

# Robots in Retirement Homes: Person Search and Task Planning for a Group of Residents by a Team of Assistive Robots\*

Kyle E. C. Booth, Sharaf C. Mohamed, Sanjiv Rajaratnam, Goldie Nejat,  
and J. Christopher Beck

Department of Mechanical & Industrial Engineering, University of Toronto, Toronto, Ontario M5S 3G8, Canada  
[{kbooth, smohamed, sraj, nejat, jcb}@mie.utoronto.ca](mailto:{kbooth, smohamed, sraj, nejat, jcb}@mie.utoronto.ca)

## Abstract

We present a general *multi-robot task planning and execution* (MRTPE) architecture for a team of heterogeneous mobile robots that interact with multiple human users. The designed architecture is implemented within an environment where such robots provide daily assistance to residents in a retirement home setting. The robots are able to allocate and schedule activities throughout the day and find the appropriate residents with whom to engage in the assistive activities. At the beginning of the day, the *robot person search* (RPS) system autonomously searches for, and finds, multiple human users in an environment within a required time interval. Upon finding users, the robot queries each user for his/her availability and interest in various activities. We then use *constraint programming* (CP) within a *multi-robot task allocation and scheduling* (MRTA) system to plan and schedule the robot team to facilitate individual and group activities, ensuring consistency with user-expressed availability and activity preferences. We test the components of the architecture on a physical multi-robot system to verify the utility of the design. Experiments indicate the design can effectively plan and execute assistive activities for multiple users.

**Keywords:** Robotics, scheduling, constraint satisfaction, multiagent systems, user/machine systems

## 1 Introduction

In response to a rapidly aging global population, the design of socially assistive robotic systems for healthcare, specifically for eldercare, has been an active area of research for the past decade [2]. These systems have been developed to assist and support elderly individuals with physical and cognitive impairments, as well as aid in the day-to-day management of the healthcare environment in order to alleviate the workload pressures of an already strained elderly care labour force.

Our research focuses on the development and deployment of a general *multi-robot system* (MRS) architecture for a team of mobile robots that interact with human users. Within this work, the designed *multi-robot task planning and execution* (MRTPE) architecture is implemented to plan and

---

\*This research represents an extension of the work presented in Vaquero et al. [1] at the AAAI Workshop on Artificial Intelligence Applied to Assistive Technologies and Smart Environments.

facilitate assistive activities for multiple human users within a retirement home environment. At the beginning of the day, the team of robots must autonomously search for and find users in the environment, eliciting their availability and preferences for activities using the developed *robot person search* (RPS) system. The MRS then uses the *multi-robot task allocation and scheduling* (MRTA) system we have developed to allocate and schedule these assistive activities over the remainder of the day. In addition to the integration of the novel components we propose herein, our architecture utilizes existing technology to achieve core autonomous robot functions (e.g. navigation, localization, and mapping).

Previous work on this application [1, 3, 4] focused primarily on implementing and testing single-robot architectural components in isolation within simulated environments; contributions for MRTA did not address finding users in uncertain environments, nor were the approaches implemented within a physically deployed MRS. The architecture presented in this paper utilizes an RPS procedure to find users within the environment, expanding on the single robot work in Mohamed and Nejat [3] by modeling the search as a *travelling thief problem* solved with dynamic programming to generate promising user search plans. While previous approaches to MRTA problems have primarily used decentralized auction-based techniques [5] or centralized mixed-integer programming [6], we make novel use of *constraint programming* (CP) within our centralized MRTPE architecture to produce high-quality, and often optimal, activity schedules. The MRTA component of the proposed architecture extends the work in Booth et al. [4] and integrates it with the RPS of Mohamed and Nejat [3]. We test the architecture on a physical MRS and present experimental results on a number of retirement home scenarios with heterogeneous robots, concluding that the system is able to plan and execute assistive activities for multiple users within a multi-region environment. As a result of our experiments, we believe the MRTPE architecture represents a promising general framework for alternate applications that involve mobile robots interacting with human users.

## 2 Problem Definition

Our problem concerns a team of mobile robots that must perform various *human-robot interactions* (HRI), in the form of assistive activities, with elderly users in a retirement home. The activities must be allocated and scheduled over a single 12-hour day (7:00 AM to 7:00 PM). The MRS must autonomously plan and facilitate these activities while adhering to problem-specific constraints, including user availability and location, robot energy consumption, activity precedence, and robot-user activity synchronization. Prior to the scheduling of activities, the users must be queried regarding their individual availability and locations for the day, as well as their preferences for participation in various activities (binary ‘yes’ or ‘no’ response). Once this information is attained, the MRS creates the activity allocation and robot schedule for the day before executing the plan.

### 2.1 Users

We consider  $n$  human users,  $U := \{u_1, u_2, \dots, u_n\}$ , residing in the retirement home. These users share the environment and participate in a number of activities throughout the course of the day. Each user has a unique calendar of  $\alpha$  time intervals where they are *not* available for interaction, where the total set of such calendars is defined by the set  $\Sigma := \{\sigma_1, \sigma_2, \dots, \sigma_n\}$ , where each  $\sigma_i :=$

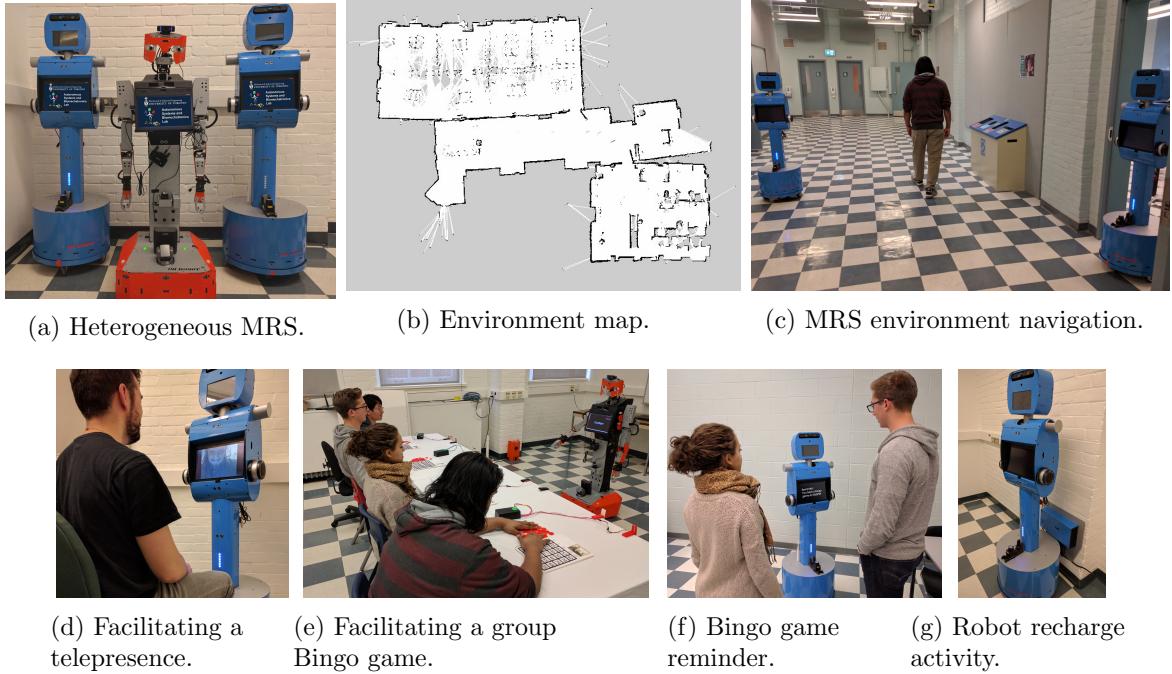


Figure 1: Problem Definition - Assistive robot fleet, test environment, and activities.

$\{[s_1^{\sigma_i}, e_1^{\sigma_i}], [s_2^{\sigma_i}, e_2^{\sigma_i}], \dots, [s_{\alpha}^{\sigma_i}, e_{\alpha}^{\sigma_i}]\}$  for user  $u_i \in U$  identifies their specific busy intervals. These busy intervals include breakfast (8:00 to 9:00 AM), lunch (12:00 to 1:00 PM), and dinner (5:00 to 6:00 PM), as well as a number of other intervals unknown *a priori* to be acquired at the start of the day. We estimate the movement speed of each user  $u_i \in U$  as  $v_i^u$  in metres/minute, which is utilized to approximate user travel time within the environment.

## 2.2 Robots

We consider  $m$  heterogeneous mobile robots,  $R := \{r_1, r_2, \dots, r_m\}$ , as shown in Figure 1a. These robots are responsible for executing the person search as well as autonomously allocating, scheduling, and facilitating the HRI activities. Each robot,  $r_k \in R$ , navigates the environment at a speed of  $v_k^r$ . Robots start and end each day at the *robot depot*, a location that houses the recharging station. Energy levels for the battery of each robot,  $r_k \in R$ , must remain between  $\beta_k^{\min}$  and  $\beta_k^{\max}$  and energy is consumed at robot and activity-specific rates.

## 2.3 Environment

The environment is divided into a number of regions that represent rooms within the facility. A sample test environment we utilize for experimentation, both map and real-world image, is illustrated in Figure 1b and Figure 1c. The set of locations,  $L$ , consists of the robot depot, games room, meals room, leisure rooms, and a personal room for each user, respectively. Distances between

any two rooms,  $a$  and  $b$ , are defined based on the shortest path as  $\delta_{(a,b)}$  in metres. Travel times between locations are then represented in minutes for each user  $u_i \in U$  as  $\Delta_i^u := \{\frac{\delta_{(a,b)}}{v_i^u} : (a, b) \in L \times L\}$ , and for each robot  $r_k \in R$  as  $\Delta_k^r := \{\frac{\delta_{(a,b)}}{v_k^r} : (a, b) \in L \times L\}$ .

## 2.4 Activities

An *activity* is either i) a direct assistive interaction with a user(s), or ii) an instance of robot recharging. They are categorized as: *telepresence* sessions (Figure 1d), *bingo games* (Figure 1e), *bingo game reminders* (Figure 1f), *robot recharges* (Figure 1g), and *information gathering sessions*. Telepresence sessions allow users to have face-to-face video calls with friends or relatives from their personal room. There is one mandatory telepresence for each user,  $P := \{p_1, p_2, \dots, p_n\}$ , each with a length of 30 minutes. Bingo game activities are group HRIs where users participate in a game of robot-facilitated bingo. Bingo games,  $G := \{g_1, g_2, \dots, g_{UB_1}\}$ , are optional activities, 60 minutes in length, and occur in the games room. The MRTA system must determine which bingo games are played, which users participate in each game, and when the games will occur. A bingo game reminder is a single user HRI where the robot reminds the user of their participation in an upcoming bingo game. The set of bingo reminder activities is then defined by  $\mathcal{M} := \bigcup_{i=1}^n M_i$ , where  $M_i := \{m_{i1}, m_{i2}, \dots, m_{iUB_1}\}$ . A reminder is required for each of the users who have been assigned to play a bingo game. Each reminder activity is two minutes in duration and must occur prior to its associated bingo game. The set of robot recharge activities is defined by  $\mathcal{C} := \bigcup_{k=1}^m C_k$ , where  $C_k := \{c_{k1}, c_{k2}, \dots, c_{kUB_2}\}$ . The upper bounds,  $UB_1$  and  $UB_2$ , associated with the activities are required by our scheduling approach to define the fixed set of activities that may be scheduled [4]. Information gathering sessions are HRIs that occur at the beginning of each day, and are used to query users regarding their availability, locations throughout the day, and activity preferences.

## 3 Multi-Robot Task Planning and Execution Architecture

The proposed MRTPE architecture represents a multi-robot extension of the single-robot system in Louie et al. [7]. In addition to the integration of multiple robot controllers through a master/slave configuration, the proposed system uses *constraint programming* (CP) instead of temporal planning for MRTA, and integrates RPS, which was not done previously. The centralized design is appropriate for the scale of problems being solved, where CP is able to produce high-quality, often optimal, activity schedules. The design allows the MRS to find users, schedule tasks, and execute assistive activities throughout the day. We design and implement the architecture within the open source *Robot Operating System* (ROS) ([www.ros.org](http://www.ros.org)) framework.

### 3.1 Architecture Design

As illustrated in Figure 2, the architecture consists of two levels: i) the Centralized Server, and ii) the Robot Controllers. The former consists of the following modules: MRTA, Global RPS, Execution and Monitoring (E&M), System World State, and System World Database. The System

World State module contains information regarding robot states (e.g., battery levels, poses) and environment states (e.g., region accessibility), while the System World Database module contains information regarding static parameters (e.g., the map). The Individual Robot Controllers include the following modules: Activity Manager, Robot World State, Robot World Database, Activity modules (e.g., Information Gathering, Bingo, etc.), Local RPS, Low-level Controllers, Actuators, and Sensors. The Activity Manager forwards commands from the E&M module to onboard Activity modules; an Activity module uses onboard sensory information to determine which robot behaviours are required to be executed via the Low-level Controllers and Actuators. The Robot World State and Database modules contain the same information as their Centralized Server equivalents, however, exclusively for their corresponding robot. At the start of each day, the System World Database updates the Robot World Database with any information required to detect and identify users.

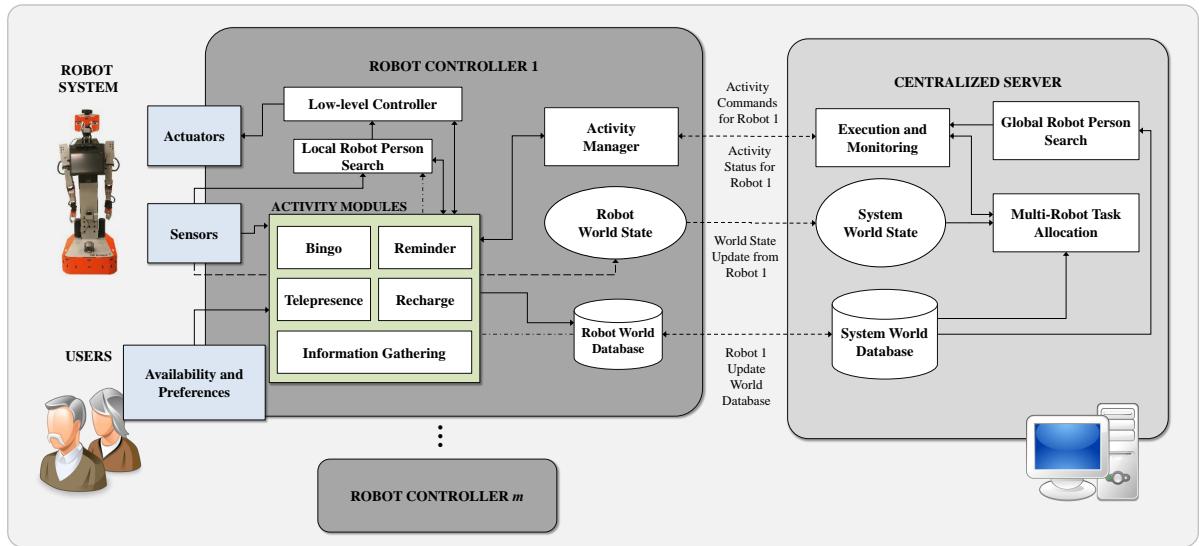


Figure 2: Proposed multi-robot task planning and execution (MRTPE) architecture.

### 3.2 Retirement Home Implementation

The Global RPS, within the Centralized Server, creates a plan for finding and gathering information from all retirement home residents between 7:00 AM and 8:00 AM, prior to breakfast, which is then sent to the E&M Module and executed by a robot. At 8:00 AM, the MRTA Module uses the gathered information to create an activity schedule for the remainder of the day (8:00 AM to 7:00 PM), which is also sent to the E&M module. Whenever a start time is reached for a planned task, the E&M module sends the request to the corresponding robot's Activity Manager. The Activity Manager then sends the activity request to the appropriate Activity Module. For example, during the information gathering period, a request is sent to the Information Gathering module for each re-

gion specified in the global plan. The Information Gathering Module requests the robot to navigate to the specified region and then upon arrival, requests the Local RPS to find target users within this region. The Local RPS reports any users it finds to the Information Gathering Module, which then requests schedule and activity preferences for the day from these users. This information is then sent to the Robot World Database Module and then the System World Database Module.

### 3.3 Robot Person Search

The RPS system is adapted from Mohamed and Nejat [3] and directly integrated into our architecture. Its unique use within our architecture allows a robot to directly obtain activity preferences and availability from the users through HRI interactions. The RPS system allows a robot to autonomously search for and find users who reside within the retirement home. The RPS is comprised of two modules: i) the Global RPS, and ii) the Local RPS. The Global RPS is utilized to determine a plan of regions (i.e., rooms) to be searched at a high-level, while the Local RPS conducts the search within each region.

#### 3.3.1 Global Robot Person Search

This module generates a *global plan* that maximizes the number of users found given the retirement home regions,  $L$ , and the *search query*: a list of target users and a specified time frame. The search query for our problem is to find all of the retirement home residents during the time period 7:00 AM to 8:00 AM. The global plan consists of: i) a subset of regions to search with their corresponding search times, and ii) the order in which to search this subset. To determine the global plan, a *Travelling Thief Problem* (TTP) is solved using a dynamic programming algorithm as detailed in Mohamed and Nejat [3]. The algorithm is given a probabilistic location model, generated for each user by using *activity patterns* (stored in the World Database), a set of time-indexed tuples defined by  $\{user, region, activity, time\ interval\}$  that are acquired *a priori* through an observation stage. The generated global plan is sent to the E&M module for execution by the robot team. The execution results in a series of information gathering tasks being sent to the Robot Controller, which in turn results in the Local RPS Module implementing a search to find users within a region.

#### 3.3.2 Local Robot Person Search

This module receives a request from the Activity Module when the robot arrives at the specified region. The region is divided into cells corresponding to the sensor range of the robot's onboard 3D camera. The robot then constructs a tour of the cells, determining within each cell if it contains any target user(s). A silhouette detection algorithm is used to compare contours in a depth image (obtained from the 3D camera) to a reference silhouette of a person (stored in the Robot World Database). Once a person is detected, the 3D point cloud of the environment, generated by the 3D camera, is used to acquire the location of the person. RGB images obtained from the robot's onboard RGB camera are then used with the *local binary patterns* face detection algorithm from OpenCV ([www.opencv.org](http://www.opencv.org)) to determine the orientation of the person with respect to the robot. Finally, the robot navigates in order to face the person for interaction, and determines his/her

identity by applying the OpenCV *local binary patterns histogram recognizer* to identify the user's facial features and compares them with the unique facial features stored for each user in the Robot World Database.

### 3.4 Multi-Robot Task Allocation and Scheduling

With user availabilities, locations, and preferences provided by the RPS, we model and solve a *multi-robot task allocation and scheduling* (MRTA) problem [8] using the MRTA module in our architecture. In this section, we summarize the decision variables, objective function, and core constraints utilized in our model.

#### 3.4.1 Constraint Programming

We model the problem as a *constraint optimization problem* (COP) defined by the tuple  $\langle X, D, C, F \rangle$ , where  $X$  is a set of decision variables,  $D$  are their associated domains (possible values in a solution),  $C$  is a set of hard constraints, and  $F$  is the problem-specific objective function. We solve the COP with *constraint programming* (CP), a model-and-solve paradigm similar to *integer programming* (IP). CP is utilized for our architecture as it was shown within Booth et al. [4] to significantly outperform an IP technique for a similar problem.

CP is more general than IP, relaxing restrictions of linearity and expressing richer variable types (e.g. interval [9] and set variables [10]) as well as constraints, termed *global constraints* [11], designed to capture frequently recurring combinatorial substructure. The combinatorial explosion of problems that CP is commonly used to solve is addressed through a branch-and-bound search algorithm that makes use of *logical inference* to reduce search effort. CP has been successfully applied to a wide range of combinatorial optimization problems, notably scheduling [12], where it often significantly outperforms IP-based approaches.

#### 3.4.2 Problem Modelling

Within the CP formalism we use *interval decision variables* [9] to model robot and user tasks. The domain of possible values for an interval variable,  $var \in X$ , is defined by  $D_{var} := \{\perp\} \cup \{[s, e] | s, e \in \mathbb{Z}, s \leq e\}$ . That is,  $var$  takes on a value that is a convex interval with integer end points  $[s, e]$ , or  $\perp$  indicating the variable is not present in the solution. The latter assignment is represented by the expression  $\text{Presence}(var)$  evaluating to 1 if  $var \in X$  is present in the solution, and 0 otherwise.  $\text{Start}(var)$ ,  $\text{End}(var)$ , and  $\text{Length}(var)$  return the integer start time, end time, and length of the interval variable  $var$ . In addition to interval variables, we also use *cumulative function expressions*; variables that model cumulative resources through the impact of interval variables.

**Decision Variables** As in Booth et al. [4] we define the decision variables for our CP formulation as follows:

$x_{ij} := (\text{interval variable})$  present, with a start time value, if user  $u_i \in U$  participates in activity  $j$  and absent, with a value of  $\perp$ , otherwise.

$y_{kj} := (\text{interval variable})$  present, with a start time value, if robot  $r_k \in R$  facilitates activity  $j$  and absent, with a value of  $\perp$ , otherwise.

$E_k := (\text{cumulative function expression})$  representing the energy level of robot  $r_k$  throughout the schedule.

**Objective Function** The objective function, (1), is to maximize bingo game user participation in order to boost the cognitive and social stimulation of users, while prioritizing schedules with fewer robot recharges.

$$\arg \max_{\{\mathbf{x}, \mathbf{y}\}} \sum_{u_i \in U} \sum_{j \in G} \text{Presence}(x_{ij}) - 0.01 \cdot \sum_{r_k \in R} \sum_{j \in C_k} \text{Presence}(y_{kj}) \quad (1)$$

**Problem Constraints** The first set of problem constraints, Constraints (2) and (3), ensure that scheduled activities do not interfere temporally on robot and user schedules. To do this, we introduce sets containing all activities potentially involving users and robots (including dummy start,  $\dot{u}, \dot{r}$ , and end,  $\ddot{u}, \ddot{r}$ , activities for sequencing), respectively, as  $T_i^u := \{\sigma_i \cup p_i \cup G \cup M_i\}$  and  $T_k^r := \{P \cup G \cup \mathcal{M} \cup C_k\}$ .

$$\text{NoOverlap}([x_{i\dot{u}}, x_{i1}, x_{i2}, \dots, x_{i|T_i^u|}, x_{i\ddot{u}}], \Delta_i^u), \quad \forall u_i \in U \quad (2)$$

$$\text{NoOverlap}([y_{k\dot{r}}, y_{k1}, y_{k2}, \dots, y_{k|T_k^r|}, y_{k\ddot{r}}], \Delta_k^r), \quad \forall r_k \in R \quad (3)$$

The `NoOverlap` constraint performs inference on interval variables, ensuring they do not interfere temporally if they are present. The next set of constraints ensures that required telepresence activities are facilitated (Constraint (4)), player bingo participation is contingent on robot facilitation (Constraint (5)), and that the end of a bingo reminder must occur before the start of the associated bingo game (Constraint (6)).

$$\text{Presence}(x_{ip_i}) = \sum_{r_k \in R} \text{Presence}(y_{kp_i}) = 1, \quad \forall u_i \in U \quad (4)$$

$$\text{Presence}(x_{ij}) \leq \sum_{r_k \in R} \text{Presence}(y_{kj}) \leq 1, \quad \forall u_i \in U; j \in G \quad (5)$$

$$\text{End}(x_{im_{ij}}) \leq \text{Start}(x_{ij}), \quad \forall u_i \in U; j \in G \quad (6)$$

Constraint (7) ensures that if a user participates in a bingo game, the corresponding reminder is facilitated. Through Constraint (8) the formulation ensures activities common to both user and robot schedules are synchronized through the use of the `StartAtStart` constraint, which synchronizes the start times of the interval variables within its scope.

$$\sum_{r_k \in R} \text{Presence}(y_{km_{ij}}) = \text{Presence}(x_{ij}), \quad \forall u_i \in U; j \in G \quad (7)$$

$$\text{StartAtStart}(x_{ij}, y_{kj}), \quad \forall u_i \in U; r_k \in R; j \in T_i^u \cap T_k^r \quad (8)$$

To represent the energy-related components of the problem, Constraints (9) through (11) model the battery level of each robot, accounting for variable-length robot recharge tasks, as well as unique robot-specific consumption rates for task  $j$ ,  $\xi_k^j$ , and robot navigation,  $\xi_k^\Delta$  for each robot  $r_k \in R$ .

**StepAtStart**, a cumulative function expression, is used to model the instantaneous impact of an interval variable on robot energy level.  $pre_j$  returns the task prior to  $j$  in a robot's schedule, and  $loc(j)$  represents the location of task  $j$ .

$$E_k = \sum_{j \in T_k^r \cup \{r\}} \text{StepAtStart}\left(j, -(\text{Length}(y_{kj}) \cdot \xi_k^j + \Delta_{(loc(pre_j), loc(j))}^r \cdot \xi_k^\Delta)\right), \quad \forall r_k \in R \quad (9)$$

$$0 \leq \text{Length}(y_{kj}) \leq \frac{\beta_k^{max} - \beta_k^{min}}{(-1) \cdot \xi_k^j}, \quad \forall r_k \in R; j \in C_k \quad (10)$$

$$\beta_k^{min} \leq E_k \leq \beta_k^{max}, \quad \forall r_k \in R \quad (11)$$

The modelled problem is solved using a *branch-and-infer* CP search, resulting in a daily schedule for users and robots identifying which activities are allocated to whom and when. Strengthening techniques to improve scheduling performance are also utilized, as detailed in Booth et al. [4].

## 4 Experiments

To validate the utility of the architecture within a physical MRS, we assess three real-world scenarios. The experiments are conducted on a multi-room floor of an engineering building at the university with multiple students representing retirement home users. The details of these scenarios are presented in Table 1, and a sample discretization of a subset of the facility regions (a total of twelve personal and general purpose rooms, with some overlapping regions) is presented in Figure 3. For example, Scenario 2 involves seven human users and three assistive mobile robots within an environment containing twelve total locations (personal rooms and general regions). For this particular scenario, two bingo games and two recharge tasks are supplied as  $UB_1$  and  $UB_2$  to the CP formulation, respectively.

We investigate the performance of the implemented architecture using computational runtime and success rates for system execution (e.g., navigation command success rate). We design our experimentation to verify the architecture through the integration of various modules. First, we validate the ability of the Global RPS module, integrated with the Local RPS module, to find and identify available users within the environment. These results are presented in Table 2. Second, we validate the capability of the MRTA module, integrated with the Local RPS module, to create consistent, high-quality activity schedules, find users, and initiate the associated activities. These results are presented in Table 3.

The first part of the experiments consider the period of time from 7AM – 8AM where the Global RPS finds users and elicits their availability. Referring to Table 2, we can see that the runtime of the Global RPS planner is consistent across all scenario sizes and takes a negligible amount of time. Plan execution, however, takes significant duration and increases as the size of the scenario grows larger: Global RPS for the largest instance considered took just under 36 minutes, which is within the one hour window allotted. The success rates for navigation, head scanning (the process of searching a specific region cell), and silhouette detection during execution range from 84.6–100% across the scenarios. The relatively inferior performance in Scenario 1 was caused by a poorly scanned area of the facility map that resulted in the need for external assistance. This was

Table 1: Real-world test scenario parameters.

Parameter	Scenario 1	Scenario 2	Scenario 3
Users ( $ U $ )	3	7	10
Robots ( $ R $ )	2	3	3
Total Regions ( $ L $ )	8	12	15
Available Bingo ( $UB_1$ )	1	2	2
Available Recharge ( $UB_2$ )	1	2	3

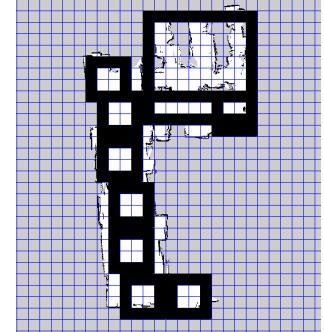


Figure 3: Discretization of regions within facility.

remedied in future scenarios by re-scanning the facility. User identification has the lowest success rate of all; users standing too close to walls/corners of regions were not consistently identified properly. This is an area we plan to improve upon in future work.

Table 2: Global and Local RPS experimental results.

Performance Metric	Scenario 1	Scenario 2	Scenario 3
Planner Runtime (s)	0.43	0.45	0.43
Plan Execution (min and s)	17:35	24:21	35:42
Region Cells Searched	13	28	42
External Assists Required	2	2	1
Navigation (%)	84.6 (11/13)	92.9 (26/28)	97.6 (41/42)
Head Scan (%)	84.6 (11/13)	100.0 (28/28)	97.6 (41/42)
Silhouette Detection (%)	84.6 (11/13)	92.9 (26/28)	97.6 (41/42)
User Identification (%)	66.7 (2/3)	62.5 (5/8)	54.5 (6/11)

The second component of experimentation centres on the creation of consistent activity schedules with the MRTA module, and their successful initiation with users found within the facility, as illustrated in Table 3. To aid experimentation, elapsed time between scheduled activities is artificially sped up. It is clear that as the scenario gets larger (and the number of activities increases), CP requires additional time to find and prove the optimality of a plan, with runtimes (and branching) increasing by roughly an order of magnitude from one scenario to the next. Once a schedule has been produced, tasks are communicated to the remainder of the architecture as their start times occur, leading to another instance of Local RPS within the user's region. Navigation, silhouette detection, and user identification bear very similar success rates to those within Table 2. Task communication and initiation maintain strong success rates due to their relatively simple

Table 3: MRTA system and Local RPS experimental results.

<b>Performance Metric</b>	<b>Scenario 1</b>	<b>Scenario 2</b>	<b>Scenario 3</b>
Activities (Bingo; Recharge)	7 (1; 0)	23 (2; 1)	31 (2; 1)
Scheduler Runtime (s)	0.01	0.82	7.66
Search Tree Branches	10	19,168	220,998
Feasible Solutions Found	1	13	18
Solution Status	<i>Optimal</i>	<i>Optimal</i>	<i>Optimal</i>
Task Communication (%)	100.0 (7/7)	100.0 (23/23)	100.0 (31/31)
Navigation (%)	100.0 (7/7)	95.6 (22/23)	90.3 (28/31)
Silhouette Detection (%)	85.7 (6/7)	95.0 (19/20)	96.4 (27/28)
User Identification (%)	57.1 (4/7)	60.0 (12/20)	60.7 (17/28)
Task Initiation (%)	100.0 (7/7)	100.0 (23/23)	100.0 (31/31)

implementation. Overall, the performance statistics presented within Table 2 and Table 3 support the ability of the architecture to find users and plan various activities within the multi-region environment.

## 5 Conclusion & Future Work

We present a general multi-robot task planning and execution architecture for a team of mobile robots interacting with multiple human users. We implement the architecture in a retirement home application where the robots provide assistance to a group of residents. A robot person search system is used to autonomously search for and find multiple users in order to obtain information regarding their availability and activity preferences. We use this information to model and solve the resultant task planning problem using constraint programming. We present a number of test scenarios using a team of heterogeneous robots and multiple users, highlighting the ability of the architecture to effectively plan various activities. As part of future work for this ongoing project, we intend to explore techniques for plan repair and re-planning in efforts to address scenarios with greater levels of embedded uncertainty.

**Acknowledgment** The authors would like to thank the Natural Sciences & Engineering Research Council of Canada (NSERC), Dr. Robot Inc., and the Canada Research Chairs (CRC) Program.

## References

- [1] Tiago Vaquero, Sharaf Christopher Mohamed, Goldie Nejat, and J Christopher Beck. The implementation of a planning and scheduling architecture for multiple robots assisting multiple users in a retirement home setting. In *Workshops at the Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.
- [2] Reza Kachouie, Sima Sedighadeli, Rajiv Khosla, and Mei-Tai Chu. Socially assistive robots in elderly care: a mixed-method systematic literature review. *International Journal of Human-Computer Interaction*, 30(5):369–393, 2014.
- [3] Sharaf Christopher Mohamed and Goldie Nejat. Autonomous search by a socially assistive robot in a residential care environment for multiple elderly users using group activity preferences. In *Workshop on Planning and Robotics*, pages 58–66, 2016.
- [4] Kyle EC Booth, Goldie Nejat, and J Christopher Beck. A constraint programming approach to multi-robot task allocation and scheduling in retirement homes. In *International Conference on Principles and Practice of Constraint Programming*, pages 539–555. Springer, 2016.
- [5] M Bernardine Dias. *Traderbots: A new paradigm for robust and efficient multirobot coordination in dynamic environments*. PhD thesis, Carnegie Mellon University Pittsburgh, 2004.
- [6] G Ayorkor Korsah, Anthony Stentz, and M Bernardine Dias. A comprehensive taxonomy for multi-robot task allocation. *The International Journal of Robotics Research*, 32(12):1495–1512, 2013.
- [7] Wing-Yue Geoffrey Louie, Tiago Vaquero, Goldie Nejat, and J Christopher Beck. An autonomous assistive robot for planning, scheduling and facilitating multi-user activities. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5292–5298. IEEE, 2014.
- [8] Brian P Gerkey and Maja J Matarić. A formal analysis and taxonomy of task allocation in multi-robot systems. *The International Journal of Robotics Research*, 23(9):939–954, 2004.
- [9] Philippe Laborie. Ibm ilog cp optimizer for detailed scheduling illustrated on three problems. In *International Conference on AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, pages 148–162. Springer, 2009.
- [10] Carmen Gervet. Constraints over structured domains. *Foundations of Artificial Intelligence*, 2:605–638, 2006.
- [11] Francesca Rossi, Peter Van Beek, and Toby Walsh. *Handbook of constraint programming*. Elsevier, 2006.

- [12] Philippe Baptiste, Claude Le Pape, and Wim Nuijten. *Constraint-based scheduling: applying constraint programming to scheduling problems*, volume 39. Springer Science & Business Media, 2012.

**Kyle E. C. Booth** is a Ph.D. candidate in the Department of Mechanical & Industrial Engineering at the University of Toronto. He is a member of the Toronto Intelligent Decision Engineering Laboratory (TIDEL) and the Autonomous Systems and Biomechatronics Laboratory (ASBLab). His research interests include multi-robot coordination, constraint satisfaction, and automated scheduling. He received his B.A.Sc. in Mechanical Engineering at the University of Toronto.

**Sharaf C. Mohamed** is a Ph.D. student in the Department of Mechanical & Industrial Engineering at the University of Toronto. He is a member of ASBLab. His research interests include multi-robot coordination, human-robot interaction, embedded systems, and autonomous robotics. He received his B.A.Sc. in Electrical and Computer Engineering at the University of Toronto.

**Sanjif Rajaratnam** is an M.A.Sc. student in the Department of Mechanical & Industrial Engineering at the University of Toronto. He is a member of ASBLab. His research interests are robotics and autonomous systems. He received his B.A.Sc. in Mechanical Engineering from the University of Waterloo.

**Goldie Nejat** (S'03-M'06) is an Associate Professor in the Department of Mechanical & Industrial Engineering at the University of Toronto, and the Founder and Director ASBLab. She is also an Adjunct Scientist at the Toronto Rehabilitation Institute. Her research interests include intelligent assistive/service robots, human-robot interactions, and semi-autonomous/autonomous control. She received her B.A.Sc. and Ph.D. degrees in Mechanical Engineering at the University of Toronto.

**J. Christopher Beck** is a Professor in the Department of Mechanical & Industrial Engineering at the University of Toronto, and the Founder and Director of TIDEL. His research interests include scheduling and planning, constraint programming, mathematical programming, operations research and artificial intelligence. He received his Ph.D. from the Department of Computer Science, University of Toronto in 1999 in constraint-directed scheduling.