandro Saffiotti. 2008. Robot task planning using semantic maps. *Robotics and autonomous systems* 56, 11 (2008), 955–966.

Stephen C Graves. 1981. A review of production scheduling. *Operations research* 29, 4 (1981), 646–675.

Michael X Grey, Caelan R Garrett, C Karen Liu, Aaron D Ames, and Andrea L Thomaz. 2016. Humanoid manipulation planning using backward-forward search. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 5467–5473.

V Daniel R Guide Jr. 2000. Production planning and control for remanufacturing: industry practice and research needs. *Journal of Operations Management* 18, 4 (2000), 467–483.

Sehoon Ha, Stelian Coros, Alexander Alspach, Joohyung Kim, and Katsu Yamane. 2017. Joint Optimization of Robot Design and Motion Parameters using the Implicit Function Theorem.. In *Robotics: Science and systems*, Vol. 8.

Edward Twitchell Hall. 1966. *The hidden dimension*. Vol. 609. Garden City, NY: Doubleday.

Ruizhen Hu, Zeyu Huang, Yuhan Tang, Oliver van Kaick, Hao Zhang, and Hui Huang. 2020. Graph2Plan: Learning Floorplan Generation from Layout Graphs. *ACM Trans. on Graphics* 39, 4 (2020).

Annette Kämpf-Dern and Jennifer Konkol. 2017. Performance-oriented office environments–framework for effective workspace design and the accompanying change processes. *Journal of Corporate Real Estate* (2017).

Benjamin Lafreniere, Tovi Grossman, Fraser Anderson, Justin Matejka, Heather Kerrick, Danil Nagy, Lauren Vasey, Evan Atherton, Nicholas Beirne, Marcelo H Coelho, et al. 2016. Crowdsourced fabrication. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*. 15–28.

Wendy MacDonald. 2003. The impact of job demands and workload on stress and fatigue. *Australian Psychologist* 38, 2 (2003), 102–117.

Behnam Malakooti. 2014. *Operations and production systems with multiple objectives*. John Wiley & Sons.

Mollie McBride and David Metcalfe. 1995. General practitioners' low morale: reasons and solutions. *The British Journal of General Practice* 45, 394 (1995), 227.

Paul Merrell, Eric Schkufza, and Vladlen Koltun. 2010. Computer-Generated Residential Building Layouts. *ACM Trans. Graph.* 29, 6, Article 181 (Dec. 2010), 12 pages.

Paul Merrell, Eric Schkufza, Zeyang Li, Maneesh Agrawala, and Vladlen Koltun. 2011. Interactive furniture layout using interior design guidelines. *ACM transactions on graphics* 30, 4 (2011), 1–10.

Gabor Nagy, Jay L Brand, David A Fik, Sally J Augustin, Haijie Ding, David S Meyer, and Rick A Perkins. 2013. System of evaluating work characteristics and providing workspace design suggestions. US Patent App. 14/003,611.

N Nishida. 1991. Practical Use of Operations Research for Production Planning. *IFAC Proceedings Volumes* 24, 12 (1991), 33–36.

Pawel Pawlewski. 2015. DES/ABS approach to simulate warehouse operations. In *International Conference on Practical Applications of Agents and Multi-Agent Systems*. Springer, 115–125.

Chi-Han Peng, Yong-Liang Yang, Fan Bao, Daniel Fink, Dong-Ming Yan, Peter Wonka, and Niloy J Mitra. 2016. Computational network design from functional specifications. *ACM Transactions on Graphics* 35, 4 (2016), 1–12.

Anetta Pizag. 2015. *Create a Thriving Workspace: 7 Essential Design Principles for Positive High-Performance Physical Work Environments*. Pizag.

Patrizia Ribino, Massimo Cossentino, Carmelo Lodato, and Salvatore Lopes. 2018. Agent-based simulation study for improving logistic warehouse performance. *Journal of Simulation* 12, 1 (2018), 23–41.

Daniel Ritchie, Kai Wang, and Yu-an Lin. 2019. Fast and flexible indoor scene synthesis via deep convolutional generative models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 6182–6190.

Michelle M Robertson, Yueng-Hsiang Huang, Michael J O'Neill, and Lawrence M Schleifer. 2008. Flexible workspace design and ergonomics training: Impacts on the psychosocial work environment, musculoskeletal health, and work effectiveness among knowledge workers. *Applied ergonomics* 39, 4 (2008), 482–494.

Nancy Ruiz, Adriana Giret, and Vicente Botti. 2006. Towards an agent-based simulation tool for manufacturing systems. In *2006 IEEE Conference on Emerging Technologies and Factory Automation*. IEEE, 797–804.

Manolis Savva, Angel X Chang, Pat Hanrahan, Matthew Fisher, and Matthias Nießner. 2016. PiGraphs: Learning interaction snapshots from observations. *ACM Transactions on Graphics* 35, 4 (2016), 1–12.

Markus Schneider and Michael Ettl. 2012. Lean factory design. *ZWF Zeitschrift für wirtschaftlichen Fabrikbetrieb* 107, 1-2 (2012), 61–66.

Hrishi Shah, Madusudanan Sathia Narayanan, and Venkat N Krovi. 2010. CAD-enhanced workspace optimization for parallel manipulators: A case study. In *International Conference on Automation Science and Engineering*. IEEE, 816–821.

H Randolph Thomas, David R Riley, and Sunil K Sinha. 2006. Fundamental principles for avoiding congested work areas—A case study. *Practice Periodical on Structural Design and Construction* 11, 4 (2006), 197–205.

Inc TranSafety. 1997. Study compares older and younger pedestrian walking speeds. *Road Management & Engineering Journal* (1997).

Kimberly A Ulfig and Patricia A Venetucci. 1997. Kitchen layout, system. US Patent 5,590,586.

Hanqing Wang, Wei Liang, and Lap-Fai Yu. 2020. Scene Mover: Automatic Move Planning for Scene Arrangement by Deep Reinforcement Learning. *ACM Transactions on Graphics* 39, 6 (2020).

Kai Wang, Yu-An Lin, Ben Weissmann, Manolis Savva, Angel X Chang, and Daniel Ritchie. 2019. Planit: Planning and instantiating indoor scenes with relation graph and spatial prior networks. *ACM Transactions on Graphics* 38, 4 (2019), 1–15.

Kai Wang, Manolis Savva, Angel X Chang, and Daniel Ritchie. 2018. Deep convolutional priors for indoor scene synthesis. *ACM Transactions on Graphics* 37, 4 (2018), 1–14.

Wenming Wu, Lubin Fan, Ligang Liu, and Peter Wonka. 2018. MIQP-based Layout Design for Building Interiors. In *Computer Graphics Forum*, Vol. 37. 511–521.

Wenming Wu, Xiao-Ming Fu, Rui Tang, Yuhan Wang, Yu-Hao Qi, and Ligang Liu. 2019. Data-driven interior plan generation for residential buildings. *ACM Transactions on Graphics* 38, 6 (2019), 1–12.

Jie Xu, Tao Du, Michael Foshey, Beichen Li, Bo Zhu, Adriana Schulz, and Wojciech Matusik. 2019. Learning to fly: computational controller design for hybrid uavs with reinforcement learning. *ACM Transactions on Graphics (TOG)* 38, 4 (2019), 1–12.

Yong-Liang Yang, Jun Wang, Etienne Vouga, and Peter Wonka. 2013. Urban pattern: Layout design by hierarchical domain splitting. *ACM Transactions on Graphics* 32, 6 (2013), 1–12.

Yi-Ting Yeh, Lingfeng Yang, Matthew Watson, Noah D Goodman, and Pat Hanrahan. 2012. Synthesizing open worlds with constraints using locally annealed reversible jump mcmc. *ACM Transactions on Graphics* 31, 4 (2012), 1–11.

Lap-Fai Yu, Sai Kit Yeung, Chi-Keung Tang, Demetri Terzopoulos, Tony F. Chan, and Stanley Osher. 2011. Make it home: automatic optimization of furniture arrangement. *ACM Trans. Graph.* 30, 4 (2011), 86.

Y. Zhang, S. Sreedharan, A. Kulkarni, T. Chakraborti, H. H. Zhuo, and S. Kambhampati. 2017. Plan explicability and predictability for robot task planning. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*. 1313–1320.

Yixin Zhu, Yibiao Zhao, and Song Chun Zhu. 2015. Understanding tools: Task-oriented object modeling, learning and recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2855–2864.