# Joint Computational Design of Workspaces and Workplans(Supplementary Material)

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

**ACM Reference Format:**
Yongqi Zhang, Haikun Huang, Erion Plaku, and Lap-Fai Yu. 2021. Joint Computational Design of Workspaces and Workplans(Supplementary Material). *ACM Trans. Graph.* 40, 6, Article 228 (December 2021), 17 pages. https://doi.org/10.1145/3478513.3480500

## 1 ABSTRACT

This document is the supplementary material that our main paper refers to.

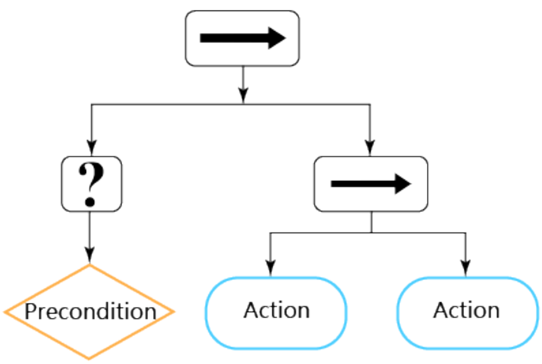



Fig. 1. A common behavior tree structure. It acontains a selector node (represented by the question mark) that describe preconditions (in orange), as well as action nodes (in blue) that encode execution and task destination.

## 2 BEHAVIOR TREE

We use behavior trees to encode human behaviors for our simulations. Behavior trees are flexible for authoring behaviors for agents. They are also a popular choice in the industry for designing agent behaviors for games and commercial applications. Figure 1 shows a common behavior tree structure. Generally, it consists of selector nodes that describe preconditions (in orange) and action nodes(in blue) that represent task destination and execution time.

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-ART228 $15.00
https://doi.org/10.1145/3478513.3480500

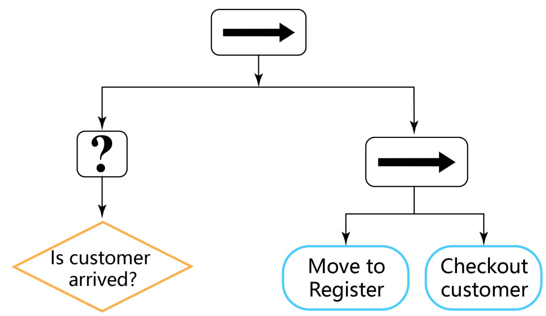



Fig. 2. The behavior tree example of the "serve at cashier" task for the supermarket scenario.

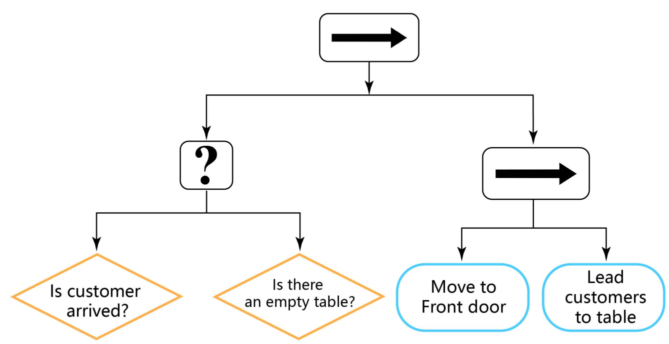



Fig. 3. The behavior tree example of the "serve customer" task for the restaurant scenario.

Figure 2, 3 and 4 show the behavior trees of the "serve at cashier" task, "serve customer" task, and "get items from pallet" task for the supermarket, restaurant and donation center examples, respectively. These behavior trees follow a similar structure as that of the common behavior tree structure in Figure 1.

## 3 ADDITIONAL DETAILS OF COARSE-TO-FINE STRATEGY

In workspace optimization, we employ a coarse-to-fine strategy to accelerate optimization. In this strategy, each equipment object is

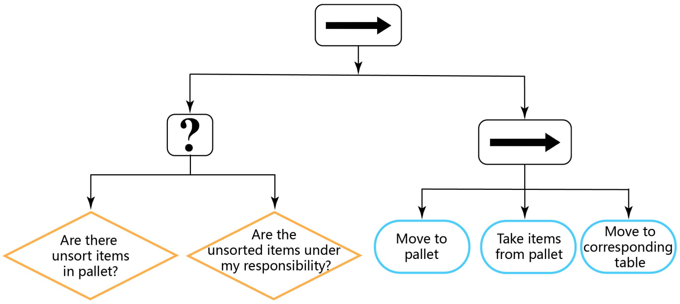
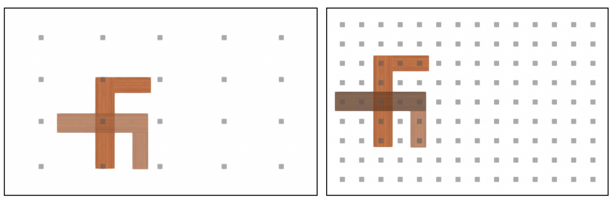



Fig. 4. The behavior tree example of the "get items from pallet" task for the donation center scenario.



(a) Coarse grid (b) Fine grid

Fig. 5. Illustration of the coarse-to-fine strategy used in workspace optimization. Each location in the grid represents a location that an equipment object's center can land on.

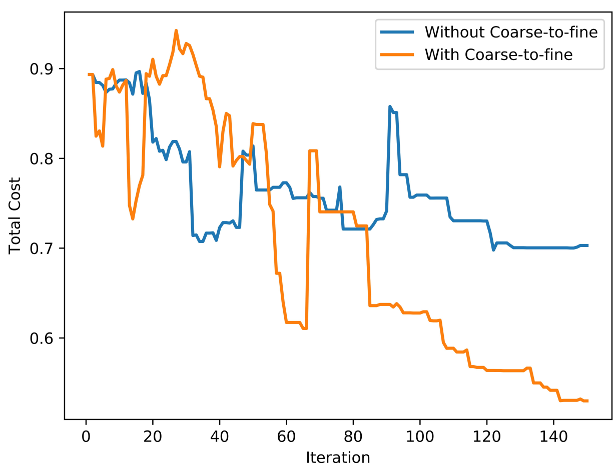



Fig. 6. Overall cost comparison for workspace optimization with and without using the coarse-to-fine strategy.

associated with a grid of locations that its center can land on. In the first round of alternating optimization, a coarse grid with larger intervals between locations is used as shown in Figure 5(a). 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 locations, allowing the optimizer to refine the object placement configuration as shown in Figure 5(b).

## 3.1 Comparison with baseline

We conduct an ablation study to show the effectiveness of the coarse-to-fine strategy. We compare two approaches, one approach with the coarse-to-fine strategy and the other without the coarse-to-fine strategy (baseline). For a fair comparison, we run workspace optimization for 150 iterations while keeping the workplan fixed. In the baseline approach, we use move types described in Section 8 of the main paper. In the coarse-to-fine strategy approach, a coarse grid is used to move objects for the first 50 iterations and a finer grid is used in the last 100 iterations. We include efficiency, walk effort, turn effort and wall proximity objective for evaluation and set each objective weight as 0.3, 0.25, 0.25, and 0.2, respectivevly.

The overall cost comparison graph is shown in Figure 6. The final cost details comparison is shown in Table 1. Based on the observation, the baseline approach got trapped at a local minimum where the wall proximity cost was not fully optimized. One possible reason is that our solution space is complex with many local minimums so it is difficult for the optimizer to locate a good configuration within 150 iterations. In contrast, the coarse stage of our approach helps the optimizer quickly jump close to a good local optimum, reducing the total number of iterations needed.

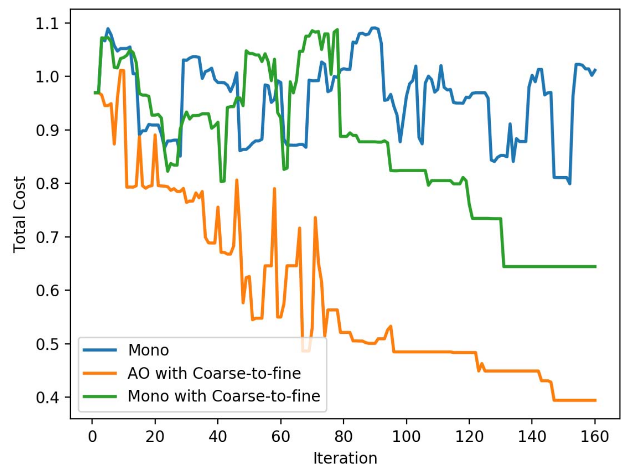



Fig. 7. Total cost comparison of the two baseline approaches and the alternative optimization approach.

# 4 ADDITIONAL DETAILS OF ALTERNATIVE OPTIMIZATION

To demonstrate the effectiveness of alternative optimization, we compare our alternative optimization approach with two baseline approaches: monolithic approach (Mono) and monolithic approach with coarse-to-fine strategy.

For a fair comparison, we run workspace and workplan optimization for each approach with a total of 160 iterations. In the monolithic approach, we optimize the workspace and workplan simultaneously.

Table 1. Detail cost comparison for workspace optimization with and without the coarse-to-fine strategy. Weighted costs for efficiency, walk effort, turn effort and wall proximity are shown with weights 0.3, 0.25, 0.25, and 0.2.

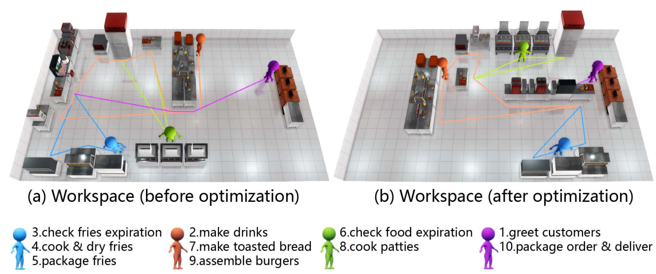
| | Total Cost | Efficiency Cost | Walk Effort Cost | Turn Effort Cost | Wall Proximity Cost |
|---|---|---|---|---|---|
| Without Coarse-to-fine strategy | 0.70 | 0.17 | 0.10 | 0.23 | 0.2 |
| With Coarse-to-fine strategy | 0.53 | 0.19 | 0.11 | 0.23 | 0 |

Table 2. Cost comparison details before and after workspace optimization (Figure8) and workplan optimization (Figure 9).

| | Total Cost | Efficiency | Congestion | Collision | Walk Effort | Turn Effort | Walk Balance | Turn Balance | Wall Proximity | Object Alignment |
|---|---|---|---|---|---|---|---|---|---|---|
| Before workspace optimization | 0.507 | 0.16 | 0.14 | 0 | 0.036 | 0.041 | 0.01 | 0.02 | 0 | 0.10 |
| After workspace optimization | 0.320 | 0.14 | 0.01 | 0 | 0.021 | 0.060 | 0.01 | 0.04 | 0 | 0.04 |
| Before workplan optimization | 0.49 | 0.2 | 0 | 0 | 0.046 | 0.083 | 0.046 | 0.083 | 0 | 0.03 |
| After workplan optimization | 0.40 | 0.11 | 0.2 | 0 | 0.013 | 0.044 | 0.0025 | 0.002 | 0 | 0.03 |

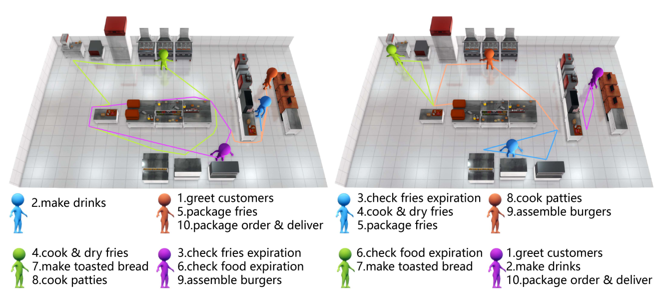
In the monolithic approach with the coarse-to-fine strategy, we follow a similar manner: in the first 80 iterations, a coarse grid is used to move equipment objects; and in the last 80 iterations, a finer grid is used. In the alternative optimization approach, we have two rounds of optimization and each (workspace or workplan) optimization runs for 40 iterations. In addition, we apply the coarse-to-fine strategy when running the workspace optimization.

For evaluation, we include efficiency, walk effort, turn effort and wall proximity objective cost with cost weight 0.3, 0.25, 0.25, and 0.2, respectively. The final cost results are shown in Figure 7. We observed that the monoliphic approach failed to converge to a local minimum. When adding the coarse-to-fine strategy to the monoliphic approach, the optimizer tends to get trapped at a local minimum. In contrast, the alternative optimization approach can locate a optimal solution in the end. This is probably because the solution space is complex with many local minimums; therefore it is hard to sample a move to result in a good configuration. The alternating optimization approach reduces the problem complexity and difficulty of search for a good solution. In addition to an alternative scheme, we also design moves specific to the workspace and workplan optimizations so that our optimizer can explore the solution space more efficiently.




(a) Workspace (before optimization) (b) Workspace (after optimization)

Fig. 8. Workspace optimization. Given a fixed workplan, our approach optimizes the workspace by moving the equipment objects.




(a) Workplan (before optimization) (b) Workplan (after optimization)

Fig. 9. Workplan optimization. Given a fixed workspace, our approach optimizes the workplan by assigning different tasks to the staff agents.

## 4.1 Fixed workplan in workspace optimization and vice versa

We demonstrate the impact of fixing workspace (or workplan) before and after one round of optimization in Figure 9 and Figure 8. When the workplan is fixed, our workspace optimizer rearranged equipment objects used by an agent when executing his sequence of tasks. When the workspace is fixed, our workplan optimizer reassigned agents with tasks based on the staff skill and location of objects. Table 2 shows cost comparison before and after the workspace (or workplan) optimization.

# 5 ADDITIONAL DETAILS OF PSA

Pseudo code for alternative optimization and modified version of PSA (Duh and Brown,2007) are shown in Algorithm 1 and Algorithm 2.

**Algorithm 1:** Alternative Optimization

**Result:** A set of pareto front solution
input: a initial set of solution S; a initial set of Pareto front solution M;
alternative iteration O;List of temperature T;
MoveMode ← Workspace Optimization;
counter ← 0
**while** *counter* < *O* **do**
  **if** *Workspace OPtimization* **then**
    PSA(S,M, t, MoveMode);
    MoveMode ← Workplan Optimization
  **else**
    PSA(S,M, t, MoveMode);
    MoveMode ← Workspace Optimization
  **end**
  increasee counter ; $t \leftarrow T[counter]$
**end**

**Algorithm 2:** Pareto Simulated Annealing (PSA)

**Result:** A set of pareto front solution M
input: a initial set of solutions S, a initial set of Pareto front solution M, MoveMode; initial temperature T0;
set current temperature T to initial temperature T0
**while** *stop conditions are not fulfilled* **do**
  **for** *all* $\phi \in S$ **do**
    Perturb a solution $\phi'$ based on MoveMode;
    **if** $\phi'$ *is not dominated by* $\phi$ **then**
      update set M with $\phi'$ ;
    **end**
    select $\phi^* \in$ S nearest (non-dominated) to $\phi$
    **if** $\phi^*$ *does not exist (or in the first iteration)* **then**
      then assign weights at random,
      assuring $\forall i\ \mathbf{w}_i^* \geq 0$ and $\sum_i \mathbf{w}_i^* = 1$ ;
    **else**
      **for** *all objectives* $\hat{f}_i$ **do**

$$\mathbf{w}_i^* = \begin{cases} \alpha \mathbf{w}_i, \text{if } \hat{f}(\phi) < \hat{f}(\phi^*) \\ \mathbf{w}_i/\alpha, \text{if } \hat{f}(\phi) \geq \hat{f}(\phi^*) \end{cases}$$

      **end**
      normalize the weights such that $\sum_i \mathbf{w}_i^* = 1$
    **end**
    update $\phi$ with $\phi'$, given $P(\phi'|\phi, \mathbf{w}^*, T)$
  **end**
  decrease Temperature T;
**end**

## 6 ADDITIONAL DETAILS OF ANYLOGIC

AnyLogic is commercial simulation software widely used to simulate traffic, retail operations, supply chains, and logistics for research and business purposes. It provides a risk-free environment, high-quality visualization, and capability to handle uncertainty in simulation models. In particular, we utilized its process modeling library to model the fast food restaurant system in terms of process (e.g., job order), entities going through the process flow (e.g., customers), and resources that entities use to perform action (e.g., object equipments). By using the same parameters (e.g., equipment locations, staff agents' task sequences) as in our own simulations, we obtained the simulation times in the workspaces via AnyLogic.

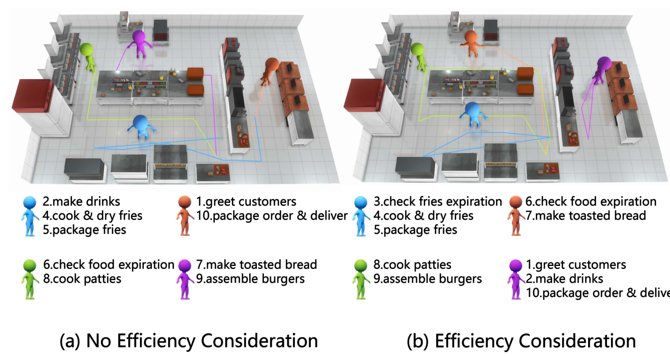



Fig. 10. Given the same workspace, efficiency consideration encourages our optimizer to assign tasks based on the agents' skills to improve efficiency. For example, with the efficiency consideration in (b), Agent 2 (orange) is assigned with Task 6 and 7 as it is good at doing Task 5-9.

## 7 ABLATION STUDY

Figure 10, 11, 12, and 13 depict the importance of each objective cost used in our optimization approach. These optimization objectives encode the performance and workload considerations. To demonstrate the effectiveness of each cost, we conduct an ablation study as follows: to investigate the efficiency, congestion, and workload balance criteria, we fixed the workspace and ran workplan optimization with the corresponding cost term being omitted; to investigate the effort criterion, we fixed the workplan and ran the workspace optimization with the effort cost term being omitted.

*Efficiency.* Its goal is to improve the time efficiency of serving work orders. With this consideration, our optimizer tends to assign staff with the tasks they are good at to shorten food-cooking time and service time. For instance, while Agent 2 (orange) is good at doing Task 6 to 9, it was assigned to do Task 1 and 10 if efficiency is not considered, as shown in the result of Figure 10 (a). If efficiency is considered, that agent is assigned with Task 6 and 7.

*Congestion.* The congestion consideration is to avoid congested locations induced by the workspace and workplan design. The congestion consideration increases the chance of assigning tasks that use nearby equipment to the same agent. For example, both Task 3 and

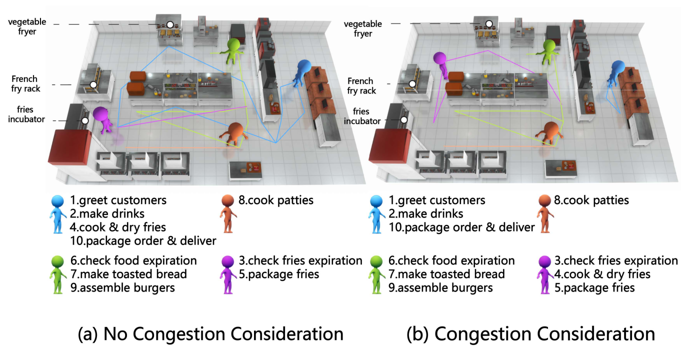



Fig. 11. Given the same workspace, our optimizer tends to assign tasks that use nearby equipment to the same agent. In this example, both Task 3 and Task 5 use fries incubator and are assigned to Agent 4 (purple). With the congestion consideration as shown in (b), Task 4 which uses the french fries rack near the fries incubator is assigned to Agent 4 (purple) instead of Agent 1 (blue). As a result, Agent 1 and Agent 4's movement become more localized. There is less overlap in the agents' paths and hence less congestion happens.

Task 5 use the fries incubator and are assigned to Agent 4 (purple) in Figure 11 (a). If congestion is considered, as shown in Figure 11 (b), Agent 4 (purple) instead of Agent 1 (blue) is assigned with Task 4 which uses the french fry rack that is near the fries incubator. Overall, with the congestion consideration, Agent 1 and Agent 4's movements become more localized, and the agents' walking paths tend to overlap less as shown in Figure 11 (b), and hence less congestion happens.

*Effort.* This cost evaluates the work experience encountered by the staff at the workspace. Figure 12 shows the effects of effort considerations. Given the same workplan for staff agents, with the effort consideration, our optimizer tends to arrange objects of related tasks to stay together to reduce walk and turn efforts. For example, the fries equipment objects are moved closer to the shelf since such equipment objects are used in Task 4 and Task 5 by Agent 1 (blue).

*Workload Balance.* This cost evaluates the fairness in distributing the workload among the staff. With the workload balance consideration, our optimizer encourages a more even workload distribution among the staff agents. As shown in Figure 13, the walk and turn efforts are more evenly distributed among the staff agents, comparing the agents' paths in (b) with their paths in (a).

## 8 ADDITIONAL DETAILS OF WORKPLACE INPUTS

Given an input space, a set of work equipment, staff agents with their properties, and a task list, our approach is capable of synthesizng an appropriate workspace and workplan that respect individual work experience and achieve workspace production goals. Below are additional details of input of different workplaces.

*Fast Food Kitchen.* Figure 18 shows the front view of work equipment of the fast food kitchen example. Table 4 shows the equipment objects used in each task. Table 5 shows details of the three customer meal orders used in the work simulation.

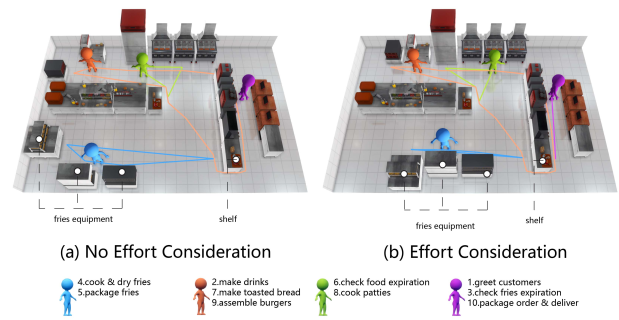



Fig. 12. Given the same workplan for the staff agents, the effort consideration prompts the equipment objects of related tasks to stay together to reduce walk and turn efforts. For example, the fries equipment objects used by Agent 1 (blue) are moved closer to the shelf so as to reduce the walk and turn efforts of Agent 1.

*Supermarket.* We show equipment objects of the supermarket in Figure 19 and list equipment objects used in each task in Table 6. We also include details of the customer shopping lists in Table 7.

*Restaurant.* Figure 20 shows equipment objects used in the restaurant example. Table 8 describes equipment objects used for each task. Table 9 reveals each customer group's orders.

*Donation Center.* Figure 21 shows equipment used in the donation center example.

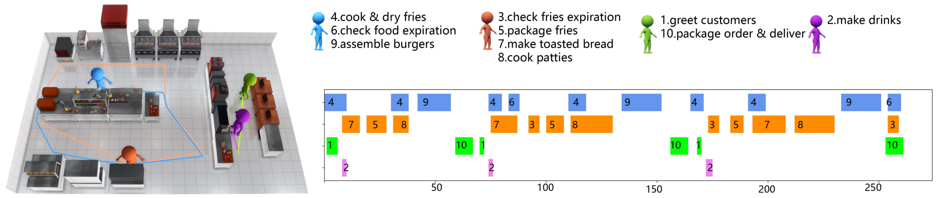
## 9 ADDITIONAL DETAILS OF OTHER SCENARIOS

As described in the main paper, our approach can be extended to handle other practical considerations such as dynamic workplan and incorporation of staff members with different capabilities in the team. We include additional details for other scenarios.

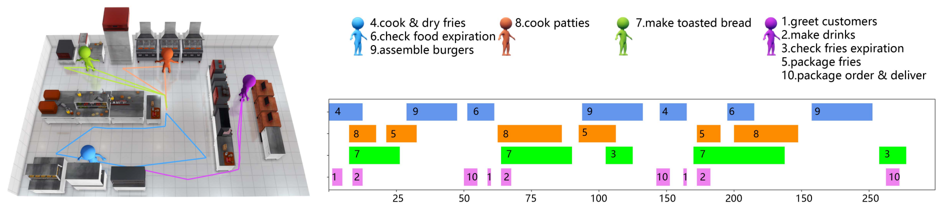
For ease of investigation, we use the basic weighted sum formulation instead of PSA to optimize the total cost $C_{\text{Total}}(\phi)$ (Equation (1) in main paper) in synthesizing these results. We set the weights as $w_{\text{Efficiency}} = 0.2$, $w_{\text{Congestion}} = 0.05$, $w_{\text{Collision}} = 0.05$, $w_{\text{Walk Effort}} = 0.25$, $w_{\text{Turn Effort}} = 0.25$, $w_{\text{Walk Balance}} = 0.05$, $w_{\text{Turn Balance}} = 0.05$, $w_{\text{Wall}} = 0.05$, and $w_{\text{Align}} = 0.05$.

*Dynamic Workplan.* We show all six agents' properties of the dynamic workplan example in Figure 22 and the customers' shopping lists for the morning and afternoon sessions in Table 10 and Table 11. As we could observe, Agent 5 (cyan) can walk very fast, so our optimizer sends him to help load supplies with Agent 6 (yellow) in the morning; and sends him to help out the seafood counter and butcher counter in the afternoon.

*Wheelchair Member with Limited Mobility.* . We include a wheelchair volunteer agent (yellow) in the scene. All agent properties are shown in Figure 24. Different from the donation center scenario, there are five tables of different sizes and the six volunteer agents are asked to work together to sort donation items based on five categories: canned food (A), toys (B), clothes (C), necessary products (D), and pet food (E). 40% is canned food, 30% is toys, 12% is clothes, 12% is necessary products, and 6% is pet food among the unsorted items.




(a) No Balancing




(b) Balancing

Fig. 13. With workload balancing, the walk and turn efforts are distributed among the staff agents as shown by the paths in (b) compared with the paths in (a).

*Rotbot Assistant.* We include a robot (Agent 5) in the restaurant example as a staff member. Figure 23 shows all five agents' properties and the synthesis result. Table 12 shows all customer groups' orders. The robot agent has a high walking speed of 1.50ms$^{-1}$ and a walk intolerance set as zero. As a result, our optimizer assigned the robot with tasks that involve high movement to relieve the workload of the other staff agents.

## 10 REAL HUMAN WORKPLACE SIMULATION

### 10.1 Setup

*Overview.* To evaluate the effectiveness of our approach for generating a feasible workspace and workplan in comparison to user's design, we also conducted a preliminary user study involving real human workplace simulation. The goal of this experiment was to simulate working in a mini warehouse where there were many unsorted objects that needed to be recorded in a computer database and then put into storage according to their types.

*Participants.* We recruited 18 pairs of participants (36 people in total) to simulate working in a mini warehouse. The participants were university students aged 19 to 25, with about 80% of males and 20% of females. The user study was IRB-approved.

*Conditions.* The user study consisted of two conditions of experiments given in a random order for counterbalancing. In one condition of the experiment, the participants would first design a mini warehouse workspace and workplan, then execute the tasks accordingly. In the other condition of the experiment, we asked the same group of participants to carry out the tasks following the workplace and workplan generated by our approach. The participants would first design the workplan/workspace, then perform both the manually- and automatically-generated scenarios. We gave the participants a five-minute warm-up session at the beginning of each condition to help them get familiar with the experiment. At the end, we asked the participants to fill out a survey.

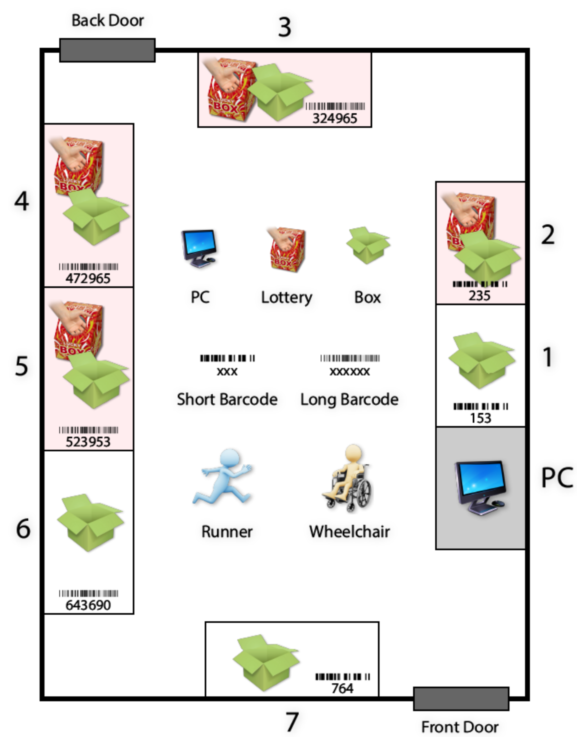



Fig. 14. The mini warehouse layout. The PC's location is fixed in the layout. In total, we have six boxes plus one lottery box to be assigned to seven locations (Loc 1,...,Loc 7). Note that the lottery box can only be placed at locations 2,3,4 or 5. Depends on the box placement location, a different length of the barcode of that object type will be used for data entry in Task 1. In particular, for boxes at locations 1, 2, or 7, objects with three-digit barcodes of that object type will be placed in the lottery box at the beginning. Otherwise, objects with six-digit barcodes of that object type will be placed in the lottery box.

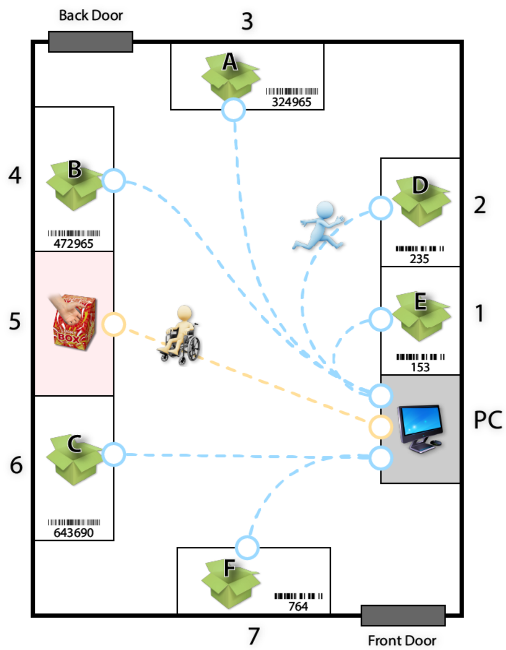



Fig. 15. The workspace generated by our approach. In the corresponding generated workplan, the runner is assigned to do Task 2 and the wheelchair person is assigned to do Task 1.

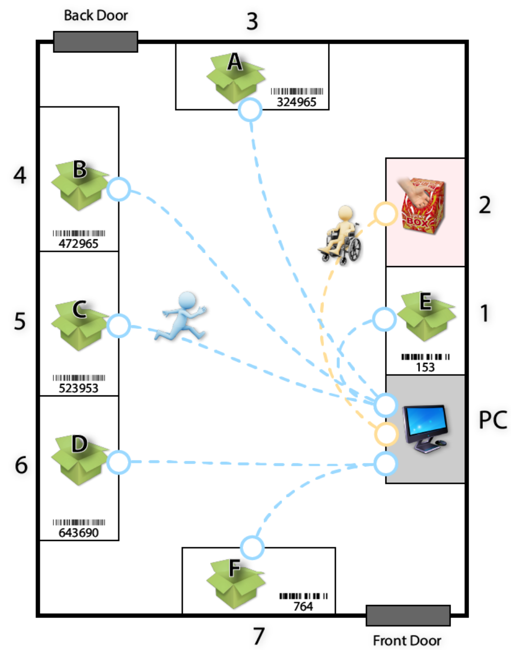



Fig. 16. An example workspace designed by a pair of participants. The wheelchair person decided to do Task 1 and the runner decided to do Task 2. Refer to Figure 17 for a screenshot of the workplace simulation.

*Tasks.* Following the workplace and workplan either designed by a pair of participants or generated by our approach, the two participants collaborate to complete two tasks. In the first task, one participant went to the lottery box, which stored unsorted objects, and took one or multiple object(s) to a PC station to record the object(s)' barcode. In the second task, the other participant took the recorded object(s) and put them in the correct boxes according to their types.

*Roles.* There were two roles in the experiments, namely, runner and wheelchair person. The runner could walk but could only carry one object at a time. The wheelchair person moved using a wheelchair but could carry three objects at a time. In the condition with our generated workplan, the two participants were assigned two different roles by our approach. In the condition where the participants designed the workplan, the two participants decided among themselves which role to take up for each person.

*Workplace Design.* The floor plan of the mini warehouse is shown in Figure 14. There were six types of objects for warehousing, therefore, there were six box types (Box A, Box B, Box C, Box D, Box E, and Box F) plus one lottery box. These boxes needed to be distributed among 7 locations in the mini warehouse. The rules for workspace design are as follows:

- The PC station is fixed in the floor plan.
- The lottery box can be placed only at the locations 2,3,4 or 5.
- There are 40 unsorted objects of different types in the lottery box at the beginning. Among these objects, 5% belong to Box A, 8% to Box B, 8% to Box C, 10% to Box D, 33% to Box E, and 36% to Box F. Each object either has a six-digit barcode or a three-digit barcode.
- Depending on the location of the box types placement, either a six-digit barcode or a three-digit barcode object of that type will be used. In particular, for boxes placed at location 1, 2 or 7, three-digit barcode object of that type will be put into the lottery box at the beginning. Otherwise, six-digit barcode object of that type will be used.

In each experiment, the pair of participants were asked to come up with a workpspace and workplan design to minimize the time to finish the tasks. An example workspace and workplan design by a pair of participants is shown in Figure 16, with the corresponding real-world simulation shown in Figure 17.

In this example, the wheelchair person decided to do Task 1 and the runner decided to do Task 2. Since they put Box E and Box F at location 1 and location 7, the barcodes of objects of type E and F were three-digit long, and the barcodes of other types of objects were six-digit long. As a result, the wheelchair person did not spend too much time in typing the barcodes of objects or type E and F when doing Task 1. Besides, they placed the lottery box at the same side of the PC. As a result, the wheelchair person traveled back and forth between these two locations to get objects from the lottery box, bringing the objects over to the PC for entering the barcode.

Please refer to our supplementary video for the actual workplace simulation.

Table 3. Real human workplace simulation results. The task completion time following the participants' designs and the generated design are shown. A design is encoded as a string with 7 characters which refer to the boxes put at locations 1 to 7. For example, "ELABCDF" (denoting the participants' design shown in Figure 16) means that Box E is placed at location 1, the lottery box is placed at location 2, etc. The generated design "EDABLCF" is shown in Figure 15.

| ID | Participants' design | Tasks For Wheelchair person and Runner | Completion Time(s) | Generated design | Tasks For Wheelchair person and Runner | Completion Time(s) |
|---|---|---|---|---|---|---|
| 1 | ELABCDF | 1,2 | 382 | EDABLCF | 1,2 | 366 |
| 2 | ELDABCF | 1,2 | 444 | EDABLCF | 1,2 | 346 |
| 3 | FLABCDE | 1,2 | 533 | EDABLCF | 1,2 | 421 |
| 4 | ELBADFC | 1,2 | 344 | EDABLCF | 1,2 | 312 |
| 5 | ELABCDF | 1,2 | 430 | EDABLCF | 1,2 | 369 |
| 6 | FLABCDE | 1,2 | 414 | EDABLCF | 1,2 | 382 |
| 7 | FLABCDE | 1,2 | 474 | EDABLCF | 1,2 | 430 |
| 8 | EFLCBAD | 1,2 | 456 | EDABLCF | 1,2 | 353 |
| 9 | FLDABCE | 1,2 | 452 | EDABLCF | 1,2 | 392 |
| 10 | FLABCDE | 1,2 | 396 | EDABLCF | 1,2 | 329 |
| 11 | FLACDBE | 1,2 | 483 | EDABLCF | 1,2 | 318 |
| 12 | DLBCEFA | 1,2 | 359 | EDABLCF | 1,2 | 316 |
| 13 | FLDCBAE | 1,2 | 405 | EDABLCF | 1,2 | 368 |
| 14 | FLDBCEA | 1,2 | 328 | EDABLCF | 1,2 | 271 |
| 15 | ELDCBAF | 2,1 | 373 | EDABLCF | 1,2 | 340 |
| 16 | FLABCDE | 2,1 | 365 | EDABLCF | 1,2 | 347 |
| 17 | FLABCDE | 1,2 | 340 | EDABLCF | 1,2 | 285 |
| 18 | ELCABDF | 1,2 | 474 | EDABLCF | 1,2 | 365 |

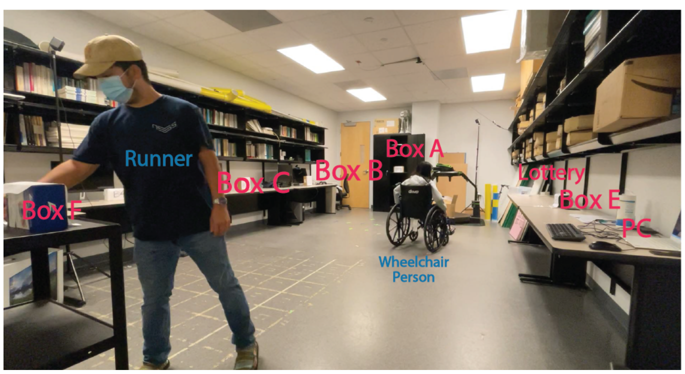



Fig. 17. A screenshot of the workplace simulation.

## 10.2 Workspace and Workplan Synthesis

To validate our approach, we synthesized a feasible workspace and workplan design for this user study. We mainly considered efficiency and workload for this mini warehouse optimization. We set the wheelchair person to have a walking speed of $0.7\text{ms}^{-1}$ and a low intolerance to walking and turning. In contrast, we set the runner person to have a walking speed of $1.25\text{ms}^{-1}$ and a high intolerance in walking and turning. We also estimate the typing speed for barcodes with different lengths in the simulation. We assume that typing a three-digit barcode takes 1.25s and typing a six-digit barcode takes 2.50s.

Figure 15 shows the workspace design generated by our approach. To minimize walk and turn effort of both agents, our synthesized result let the wheelchair person do Task 1, and put the lottery on the opposite side of the PC. The analysis section contains more discussion.

## 10.3 Analysis

We measure the completion time for each condition of experiment, comparing the participants' designs with the design generated by our approach. The completion times and the participants' workspace designs are shown in Table 3. Most participants assigned the wheelchair person to do Task 1 and the runner to do Task 2.

We performed paired t-test to compare completion time under the participants' design and the generated design. There was a significant difference in completion time for the participants' design (414s ± 56s) and the generated design (351s ± 41s); $t(17)=6.84$, $p < 0.001$. It is important to note that the average completion time following the generated plan is lower than that of the participants' designs.

This relative large amount of time difference is probably due to the fact that the participants underestimated or neglected the turning difficulty of the wheelchair person in their planning. In the survey, some participants revealed that they put boxes that contained a large amount of objects near the PC station so as to shorten the travel distance of the runner. They also aimed to minimize the distance between the lottery box and the PC station for the wheelchair person. In contrast, our generated design considered the turning effort for the wheelchair person and put the lottery box opposite to the PC station.

Please refer to our supplementary video for the actual workplace simulation comparison.

## 11 ADDITIONAL DETAILS OF EVALUATION ON WORKPLACE DESIGN

As described in the main paper, we invited 15 participants to design fast food kitchen workspaces, which are compared with our synthesis results. All participants have some layout design background. They are aged between 20 to 45; 60% of them are males and 40% are females. We asked the participants to use our application to design an aesthetic and efficient workspace for a given fast food kitchen work-plan. Figure 25 shows the screenshot of our application. Staff properties and list of tasks they need to complete are shown on the right. During the desing process, they could run a simulation to see how the staff agents worked in the current workspace. The evaluation results are shown in the lower right of the application. Figure 26 shows screenshot of their design and Table 13 show their simulation results.

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. Table 14 show simulation results.

### 11.1 Discussion of Workplace Design Evaluation

We surveyed our participants about the usage of our application. Many of them think our simulation model help them better understand how staff agents will interact at the workplace. Instead of estimating performance metrics manually, they could obtain these measurements precisely from our simulation model. For improvement, one participant suggested generating several variations of the current configuration emphasizing different objectives. He believed that adding this one functionality to our tool will speed up the refinement process.

With customized human behaviors, our tool can help designers demonstrate and discuss their designs more effectively. They believed this could save them lots of time in revising their design that targets customer needs.

(a) Vegetable Fryer (b) French Fry Rack (c) Fries Incubator (d) Bun Pan Rack (e) Bun Grill Toaster (f) Freezer (g) Grill Station

(h) Cooked Food Incubator (I) Burger-making Table (j) Drink & Food Prep (k) Register

Fig. 18. Fast food kitchen equipment.

Table 4. Equipment used for each task in the fast food kitchen example.

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

Table 5. Fast food order

| | #Drinks | #Fries | #Burger |
|---|---|---|---|
| Order 1 | 1 | 2 | 1 |
| Order 2 | 3 | 2 | 2 |
| Order 3 | 3 | 3 | 3 |


Fig. 19. Supermarket equipment.

Table 6. Equipment used for each task in the supermarket example.

| Tasks | Equipment Used |
|---|---|
| (1) Serve at cashier | (a) Register |
| (2) Go around shelves | (b)-(k) Shelves |
| (3) Serve at seafood counter | (l) Seafood |
| (4) Serve at bakery counter | (m) Bakery |
| (5) Serve at butcher counter | (n) Butcher |

Table 7. Supermarket customer shopping lists.

| | Customer 1 | Customer 2 | Customer 3 | Customer 4 | Customer 5 |
|---|---|---|---|---|---|
| Food Section | Bakery | Butcher,<br>Seafood | Bakery | Butcher | Butcher,<br>Bakery |
| Item Type | Fruit,<br>Dairy Products,<br>Personal Care,<br>Drinks,Dry Goods,<br>Toys | Electronics,<br>Dairy Products | Personal Care,Drinks,<br>Dry Goods,Toys | Electronics,<br>Pet Food,<br>Dry Goods,<br>Toys, Fruit,<br>Dairy Products | Fruit,<br>Dairy Products |
| #Item Wanted | 1,2,2,2,1,3,2 | 1,1,2,2 | 1,3,3,1,2 | 1,2,2,2,1,1,3,1 | 1,2,2,2 |
| | Customer 6 | Customer 7 | Customer 8 | Customer 9 | Customer 10 |
| Food Section | Bakery,<br>Butcher | Butcher,<br>Seafood | Butcher | Butcher,<br>Seafood | Bakery |
| Item Type | N/A | Fruit,<br>Dairy Products,<br>Personal Care,<br>Drinks | Electronics,<br>Dry Goods,<br>Toys | N/A | N/A |
| #Item Wanted | 1,1 | 1,3,2,1,1,1 | 1,4,3,3 | 1,3 | 1 |


Fig. 20. Restaurant equipment.

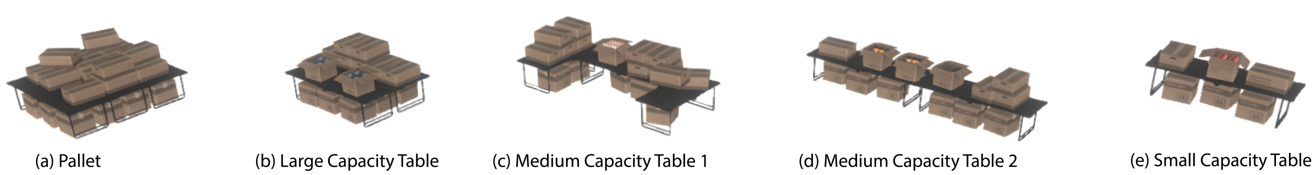
Table 8. Equipment used for each task in the restaurant example.

| **Tasks** | **Equipment Used** |
|---|---|
| (1) Serve customer | (c)-(e) Tables |
| (2) Go around with cart | (f) Cart |
| (3) Serve at food counter | (a) Food Counter |
| (4) Checkout with customer | (b) Checkout Counter |

Table 9. Details of customer groups' orders in the restaurant example.

| | Group 1 | Group 2 | Group 3 | Group 4 | Group 5 | Group 6 | Group 7 | Group 8 |
|---|---|---|---|---|---|---|---|---|
| #People | 1 | 3 | 2 | 11 | 3 | 6 | 4 | 2 |
| #Dim sum | 1 | 2 | 2 | 7 | 2 | 5 | 3 | 1 |
| #Fried noodle | 1 | 2 | 2 | 4 | 2 | 5 | 3 | 3 |
| | Group 9 | Group 10 | Group 11 | | | | | |
| #People | 10 | 4 | 7 | | | | | |
| #Dim sum | 6 | 2 | 5 | | | | | |
| #Fried noodle | 6 | 2 | 1 | | | | | |

Fig. 21. Donation center equipment.



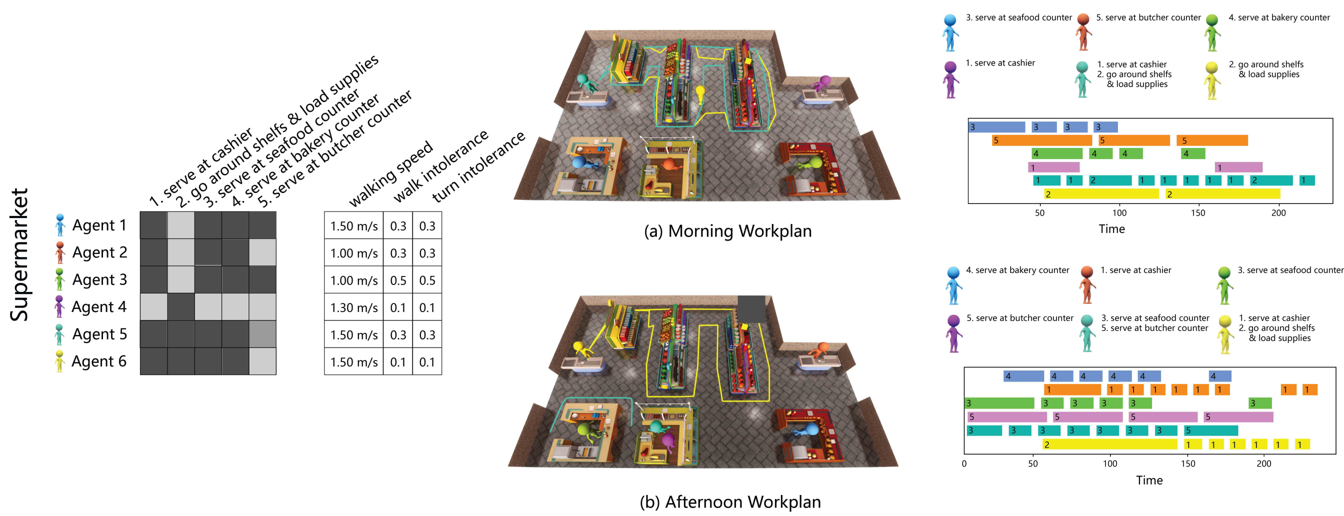



Fig. 22. Additional details of the dynamic workplan example.

Table 10. Details of supermarket customers' shopping lists in the morning. In general, customers shop shelf items more often than going to the food section.

| | Customer 1 | Customer 2 | Customer 3 | Customer 4 | Customer 5 |
|---|---|---|---|---|---|
| Food Section | Seafood | Seafood, Bakery | Butcher | N/A | N/A |
| ItemType | N/A | Fruit, Books, Sport, Electronics | Dairy Products, Pet Food, Personal Care, Drinks, Dry Goods, Toys | Dairy Products, Pet Food, Personal Care, Drinks, Dry Goods, Toys | Fruit, Books, Sport, Electronics |
| #Item Wanted | 1 | 1,3,2,2,4,1 | 1,3,1,3,4,1,2 | 1,1,5,1,3,2 | 1,1,4,3 |
| | Customer 6 | Customer 7 | Customer 8 | Customer 9 | Customer 10 |
| Food Section | Bakery | Butcher, Seafood | Butcher, Bakery | Bakery | Seafood |
| ItemType | Dairy Products, Pet Food, Personal Care, Drinks, Dry Goods, Toys | N/A | N/A | Fruit, Books, Sport, Electronics | |
| #Item Wanted | 1,3,2,5,3,4 | 1,1 | 1,1 | 1,1,2,3,2,1 | 1 |

Table 11. Supermarket customers' shopping lists in the afternoon. Since seafood and meat are on sale, customers tend to go to either of these sections to buy food.

| | Customer 1 | Customer 2 | Customer 3 | Customer 4 | Customer 5 |
|---|---|---|---|---|---|
| Food Section | Seafood | Seafood,Bakery | Seafood | Seafood,Butcher | Seafood,Bakery |
| ItemType | N/A | Fruit,Books, Sport, Electronics | Dairy Products, Pet Food, Personal Care, Drinks, Dry Goods, Toys | | |
| #Item Wanted | 1 | 1,1,2,1,3,1 | 1,3,1,1,2,1,3 | 1,1 | 1,1 |
| | Customer 6 | Customer 7 | Customer 8 | Customer 9 | Customer 10 |
| Food Section | Seafood,Butcher | Butcher,Seafood | Butcher,Bakery | Bakery | Seafood,Bakery |
| ItemType | | | | | |
| #Item Wanted | 1,2 | 1,1 | 1,1 | 1 | 1,2 |
| | Customer 11 | Customer 12 | Customer 13 | Customer 14 | Customer 15 |
| Food Section | Seafood | Seafood,Butcher | Seafood,Bakery | Seafood | Seafood |
| ItemType | N/A | N/A | N/A | N/A | N/A |
| #Item Wanted | | 1,2 | 1,3 | 3 | 1 |

Table 12. Customer orders in the robot assistant example. In general, the more people a group has, the more food (either fried noodles or dim sum) the group will order.

| | Group 1 | Group 2 | Group 3 | Group 4 | Group 5 | Group 6 | Group 7 | Group 8 |
|---|---|---|---|---|---|---|---|---|
| #People | 1 | 3 | 2 | 11 | 3 | 6 | 4 | 2 |
| #Dim sum | 1 | 2 | 2 | 4 | 2 | 5 | 3 | 1 |
| #Fried noodle | 1 | 2 | 2 | 6 | 2 | 3 | 4 | 2 |
| | Group 9 | Group 10 | Group 11 | | | | | |
| #People | 10 | 4 | 7 | | | | | |
| #Dim sum | 3 | 2 | 5 | | | | | |
| #Fried noodle | 6 | 2 | 1 | | | | | |

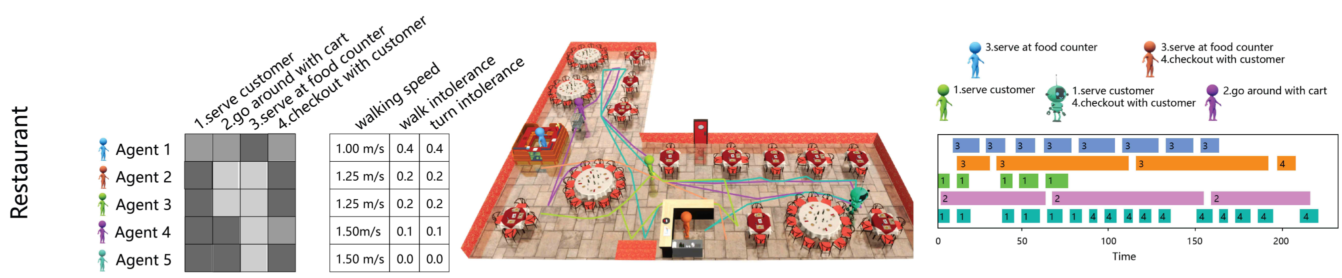



Fig. 23. Details of inputs and results of Robot Assistant scene. Our optimizer assigned the robot with tasks that involve high movement to relieve the workload of the other staff agents since the robot has a high walking speed of $1.50\text{ms}^{-1}$ and a walk intolerance set as zero.

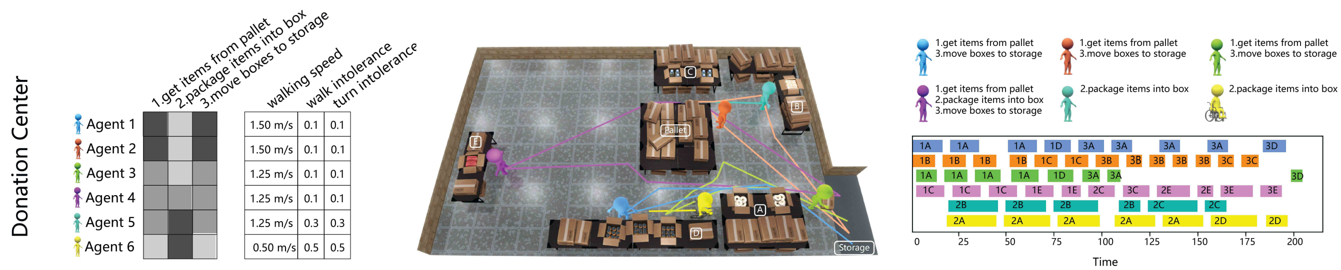



Fig. 24. Details of inputs and results of the Physical Challenged Staff Member example.

Table 13. Evaluation results. After the participants finished their design, we run simulations on all the workspaces using the Unity game engine to obtain the simulation time, total walk distance, and total body rotation. We also record number of moves, rotations, and evaluations required for creating one design. We also run AnyLogic simulations to compute the simulation times.

| #Move | #Rotation | # Evaluation | AnyLogic Simulation Time (s) | Unity Simulation Time (s) | Total Walk Distance (m) | Total Body Rotation (deg) |
|---|---|---|---|---|---|---|
| 261 | 61 | 24 | 83 | 98 | 50.86 | 1,148 |
| 153 | 32 | 19 | 96 | 113 | 71.51 | 1,145 |
| 249 | 73 | 3 | 94 | 111 | 62.63 | 1,242 |
| 227 | 65 | 12 | 83 | 98 | 50.82 | 1,280 |
| 42 | 27 | 2 | 92 | 109 | 66.21 | 1,380 |
| 118 | 37 | 10 | 86 | 102 | 50.41 | 1,348 |
| 232 | 79 | 24 | 94 | 111 | 63.07 | 1,165 |
| 104 | 23 | 7 | 95 | 112 | 72.65 | 926 |
| 88 | 12 | 5 | 82 | 97 | 55.58 | 1,302 |
| 169 | 39 | 14 | 91 | 107 | 64.21 | 1,374 |
| 59 | 46 | 9 | 92 | 109 | 66.68 | 1,157 |
| 83 | 54 | 4 | 87 | 103 | 61.35 | 946 |
| 106 | 24 | 3 | 87 | 103 | 64.29 | 1,278 |
| 95 | 15 | 9 | 85 | 100 | 62.51 | 965 |
| 174 | 63 | 11 | 93 | 110 | 58.91 | 1,370 |

Table 14. Simulation results of the 15 workspaces synthesized by our workspace optmization given the same workplan used in the evaluation.

| AnyLogic Simulation Time (s) | Unity Simulation Time (s) | Total Walk Distance (m) | Total Body Rotation (deg) |
|---|---|---|---|
| 86 | 102 | 50.84 | 1,157 |
| 85 | 100 | 52.83 | 1,101 |
| 82 | 97 | 57.91 | 1,109 |
| 85 | 100 | 48.80 | 890 |
| 92 | 108 | 57.15 | 975 |
| 86 | 101 | 59.45 | 995 |
| 90 | 106 | 58.14 | 911 |
| 81 | 96 | 47.21 | 963 |
| 83 | 98 | 62.11 | 1,119 |
| 81 | 96 | 49.42 | 940 |
| 85 | 100 | 52.36 | 1,388 |
| 90 | 106 | 53.76 | 1,008 |
| 82 | 97 | 50.06 | 965 |
| 86 | 102 | 54.28 | 961 |
| 85 | 100 | 64.74 | 896 |

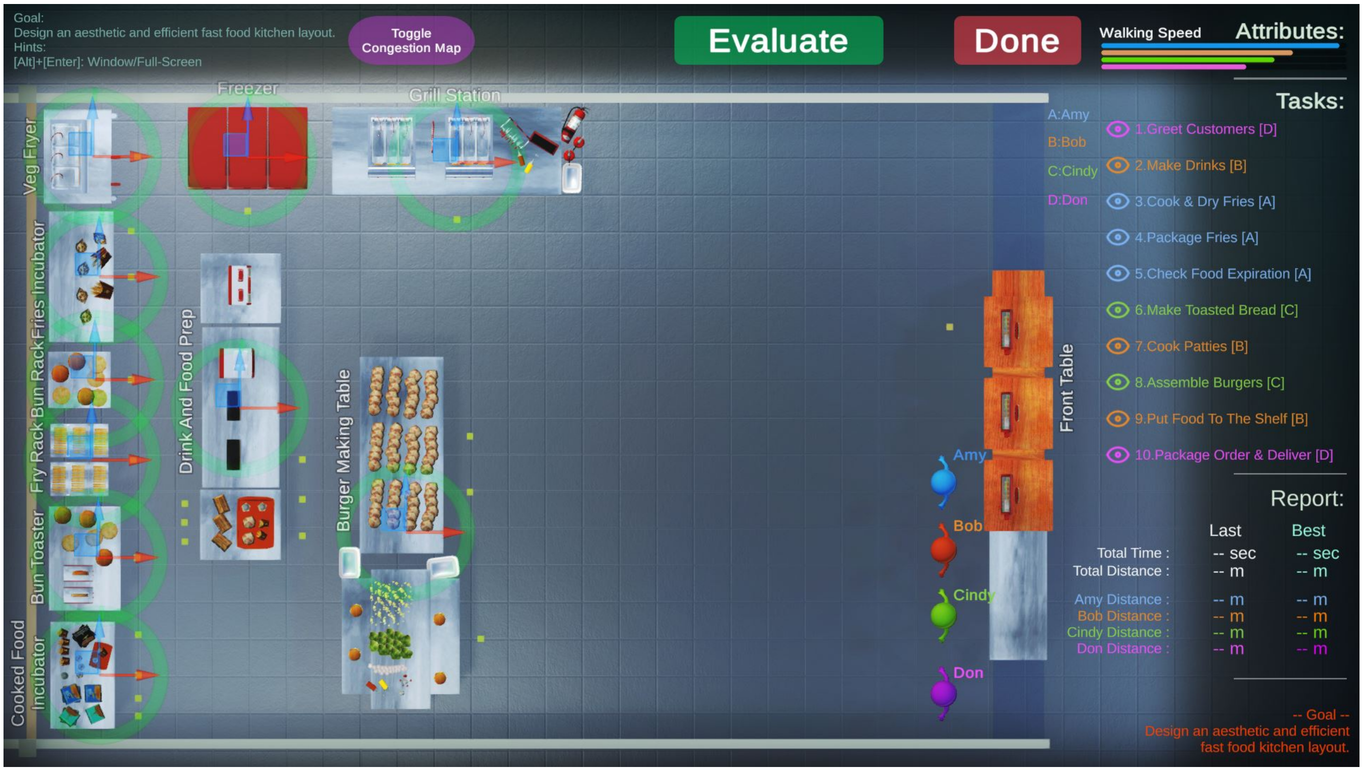



Fig. 25. Screenshot of the application used in the evaluation. Staff properties and the list of tasks assigned to the staff agents are shown on the right. The evaluation results are shown at the lower right of the application.

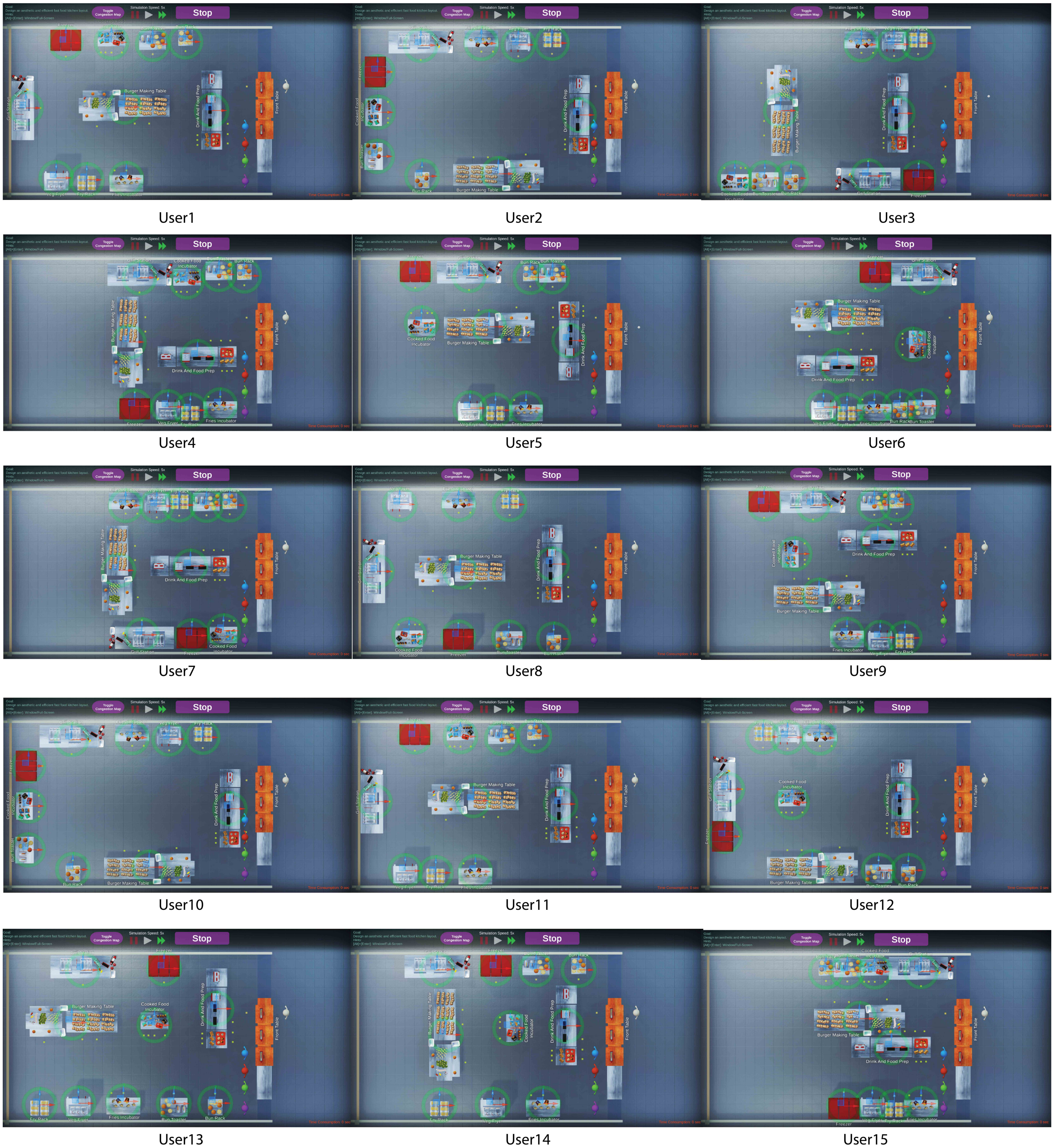



Fig. 26. Screenshots of the participants' designs in the evaluation.