Optimal Multi-Robot Path Planning for Trash Pick and Drop in Hospitals

Ratan Lal
School of Computer Science and Information Systems
Northwest Missouri State University
Maryville, USA
email: rlal@nwmissouri.edu

Rehaman Naguru Abdur
School of Computer Science and Information Systems
Northwest Missouri State University
Maryville, USA
email: s544913@nwmissouri.edu

***Abstract*—In this paper, we consider a hospital environment, where each patient's room is very contagious. Hence, we consider the problem of picking trash from each patient's room and dropping it in a big container through multiple robots. Here, we assume that all the robots are small and can pick only one trash bag at a time. Our main objective is to find a plan for the robots that can minimize the total consumed energy (distance, time). Our broad approach is to express the environment in the form of a graph and reduce the problem as an instance of the Multiple Traveling Salesman problem. Then, we encode the reduced problem into the Mixed-Integer Linear Programming (MILP) and solve the encoding using the MILP solver. Next, we perform our approach for hospitals of varied sizes and pick-drop tasks. Our experimental results show that our method is scalable. Finally, we simulate an execution of the optimal plan in the Virtual Robot Experimentation Platform (V-REP) simulator.**

***Index Terms*—multi robots; path planning; trash pick and drop; mixed integer linear programming.**

I. Introduction

Path planning [32] is a well-known problem which is widely used in various applications, such as task spanning [10], evacuation [24], search and rescue [2, 3, 13], coverage [1, 4, 23], precision weeding [31], pesticide spraying [8, 17, 18], transportation in the hospital [21, 28] and medicine delivery [14, 15, 22]. The planning algorithm has been extended from single mobile robot to multiple robots through different techniques, such as cell decomposition approaches [11, 12, 23, 26, 35], potential field approaches [7, 27, 29], and road map approaches [5, 6, 16]. Although different aspects of multi-robot path planning for a hospital have been explored, there is a pressing need of multi-robot for picking dustbin bags from the patient's room and dropping them into a big container to avoid contagious diseases, such as COVID-19.

In this paper, we consider a multi-robot path planning for trash picking and dropping in a hospital environment motivated by a real need. Here, we want to use multiple robots to pick trash bags from the desired patient rooms and drop them into a big container. We assume that robots are small and can collect the garbage from only one dustbin bag at a time. Our objective is to find a plan for the robots that covers all the desired trash bags while minimizing the total distance traveled by all the robots. Hence, this is an optimization problem.

Our broad approach is to reduce the problem into an instance of the Multiple Traveling Salesman (MTS) problem. The MTS problem is a well-known problem where given a graph between cities, multiple salesmen need to visit all the cities exactly once and return to their initial position with the minimum traveling distance. Our approach for reduction is based on the shortest distance graph algorithm. First, we transform a given hospital environment $\mathcal{E}$ into a weighted graph $\mathcal{G}_{\mathcal{E}}$. In the graph $\mathcal{G}_{\mathcal{E}}$, we capture all the valid line segments of the environment where robots could move. Note that the size of $\mathcal{G}_{\mathcal{E}}$ depends on the types of hospitals. Here, we consider three types of hospitals, namely, small, medium and large. Next, we capture each pick and drop task as a pair of one trash bag and one big container. Then, we transform the graph $\mathcal{G}_{\mathcal{E}}$ into another weighted graph $\mathcal{G}_{\mathcal{E}}^{\Upsilon}$ based on a given set of tasks Υ. In the graph $\mathcal{G}_{\mathcal{E}}^{\Upsilon}$, we capture only robots' initial locations, all the desired trashes and containers' location and create the following edges: (a) edges from robots' location to trashes' location; (b) edges between trashes' location and containers' location. We compute the weight of the edges by applying the shortest distance graph algorithm on $\mathcal{G}_{\mathcal{E}}$. Note that the size of the graph $\mathcal{G}_{\mathcal{E}}^{\Upsilon}$ depends on the number of tasks. Also, for the MTS problem, each city (vertex) must be visited exactly once by one of the salesmen. However, in the graph $\mathcal{G}_{\mathcal{E}}^{\Upsilon}$, a robot may need to visit the same container more than once. So, we need to ensure that each vertex corresponding to a big container is visited exactly once by one of the robots. Therefore, we transform $\mathcal{G}_{\mathcal{E}}^{\Upsilon}$ into another graph $\mathcal{G}_M$, where if a big container is common among multiple tasks, we create a copy of the vertex corresponding to the container and add the respective edges to the copied vertex. Next, we introduce a dummy vertex to $\mathcal{G}_M$ for the robots to return to their initial location. Finally, we encode the reduced problem into the Mixed-Integer Linear Programming (MILP) similar to the one given in [19].

We have implemented our method in the Python toolbox for finding the optimal plan for the robots, where we have used the NetworkX tool for the graph construction and the GNU Linear Programming Kit (GLPK) to solve the MILP encoding for an instance of the MTS problem. Next, we extract a real optimal plan for the robots from a solution returned by the GLPK solver and by applying the shortest path graph

 ISBN: 978-1-61208-964-5

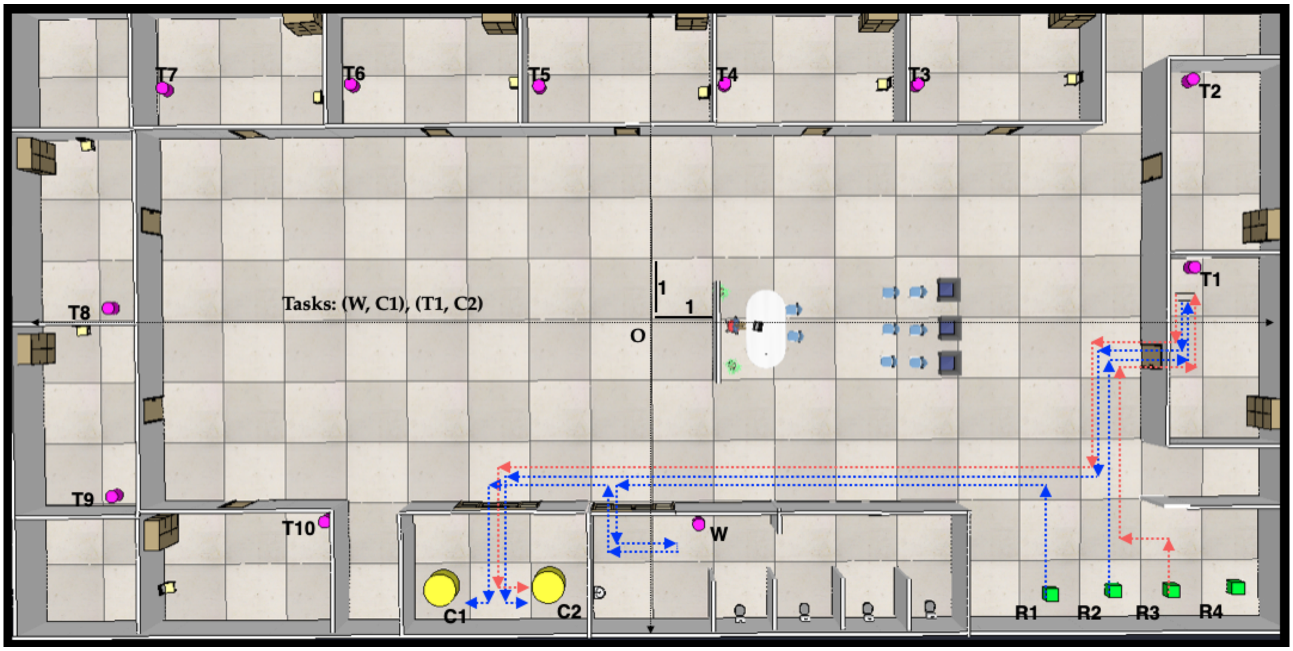

Figure 1: Hospital Environment

algorithm on $\mathcal{G}_{\mathcal{E}}$. Finally, we deploy the plan in the Virtual Robot Experimentation Platform (V-REP) simulator [25].

Our main contributions of the paper are given below.

- We have presented a task-based graph reduction for reducing the pick and drop problem into an instance of the MTS problem.
- Since the task-based graph depends on the tasks, the size of encoding is less and the GLPK solver returns faster.
- We have simulated the plan in the V-REP simulator.

Remark 1: We use the collision avoidance protocol integrated with the V-REP simulator while executing the plan.

II. Related Work

Multi-robot path planning has been explored for hospitals. Different techniques, such as sub-dimensional expansion [30], Integer Linear Programming [33], Artificial Potential Field (AFP) [29], and Enhanced Genetic Algorithm (EGA) [20] have been explored.

In this paper, we explore selected trash pick and drop tasks for the hospitals. Some of the research works in the area of trash collection have been explored. A prototype for the garbage collection based on Convolutional Neural Networks (CNNs) has been studied [9], where a CNN is integrated with a robot to detect and classify different types of garbage. However, our work is focused on finding an optimal path for the robots to pick the trash from patient's room and drop into a big container. Although the pick and drop problem has been studied in urban settings [34], there are limited works in this direction. Hence, further investigation is needed to avoid contagious diseases.

III. Motivation

In this section, we demonstrate the problem by considering a hospital environment shown in Figure 1, where there are eleven small dustbin containers, namely, T1, T2, ..., T10 located in each patient's room and W situated in the washroom represented by magenta color; two big trash containers, namely, C1 and C2 represented by yellow color; and four robots, namely, R1, R2, R3, and R4 represented by green color. Assume that each container has a small dust bag, and each (small) robot can pick only one bag at a time. Note that all the bags do not always have dust. Hence, we consider the selected pick and drop problem, where we want the robots to pick the bags from only required dustbins and drop them into the big containers. For example, in Figure 1, we want the robots to pick the bag from washroom W and drop it into C1 and pick the bag from patient dustbin T1 and drop it into C2. For the problem, each pair of pick and drop points is considered a task. For example, (W, C1) is a task.

Here, given a list of tasks (W, C1), (T1, C2), we want to find a path for each robot to solve the selected pick and drop problem. Note that multiple paths may exist for the robots. For example, for the tasks, the following two paths solve the problem, namely, (a) blue path from R1 and R2; (b) blue path from R1 and red path from R3. Hence, our objective is to find a path for each robot such that the total distance traveled by all the robots can be minimized.

IV. Preliminaries

Notations: Let $\mathbb{R}_{\geq 0}$, $\mathbb{R}$ and $\mathbb{N}$ denote the set of positive real numbers, the set of real numbers, and the set of natural numbers, respectively. We use $[n]$ to denote the set

 ISBN: 978-1-61208-964-5

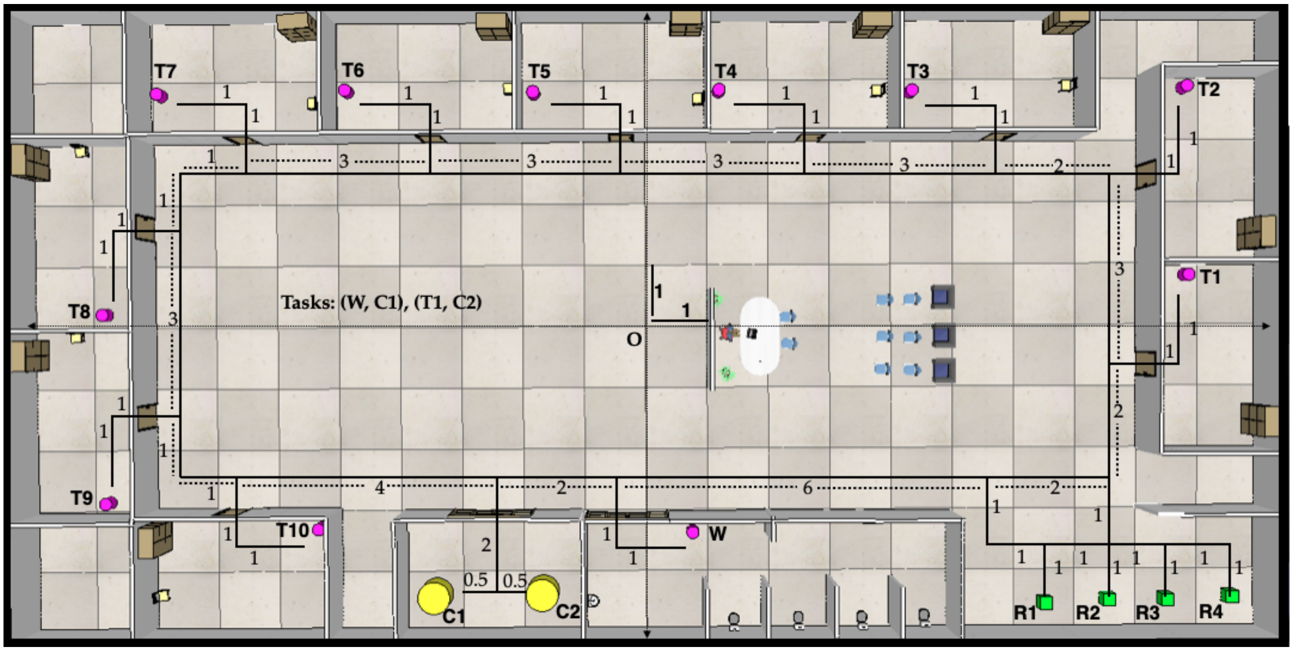

Figure 2: Hospital Environment ($\mathcal{E}$)

$\{1, 2, \ldots, n\}$. Given a set $\mathcal{S}$, we use $|\mathcal{S}|$ to denote the number of elements in $\mathcal{S}$.

Euclidean and Manhattan Distance: Given two 2−dimensional points $p_1 = (x_1, y_1)$ and $p_2 = (x_2, y_2)$, the Euclidean distance between p_1 and p_2 denoted by $d_E(p_1, p_2)$, is defined as follows:

$$d_E(p_1, p_2) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}.$$

The Manhattan distance between p_1 and p_2 denoted by $d_M(p_1, p_2)$, is defined as follows:

$$d_M(p_1, p_2) = |x_1 - x_2| + |y_1 - y_2|.$$

Weighted Graph: A weighted graph is defined as a tuple $\mathcal{G} = (V_0, V, E, W)$ where

- V is a set of vertices;
- $V_0 \subseteq V$ is a set of initial vertices;
- $E \subseteq V \times V$ is a set of edges;
- $W : E \to \mathbb{R}_{\geq 0}$ is a weight function that captures the length for each edge.

A path denoted by σ for a given weighted graph $\mathcal{G}$ is a sequence of vertices $v_0, v_1, v_2, \ldots, v_n \in V$ such that $(v_{i-1}, v_i) \in E$ for $i \in [n]$. We use $cost(\sigma)$ to denote the cost of the path σ, that is,

$$cost(\sigma) = \sum_{i=1}^{n} W(v_{i-1}, v_i).$$

Complete Paths: Given a weighted graph $\mathcal{G}$, a complete path is a set of paths $\varrho = \{\sigma_i\}_{i=1}^{n}$ satisfying the following conditions:

- $n = |V_0|$;
- for each path $\sigma_i = v_0^i, v_1^i, \ldots, v_m^i$, $v_0^i \in V_0$.

Succinctly, we use $cost(\varrho)$ to denote the cost of the complete path ϱ, that is,

$$\sum_{\sigma \in \varrho} cost(\sigma).$$

V. Problem Formulation

In this section, we formally describe the pick and drop problem. First, we define an environment for the hospitals given as below.

Definition 1: [Environment] An environment is a tuple $\mathcal{E} = (\mathcal{R}, \mathcal{T}, \mathcal{C}, Edges, \mathcal{D})$, where

- $\mathcal{R} \subseteq \mathbb{R}^2$ is a set of robots' initial location;
- $\mathcal{T} \subseteq \mathbb{R}^2$ is a set of dustbins' location;
- $\mathcal{C} \subseteq \mathbb{R}^2$ is a set of big trash containers' location;
- $Edges \subseteq \mathcal{P} \times \mathcal{P}$ where $\mathcal{P} = \mathcal{R} \cup \mathcal{T} \cup \mathcal{C}$, that captures a set of edges on which robots could move;
- $\mathcal{D} : Edges \to \mathbb{R}_{\geq 0}$ is a distance function that determines the length for each edge.

Example 1: Consider the hospital environment shown in Figure 2. It can be represented as a tuple $\mathcal{E} = (\mathcal{R}, \mathcal{T}, \mathcal{C}, Edges, \mathcal{D})$, where

- $\mathcal{R} = \{(6.5, -4.5), (7.5, -4.5), (8.5, -4.5), (9.5, -4.5)\}$;
- $\mathcal{T} = \{(8.5, 0.5), (8.5, 3.5), (4.5, 3.5), (1.5, 3.5), (-1.5, 3.5), (-4.5, 3.5), (-7.5, 3.5), (-8.5, 0.5), (-8.5, -2.5), (-5.5, -3.5), (0.5, -3.5)\}$;
- $\mathcal{C} = \{(-3.5, -4.5), (-1.5, -3.5)\}$;
- $Edges$ are all the pairs of end points of each solid black line segment shown in Figure 2;
- For each line segment $e \in Edges$, $\mathcal{D}(e)$ is shown in Figure 2. For example, $\mathcal{D}((6.5, -4.5), (6.5, -3.5)) = 1$.

Next, we define a pick and drop task given as below:

 ISBN: 978-1-61208-964-5

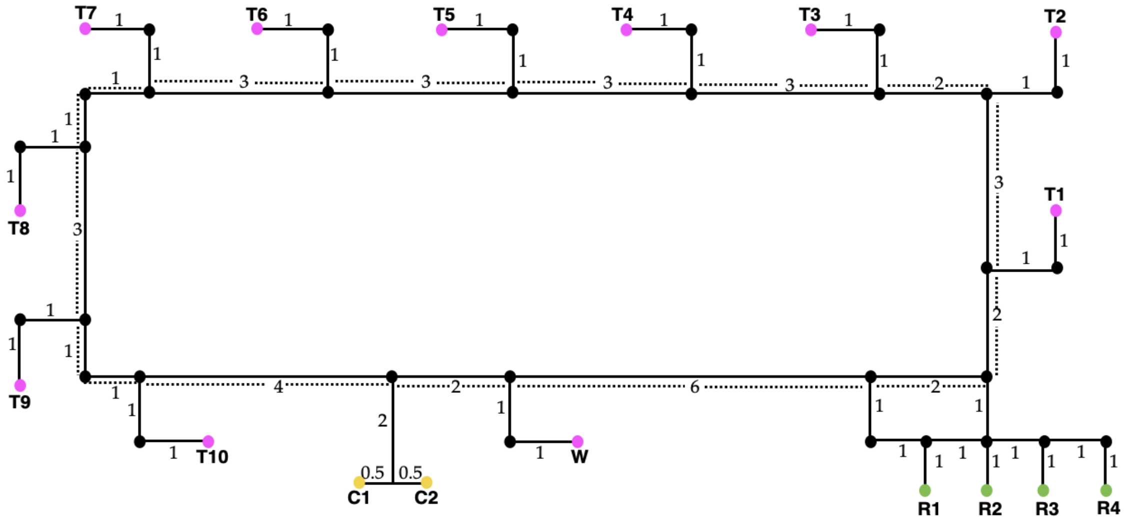

Figure 3: Graph ($\mathcal{G}_{\mathcal{E}}$)

Definition 2: [Task] Given an environment $\mathcal{E} = (\mathcal{R}, \mathcal{T}, \mathcal{C}, Edges, \mathcal{D})$, a task is defined as a pair $\tau = (t, c)$, where

- $t \in \mathcal{T}$ is a location of a dustbin;
- $c \in \mathcal{C}$ is a location of a big container.

Next, we define a plan for a robot to accomplish a set of tasks in an environment given as below.

Definition 3: [Plan] Given an environment $\mathcal{E} = (\mathcal{R}, \mathcal{T}, \mathcal{C}, Edges, \mathcal{D})$, a set of tasks $\Upsilon = \{(t_k, c_k)\}_{k=1}^{p}$, an initial robot's location $r \in \mathcal{R}$, a plan for the robot is a sequence of locations $\rho = l_0, l_1, l_2, \ldots, l_n$ satisfying the following conditions:

- $l_0 = r$ and $(l_{i-1}, l_i) \in Edges$ for $i \in [n]$;
- for each $l_i = t_k$ for some $i \in [n]$ and $k \in [p]$, $\nexists\ j \in [n]$, satisfying $j \neq i$ and $l_j = t_k$;
- for each $l_i = t_k$ for some $i \in [n]$ and $k \in [p]$, $\exists\ j$, $j > i$ satisfying $l_j = c_k$ and $\nexists\ m$, $i < m < j$ such that $l_m = t_{k'}$ for some $k' \in [p]$.

The cost of the plan ρ can be computed by the following formula:

$$cost(\rho) = \sum_{i=1}^{n} \mathcal{D}((l_{i-1}, l_i)).$$

Definition 4: [Complete Plan] Given an environment $\mathcal{E} = (\mathcal{R}, \mathcal{T}, \mathcal{C}, Edges, \mathcal{D})$, a set of tasks $\Upsilon = \{(t_k, c_k)\}_{k=1}^{p}$, a complete plan is defined as a sequence of plan $\Gamma = \{\rho_i\}_{i=1}^{|\mathcal{R}|}$ such that for each task (t_k, c_k), there exists a plan $\rho = l_0, l_1, \ldots, l_n \in \Gamma$ starting from some robot's initial location $r \in \mathcal{R}$ such that $l_i = t_k$ for some $i \in [n]$.

The cost of the complete plan $\Gamma = \{\rho_i\}_{i=1}^{|\mathcal{R}|}$ can be computed by the following expression:

$$\sum_{i=1}^{|\mathcal{R}|} cost(\rho_i).$$

Note that multiple complete plans are possible. Hence, our aim is to find a complete plan whose cost is minimum. We formally define the optimization problem given as below.

Problem 1: [Problem] Given an environment $\mathcal{E} = (\mathcal{R}, \mathcal{T}, \mathcal{C}, Edges, \mathcal{D})$ and a set of tasks $\Upsilon = \{\tau_k\}_{k=1}^{p}$, find a complete plan $\Gamma = \{\rho_i\}_{i=1}^{|\mathcal{R}|}$ such that

$$\sum_{i=1}^{|\mathcal{R}|} cost(\rho_i) \text{ is minimum.}$$

VI. Our Approach

In this section, we present a procedure to reduce the problem as an instance of the MTS problem. The reduction procedure consists of three steps. First, we express a given environment $\mathcal{E}$ as a weighted graph $\mathcal{G}_{\mathcal{E}}$ that captures all possible robots' movements, where the set of initial vertices will be all robots' initial location. Second, given a set of tasks $\Upsilon = \{(t_k, c_k)\}_{k=1}^{p}$, we reduce $\mathcal{G}_{\mathcal{E}}$ into another graph $\mathcal{G}_{\mathcal{E}}^{\Upsilon}$ that consists of only those vertices, which are either t_i, c_j, or robots' initial location for $i, j \in [p]$. The edges for $\mathcal{G}_{\mathcal{E}}^{\Upsilon}$ are constructed as follows: (a) for each vertex t_i, we create edges from t_i to all c_j's; (b) for each vertex c_j, we create edges from c_j to all t_i's; (c) for each robots' initial location r, we create edges from r to all c_j's. Then, we compute the weight for each edge by computing the shortest distance between the end points of the edge in the graph $\mathcal{G}_{\mathcal{E}}$. Finally, we reduce the problem as an instance of the MTS problem by appropriate modification to $\mathcal{G}_{\mathcal{E}}^{\Upsilon}$. Next, we provide the details about each step.

A. Graph Representation for Environments

In this section, we provide a formal construction of the weighted graph capturing all the line segments of an environment in which robots could move.

Definition 5: [Construction of $\mathcal{G}_{\mathcal{E}}$] Given an environment $\mathcal{E} = (\mathcal{R}, \mathcal{T}, \mathcal{C}, Edges, \mathcal{D})$, the construction of $\mathcal{G}_{\mathcal{E}} = (V_0, V, E, W)$ is given below:

- $V_0 = \mathcal{R}$;
- $V = \bigcup_{(u,v) \in Edges} \{u, v\}$;
- $E = Edges$;
- $W(e) = \mathcal{D}(e)$.

Next, we demonstrate the construction of $\mathcal{G}_\mathcal{E}$ with an example.

Example 2: Consider the hospital environment $\mathcal{E}$ shown in Figure 2. The constructed graph $\mathcal{G}_\mathcal{E}$ corresponding to the environment $\mathcal{E}$ is shown in Figure 3.

B. Tasks based Reduction of $\mathcal{G}_\mathcal{E}$

In this section, we reduce the graph $\mathcal{G}_\mathcal{E}$ based on a given set of tasks. Given a set of tasks Υ, and a graph $\mathcal{G}_\mathcal{E}$ corresponding to an environment $\mathcal{E}$, we construct a task-based graph $\mathcal{G}_\mathcal{E}^\Upsilon$ from the graph $\mathcal{G}_\mathcal{E}$ that consists of only those vertices related to robots' initial location, trashes' location, containers' location. The formal construction of the graph $\mathcal{G}_\mathcal{E}^\Upsilon$ is given as below.

Definition 6: [Construction of $\mathcal{G}_\mathcal{E}^\Upsilon$] Given a graph $\mathcal{G}_\mathcal{E} = (V_0, V, E, W)$ corresponding to an environment $\mathcal{E}$ and a set of tasks $\Upsilon = \{(t_k, c_k)\}_{k=1}^{p}$, the construction of $\mathcal{G}_\mathcal{E}^\Upsilon = (V'_0, V', E', W')$ is given as below:

- $V'_0 = V_0$;
- $V = V'_0 \cup T \cup C$, where $T = \cup \bigcup_{(t,c) \in \Upsilon} \{t\}$ and $C = \cup \bigcup_{(t,c) \in \Upsilon} \{c\}$;
- $E = E_1 \cup E_2 \cup E_3$ where
  - $E_1 = \{(v, t) \mid v \in V_0, t \in T\}$ captures edges from robots' initial location to the dustbins associated with the tasks;
  - $E_2 = \{(t, c) \mid t \in T,\ c \in C\}$ captures edges between each dustbin and container associated with the tasks;
  - $E_3 = \{(c, t) \mid c \in C,\ t \in T, \}$ captures edges between each container and dustbin associated with the tasks;
- for each edge $(u, v) \in E'$, $W'(u, v)$ captures the shortest distance between u and v in $\mathcal{G}_\mathcal{E}$.

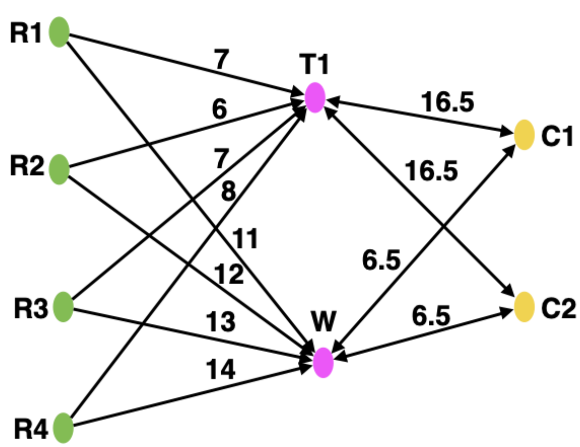

Figure 4: Task based Graph ($\mathcal{G}_\mathcal{E}^\Upsilon$)

Next, we illustrate the construction of the task-based graph with an example.

Example 3: Consider the graph $\mathcal{G}_\mathcal{E}$ shown in Figure 3 corresponding to the hospital environment $\mathcal{E}$ shown in Figure 2 and a set of tasks $\Upsilon = \{(W, C_1), (T1, C2)\}$. The constructed graph $\mathcal{G}_\mathcal{E}^\Upsilon$ corresponding to the graph $\mathcal{G}_\mathcal{E}$ and Υ is shown in Figure 4.

Note that for the MTS problem, each vertex has to be visited exactly once by the salesmen. However, in the pick and drop problem, a vertex corresponding to the big container may need to be visited more than once. Hence, we convert the graph $\mathcal{G}_\mathcal{E}^\Upsilon$ into another graph $\mathcal{G}_M$ for reducing the pick and drop problem as an instance of the MTS problem.

C. Construction of $\mathcal{G}_M$ from $\mathcal{G}_\mathcal{E}^\Upsilon$

In this section, we present the construction of the graph $\mathcal{G}_M$ from the graph $\mathcal{G}_\mathcal{E}^\Upsilon$. First, for each vertex in the graph $\mathcal{G}_\mathcal{E}^\Upsilon$ corresponding to a big container c associated with the task, if the maximum number of tasks having c is k, then we create $k-1$ copies of vertex c. Then, we add all the incoming and outgoing edges associated with the container c to all the copied vertices. Next, we introduce a dummy vertex d for all the robots to have a common initial point. Finally, we add edges between d and all the robots' location and from big containers to d with distance 0. The formal construction for the graph $\mathcal{G}_M$ is given below.

Definition 7: [Construction of $\mathcal{G}_M$] Given a task based graph $\mathcal{G}_\mathcal{E}^\Upsilon = (V'_0, V', E', W')$, the construction of $\mathcal{G}_M = (V_0^m, V^m, E^m, W^m)$ is given as below. Let $C = \cup \bigcup_{(t,c) \in \Upsilon} \{c\}$.

- $V_0^m = \{d\}$;
- $V^m = V_0^m \cup V' \cup V_c$, where $V_c = \bigcup_{c \in C} \{c_i \mid 1 \leq i < k,$ k is the number of tasks in which c appears$\}$;
- $E^m = E' \cup \{(u, v) \mid u \in V_0^m,\ v \in V'_0\} \cup \{(u, v) \mid u \in V'_0 \cup V_c \cup C,\ v \in V_0^m\}$;
- $W^m(e) = W(e)$ if $e \in E'$ otherwise $W^m(e) = 0$.

Example 4: Consider the task based graph $\mathcal{G}_\mathcal{E}^\Upsilon$ shown in Figure 4. The constructed graph $\mathcal{G}_M$ corresponding to $\mathcal{G}_\mathcal{E}^\Upsilon$ is shown in Figure 8.

Now, the graph $\mathcal{G}_M$ can be used for an instance of the MTS problem, where vertices corresponding to the robots can be considered as salesmen. The dummy vertex can be treated as a source and target vertex for each robot. Finally, we use the graph $\mathcal{G}_M$ to encode the pick and drop problem into the Mixed Integer Linear Programming (MILP) and extract the optimal plan for the robots.

VII. Experimental Analysis

In this section, we present the analysis of our method for different number of tasks in three kinds of hospitals, namely, small, medium, and large shown in Figures 5, 6, and 7, respectively. We have implemented our method in the Python toolbox, where we have used the NetworkX tool for the graph construction; the GLPK solver for solving the optimization problem. Finally, we use the V-REP [25] simulator to simulate the optimal plan.

In Table I, *#Robots* denotes the number of robots for small, medium, and large hospitals. *#Tasks* represents the number

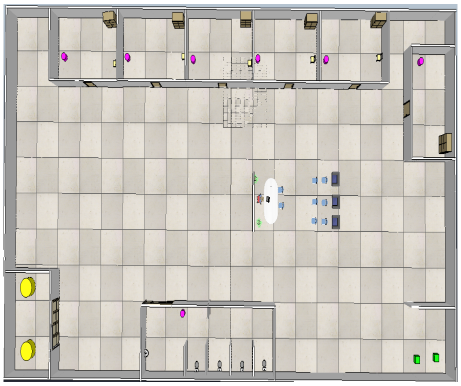

Figure 5: Small Hospital

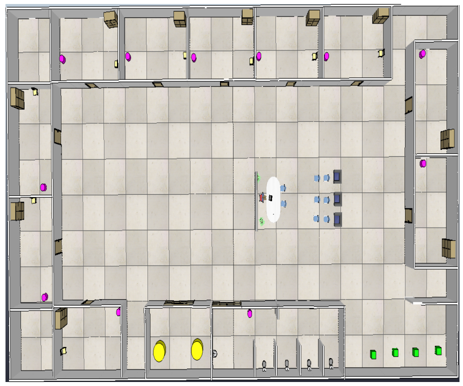

Figure 6: Medium Hospital

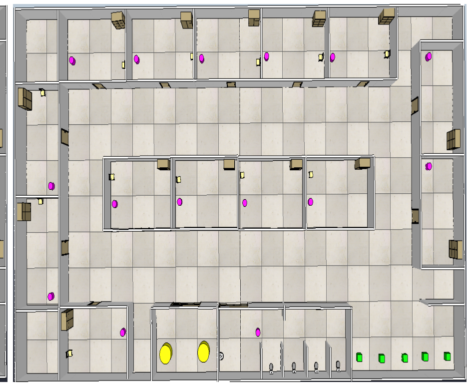

Figure 7: Large Hospital

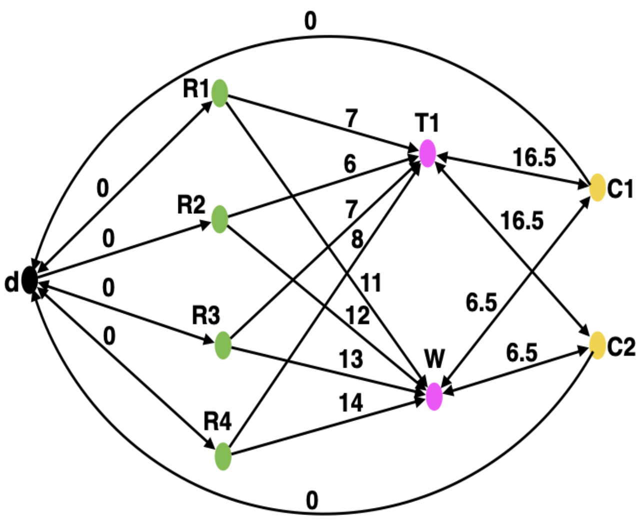

Figure 8: Graph ($\mathcal{G}_M$)

of tasks. $T_{\mathcal{G}_\mathcal{E}}$, $T_{\mathcal{G}_\mathcal{E}^\Upsilon}$, T_{opt}, and T_p are the times taken for constructing the graph corresponding to a given environment, a task-based graph, and finding the optimal solution by the GLPK solver, and extracting an optimal path for the robots, respectively. *cost* denotes the optimal cost, that is, the total distance traveled by all the robots. The experimental results are presented in Table I.

TABLE I: COMPUTATIONAL ANALYSIS

#Robots	*#Tasks*	$T_{\mathcal{G}_\mathcal{E}}$(sec.)	$T_{\mathcal{G}_\mathcal{E}^\Upsilon}$(sec.)	T_{opt}(sec.)	T_p(sec.)	*cost*
2 (small)	2	0.28	0.18	0.001	0.13	72
	3	0.28	0.18	0.005	0.16	113.5
	4	0.28	0.19	0.041	0.20	136
4 (med.)	2	0.32	0.23	0.004	0.15	62
	3	0.32	0.23	0.031	0.22	101.5
	4	0.32	0.24	0.255	0.26	142
6 (large)	2	0.38	0.25	0.019	0.21	60
	3	0.38	0.25	0.216	0.24	98.5
	4	0.38	0.26	1.337	0.27	138

In Table I, we have observed that the time taken to construct the graph $\mathcal{G}_E$ gradually grows for a fixed number of tasks when we increase the size of hospitals, as can be seen in the first row of small, medium, and large hospital, respectively. A similar observation can be seen for the task-based graph. The time taken by the GLPK solver slowly grows when we increase the number of tasks for a fixed environment, as can be seen in the first three rows for the small hospital. A similar observation can be seen for the path extraction. Also, *cost* increases when we increase the number of tasks for the same environment. We have plotted the data in Figure 9 corresponding to the data presented in Table I, where all the observations mentioned above can be clearly seen. Overall, our method is scalable.

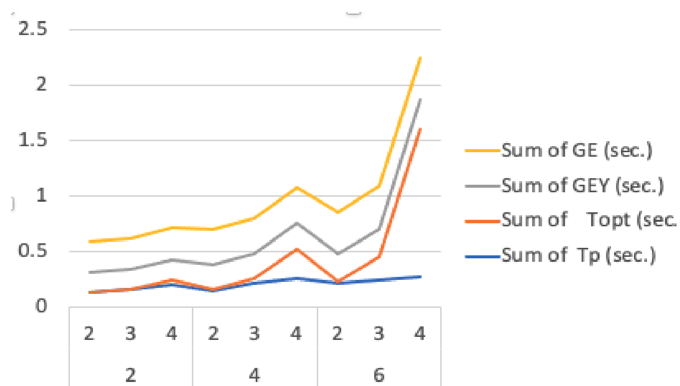

Figure 9: Computational Analysis

VIII. Conclusion

In this paper, we have investigated optimal multi-robot path planning for a selected pick and drop problem. We have reduced the problem as an instance of the MTS problem by representing a hospital environment as a weighted graph and transforming the weighted graph into a task-based graph with the help of the shortest distance graph algorithm. We have used the GLPK solver to solve the optimization problem. Finally, we have performed the experiments for different types of hospitals, namely, small, medium, and large. In the future, we will investigate optimal multi-robot path planning for multi-objective tasks.

IX. Acknowledgments

This work is partially supported by Northwest Missouri State University.

References

[1] A. Barrientos, et al., "Aerial remote sensing in agriculture: A practical approach to area coverage and path planning for fleets of mini aerial robots," Journal of Field Robotics vol. 28, pp. 667–689, 2011.

[2] J. L. Baxter, E. Burke, J. M. Garibaldi, and M. Norman, "Multi-robot search and rescue: A potential field based approach," Autonomous robots and agents, pp. 9–16, 2007.

[3] G. Bevacqua, J. Cacace, A. Finzi, and V. Lippiello, "Mixed-initiative planning and execution for multiple drones in search and rescue missions," International Conference on Automated Planning and Scheduling, vol. 25, pp. 315–323, 2015.

[4] H. Choset, "Coverage for robotics–a survey of recent results," Annals of mathematics and artificial intelligence, vol. 31, pp. 113–126, 2001.

[5] C. M. Clark, S. M. Rock, and J.-C. Latombe, "Motion planning for multiple mobile robots using dynamic networks," IEEE International Conference on Robotics and Automation, vol. 3, pp. 4222–4227, 2003.

[6] A. Cowley, C. J. Taylor, and B. Southall, "Rapid multi-robot exploration with topometric maps," IEEE International Conference on Robotics and Automation, pp. 1044–1049, 2011.

[7] R. Gayle, W. Moss, M. C. Lin, and D. Manocha, "Multi-robot coordination using generalized social potential fields," IEEE International Conference on Robotics and Automation, pp. 106–113, 2009.

[8] M. Gonzalez-de-Soto, L. Emmi, M. Perez-Ruiz, J. Aguera, and P. Gonzalez-de-Santos, "Autonomous systems for precise spraying–Evaluation of a robotised patch sprayer," Biosystems engineering, vol. 146, pp. 165–182, 2016.

[9] S. Gupta, H. Kruthik, C. Hegde, S. Agrawal, and S. B. Prashanth, "GarBot: Garbage Collecting and Segregating Robot," Journal of Physics: Conference Series, vol. 1950, pp. 012023, 2021.

[10] D. Halperin, J.-C. Latombe, and R. H. Wilson, "A general framework for assembly planning: The motion space approach," vol. 26, pp. 577–601, 2000.

[11] N. Hazon and G. A. Kaminka, "Redundancy, efficiency and robustness in multi-robot coverage," IEEE international conference on robotics and automation, pp. 735–741, 2005.

[12] N. Hazon, F. Mieli, and G. A. Kaminka, "Towards robust on-line multi-robot coverage," IEEE International Conference on Robotics and Automation, pp. 1710-1725, 2006.

[13] J. S. Jennings, G. Whelan, and W. F. Evans, "Cooperative search and rescue with a team of mobile robots," International Conference on Advanced Robotics, pp. 193–200, 1997.

[14] A. Joy, R. Varghese, A. Varghese, A. M. Sajeev, J. Raveendran, A. Thomas, and K. Saran, "Medrobo medicine delivering and patient parameter monitoring robot," International Conference on Advanced Computing and Communication Systems, vol. 1, pp. 1808–1812, 2021.

[15] S. Kavirayani, D. S. Uddandapu, A. Papasani, and T. V. Krishna, "Robot for delivery of medicines to patients using artificial intelligence in health care," IEEE Bangalore Humanitarian Technology Conference, pp. 1–4, 2020.

[16] S. Kumar and S. Chakravorty, "Multi-agent generalized probabilistic RoadMaps: MAGPRM," IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 3747–3753, 2012.

[17] R. Lal and P. Prabhakar, "Time-optimal multi-quadrotor trajectory planning for pesticide spraying," IEEE International Conference on Robotics and Automation, pp. 7965–7971, 2021.

[18] R. Lal, A. Sharda, and P. Prabhakar, "Optimal multi-robot path planning for pesticide spraying in agricultural fields," IEEE 56th Annual Conference on Decision and Control, pp. 5815–5820, 2017.

[19] C. E. Miller, A. W. Tucker, and R. A. Zemlin, "Integer programming formulation of traveling salesman problems," Journal of the ACM, vol. 7, pp. 326–329, 1960.

[20] M. Nazarahari, E. Khanmirza, and S. Doostie, "Multi-objective multi-robot path planning in continuous environment using an enhanced genetic algorithm," Expert Systems with Applications, vol. 115, pp. 106–120, 2019.

[21] A. G. Ozkil, Z. Fan, S. Dawids, H. Aanes, J. K. Kristensen, and K. H. Christensen, "Service robots for hospitals: A case study of transportation tasks in a hospital," IEEE international conference on automation and logistics, pp. 289–294, 2009.

[22] A. Patel, P. Sharma, and P. Randhawa, "MedBuddy: The Medicine Delivery Robot," International Conference on Reliability, Infocom Technologies and Optimization, pp. 1–4, 2021.

[23] I. Rekleitis, V. Lee-Shue, A. P. New, and H. Choset, "Limited communication, multi-robot team based coverage," IEEE International Conference on Robotics and Automation, vol. 4, pp. 3462–3468, 2004.

[24] S. Rodriguez and N. M. Amato, "Behavior-based evacuation planning," IEEE International Conference on Robotics and Automation, pp. 350–355, 2010.

[25] E. Rohmer, S. P. N. Singh, and M. Freese, "V-REP: A versatile and scalable robot simulation framework," IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1321–1426, 2013.

[26] T. Siméon, S. Leroy, and J.-P. Lauumond, "Path coordination for multiple mobile robots: A resolution-complete algorithm," IEEE transactions on robotics and automation, vol. 18, pp. 42–49, 2002.

[27] P. Song and V. Kumar, "A potential field based approach to multi-robot manipulation," IEEE International Conference on Robotics and Automation, vol. 2, pp. 1217–1222, 2002.

[28] M. Takahashi, T. Suzuki, H. Shitamoto, T. Moriguchi, and K. Yoshida, "Developing a mobile robot for transport applications in the hospital domain," Robotics and Autonomous Systems, vol. 58, pp. 889–899, 2010.

[29] H. G. Tanner and A. Kumar, "Towards decentralization of multi-robot navigation functions," IEEE International Conference on Robotics and Automation, pp. 4132–4137, 2005.

[30] G. Wagner and H. Choset, "M*: A complete multirobot path planning algorithm with performance bounds," IEEE/RSJ international conference on intelligent robots and systems, pp. 3260–3267, 2011.

[31] M. Weis, et al., "Precision farming for weed management: techniques," Gesunde Pflanzen, vol. 60, pp. 171–181, 2008.

[32] Z. Yan, N. Jouandeau, and A. A. Cherif, "A survey and analysis of multi- robot coordination," International Journal of Advanced Robotic Systems, vol. 10, pp. 399, 2013.

[33] J. Yu and S. M. LaValle, "Optimal multirobot path planning on graphs: Complete algorithms and effective heuristics," IEEE Transactions on Robotics, vol. 32, pp. 1163–1177, 2006.

[34] S. Yu, J. Puchinger, and S. Sun, "Van-based robot hybrid pickup and delivery routing problem," European Journal of Operational Research, vol. 298, pp. 894–914, 2022.

[35] X. Zheng, S. Jain, S. Koenig, and D. Kempe, "Multi-robot forest coverage," IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 3852–3857, 2005.