This article has been accepted for publication in IEEE Robotics and Automation Letters. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/LRA.2024.3519863

Improving Human-Robot Collaboration via Computational Design

Jixuan Zhi[1] and Jyh-Ming Lien[2]

***Abstract*—When robots enter our day-to-day lives, the shared space surrounding humans and robots is critical for facilitating Human-Robot collaboration. The design of shared space should satisfy humans' preferences and robots' efficiency. This work uses the kitchen as an example to illustrate the importance of good space designs in enhancing collaboration. Given the kitchen boundary, food stations, counters, and recipes, the proposed method determines the optimal placement of stations and counters that meet the requirements of kitchen design rules and improve performance. The key technical challenge is that the optimization method usually evaluates thousands of designs, and each evaluation analyzes the traffic flow of the space, which must solve many motion planning problems. To address this technical challenge, we use a decentralized motion planner that can solve multi-agent motion planning efficiently. Our results indicate that optimized kitchen designs can provide noticeable performance improvement to Human-Robot collaboration.**

***Index Terms*—Service Robotics, Human-Aware Motion Planning, Simulation and Animation**

I. Introduction

As Human-Robot collaboration becomes more prevalent, the design of spaces for these tasks is crucial for effective and safe interaction. Although most robots are developed for humans in the existing spaces, we foresee that the common space will evolve in the coming decades as humans and robots work closer together. This paper discusses a space design problem that enhances Human-Robot collaboration.

Although space optimization is not new, previous work has overlooked tasks requiring close human-robot collaboration and behavioral adaptation, and the role of optimizing physical spaces for collaboration has not been explored. Behavioral adaptation allows robots to respond to human actions in real time, and environmental optimization ensures that the space is designed to facilitate collaboration. This paper combines them together to improve task efficiency and enhance safety in human-robot environments.

Inspired by our day-to-day chores, making meals requires humans to work together. Similar to the gameplay in Overcooked [1], we use kitchen scenarios to evaluate the proposed ideas. Specifically, cooking tasks consist of organized subtasks, allowing humans and robots to work concurrently; and a better design of the kitchen should improve collaboration. These improvements can be measured by the travel distance, path intersections, and task completion time. Fig. 1 presents a comparison between two kitchens designed with and without Human-Robot collaboration taken into account. The choice of a kitchen scenario serves as proof for broader applications where the dynamics of space can impact collaboration.

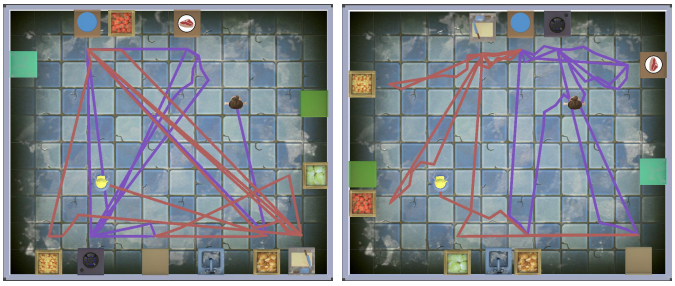


Fig. 1: In this example, a human and a robot are tasked to make two burgers together. The kitchen layout on the left leads to worse performance than that on the right generated by the proposed method. The red and purple lines represent the paths of the robot and the human, respectively.

The kitchen design problem for Human-Robot collaboration can be formulated as an optimization problem. In this framework, the decision variables are the layout of counters, and the paths taken by the robot and human to finish a meal together. Hence the objective function encodes layout information and path quality as the two most important factors to measure the performance of the collaboration. The key technical bottleneck is the optimization method that usually invokes the motion planner thousands of times during the evaluation step. Therefore, efficient methods for motion planning are important.

The main contribution of this work is adapting and integrating established methods to optimize shared spaces in Human-Robot Collaboration (HRC) environments. This integrated framework ensures that both spatial constraints (layout of counters) and dynamic constraints (robot-human motion paths) are considered simultaneously. The proposed framework can generate kitchen designs that enhance Human-Robot collaboration in a reasonable amount of time.

II. Related Work

While most papers focused on the design of different spaces, there has been a limited effort in designing Human-Robot collaboration space [2], [3]. Table I highlights the key advantages of the proposed framework compared to existing methods for space design in enhancing collaboration.

Manuscript received: July, 16, 2024; Revised October, 23, 2024; Accepted November, 26, 2024.

This paper was recommended for publication by Editor Gentiane Venture upon evaluation of the Associate Editor and Reviewers' comments.

[1]Jixuan Zhi is with the Department of Computer Science, George Mason University, 4400, University Drive MSN 4A5, Fairfax, VA 22030 USA, jzhi@gmu.edu

[2]Jyh-Ming Lien is an independent scholar, lienjyhming@gmail.com

Digital Object Identifier (DOI): see top of this page.

© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Robotics and Automation Letters. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/LRA.2024.3519863

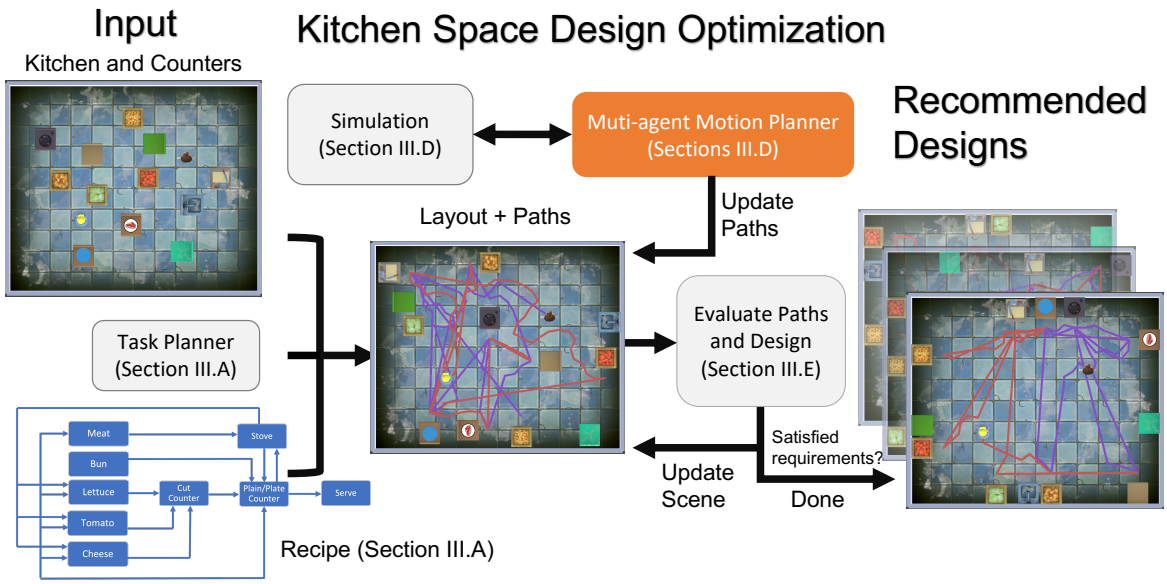


Fig. 2: The overview of the proposed optimization method for kitchen space design. Given the recipes of meals that human and robot will make together, our method generates kitchen layouts that enhance the Human-Robot collaboration.

A. Computational Method for Space Design

Researchers formulated the space design as optimization problems. Yu et al. [4] developed an optimization framework to change the layout. Zhang et al. [5] proposed a computational method that can design a workspace and a workplan together. Researchers also applied learning-based methods to synthesize scenes [6] [7].

Some papers focus on the interaction between scenes and actions [8] [9]. Action graph [10], activity-associated object relation graphs [11], stochastic grammar [12], and diffusion model [13] are introduced to generate scenes.

Some works discuss the relationship between functionality and space design. Researchers proposed frameworks for workspace design to enhance task performance [14] [15]. Ribino et al. [16] used agent-based simulations to improve warehouse performance. Chaeibakhsh et al. [17] optimized hospital room layouts to reduce the risk of patient falls. Researchers also explored the relationship between workspace design and workers' health [18] [19], and patient safety [20].

Unlike these approaches, our work specifically targets the optimization of space to facilitate and enhance HRC.

B. Human-Robot Collaboration in the Existing Spaces

Bauer et al. [21] proposed a Human-Robot collaboration framework involving perception, intention estimation, joint intention, action planning, and joint action. Alami et al. [22] proposed a decision-making framework for robot control that considers humans. Carroll et al. [23] trained the agents by self-play and human data in a kitchen environment and studied Human-Robot collaboration.

However, their studies are limited to behavioral adaptation rather than environmental optimization. Our research takes a different approach by emphasizing the redesign and optimization of the environment to enhance collaboration.

C. Multi-Agent Communication and Motion Planning

Human-robot collaboration needs communication and planning. Communication includes explicit and implicit communication [24]. For implicit communication, the agent infers another agent's intention by observing their actions. This concept can be integrated into Human-Robot collaboration, allowing robots to infer human intentions [25] [26] [27]. A data-driven model [28] and a method based on online clustering [29] have been developed for communication recently.

In multi-agent motion planning, centralized planner plans in joined configuration space with exponential computational cost; decentralized planning methods, like prioritized planning [30], are more piratical. In this approach, the agent with higher priority is planned first, other agents are then planned by treating the first as a moving obstacle.

These studies overlook the role of the physical environment in improving collaboration. Our work addresses this by exploring how environmental design supports communication and planning in Human-Robot collaboration.

III. Optimization Framework of Kitchen Design

Fig. 2 sketches the optimization framework that uses the designed functionality to analyze the layout. The functionality is defined by a set of meals provided. Given a list of meals intended for collaborative work, the task planner chooses respective sub-tasks for each during the optimization. Subsequently, a motion planner generates individualized paths for each of these sub-tasks. To ensure the validity of paths, a simulator assesses the paths using collision detection. The framework evaluates the quality of the design by a cost function encoding the layout information and path quality metric which measures the collaboration.

© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Robotics and Automation Letters. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/LRA.2024.3519863

TABLE I: Benchmarking of Proposed Framework versus Existing Approaches

Feature	Proposed Framework	Existing Frameworks
Integrated Space and Path Optimization	Combines space layout design with motion planning for robots to optimize both simultaneously	Focuses on either space layout or robot path optimization, not both. [8], [10], [11], [15]
Decentralized Multi-Agent Motion Planning	Uses decentralized algorithms for efficient multi-agent collaboration	Single agent path planning without considering collision or no path planning for robots [2], [3], [5]
Real-Time Adaptability to Human Behavior	Incorporates real-time feedback to adjust robot movements based on human actions	Limited or no real-time adaptability to human behavior [5], [22]

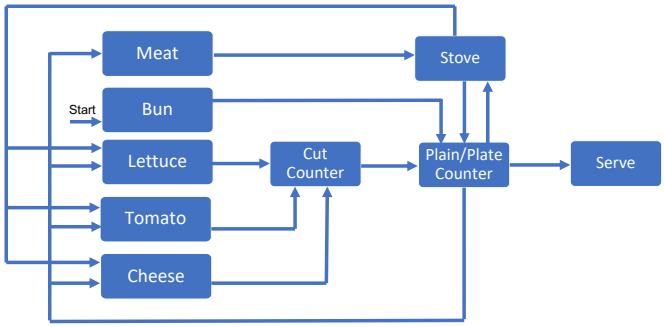


Fig. 3: The recipe and sub-tasks for making a burger

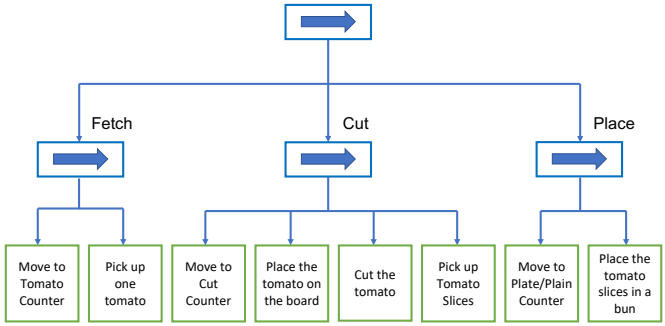


Fig. 4: The behavior tree of tomato cutting task

A. Recipe and Task Planning

Collaborative tasks must be delegated appropriately. Take a burger as an example: essential components include a bun and meat, while cheese, lettuce, and tomato are optional. Making a burger involves bun preparation, meat grilling, and adding optional ingredients. A sub-task, such as grilling meat, also comprises multiple steps. We use two burgers for simplicity in the illustration. Humans and robots alternate in selecting sub-tasks and collaboratively preparing the burgers. Fig. 3 gives an example of making a whole burger and the counters that need to be visited. Beginning with the bun, the agent can select sub-tasks like preparing meat or adding sides. Multiple arrows signify the multiple possibilities available for selection. Once meat grilling commences, the agent's subsequent tasks only focus on side preparation.

B. Agent Behaviour Design

For agent behavior, we integrate two techniques: finite state machine and the behavior tree [31]. The combination allows us to model the complex interaction between humans and robots. The finite state machine defines states and actions, and the behavior tree is a directed tree structure with different node types, directing the agent's actions from root to leaf nodes. The finite state machine validates paths for human and robot movements in a virtual simulator, as detailed in III-D. The behavior tree is employed in a real simulator to illustrate human-robot collaboration. Fig. 4 delineates a behavior tree for a tomato cutting task, comprising "Fetch", "Cut", and "Place" sub-tasks. The right-pointing arrows represent sequential nodes in the behavior tree, guiding the sequential task execution.

C. Touring the Kitchen

Cooking with a recipe needs a human or robot to traverse various kitchen counters to complete each sub-task. We only associate one valid configuration with each counter, and the configurations are near the counter and have an orientation to ensure the agent is oriented toward the counter. Given the configurations of each counter $\{c_i, c_j, ..., c_k\}$ and a counter visit order for each sub-task, we intend to determine a path that encompasses all counters with no collision.

D. Multi-agent Motion Planning

We outline the multi-agent motion planner, which models both the human and robot as analogous agents, using Rapidly-exploring Random Trees (RRT) [32] search policy.

Overview. Fig. 5 presents the overview of motion planning with a decentralized policy. The task planner chooses the sub-task for human and robot randomly, then the robot may image the paths of the human when needed. We randomly selected tasks to assess the system's adaptability in varied scenarios without fixed sequences. In addition, this collaboration scenario falls within the domain of human-aware motion planning and we need to prioritize the safety of the human. Therefore, we employ prioritized planning, specifically, we plan the human path initially, then treat the human as a moving obstacle during the robot motion planning. We assume identical speeds for the human and robot during the simulation and assign a specific duration time for the cooking task.

Four States in Simulation. In the simulation, a finite state machine with four states is incorporated: Moving, Engaged at a Counter, Requiring New Task, and Idle. When an agent is in the "Requiring New Task" state, a new sub-task is randomly selected. Each sub-task needs several paths and operations, correlating to the 'Moving' and 'Engaged at a Counter' states. When no remaining sub-tasks are identified, they are in the "Idle" state. Once the system verifies the completion of all sub-tasks and success in pathfinding, the results are stored for subsequent comparison. The pause for simulations has several reasons: needing a new task for the agent, needing re-planning for the robot to avoid a collision, and reporting failure when collision avoidance is unfeasible.

© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Robotics and Automation Letters. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/LRA.2024.3519863

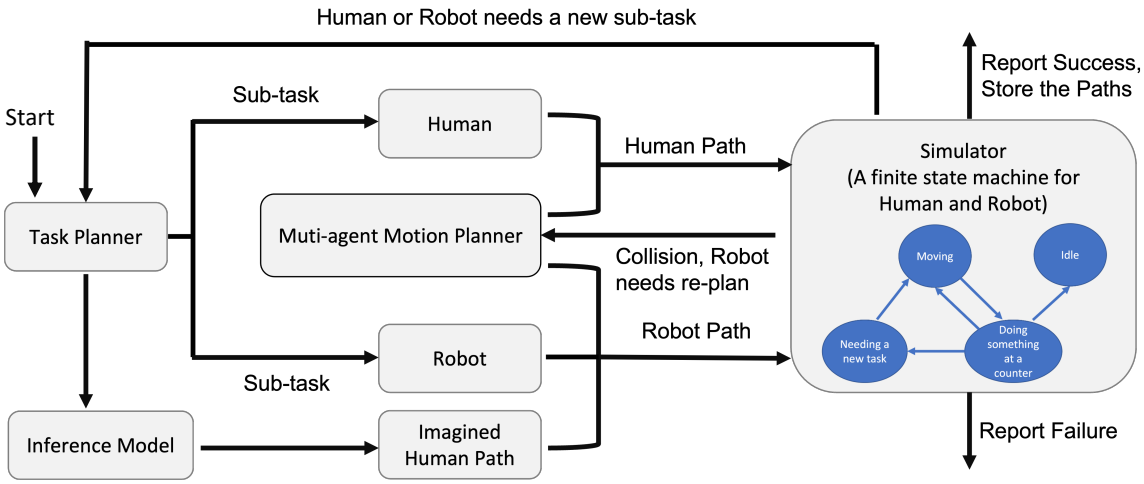


Fig. 5: An overview of multi-agent motion planning

Planning Human Motion. Given the higher priority of the human, the task planner initially assigns a sub-task to the human, and the motion planner generates a path using a modified RRT method with a time component. This implies that state x is represented as $x = (q, t)$, where q denotes the configuration, and t represents the time component.

The method starts by generating a random configuration with an 80% probability of choosing the goal configuration. It then finds the nearest tree node using Euclidean distance and connects it to this random configuration. If the goal configuration is selected, the planner adds the entire path segment until it encounters the invalid configuration. If not, up to five path steps are added to the tree [33]. Subsequently, the planner only adds collision-free segments and nodes into the tree by collision detection. This process is repeated until the search tree reaches the goal configuration. Each valid node is assigned a unique timestamp based on its expansion steps from the root during expansion and the root node is assigned a default zero. When the agent is proximate to the counter, additional time is required to complete cooking tasks. This time cost is also incorporated into the timestamp of the node. Finally, we find a valid path for the human to traverse counters and complete the assigned sub-task.

Planning Robot Motion. Now the planner needs to pick a sub-task and find a valid path for the robot. Here we consider multi-agent coordination with two methods: reactive planning and a probability inference model [25].

The inference model allows the robot to infer human intentions without communication. Initially, we assign equal probability to all sub-tasks. As each sub-task is completed, the system updates the probabilities of the remaining sub-tasks. By observing the human's position, the robot can infer the human's next step. For example, when a human is at the stove counter with three incomplete sides, each side has an equal selection chance. The robot predicts the human's choice, allowing the planner to form a virtual path with timestamps. By treating this path as a moving obstacle, the robot plans its own path. However, our experiments indicate that planning robot motion based on an imaged human path results in a 20% increase in time cost compared to planning the robot motion directly. This discrepancy arises because the virtual human path may affect the robot's path even if the real human path does not. Consequently, we only consider the virtual path when the simulator reports a collision.

While planning the robot's motion, simulations are necessary to check the collision. The simulator operates on a global time scale and the RRT expansion maintains the same step for both agents. The global timestamp of each node is found for collision detection. Upon detecting a collision, the simulation is paused, and the robot employs reactive planning to find a new collision-free path with the RRT method and a virtual human path. The simulation is then resumed to check the new path. If the planner cannot resolve the collision, the simulator reports failure and stops the simulation.

Iteration. After the robot or human completes a sub-task, the inference model is updated. The task planner selects a new sub-task for human or robot. This process continues until there are no tasks left or a failure is reported.

Discussion. RRT was selected because it can explore high-dimensional spaces, and adapt to dynamics and multi-agent environments, although it doesn't always find the shortest path. While A* guarantees the shortest path, it is computationally expensive in continuous spaces and less flexible in dynamic environments. D* improves on A* with incremental updates but with a higher computational cost. Future work will compare RRT with A* and D* in HRC environments.

E. Evaluating a Space Design

We evaluate kitchen design quality through two criteria: counter placement and path quality for task completion. The criteria are encoded as cost terms. Specifically, The cost of design can be written as C_{total}:

$$C_{total} = \mathbf{C_L}\mathbf{w_L^T} + \mathbf{C_P}\mathbf{w_P^T}\ , \tag{1}$$

where $\mathbf{C_L}$ and $\mathbf{C_P}$ are vectors of layout costs and paths costs, and $\mathbf{w_L}$ and $\mathbf{w_P}$ represent vectors of weights, and $\mathbf{w_L}$ is determined by user-defined preferences. The result is a dot product. A smaller C_{total} means a better space design.

© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Robotics and Automation Letters. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/LRA.2024.3519863

1) Kitchen layout cost: This cost measures the quality of counter placement. The placements can be described by position and orientation. We assume all the counters are aligned along the wall.

Distance Cost C_L^d. Each counter has a user-defined distance to its closest wall. We define the cost here:

$$C_L^d = \sum_{\forall i \in C} (||c_i - w|| - d_i)^2 \ , \tag{2}$$

where c_i is the position of counter i, w is the nearest wall position to the c_i, and d_i refers to the target distance.

Rotation Cost C_L^r. We assume each counter only has four directions and should have the same direction as its nearest wall. Thus we define the rotation cost as follows:

$$C_L^r = \sum_{\forall i \in C} (||\theta_i - \theta_w||)^2 \tag{3}$$

where θ_i is the orientation of the counter i , the θ_w is the orientation of the nearest wall to the counter i.

Kitchen Layout Cost Weight $\mathbf{w_L}$. The weights $\mathbf{w_L}$ are defined as $\mathbf{w_L} = \alpha[w_L^d, w_L^r]$, where w_L^d, w_L^r that reflects the significance of the distance and rotation in the framework. We set α, w_L^d, w_L^r to be 1 here.

2) Path Cost: For path cost, we have the formula $\mathbf{C_P} = [C_P^l, C_P^n, C_P^t]$, which measures the length of the path, the narrowness of the path, and task completed time along this path. A path is considered superior if it is shorter and wider, with less completion time along this path. Every cost term is a scale from 0 to 1. Suppose there is a path consisting of N nodes and p_i is the position of the i-th node.

Path Length Cost C_P^l. For path length, we have human length and robot length with the same weight 0.5. We add up the distances between successive nodes along the path:

$$C_P^l = \sum_{i=1}^{N-1} ||p_{i+1} - p_i|| \ . \tag{4}$$

Path Narrowness Cost C_P^n. We use narrowness cost to describe the minimum width along the path.

$$C_P^n = 2 * \min\left(||p_i - q_i||, ||p_i - w_i||\right) \ . \tag{5}$$

For each node with position p_i, the orientation of p_i is directed towards the next node p_{i+1}, then we can find the closest object of the node on the left and right sides, with positions denoted as q_i and w_i respectively.

Path Time Cost C_P^t. We focus on the completed time of the meal, and then the timestamp of the submission counter is the path time cost. This cost considers the path length and the task assignment. If a task plan is unbalanced between the human and the robot, the completion time will be longer because one agent will be busy while the other remains idle.

$$C_P^r = T_{finished} \ . \tag{6}$$

Path Cost Weight $\mathbf{w_P}$. The weights can be defined as $\mathbf{w_P} = [w_P^l, w_P^n, w_P^t]$, where w_P^l, w_P^n, w_P^t are parameters representing the importance of the path length narrowness and time. For simplicity, We set w_P^l, w_P^n, w_P^t to be $1/3$.

3) Weighting Scheme Discussion: Different environments may prioritize different weights. In fast-paced settings like assembly lines, minimizing task time might matter more than path length or spacing. In healthcare, safety and comfort could be prioritized by ensuring wider paths for navigation. Increasing the weight on path length cost would prioritize shorter paths, leading to a layout where objects are placed closer together, reducing the time spent moving around. Giving more weight to path narrowness would focus on safety, lowering the chances of collisions. Prioritizing task completion time would speed up task execution, but it may create layouts that allow quick task transitions but might not use the space as effectively.

F. Design Optimization

The simulated annealing algorithm [34] with the Metropolis criterion [35] is utilized for minimizing costs. An objective function for a current state (P, L) is defined:

$$f(P, L) = exp(-\frac{1}{t} C_{total}(P, L)) \ , \tag{7}$$

where P is the set of all paths, L is the placement of all counters, and t is the temperature. The optimization method has two steps, layout optimization and task path optimization. In every step, the framework makes a change from the current state (P, L) to a new state (P', L) or (P, L'). We define two types of moves in the layout part: (1) change the position and orientation of a counter, and (2) swap two counters. For the path part, the planner generates a new order of tasks and paths. Then we have a new design L', or a new set of paths P'. The framework accepts the new state with a probability:

$$Prob(newConf|oldConf) = min(1, \frac{f(newConf)}{f(oldConf)}) \ . \tag{8}$$

By tuning parameters such as an initial temperature of 500 and a linear cooling schedule (decreasing by 1 per step, 1 iteration each step), the algorithm balances layout optimization and path optimization.

G. Solution Set

The linear weighted sum method might not balance layout and path costs. To address this, we implemented the Non-dominated sorting method [36], which sorts solutions based on dominance levels. We maintained a fixed-size (5) Pareto set. In each iteration, if a new solution dominates any in the Pareto set, we replace the dominated ones. When the set is full, we remove the solution with the highest path cost.

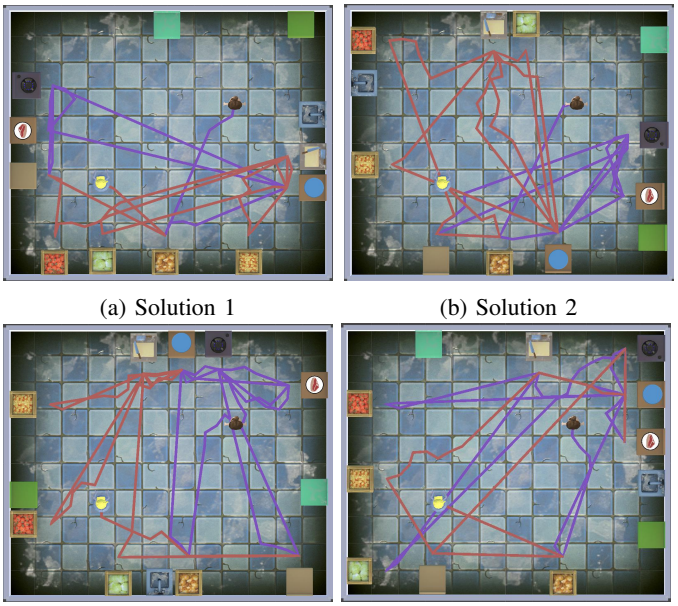
IV. Experiments and Result for Kitchen Design

In this work, no pre-existing simulation datasets were used. Instead, we created custom environments focused on optimizing HRC in kitchen layouts. We implemented the framework and the motion planner in the Unity game engine. For replication concerns, we conduct the experiments in a 12×10 space, the coordinate of the room center is $(0, 0)$, and the robot and human are fixed at $(2.5, 1.5)$ and $(-2.5, -1.5)$ respectively, we have 11 different counters for arrangement,

© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Robotics and Automation Letters. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/LRA.2024.3519863

each counter has the same size $1 \times 1 \times 1$. The experiments were repeated 10 times for each room configuration and got a total of 50 Pareto optimal solutions. The experiments are in a MacBook Pro laptop with Apple M1 Pro and 16 GB memory.



(a) Solution 1 (b) Solution 2

(c) Solution 3 (d) Solution 4

Fig. 6: Four optimized kitchen designs. The red and purple lines represent the paths of the robot and the human, resp.

A. Results Analysis

Two burgers are prepared: one with all four ingredients; and the other with beef only. There are seven sub-tasks: bun-1, meat-1, tomato-1, lettuce-1, cheese-1, bun-2, and meat-2. Fig. 6 shows four different kitchen layouts, while Fig. 7 shows the different path and layout costs across these configurations. $PathLenCost$ is the average of $humanLenCost$ and $robotLenCost$; $pathCost$ is the average of $pathLenCost$, $pathNarCost$ and $pathTimeCost$; $TotalCost$ is the sum of $pathCost$ and $layoutCost$. The task planning selections are shown in Table II.

Given the same tasks, Solution 1 prefers a short robot path, whereas Solution 2 prefers a short human path. Both Solution 3 and Solution 4 strike a balance between robot and human path length. The figure shows that path narrowness costs are zero for all solutions, indicating that path narrowness may not be a significant parameter. In addition, Solution 4 has the highest path cost among the four solutions. This difference may arise due to distinct task plans for the human and the robot. See Table II. Our findings indicate that 72% of the task plans resemble Solutions 1, 2, and 3, such a task plan facilitates the finding of optimal layouts and paths.

B. Comparison with Random Configurations

We compare the proposed method to kitchens with random layouts. We randomly generate the kitchen layout until the layout cost is less than a threshold, then perform the task planning and pathfinding in the random layout with the defined recipes. Fig. 1 presents a comparison between two kitchens: A random one (on the left) and an optimized one (on the right) with the proposed method. Given the same tasks, the right layout exhibits better path quality, signifying a more conducive environment for collaboration.

TABLE II: Task Planning for Each Solution

	Human Task	**Robot Task**
Solution 1	bun-1, meat-1, meat-2	bun-2, tomato-1, cheese-1, lettuce-1
Solution 2	bun-1, meat-1, meat-2	bun-2, tomato-1, cheese-1, lettuce-1
Solution 3	bun-1, meat-2, meat-1	bun-2, lettuce-1, cheese-1, tomato-1
Solution 4	bun-1, meat-1, lettuce-1, tomato-1	bun-2, cheese-1, meat-2

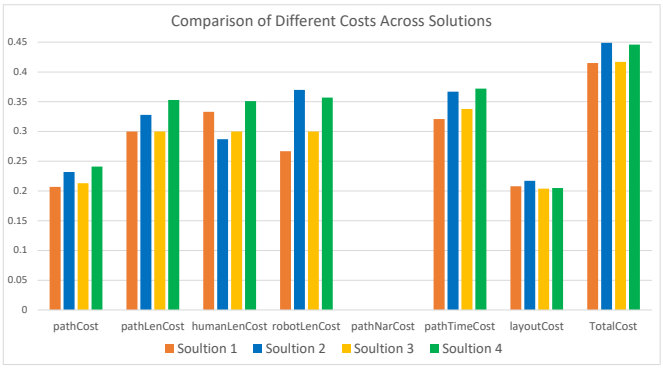


Fig. 7: different costs of four designs in Fig. 6

In Fig. 8, a comparison between the proposed method and randomly generated kitchen layouts is illustrated through various path and layout costs. Except for path narrowness cost and layout cost, the random configurations consistently yield over 10% higher costs, indicating inferior path quality of random layouts, while the proposed method can rule out the layouts that can not lead to high-quality paths.

We also performed a t-test and computed confidence intervals for cost metrics. The results are shown in Fig. 9. Except for path narrowness cost, where most data points are zero, all other metrics show significant differences. The proposed method achieves a better balance between path and layout

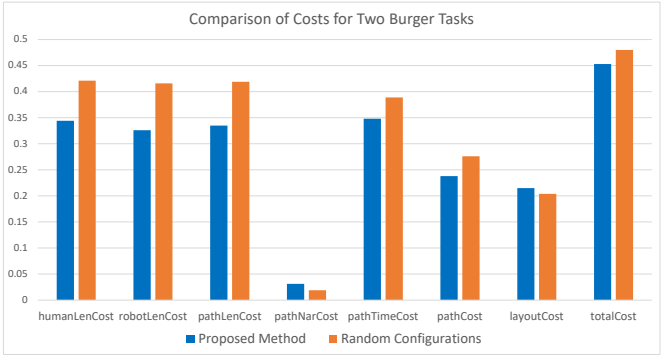


Fig. 8: Comparison of costs for the same two burger tasks

© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Robotics and Automation Letters. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/LRA.2024.3519863

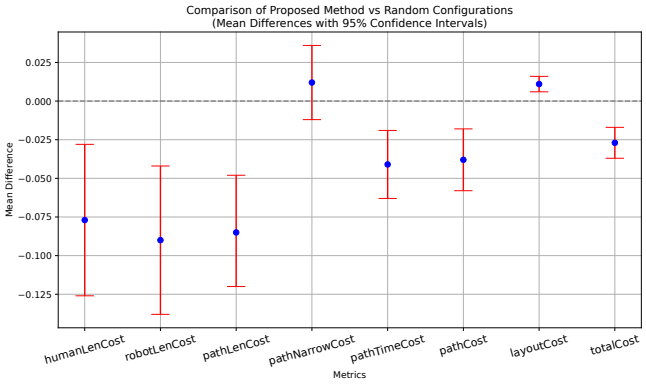


Fig. 9: Comparison of Proposed Method vs Random Configurations (Mean Differences with 95% Confidence Intervals)

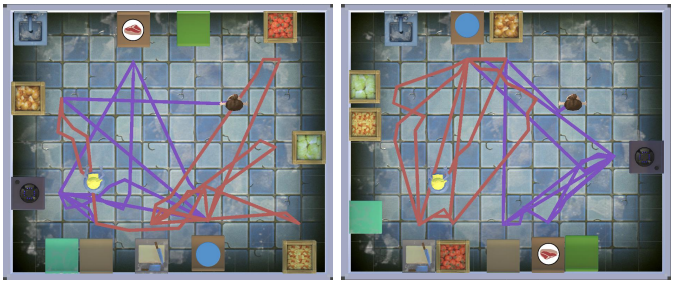


Fig. 10: Two kitchen designs with a smaller area

costs with significantly lower total costs.

We conducted a runtime analysis and found that the total runtime for the proposed framework ranged from 3 to 4 minutes. Motion planning tasks accounted for about 95% of the total computation time. We plan to reuse the paths from earlier iterations and the RRT tree to reduce computation time.

C. Design a Small Kitchen

To demonstrate the robustness of the proposed method, we apply the method to design a more compact kitchen and an L-shaped room with the same task. For the compact space, the boundary length of the kitchen is reduced by 20%. Fig. 10 shows two layouts of a compact kitchen designed for making two burgers. The designs closely resemble those of the larger kitchen. Fig. 12 shows the different types of costs in various kitchens with the same task. In every case, the costs of the smaller room are higher than the costs of the regular room, indicating the challenges of working within a limited space.

D. Design an L-shaped Kitchen

Fig. 11 shows two different layouts of an L-shape of this kitchen intended for two burgers. Fig. 12 shows that the total path cost of an L-shaped kitchen is higher than the costs of a regular kitchen given the same task, especially for path narrow cost. This is because the path is more constrained in a non-convex space. The L-shape room result shows lower layout costs and total costs, indicating that the method can adapt to complex room shapes effectively. Overall, Fig. 12 shows that the proposed method provides consistent costs for several categories across different room configurations.

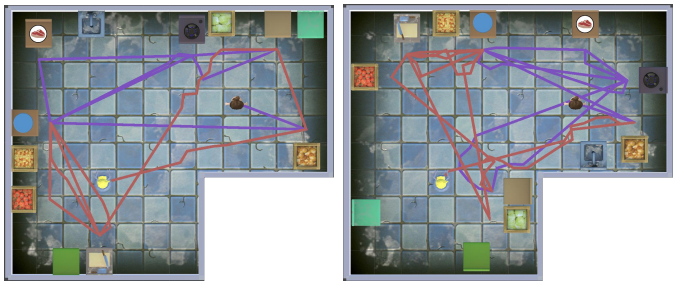


Fig. 11: Two kitchen designs with an L-shape room

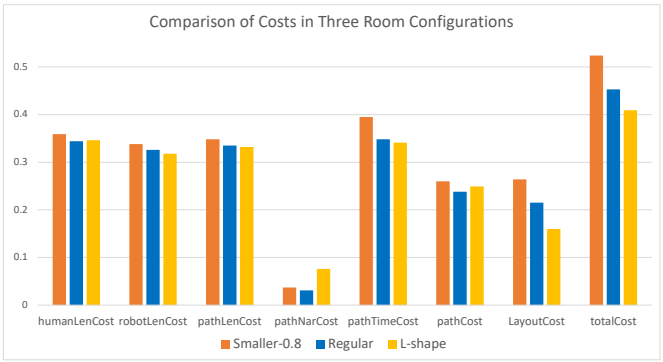


Fig. 12: Comparison of costs in three room configurations: Smaller-0.8, Regular, and L-shape.

E. Qualitative analysis

This section provides qualitative analysis to explain strategy performance in specific contexts. In dynamic collaboration scenarios, Task completion time is important. In poorly optimized layouts, humans and robots often have to wait for each other, leading to idle times. The framework's ability to optimize counter placements and task assignments ensures efficient task execution without unnecessary delay.

V. Generalizability of the Proposed Framework

While the proposed optimization framework has been demonstrated in a kitchen setting, its principles can extend to various environments. In industrial settings, such as warehouses and assembly lines, the framework can optimize the layout of workstations, equipment, and pathways to enhance task execution, minimize human-robot interference, and improve the overall flow of operations. For example, in a warehouse, optimizing the placement of inventory shelves, workstations, and robotic routes could reduce travel time for both humans and robots. Similarly, in an assembly line, designing the layout of robot workstations and human-operated machinery could reduce idle time, and ensure that both humans and robots are operating efficiently.

VI. Conclusion, Limitation, and Discussion

We developed a computational method for kitchen layout design that improves Human-Robot collaboration. Using a decentralized motion planner, we efficiently coordinate human and robot movements. Results show that our method creates practical layouts that boost productivity in challenging

© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Robotics and Automation Letters. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/LRA.2024.3519863

tasks and consistently delivers similar designs across various scenarios. One limitation is that the design efficiency is tied to specific tasks; if the tasks change, the room configuration needs to be recalculated. Additionally, the proposed method has only been applied to kitchen design. We aim to include other scenarios, like industrial environments. Finally, identifying the optimal weights for path cost and layout cost could be explored in future work.

The framework is efficient for kitchen-sized environments, but larger rooms or more complex tasks may increase computational demands. To address this, hierarchical optimization can be applied by dividing the layout into smaller sections for localized optimization and then combining them into a complete design. Additionally, modifying the RRT algorithm with adaptive sampling to prioritize areas with high collision risks can further improve efficiency. In addition, task allocation is also important. Future research should focus on developing algorithms that personalize task assignments based on individual preferences, use probabilistic models to handle uncertainty, and improve multi-agent coordination.

REFERENCES

[1] G. T. Games, "Overcooked," 2016. 1

[2] J. Zhi, L.-F. Yu, and J.-M. Lien, "Designing human-robot coexistence space," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7161–7168, 2021. 1, 3

[3] W. Wang, Z. Zhao, Z. Jiao, Y. Zhu, S.-C. Zhu, and H. Liu, "Rearrange indoor scenes for human-robot co-activity," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 11 943–11 949. 1, 3

[4] L. F. Yu, S. K. Yeung, C. K. Tang, D. Terzopoulos, T. F. Chan, and S. J. Osher, "Make it home: automatic optimization of furniture arrangement," *ACM Transactions on Graphics (TOG)-Proceedings of ACM SIGGRAPH 2011, v. 30,(4), July 2011, article no. 86*, vol. 30, no. 4, 2011. 2

[5] Y. Zhang, H. Huang, E. Plaku, and L.-F. Yu, "Joint computational design of workspaces and workplans," *ACM Transactions on Graphics (TOG)*, vol. 40, no. 6, pp. 1–16, 2021. 2, 3

[6] Z. Zhang, Z. Yang, C. Ma, L. Luo, A. Huth, E. Vouga, and Q. Huang, "Deep generative modeling for scene synthesis via hybrid representations," *ACM Transactions on Graphics (TOG)*, vol. 39, no. 2, pp. 1–21, 2020. 2

[7] J. Liu, Z. Qiu, L. Wang, P. Liu, G. Cheng, and Y. Chen, "Intelligent floor plan design of modular high-rise residential building based on graph-constrained generative adversarial networks," *Automation in Construction*, vol. 159, p. 105264, 2024. 2

[8] M. Fisher, M. Savva, Y. Li, P. Hanrahan, and M. Nießner, "Activity-centric scene synthesis for functional 3d scene modeling," *ACM Transactions on Graphics (TOG)*, vol. 34, no. 6, pp. 1–13, 2015. 2, 3

[9] C. Li and L.-F. Yu, "Generating activity snippets by learning human-scene interactions," *ACM Transactions on Graphics (TOG)*, vol. 42, no. 4, pp. 1–15, 2023. 2

[10] R. Ma, H. Li, C. Zou, Z. Liao, X. Tong, and H. Zhang, "Action-driven 3d indoor scene evolution," *ACM Transactions on Graphics (TOG)*, vol. 35, no. 6, pp. 1–13, 2016. 2, 3

[11] Q. Fu, X. Chen, X. Wang, S. Wen, B. Zhou, and H. Fu, "Adaptive synthesis of indoor scenes via activity-associated object relation graphs," *ACM Transactions on Graphics (TOG)*, vol. 36, no. 6, pp. 1–13, 2017. 2, 3

[12] S. Qi, Y. Zhu, S. Huang, C. Jiang, and S.-C. Zhu, "Human-centric indoor scene synthesis using stochastic grammar," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5899–5908. 2

[13] Y. Yang, B. Jia, P. Zhi, and S. Huang, "Physcene: Physically interactable 3d scene synthesis for embodied ai," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 16 262–16 272. 2

[14] A. Kämpf-Dern and J. Konkol, "Performance-oriented office environments–framework for effective workspace design and the accompanying change processes," *Journal of Corporate Real Estate*, 2017. 2

[15] W. Liang, J. Liu, Y. Lang, B. Ning, and L.-F. Yu, "Functional workspace optimization via learning personal preferences from virtual experiences," *IEEE transactions on visualization and computer graphics*, vol. 25, no. 5, pp. 1836–1845, 2019. 2, 3

[16] P. Ribino, M. Cossentino, C. Lodato, and S. Lopes, "Agent-based simulation study for improving logistic warehouse performance," *Journal of Simulation*, vol. 12, no. 1, pp. 23–41, 2018. 2

[17] S. Chaeibakhsh, R. S. Novin, T. Hermans, A. Merryweather, and A. Kuntz, "Optimizing hospital room layout to reduce the risk of patient falls," in *Proceedings of the 10th International Conference on Operations Research and Enterprise Systems - ICORES*, 2021, pp. 36–48. 2

[18] O. Broberg, "Workspace design: a case study applying participatory design principles of healthy workplaces in an industrial setting," *International Journal of Technology Management*, vol. 51, no. 1, pp. 39–56, 2010. 2

[19] M. M. Robertson, Y.-H. Huang, M. J. O'Neill, and L. M. Schleifer, "Flexible workspace design and ergonomics training: Impacts on the psychosocial work environment, musculoskeletal health, and work effectiveness among knowledge workers," *Applied ergonomics*, vol. 39, no. 4, pp. 482–494, 2008. 2

[20] A. Hundt, P. Carayon, and C. Alvarado, "Reducing workload and increasing patient safety through work and workspace design," Tech. Rep., 2003. 2

[21] A. Bauer, D. Wollherr, and M. Buss, "Human–robot collaboration: a survey," *International Journal of Humanoid Robotics*, vol. 5, no. 01, pp. 47–66, 2008. 2

[22] R. Alami, A. Clodic, V. Montreuil, E. A. Sisbot, and R. Chatila, "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*, 2005, pp. 81–85. 2, 3

[23] M. Carroll, R. Shah, M. K. Ho, T. Griffiths, S. Seshia, P. Abbeel, and A. Dragan, "On the utility of learning about humans for human-ai coordination," *Advances in Neural Information Processing Systems*, vol. 32, pp. 5174–5185, 2019. 2

[24] T. Balch and R. C. Arkin, "Communication in reactive multiagent robotic systems," *Autonomous robots*, vol. 1, pp. 27–52, 1994. 2

[25] S. A. Wu, R. E. Wang, J. A. Evans, J. B. Tenenbaum, D. C. Parkes, and M. Kleiman-Weiner, "Too many cooks: Bayesian inference for coordinating multi-agent collaboration," *Topics in Cognitive Science*, vol. 13, no. 2, pp. 414–432, 2021. 2, 4

[26] M. Shum, M. Kleiman-Weiner, M. L. Littman, and J. B. Tenenbaum, "Theory of minds: Understanding behavior in groups through inverse planning," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, no. 01, 2019, pp. 6163–6170. 2

[27] X. Puig, T. Shu, S. Li, Z. Wang, Y.-H. Liao, J. B. Tenenbaum, S. Fidler, and A. Torralba, "Watch-and-help: A challenge for social perception and human-ai collaboration," in *International Conference on Learning Representations*, 2021. 2

[28] Y. Qin, M. Kumon, and T. Furukawa, "A data-driven multiple model framework for intention estimation," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 6458–6464. 2

[29] J. Chen, T. Lan, and C. Joe-Wong, "Rgmcomm: Return gap minimization via discrete communications in multi-agent reinforcement learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 16, 2024, pp. 17 327–17 336. 2

[30] J. P. Van Den Berg and M. H. Overmars, "Prioritized motion planning for multiple robots," in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2005, pp. 430–435. 2

[31] M. Colledanchise and P. Ögren, *Behavior trees in robotics and AI: An introduction*. CRC Press, 2018. 3

[32] S. M. LaValle, "Rapidly-exploring random trees: A new tool for path planning," 1998. 3

[33] J. J. Kuffner and S. M. LaValle, "Rrt-connect: An efficient approach to single-query path planning," in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, vol. 2, 2000, pp. 995–1001. 4

[34] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *science*, vol. 220, no. 4598, pp. 671–680, 1983. 5

[35] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *The journal of chemical physics*, vol. 21, no. 6, pp. 1087–1092, 1953. 5

[36] N. Srinivas and K. Deb, "Muiltiobjective optimization using nondominated sorting in genetic algorithms," *Evolutionary computation*, vol. 2, no. 3, pp. 221–248, 1994. 5

© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.