# 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, Sanjif Rajaratnam, Goldie Nejat, and J. Christopher Beck,** *University of Toronto*

*A general multirobot task planning and execution architecture for a team of robots that interact with multiple human users is implemented in an environment where such robots provide daily assistance to residents in a retirement home setting.*

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.[1] 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 to alleviate the workload pressures of an already strained elderly care labor force.

Our research focuses on the development and deployment of a general multirobot system (MRS) architecture for a team of mobile robots that interact with human users. Within this work, the designed multirobot task planning and execution (MRTPE) architecture is implemented to plan and facilitate assistive activities for multiple human users in 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 multirobot task allocation and scheduling (MRTA) system we 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 (navigation, localization, and mapping).

Previous work on this application[2–4] focused primarily on implementing and testing single robot architectural components in isolation within simulated environments; contributions for MRTA didn't address finding users in uncertain environments, nor were the approaches implemented in a physically deployed MRS. The architecture presented in this article utilizes an RPS


1541-1672/17/$33.00 © 2017 IEEE
*Published by the IEEE Computer Society*

procedure to find users in the environment, expanding on single robot work[3] by modeling the search as a traveling 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) in our centralized MRTPE architecture to produce high-quality, and often optimal, activity schedules. The MRTA component of the proposed architecture extends[4] and integrates[3] other work. Here, we test the architecture on a physical MRS and present experimental results on several retirement home scenarios with heterogeneous robots, concluding that the system is able to plan and execute assistive activities for multiple users within a multiregion 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.

## Problem Definition

Our problem concerns a team of mobile robots that must perform various human-robot interactions (HRIs), 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, users must be queried regarding their individual availability and locations for the day, as well as their preferences for participation in various activities (a 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.

### Users

We consider $n$ human users, $U := \{u_1, u_2, \ldots, u_n\}$, residing in the retirement home. These users share the environment and participate in several activities throughout the course of the day. Each user has a unique calendar of $\alpha$ time intervals where they aren't available for interaction, where the total set of such calendars is defined by the set $\Sigma := \{\sigma_1, \sigma_2, \ldots, \sigma_n\}$, where each $\sigma_i := \left\{\left[s_1^{\sigma_i}, e_1^{\sigma_i}\right], \left[s_2^{\sigma_i}, e_2^{\sigma_i}\right], \ldots, \left[s_\alpha^{\sigma_i}, e_\alpha^{\sigma_i}\right]\right\}$ for user $u_i \in U$ identifies their specific busy intervals. These busy intervals include breakfast (8:00 AM to 9:00 AM), lunch (12:00 PM to 1:00 PM), and dinner (5:00 PM to 6:00PM), as well as several 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 meters/minute, which is utilized to approximate user travel time within the environment.

### Robots

We consider $m$ heterogeneous mobile robots, $R := \{r_1, r_2, \ldots, r_m\}$, 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.

### Environment

The environment is divided into regions that represent rooms in the facility. A sample test environment we utilize for experimentation, both map and real-world image, is illustrated in Figures 1b and 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 meters. Travel times between locations are then represented in minutes for each user $u_i \in U$ as $\Delta_i^u := \left\{\frac{\delta_{(a,\ b)}}{v_i^u} : (a,b) \in L \times L\right\}$, and for each robot $r_k \in R$ as $\Delta_k^r := \left\{\frac{\delta_{(a,\ b)}}{v_k^r} : (a,b) \in L \times L\right\}$.

### Activities

An *activity* is either a direct assistive interaction with a user or 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, $P := \{p_1, p_2, \ldots, p_n\}$, each with a length of 30 minutes. Bingo game activities are group HRIs in which users participate in a game of robot-facilitated bingo. Bingo games, $G := \left\{g_1, g_2, \ldots, g_{UB_1}\right\}$ 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 his or her participation

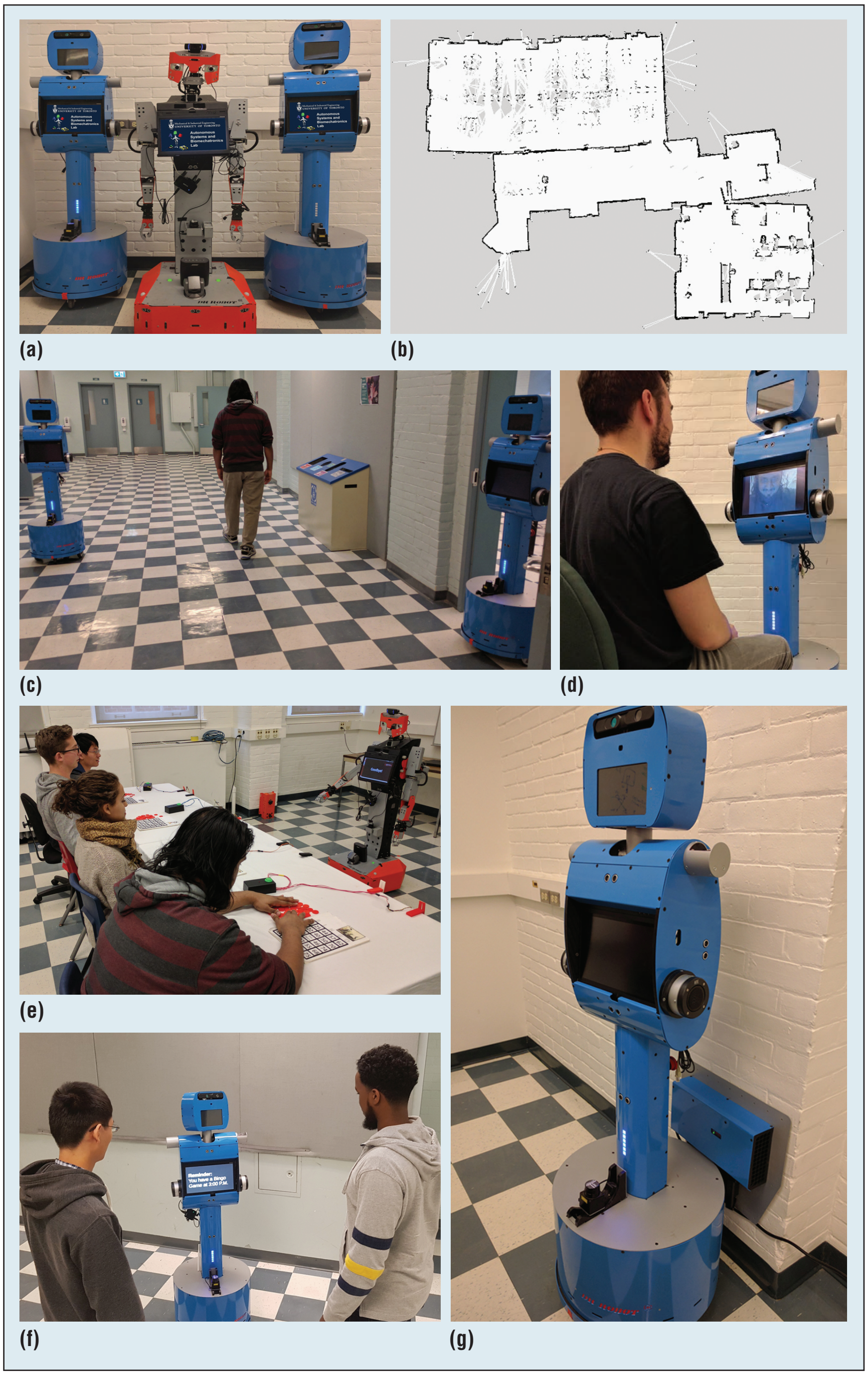


**Figure 1. Problem definition. Assistive robot fleet, test environment, and activities: (a) heterogeneous multirobot system (MRS), (b) environment map, (c) MRS environment navigation, (d) facilitating a telepresence, (e) facilitating a group bingo game, (f) bingo game reminder, and (g) robot recharge activity.**

in an upcoming bingo game. The set of bingo reminder activities is then defined by $\mathcal{M} := \bigcup_{i=1}^{n} M_i$, where $M_i := \{m_{i1}, m_{i2}, \ldots, m_{iUB_1}\}$. 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 $C_k := \{c_{k1}, c_{k2}, \ldots, c_{kUB_2}\}$. 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 can be scheduled.[4] Information-gathering sessions are HRIs that occur at the beginning of each day to query users about their availability, locations throughout the day, and activity preferences.

## Multirobot Task Planning and Execution Architecture

The proposed MRTPE architecture represents a multirobot extension of a single-robot system.[7] In addition to the integration of multiple robot controllers through a master/slave configuration, the proposed system uses 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) framework.

### Architecture Design

As illustrated in Figure 2, the architecture consists of two levels: the centralized server, and 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 (battery levels, poses) and environment states (region accessibility), while the

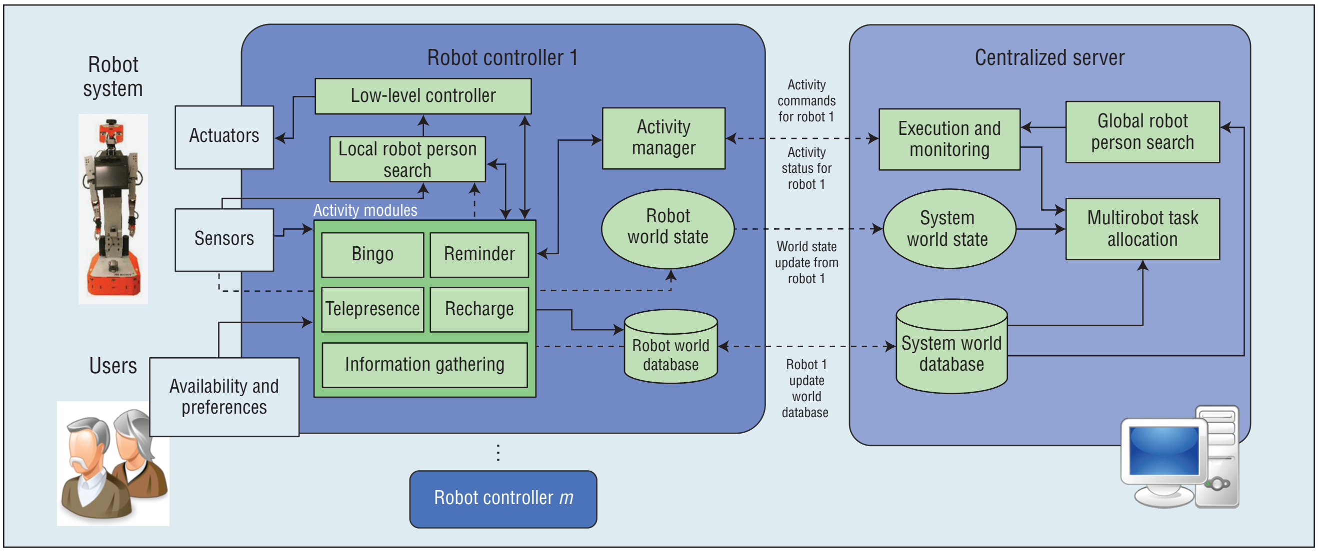


**Figure 2. Proposed multirobot task planning and execution (MRTPE) architecture.**

system world database module contains information regarding static parameters, such as the map. The individual robot controllers include the following modules: activity manager, robot world state, robot world database, activity modules (information gathering, bingo, and so on), 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 behaviors 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, only 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.

## 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, which 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 region specified in the global plan. This 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 the system world database module.

## Robot Person Search

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

***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 the retirement home residents during the time period 7:00 AM to 8:00 AM. The global plan consists of a subset of regions to search with their corresponding search times, and the order in which to search this subset. To determine the global plan, a traveling thief problem is solved using a dynamic programming algorithm.[3] This 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 as ⟨*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 sent to the robot controller, which in turn results in the local RPS module implementing a search to find users within a region.

***Local Robot Person Search.*** This module receives a request from the activity module when the robot arrives at the specified region, which 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 a target user. A silhouette detection algorithm compares 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 his or her location. 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) to determine the person's orientation with respect to the robot. Finally, the robot navigates to face the person for interaction and determines their identity by applying the OpenCV local binary patterns histogram recognizer to identify facial features and compare them with the unique facial features stored for each user in the robot world database.

## Multirobot Task Allocation and Scheduling

With user availabilities, locations, and preferences provided by the RPS, we model and solve an MRTA problem[8] using the MRTA module in our architecture.

***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 CP, a model-and-solve paradigm similar to integer programming (IP). CP is utilized for our architecture as it was shown to significantly outperform an IP technique for a similar problem.[4]

CP is more general than IP, relaxing restrictions of linearity and expressing richer variable types (for example, 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.

***Problem modeling.*** 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 $D_{var} := \{\perp\}\cup\{[s,e) | s,e \in \mathbb{Z}, s \le e\}$. That is, $var$ takes on a value that is a convex interval with integer start and end points, $s$ and $e$, respectively, or $\perp$ indicating the variable isn't present in the solution. The latter assignment is represented by the expression Presence($var$) evaluating to 1 if $var \in X$ is present in the solution and 0 otherwise. Start($var$), End($var$), and 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*, which are variables that model cumulative resources through the impact of interval variables.

***Decision variables.*** As in previous work,[4] we define the decision variables for our CP formulation as follows:

$x_{ij}$ := (*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.
$y_{kj}$ := (*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.
$E_k$ := (*cumulative function expression*) representing the energy level of robot $r_k \in R$ throughout the schedule.

***Objective function.*** The objective function, Equation 1, is to maximize bingo game user participation to boost users' cognitive and social stimulation, while prioritizing schedules with fewer robot recharges:

$$\begin{aligned}\underset{\{x,y\}}{\operatorname{argmax}} & \sum_{u_i \in U} \sum_{j \in G} \text{Presence}(x_{ij}) \\ & - 0.01 \cdot \sum_{r_k \in R} \sum_{j \in C_k} \text{Presence}(y_{kj})\end{aligned}. \quad (1)$$

***Problem constraints.*** The first set of problem constraints, Equations 2 and 3, ensure that scheduled activities don't 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 $T_k^r := \{P \cup G \cup \mathcal{M} \cup C_k\}$:

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

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

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

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

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

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

Equation 7 ensures that, if a user participates in a bingo game, the corresponding reminder is facilitated. Through Equation 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:

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

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

To represent the problem's energy-related components, Equations 9 through 11 model each robot's battery level, 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. The term $pre_j$ returns the task prior to $j$ in a robot's schedule, and $loc(j)$ represents the location of task $j$:

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

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

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

The modeled 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[4] to improve scheduling performance are also utilized.

## Experiments

To validate the architecture's utility in a physical MRS, we assess three real-world scenarios. The experiments are conducted on a multiroom floor of an engineering building at the university with multiple students representing retirement home users. Table 1 presents details of these scenarios, and Figure 3 shows a sample discretization of a subset of the facility regions (a total of 12 personal and general-purpose rooms, with some overlapping regions). For example, Scenario 2 involves 7 human users and 3 assistive mobile robots in an environment containing 12 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

**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 |

**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) |

**Table 3. MRTA system and local RPS experimental results.**

| Performance metric | Scenario 1 | Scenario 2 | Scenario 3 |
|---|---|---|---|
| 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 |
| Final solution status | *Optimal* | *Optimal* | *Optimal* |
| 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) |

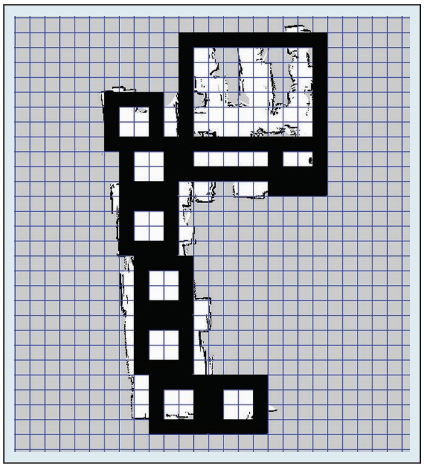

**Figure 3. Discretization of regions within the experimental facility.**

for system execution (such as 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 in the environment. Table 2 presents the results. 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. Table 3 presents these results.

The first part of the experiments consider the period of time from 7:00 AM to 8:00 AM, 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 to 100 percent across the scenarios. The relatively inferior performance in Scenario 1 was caused by a poorly mapped area of the facility that resulted in the need for external assistance. This was remedied in future scenarios by re-mapping the facility. User identification has the lowest success rate of all; users standing too close to walls/corners of regions weren't consistently identified properly. This is an area we plan to improve on in future work.

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

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. R. Kachouie et al., “Socially Assistive Robots in Elderly Care: A Mixed-Method Systematic Literature Review,” *Int’l J. Human-Computer Interaction*, vol. 30, no. 5, 2014, pp. 369–393.
2. T. Vaquero et al., “The Implementation of a Planning and Scheduling Architecture for Multiple Robots Assisting Multiple Users in a Retirement Home Setting,” *Proc. Workshops 29th AAAI Conf. Artificial Intelligence*, 2015, pp. 47–52.
3. S.C. Mohamed and G. Nejat, “Autonomous Search by a Socially Assistive Robot in a Residential Care Environment for Multiple Elderly Users Using Group Activity Preferences,” *Proc. Workshop Planning and Robotics*, 2016, pp. 58–66.
4. K.E.C. Booth, G. Nejat, and J.C. Beck, “A Constraint Programming Approach to Multirobot Task Allocation and Scheduling in Retirement Homes,” *Proc. Int’l Conf. Principles and Practice of Constraint Programming*, 2016, pp. 539–555.
5. M.B. Dias, “Traderbots: A New Paradigm for Robust and Efficient Multirobot Coordination in Dynamic Environments,” PhD thesis, The Robotics Institute, Carnegie Mellon Univ., 2004.
6. G.A. Korsah, A. Stentz, and M.B. Dias, “A Comprehensive Taxonomy for Multirobot Task Allocation,” *Int’l J. Robotics Research*, vol. 32, no. 12, 2013, pp. 1495–1512.
7. W.-Y. Geoffrey Louie et al., “An Autonomous Assistive Robot for Planning, Scheduling and Facilitating Multi-User Activities,” *Proc. IEEE Int’l Conf. Robotics and Automation*, 2014, pp. 5292–5298.
8. B.P. Gerkey and M.J. Matarić, “A Formal Analysis and Taxonomy of Task Allocation in Multirobot Systems,” *Int’l J. Robotics Research*, vol. 23, no. 9, 2004, pp. 939–954.
9. P. Laborie, “IBM ILOG CP Optimizer for Detailed Scheduling Illustrated on Three Problems,” *Proc. Int’l Conf. AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, 2009, pp. 148–162.
10. C. Gervet, “Constraints over Structured Domains,” *Foundations of Artificial Intelligence*, vol. 2, 2006, pp. 605–638.
11. F. Rossi, P. Van Beek, and T. Walsh, *Handbook of Constraint Programming*, Elsevier, 2006.
12. P. Baptiste, C. Le Pape, and W. Nuijten, *Constraint-Based Scheduling: Applying Constraint Programming to Scheduling Problems*, Springer Science & Business Media, 2012.


## THE AUTHORS


**Kyle E.C. Booth** is a PhD candidate in the Department of Mechanical & Industrial Engineering at the University of Toronto, where he’s a member of the Toronto Intelligent Decision Engineering Laboratory (TIDEL) and the Autonomous Systems and Biomechatronics Laboratory (ASBLab). His research interests include multirobot coordination, constraint satisfaction, and automated scheduling. Contact him at kbooth@mie.utoronto.ca.

**Sharaf C. Mohamed** is a PhD student in the Department of Mechanical & Industrial Engineering at the University of Toronto, where he’s a member of the ASBLab. His research interests include multirobot coordination, human-robot interaction, embedded systems, and autonomous robotics. Contact him at smohamed@mie.utoronto.ca.

**Sanjif Rajaratnam** is an M.A.Sc. student in the Department of Mechanical & Industrial Engineering at the University of Toronto, where he’s a member of the ASBLab. His research interests are robotics and autonomous systems. Contact him at sraj@mie.utoronto.ca.

**Goldie Nejat** is an associate professor in the Department of Mechanical & Industrial Engineering at the University of Toronto, and founder and director of the ASBLab. She’s also an adjunct scientist at the Toronto Rehabilitation Institute. Her research interests include intelligent assistive/service robots, human-robot interactions, and semiautonomous/autonomous control. Nejat received a PhD in mechanical engineering from the University of Toronto. Contact her at nejat@mie.utoronto.ca.

**J. Christopher Beck** is a professor in the Department of Mechanical & Industrial Engineering at the University of Toronto, and founder and director of the TIDEL. His research interests include scheduling and planning, constraint programming, mathematical programming, operations research, and artificial intelligence. Beck received a PhD in constraint-directed scheduling from the University of Toronto. Contact him at jcb@mie.utoronto.ca.