Planning and Scheduling Single and Multi-Person Activities in Retirement Home Settings for a Group of Robots

Tiago Vaquero and **Goldie Nejat** and **J. Christopher Beck**
Department of Mechanical and Industrial Engineering, University of Toronto,
5 King's College Road, Toronto, Ontario, Canada M5S 3G8
{tvaquero, nejat, jcb}@mie.utoronto.ca

Abstract

Automated planning and scheduling (P&S) technology has been increasingly investigated and applied to various robotics applications. We introduce a challenging P&S problem in which multiple social robots must autonomously organize and facilitate human-robot interactions for one-on-one telepresence sessions and multi-user Bingo games. These activities need to take place throughout the day based on the individual availabilities of the residents living in a retirement home. We utilize a domain-independent P&S approach for this problem, studying different variations of a PDDL model and the performance of state-of-the-art temporal planners in five different scenarios. We demonstrate the modeling challenges and technological gap in domain-independent P&S technology for such real-world robot problems. In particular, modeling a combination of metric quantities, resources, temporal availability of residents and time constraints on cascading actions is non-trivial. Moreover, we show that the available temporal planners perform poorly on the problem and struggle with the optimization aspects of such real-world scenarios.

Introduction

Due to the rapid aging of the world's population and the shortage in healthcare professionals, robotic technologies are being increasingly developed to engage the elderly in cognitively and socially stimulating activities in eldercare environments (Pineau et al. 2003; Kidd, Taggart, and Turkle 2006; Fasola and Mataric 2012; McColl, Louie, and Nejat 2013). While some of this work has incorporated automated planning and scheduling (P&S) (Pineau et al. 2003; Pollack 2005; Cesta et al. 2011), the majority of the existing research in robotics and P&S in eldercare environments has focused on the human-robot interaction (HRI) activities of a single robot in one-on-one activities with a single user. Only a handful of works have considered robots interacting with multiple people at the same time, e.g., (Montemerlo et al. 2002). However, these robots have not actively distinguished between users to provide personalized interactions during multi-user activities. Given the variety of users' abilities and availabilities, multi-user assistance activities require robots to plan, schedule, and customize their HRI interactions to the needs, time constraints, availability and preferences of each individual during the day. An environment with multiple users, multiple robots, and single- and multi-person HRI activities has not been addressed in the P&S literature.

In this paper, we introduce such an environment and the associated P&S problem. A set of robots has to search for and interact with multiple residents living in a retirement home to perform a set of telepresence sessions (single-person activity), Bingo games (multi-person activity), and reminder deliveries. In addition, such activities deplete a robot's batteries and so a recharging activity may also be necessary.

The proposed problem provides a complex combination of reasoning about actions, resources (e.g., the robots), time windows (e.g., user availability), temporal constraints (e.g., activity deadlines), metric quantities (e.g., battery level), and optimization (e.g., maximizing the number of residents taking part in a Bingo game). Since the 1980s there has been a recurring discussion in the literature regarding the challenges of combining these elements, which have often been investigated independently (Fox 1999; Smith, Frank, and Jonsson 2000). However, developing solvers for P&S applications that include these features is still an open challenge.

The novelty of this work lies in: 1) presenting a new P&S problem for assistive robotics in retirement homes that considers multiple robots, users and user schedules, as well as single-user and multi-user HRI activities, and 2) an investigation of the state-of-the-art domain-independent temporal planners to solve the proposed problem.

Background

Our long-term project is the deployment of intelligent human-like mobile robots in retirement homes to engage residents daily in stimulating recreational activities (Louie, Han, and Nejat 2013; Louie et al. 2014). We use the robotic platform H20 from Dr Robot (Dr Robot 2014) and have designed the robot to: 1) navigate using a laser range finder and 3D depth sensors, 2) detect users with 2D cameras, and 3) interact with users through speech, gestures, and a touch screen. While the implementation of the robot behaviors addresses real robotics challenges (e.g., sensing, HRI, person recognition), herein we focus on the planning and scheduling of the daily activities of the social robots. Details of robot implementation can be found in (Louie et al. 2014).

We focus on two representative activities: *telepresence* and *Bingo*. In the former, the robot autonomously navigates to the user in his/her private room, prompts the user for the video call, starts the call and tracks the user during the session. For the Bingo game, the robot autonomously finds and reminds participants about the game prior to its start and then navigates to a specified location to conduct the game. During Bingo, the robot acts as the game facilitator, calling out numbers, verifying Bingo cards, prompting players to mark missed numbers and celebrating with winners. Currently, a centralized server is being designed to plan, schedule and monitor the daily activities of the robots. Specific behaviors are planned and performed locally by each individual robot platform.

The integration of planning and scheduling techniques has been investigated over the past several years in such robotic applications as container transportation robots (Alami et al. 1998), office assistant robots (Beetz and Bennewitz 1998), planetary rovers (Estlin et al. 2007), hospital assistant robots (Pecora and Cesta 2002), and eldercare robots (Pineau et al. 2003; Cesta et al. 2011). In these applications, single robot approaches are more commonly studied. With respect to HRI activities, existing work has mainly focused on automated reasoning about the schedule of a single user. For example, the Pearl robot (Pineau et al. 2003) uses the Autominder system (Pollack 2005) to reason about an elderly person's current and planned activities to determine if and when reminders should be provided. The Autominder system has not been extended to consider multiple users. The Cobot robots (Coltin, Veloso, and Ventura 2011) plan and schedule HRI activities, including semi-autonomous telepresence, and office tasks based on requests from several users. However, the planning and scheduling are managed independently and the user schedules are not considered as constraints for the robots' activities. Although multiple user schedules have been considered in other non-robotic scheduling and optimization applications (e.g., building energy conservation (Kwak et al. 2012)), in this work we focus on problems in which an integration of both planning and scheduling is required to reason about the schedules of multiple users, limited resources and metric quantities, and both single- and multi-user HRI activities.

The Problem

We define the main elements of the proposed problem: the environment in which the residents (users) and robots interact, the constraints, the goal and preferences. The constraints for the telepresence and Bingo activities were obtained from meetings with directors, healthcare professionals and residents from Toronto area retirement homes.

The Retirement Home Environment

We consider a floor in a retirement home. The environment consists of rooms, corridors and hallways that are discretized as a set of locations, L (l_1 ... l_n), within which the users and robots will interact. The set of locations and the distance between any two locations ($d_{i,j}$) are known a priori.

Users

The users are the residents of the retirement home. We consider a set of users, U (u_1 ... u_n), for which each user u_k has his/her own *profile*. The profile consists of the user's private *room* location; a minimum, att_min_{uk}, and maximum, att_max_{uk}, number of Bingo games to play in a day; and his/her own distinct *schedule* for the day, representing the user availability (in time and space) for interaction with a robot.

A day for users starts at 7am and ends at 7pm. Within this time frame, users in different locations can be either available or unavailable for interaction with a robot. All users are considered unavailable during breakfast (8am-9am), lunch (12pm-1pm), and dinner (5pm-6pm) and can have other unavailabilities already scheduled.

The Assistive Robots

We consider a set of assistive robots, R (r_1 … r_n), in which each robot r_l is able to execute the following activities: 1) *move* from one discrete location to another at a constant speed v_{rl}, 2) perform a *telepresence* session with a user, 3) perform a *Bingo* session with a group of users, 4) provide a *reminder* to each user prior to a Bingo game, and 5) *recharge* its battery at a charging station. Since battery consumption depends on the activity, whenever the robot, r_l, executes an activity, its *battery level*, bl_{rl}, must remain within bounds (i.e., $bl\_min_{rl} <= bl_{rl} <= bl\_max_{rl}$). A constant rate cr_move_{rl} is used to specify the power consumed for the moving activity (e.g., V/m). Each HRI activity has a different constant consumption rate (e.g., V/min): cr_telep_{rl}, cr_remind_{rl} and cr_bingo_{rl} for the corresponding activities. Battery power is regained through a charging station. A constant recharging rate rr_{rl} (e.g., V/min) is used to estimate the duration of a recharging process of a robot r_l. The battery of the robot can be recharged up to bl_max_{rl}.

Charging Stations

A set of charging stations, *CS* ($cs_1 \ldots cs_n$), exists for recharging. Each station is at a fixed location and can accommodate at most one robot at a time.

Telepresence Sessions

A set of *telepresence sessions*, *S* ($s_1 \ldots s_n$), must be scheduled during the day. Each session s_y is characterized by: 1) the user u_k; 2) the duration, dur_{sy}, (e,g, 30 min); and 3) the time window(s) it can occur in. The session should always take place in the user's room (l_{uk}).

Bingo Games

A set of *Bingo games*, *G* ($g_1 \ldots g_n$), should be scheduled during the day, if possible. For each game g_z, the robots will assign, find, and remind participants prior to the game and, then, play Bingo at a specific location, the games room (l_{game}), at the scheduled time. Only one game can occur at any given time. Only one robot can conduct the game, but the robots can collaborate to deliver the reminders. Each game g_z is characterized by: 1) the duration of the game, dur_{gz}, (e.g., 60 min) and of the reminder, dur_remind_{gz}; 2) the minimum and maximum number of participants, p_min_{gz} and p_max_{gz}; and 3) the time window(s) in which it can occur.

The group of participating users of a game g_z is not known a priori nor is the time of each game. Users are assigned to each game based on their schedules and attendance preferences and games are scheduled to fit the users' availabilities. Reminders must be delivered to *all* assigned users between 15-120 minutes before the game starts. It is assumed that the users will go to the games room at the time specified.

Robot Activities

We describe below the conditions and constraints of the available robot activities.

Navigate to a target location: the robot has to have enough battery power to reach the target location l_j from its current location l_i. The power consumption and the duration of the moving activity are $d_{i,j}\times cr\_move_{rl}$ and $d_{i,j} / v_{rl}$, respectively.

Recharge battery: the robot has to be in a location with an idle charging station and the battery level has to be less than the battery capacity, $bl_{rl} < bl\_max_{rl}$. The duration of the activity is $(bl\_max_{rl} - bl_{rl})/rr_{rl}$.

Perform Telepresence Session: the robot has to be in the private room of the specified user, who must be available during the entire duration (dur_{sy}) of the activity. The power consumption of the activity is $dur_{sy}\times cr\_telep_{rl}$.

Play Bingo Game: the robot has to be in the games room, no other game can be ongoing, and all users must be available during the entire duration (dur_{gz}) of the game. All assigned users must have been reminded 15-120 minutes before the game starts. The power consumption of the Bingo activity is $dur_{gz}\times cr\_bingo_{rl}$.

Remind User: the robot has to be at the same location as the user, who cannot be interacting with another robot and must be available during the entire duration (dur_remind_{gz}) of the activity. The power consumption of the reminder activity is $dur\_remind_{gz}\times cr\_remind_{rl}$.

In all the activities (except recharging), the robot has to have enough power to reach a location that has a charging station after the activity is completed.

Input, Goal, and Preferences

The *input* of the problem is the sets of locations *L,* users *U* (including their corresponding profiles), charging stations *CS*, available robots *R* (with their initial location and corresponding velocity, battery levels and limits, and consumption rates), and the requested telepresence sessions *S* and Bingo games *G* with their corresponding properties. The *goal* is to have a plan of robot activities in which: 1) all the requested telepresence sessions are scheduled, and 2) the requested Bingo games and reminders are scheduled, if possible, given that user attendance preferences have to be satisfied. All robots must be at a recharging location at the end of the day. As a *multi-objective optimization problem*, we want to: 1) perform as many Bingo games as possible, 2) have as many users playing Bingo as possible, 3) provide reminders as close as possible to the game times, and 4) expend as little battery power as possible.

An Automated P&S Approach

We address the proposed problem using a P&S approach. Herein, we use the itSIMPLE Knowledge Engineering (KE) (Vaquero et al. 2009; 2013) tool that follows an object-oriented modeling approach using the *Unified Modeling Language* (UML) (OMG 2005) and generates a PDDL model of the target problem.

Domain Modeling

A visualization of the modeled object types (classes), fluents and operators is provided in the UML class diagram in Figure 1. The most important classes are: *Location, GamesRoom*, *ChargingStation*, *Robot*, *User*, *TelepresenceSession*, *BingoGame* and *Global*. The *Location* and *GamesRoom* (a specialization of Location) represent the topology of the retirement home. The *distance* between locations, and the distance between each available charging station and these locations are represented in the class *Global.* A games room is said to be *free* when no game is taking place at the location. A *ChargingStation* is said to be *idle* when no robot is docked for charging. Moreover, *Robots* and *Users* can only be *at* one location at a time.

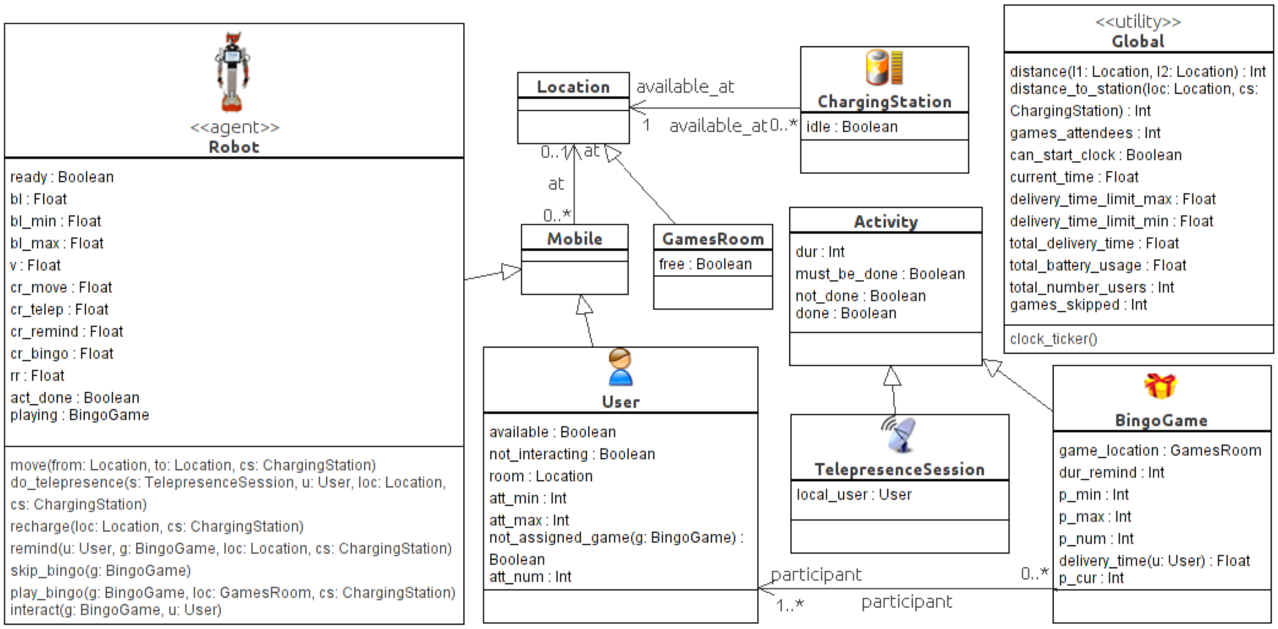

Figure 1. The UML Class diagram of the proposed problem model.

The class *User* has a set of properties to represent the user's profile. The predicate *room* specifies the user's private room while the predicate *available* is used to represent the availability of the user during the day. This availability is translated into PDDL in the form of timed initial literals (TILs) (Edelkamp and Hoffman, 2004) by assigning the *available* predicate to true or false in specific time intervals. We also represent the known locations of the user during the day with TILs. We represent the user preference on attending games (*att_min*, *att_max*), the variable for the number of games attended (*att_num*), and the predicate *not_assigned_game* to list all the games to which a user has not yet been assigned. When a user is interacting with a robot, the predicate *not_interacting* is set to false to prevent other robots from interacting.

The classes *TelepresenceSession* and *BingoGame* represent the HRI activities. Both have the properties: *dur* to represent duration; *not_done* and *done* to represent if the activity has been performed; and *must_be_done*, TILs to represent the time windows in which the activity can be performed. In addition to the properties of the sessions and games introduced in the problem description, we have added the properties *p_num* and *p_cur* to control the number of users reminded by the robots and the number of users playing the game, as well as *delivery_time* to control the time each user is reminded about the game. The difference between the reminder delivery time and the start of the game must be within 15-120 minutes.

Modeling the reminder delivery constraint is not possible without using features that have not been officially incorporated into PDDL. The planner would have to explicitly reason about continuous time during the planning process itself to determine that two actions (*reminder* and *playbingo*) are a certain time apart. This can be done by using PDDL+ which includes *processes* (Fox and Long 2006). Herein, a *process* (called *clock_ticker* in the class *Global*) models an exogenous activity that is triggered for as long as a condition holds (in this case *can_start_clock*), regardless of the action selection process. This mechanism allows us to increment the variable *current_time* in every step of the search, simulating the passage of time. If *current_time* is used in an action's precondition it will hold the exact start time of the action. We use this variable to record the time each user is reminded (*delivery_time*) and also to check if the start time of a game is within the time constraints of the reminders.

The class *Global* also holds global variables including the maximum and minimum time for delivering reminders prior to the games, the total time generated by adding all the lengths of the time intervals between the reminders and the game (*total_delivery_time*), the total amount of battery power consumed by all robots (*total_battery_usage*), the total number of games not played (*game_skipped*), and the number of target users (*total_number_users*). These variables are used to specify the cost function and are manipulated in the specification of the robot actions.

The class *Robot* has all the properties described in the problem description (e.g., velocity, battery level, etc). In addition, we have the predicates *ready*, *act_done*, and *playing*. A robot is *ready* when it is not engaged in any activity and it is *playing* when it is performing a Bingo activity. The predicate *act_done* prevents a robot from going to a location and performing no action: a robot can only move to another location if it has completed an activity in its current location. As shown in Figure 1, a robot has the following operators: *move* to a target location; *recharge* its battery; *remind* users; *do_telepresence* with a user; *play_bingo* and *interact* with a player during the game; and *skip_bingo* which removes the game from the request list.

In the *reminder* operator, the user is set as a *participant* of the game. In order to play a game after the reminders, a robot has to first start the *play_bingo* action, then it has to perform, in parallel (a required concurrency), the action *interact* with each participant. The *play_bingo* action can only finish when the robot has performed the *interact* action with all assigned players.

In the goal state all sessions and games must be *done* (Bingo games can be either performed or skipped) and the user preferences on game attendance must be satisfied. We aim to minimize the following weighted cost function f:

$$f = 500\times(games\_skipped) + 1000\times(total\_number\_users - games\_attendees) + total\_battery\_usage + total\_delivery\_time \quad (1)$$

where the weights are used to express preference on optimizing the number of games and players. Due to space limitations, we present the PDDL code for the proposed model at:

https://docs.google.com/file/d/0B3t9fqfsJqrlTGpndVNxMEdSSlU/edit?pli=1.

Model Variations

The resulting PDDL model includes features that are challenging for most existing planners: metric quantities, optimization, temporal actions, timed initial literals, concurrent actions, and processes. In particular, few planners can properly handle the required concurrency (R) and processes (P). Therefore, we decided to define model variations to investigate the performance of existing temporal planners. Model ***RP*** is our full model as described above. Model ***RN*** does not consider the reminder time constraint and therefore, does not use processes. Model ***NP*** is our full model without the required concurrency in the Bingo activity. We replace both operators *play_bingo* and *interact* with operators *play_bingo3* and *play_bingo4*, each representing a game activity with a specific number of participants. Representing an operator for each possible number of participants, in this case from 3 to 10, is impractical due to the large number of parameters and, consequently, an exponentially increasing number of action instantiations during the planning procedure. Therefore, the maximum number of Bingo game participants is 4 when using the operators *play_bingo3* and *play_bingo4*. Model ***NN*** is the full model with both required concurrency and processes removed.

Experiments

We chose five planners to investigate: COLIN (Coles et al. 2012), LPG-td (Gerevini, Saetti, and Serina 2004), OPTIC (Benton, Coles, and Coles 2012), POPF (Coles et al. 2010) and SGPlan (Hsu and Wah 2008). All these planners can potentially handle metric quantities, optimization, temporal actions, and timed initial literals. However, only OPTIC, COLIN and POPF handle the required concurrency and processes.

We consider a realistic retirement home environment in which residents have several activities in different locations (e.g., TV room, private room, garden, dining hall) during a day. We assume that each user has a number of 1-hour activities (e.g., physiotherapy, doctor's appointment, family visit, nap), in addition to the meal times, during which the robots cannot disturb him/her (herein called *non-interruptible activities*). Other activities (e.g., walk in the garden, reading in a common area) allow robot interactions (*interruptible activities*); at least one interruptible activity is assumed for each user. We analyze the selected planners for five full-day scenarios in this environment (7am-7pm) – see Table 1. For each full-day scenario, we analyze different numbers of non-interruptible activities for the users. We investigate non-interruptible activity density, defined as *Density k*, ($k = 0,1,2,3,4$), where k is the number of non-interruptible activities, in addition to the meals, that each user has per day. The different densities in particular are aimed to study the impact of the user availability constraints on the performance for the selected planners.

Table 1. The number of objects in the five scenarios.

Scenario	*Users*	*Robots*	*Telepresence*	*Bingo*
1	5	2	2	1
2	10	2	4	3
3	15	3	6	5
4	20	3	8	6
5	25	4	10	8

In all scenarios, the telepresence sessions and Bingo games are 30 and 60 minutes long, respectively with time windows from 7am-7pm. Reminders are 2 minutes long. Each game has a minimum of three participants and a maximum of ten participants (in models *NP* and *NN* the maximum is four as previously noted). Every user is willing to attend at most one Bingo game during the day (i.e., att_min = 0, att_max = 1). Each scenario was designed so that it is feasible to schedule at least one game with five participants. All robots have the following property values, estimated based on the H20 robot platform: bl_min = 0, bl = bl_max = 20, v = 20m/min, rr = 0.5, cr_move = 0.04 and cr_telep = cr_remind = cr_bingo = 0.1.

We run the planners for each model variation with each of the five scenarios and each density on a 64-bit Ubuntu Linux machine with 32 GB of memory. A 1-hour timeout was used for each planner in each scenario. We measure the solvability of the planners, the runtime, the number of states evaluated, and the number of users attending a game. Table 2 shows the number of scenarios (out of 5) for which each planner was able to generate at least one solution.

Table 2. Number of scenarios solved by each planner. The '-' indicates that the planner could not represent the model, while the '$^{(inv)}$' indicates that the planner generates invalid solutions for some scenarios. Such solutions are not included in the number of scenarios solved.

Planners	**Models**			
	RP	*RN*	*NP*	*NN*
		Density 0		
COLIN	$0^{(inv)}$	$0^{(inv)}$	$0^{(inv)}$	$0^{(inv)}$
LPG-td	-	-	-	0
OPTIC	5	5	2	2
POPF	0	$1^{(inv)}$	0	$1^{(inv)}$
SGPlan	-	-	-	0
		Density 1		
COLIN	3	3	2	2
LPG-td	-	-	-	0
OPTIC	5	5	2	2
POPF	0	5	0	2
SGPlan	-	-	-	0
		Density 2		
COLIN	4	4	2	2
LPG-td	-	-	-	0
OPTIC	5	5	2	2
POPF	0	5	0	2
SGPlan	-	-	-	0
		Density 3		
COLIN	4	4	2	2
LPG-td	-	-	-	0
OPTIC	5	5	2	2
POPF	0	5	0	2
SGPlan	-	-	-	0
		Density 4		
COLIN	4	4	2	2
LPG-td	-	-	-	0
OPTIC	5	5	2	2
POPF	0	5	0	2
SGPlan	-	-	-	0

As shown in Table 2, the majority of scenarios were solved by some of the planners with models *RP* and *RN* while few scenarios were solved with models *NP* and *NN*. The OPTIC planner was the only P&S system able to solve scenarios with all models and in all investigated densities. Across all investigated models and densities, the solutions generated by the planners for a given scenario varied with respect to the number of robots used, total battery usage, and makespan. For example, COLIN generated plans with only one robot more often than POPF and OPTIC did for *Scenarios 1* and *2*. Having less robots resulted in a lower cost, however, using multiple robots had lower makespan. The number of Bingo games scheduled also varied. The majority of the solutions did not schedule a game at all and had the following common structure for the plan: skip the Bingo games and schedule the assigned robots to implement the requested telepresence sessions, while recharging the robots when necessary; and at the end of the day, the robots assigned in the plan return to the charging station. The solutions with scheduled Bingo games had a similar structure as those with no games, however in these cases the robots assigned in the plan also scheduled reminders to users prior to the start of a Bingo game as well as the game playing session.

LPG-td and SGPlan were not able to solve any of the problem instances with model *NN*, the only model that these planners could represent. We suspect that this is due to the large number of TILs used to represent the user schedules. The COLIN planner was able to generate solutions for the four proposed models and POPF was able to generate solutions only for models *RN* and *NN*. However, none of the solutions from COLIN and POPF for the five scenarios had any Bingo games: all the games were skipped. Furthermore, these two planners generated invalid solutions in problem instances with *Density 0*, for example, scheduling telepresence activities during the breakfast period. Interestingly, neither planner generated invalid solutions at higher densities. This issue occurs when there is no non-interruptible activity in beginning of the day (7am), i.e. all users start with the variable *available* set to true in the initial state and this variable does not change until the beginning of breakfast, when it is set to false. In the problem instances with density greater than zero, some users have non-interruptible activities starting at 7am, so their corresponding variable *available* is false in the initial state. In such cases no invalid solutions were generated. With COLIN we observed that the issue seems to be related to the compression-safe action detection mechanism (Coles et al. 2012). When this mechanism is disabled, the issue no longer occurs. POPF has a similar mechanism; however, disabling it does not eliminate the issue. Given that the compression-safe action detection is a default mechanism in both planners, we decided to keep it enabled in our experiments. Further experimentation and analysis is needed.

Table 3 shows the runtime and number of states evaluated for COLIN and POPF to find a solution with the four models. In most cases, the planners stopped before the timeout. The density of non-interruptible user activities seems to have some impact on the performance of both planners. Different trends are observed in Table 3. For example, the runtime decreases as the density increases in *Scenario 3* for COLIN with models *RP* and *RN*. Moreover, the runtime increases in *Scenario 5* for POPF with model RN as the density increases.

As OPTIC had the best performance, we ran it to search for better solutions until the timeout. OPTIC was the only planner that was able to find solutions that included Bingo games. Table 4 shows the runtime, number of states evaluated and the number of users playing Bingo games in the plans found by the OPTIC planner. This table focuses on the *first* and *last* solutions found to illustrate how fast the planner can find a solution and the quality of the best solution found. For problems for which the planner generated no solution or only one solution, we show the time the planner stopped instead.

As shown in Table 4, plans with Bingo games were only found in *Scenario 1*. Most of these plans were found with models *RN* and *NN*, i.e., the models without the reminder time constraint. Problem instances from *Scenario 1* with *Density 0* are the only instances in which OPTIC generated plans with Bingo games with all models. OPTIC generated solutions for *Scenarios 3*, *4* and *5* across all non-interruptible user activity densities only with models *RP* and *RN* (models with required concurrency). Most of the solutions with the highest number of Bingo participants were generated with model *NN*, the simplest PDDL model. During the optimization process of all the scenarios, most improvements to the plan resulted in lower battery consumption. OPTIC stopped running before the timeout in most cases. While the reason is unclear but we suspect that it reached its internal memory limits.

With respect to the impact of the different non-interruptible user activity densities, Table 4 shows that both the runtime and the number of evaluated states increased as the density was increased with model *RP* and *RN* in *Scenarios 4* and *5*. An increase in runtime can also be observed with modes *NP* and *NN* in *Scenario 2*. In order to investigate whether the different non-interruptible user activity densities affected the performance of the planner on finding a solution with a Bingo game, we further investigated the very first solutions found by OPTIC in which a Bingo game was scheduled. Table 5 shows the runtime and number of evaluated states for those cases in *Scenario 1* across the four models and the five densities. The density increment tends to decrease the runtime to find a solution with a Bingo game as well as the number of evaluated states with models *RN* and *NP*. We suspect that this pattern is due to the decreasing number of time windows in which a game can be scheduled leading to a reduction in the alternatives during the search.

Discussion

The experimental results show that existing domain-independent temporal planners are not able to solve the proposed multi-robot, multi-user, single and multi-user HRI activities problem for realistic scenarios. Although some of the planners provide feasible solutions, optimal solutions do not appear to be achievable. In particular, the expected optimization of the number of Bingo players is observed in very few small-scale cases, most of the time in models that do not consider the full requirements of the problem.

The advancement of P&S technology in representing and solving problems with temporal constraints, time-windows and numeric quantities is noticeable since the 1980s (Boddy, Cesta, and Smith 2004). However, the challenges in modeling and solving problems that require integrated P&S with such a combination of complex features are evident from our experiment. From the modeling perspective, not all temporal requirements can be represented in standard PDDL. While few planners can handle the aforementioned features together, even fewer represent the PDDL+ features our problem requires. Due to the few temporal planners available for these challenging problems, the modeling process becomes driven by the planner at hand. Namely, we found ourselves faced with tailoring the model to the solver's abilities at the expense of accurately representing our problem.

From the perspective of a user of AI planning technology (e.g., a roboticist who wants to focus on the challenges of sensing, navigation, and HRI), current domain-independent AI planning technology is not up to the task. We hope that the application we have introduced can form a challenge to spur advances in this direction.

We intend to extend this work to investigate timeline-based planning (Muscettola 1994) and scheduling approaches such as constraint programming and mixed-integer programming. Our preliminary indication is that none of these technologies will be able to reliably solve these problems. If that is the case, we intend to identify the most promising technology and investigate its extension to be able to solve our real-world problem.

Conclusion

We have introduced a new planning and scheduling problem in which multiple robots have to interact with residents in a retirement home environment to perform single- and multi-user activities while considering the users' schedules. We have investigated an AI P&S approach by: 1) designing variations of a PDDL model and realistic problem instances using the itSIMPLE KE tool, and 2) studying the performance of five state-of-the-art domain-independent temporal planners. Experimental results demonstrate that current temporal planners can sometimes provide valid solutions even with a complex combination of model features. However, in most of the cases they failed to provide solutions in which both single- and multi-user activities are present, even when using simplified models. The results reinforce the existing technology gap in the AI P&S approach for both modeling and solving real problems that combine temporal, numeric and optimization requirements.

Acknowledgments

This research has been funded by a Natural Sciences and Engineering Research Council of Canada (NSERC) Collaborative Research and Development Grant and by Dr Robot Inc.

References

Alami, R.; Chatila, R.; Fleury, S.; Ghallab, M.; and Ingrand, F. 1998. An Architecture for Autonomy. *International Journal of Robotics Research*, 17: 315-337.

Beetz, M., and Bennewitz, M. 1998. Planning, Scheduling, and Plan Execution for Autonomous Robot Office Couriers, *Integrating Planning, Scheduling and Execution in Dynamic and Uncertain Environments*, Workshop Notes, 1998.

Benton, J.; Coles, A. J.; and Coles, A. I. 2012. Temporal Planning with Preferences and Time-Dependent Continuous Costs. In *Proceedings of the International Conference on Automated Planning and Scheduling* (ICAPS-12), pp. 2-10.

Boddy, M.; Cesta, A.; and Smith, S. ICAPS-04 Workshop on Integrating Planning into Scheduling. Workshop proceeding: http://pst.istc.cnr.it/wipis-at-icaps-04/WIPIS-ICAPS04-Notes.pdf

Cesta, A.; Cortellessa, G.; Rasconi, R.; Pecora, F.; Scopelliti, M.; and Tiberio, L. 2011. Monitoring Older People with the RoboCare Domestic Environment: Interaction Synthesis and User Evaluation. *Computational Intelligence*, 27(1): 60–82.

Coles, A. J.; Coles, A. I.; Fox, M.; and Long, D. 2010. Forward-Chaining Partial-Order Planning. In *Proceedings of the International Conference on Automated Planning and Scheduling* (ICAPS-10), pp. 1-8.

Coles, A. J.; Coles, A. I.; Fox, M.; and Long, D. 2012. COLIN: Planning with Continuous Linear Numeric Change. *Journal of Artificial Intelligence Research*. 44: 1-96.

Coltin, B.; Veloso, M.; and Ventura, R. 2011. Dynamic User Task Scheduling for Mobile Robots. In *Proceedings of the AAAI Workshop on Automated Action Planning for Autonomous Mobile Robots*, pp. 1-6.

Dr Robot. 2014. H20 Wireless Networked Autonomous Humanoid Mobile Robot. Dr Robot Inc. http://www.drrobot.com/products_h20.asp.

Edelkamp, S., and Hoffmann, J. 2004. PDDL2.2: The language for the classical part of the 4th International Planning Competition. Tech. rep. 195, Albert-Ludwigs-Universität Freiburg, Institut für Informatik.

Estlin, T.; Gaines, D.; Chouinard, C.; Castano, R.; Bornstein, B.; Judd, M.; Nesnas, I.; and Anderson, R. 2007. Increased Mars Rover Autonomy using AI Planning, Scheduling and Execution. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), pp. 4911-4918.

Fasola, J., and Mataric, M. 2012. Using Socially Assistive Human-Robot Interaction to Motivate Physical Exercise for Older Adults. *IEEE, Special Issue on Quality of Life Technology*, T. Kanade, ed., 100(8): 2512-2526.

Fox, M. 1994. ISIS: A Retrospective. In Zweben, M., and Fox, M. 1994. *Intelligent Scheduling*. Morgan Kaufmann, pp. 3–28.

Smith, D. E.; Frank, J.; and Jonsson, A. K. 2000. Bridging the Gap Between Planning and Scheduling. *Knowledge Engineering Review*, 15(1): 1-34.

Fox, M., and Long, D. 2006. Modelling Mixed Discrete Continuous Domains for Planning. *Journal of Artificial Intelligence Research* 27:235–297.

Gerevini, A.; Saetti, A.; and Serina, I. 2004. LPG-TD: a Fully Automated Planner for PDDL2.2 Domains. In *Proceedings of the International Conference on Automated Planning and Scheduling* (ICAPS-04), booklet of the system demo section..

Hsu, C., and Wah, B. W. 2008. The SGPlan Planning System in IPC-6. In the booklet of the International Planning Competition (IPC), International Conference on Planning and Scheduling.

Kidd, C. D.; Taggart, W.; and Turkle, S. 2006. A Sociable Robot to Encourage Social Interaction among the Elderly, In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), pp. 3972-3976.

Kwak, J.; Varakantham, P.; Maheswaran, R.; Tambe, M.; Jazizadeh, F.; Kavulya, G.; Klein, L.; Becerik-Gerber, B.; Hayes, T.; and Wood, W. 2012. SAVES: a sustainable multiagent application to conserve building energy considering occupants. In *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems*, v.1, pp. 21-28.

Louie, W. G.; Han, R.; and Nejat, G. 2013. A Socially Assistive Robot to Promote Stimulating Recreational Activities at Long-term Care Facilities, *Journal of Medical Devices-Transactions of the ASME*, 7, 030944-1.

Louie, W. G.; Vaquero, T.; Nejat, G.; and Beck, J. C. 2014 An Autonomous Assistive Robot for Planning, Scheduling and Facilitating Multi-User Activities, In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA).

McColl, D.; Louie, W. G.; and Nejat, G. 2013. Brian 2.1: A Socially Assistive Robot for the Elderly and Cognitively Impaired. *IEEE Robotics & Automation Magazine*, 20(1): 74-83.

Montemerlo, M.; Prieau, J.; Thrun, S.; and Varma, V. 2002. Experiences with a mobile robotics guide for the elderly. In *Proceedings of the AAAI National Conference on Artificial Intelligence*, pp. 587–592.

Muscettola, N. 1994. HSTS: Integrating Planning and Scheduling. *Intelligent Scheduling*, ed., Zweben, M. and Fox, M.S., M. Kauffmann.

OMG. 2005. OMG Unified Modeling Language Specification, Version 2.0.

Pecora, F., and Cesta, A. 2002. Planning and Scheduling Ingredients for a Multi-Agent System. In *Proceedings for the of UK PLANSIG'02 Workshop*, pp. 135-148.

Pineau, J.; Montemerlo, M.; Pollack, M.; Roy, N.; and Thrun, S. 2003. Towards Robotic Assistants in Nursing Homes: Challenges and Results. *Robotics and Autonomous Systems*, 42(3-4): 271-281.

Pollack, M. 2005. Intelligent technology for an aging population: The use of AI to assist elders with cognitive impairment. *AI Magazine*, 26(2): 9–24.

Vaquero, T.; Silva, J. R.; Ferreira, M.; Tonidandel, F.; and Beck, J. C. 2009. From requirements and analysis to PDDL in itSIMPLE3.0. In *Proceedings of the International Competition on Knowledge Engineering for Planning and Scheduling* (ICKEPS), at ICAPS 2009, Thessaloniki, Greece, pp. 1-8.

Vaquero, T.; Silva, J. R.; Tonidandel, F.; and Beck, J. C. 2013. itSIMPLE: Towards an Integrated Design System for Real Planning Applications. *The Knowledge Engineering Review Journal*, 28(2): 215–230.

Table 3. Runtime (s) and number of evaluated states for the planners COLIN and POPF. The '-' indicates that no solution was found and the '$^{(inv)}$' indicates the invalid solutions.

Scenarios	POPF								COLIN							
	RP		*RN*		*NP*		*NN*		*RP*		*RN*		*NP*		*NN*	
	runtime (s)	states	runtime (s)	states	runtime (s)	states	runtime (s)	states	runtime (s)	states	runtime (s)	states	runtime (s)	states	runtime (s)	states
Density 0																
1	0.04	-	0.04	11	0. 06	-	0.10	11	0.04$^{(inv)}$	8	0.06$^{(inv)}$	8	0.16$^{(inv)}$	8	0.10$^{(inv)}$	8
2	0.07	-	0.18$^{(inv)}$	30	3.12	-	312.7$^{(inv)}$	30	0.14$^{(inv)}$	16	0.14$^{(inv)}$	16	1806.6$^{(inv)}$	16	179.0$^{(inv)}$	16
3	0.17	-	1.00$^{(inv)}$	65	46.9	-	timeout	-	0.70$^{(inv)}$	30	0.64$^{(inv)}$	30	timeout	-	timeout	-
4	0.32	-	2.56$^{(inv)}$	107	97.6	-	96.5	-	1.70$^{(inv)}$	56	1.52$^{(inv)}$	56	121.4	-	120.8	-
5	0.74	-	10.0$^{(inv)}$	188	308.3	-	282.8	-	5.82$^{(inv)}$	59	4.92$^{(inv)}$	59	432.6	-	414.5	-
Density 1																
1	0.04	-	0.12	127	0.12	-	0.24	127	0.08	127	0.08	127	0.20	127	0.14	127
2	0.08	-	0.66	336	3.20	-	318.6	336	1.88	2326	1.74	2326	1890.0	2326	194.1	2326
3	0.18	-	4.60	933	46.8	-	timeout	-	906.2	199559	628.3	199559	timeout	-	timeout	-
4	0.32	-	15.2	1676	98.1	-	96.3	-	2838.7	-	2118.1	-	121.7	-	120.7	-
5	0.74	-	66.4	3164	307.9	-	284.7	-	timeout	-	timeout	-	433.2	-	416.5	-
Density 2																
1	0.05	-	0.14	139	0.14	-	0.24	139	0.08	139	0.08	139	0.22	139	0.16	139
2	0.08	-	0.72	359	3.22	-	344.4	359	0.94	1052	0.86	1052	1905.8	1052	203.9	1052
3	0.18	-	6.08	1264	46.9	-	timeout	-	111.5	27155	76.9	27155	timeout	-	timeout	-
4	0.32	-	14.1	1615	98.4	-	95.9	-	2362.9	255989	1493.7	255989	122.1	-	120.5	-
5	0.75	-	85.7	4060	307.2	-	285.0	-	timeout	-	timeout	-	432.0	-	414.0	-
Density 3																
1	0.05	-	0.14	139	0.14	-	0.28	139	0.08	139	0.08	139	0.22	139	0.16	139
2	0.08	-	0.70	359	3.26	-	324.2	359	0.96	1052	0.88	1052	2000.7	1052	158.5	1052
3	0.18	-	5.96	1228	46.9	-	timeout	-	57.2	13911	39.0	13911	timeout	-	timeout	-
4	0.32	-	13.5	1570	97.7	-	96.0	-	2405.8	256743	1548.6	256743	120.8	-	120.7	-
5	0.76	-	126.9	5375	310.9	-	283.8	-	timeout	-	timeout	-	431.9	-	414.9	-
Density 4																
1	0.06	-	0.14	139	0.14	-	0.24	139	0.08	139	0.08	139	0.22	139	0.16	139
2	0.09	-	0.72	359	3.24	-	327.2	359	0.96	1052	0.86	1052	2043.0	1052	173.6	1052
3	0.20	-	6.00	1228	47.0	-	timeout	-	55.5	13911	39.4	13911	timeout	-	timeout	-
4	0.34	-	19.9	2080	98.1	-	95.9	-	1711.9	191594	1119.9	191594	121.0	-	120.3	-
5	0.76	-	313.4	11352	308.2	-	284.6	-	timeout	-	timeout	-	432.6	-	413.0	-

Table 4. OPTIC planner performance in all models and scenarios: runtime (s), number of states evaluated and the number of Bingo game participants (part.) for the first and the last solutions found by the planner. The '*' indicates that the planner stopped at the specified time and '-' that no solution was found.

Scen-arios	***RP***						***RN***						***NP***						***NN***					
	first			last			first			last			first			last			first			last		
	runtime (s)	states	part.	runtime (s)	states	part.	runtime (s)	states	part.	runtime (s)	states	part.	runtime (s)	states	part.	runtime (s)	states	part.	runtime (s)	states	part.	runtime (s)	states	part.
Density 0																								
1	0.06	11	0	389.1	67419	3	0.06	11	0	730.6	92553	3	0.18	11	0	3197.0	94794	4	0.14	11	0	2179.6	188659	4
2	0.34	40	0	821.5	89927	0	0.32	40	0	745.2	89927	0	273.5	40	0	1772.5	6585	0	127.8	40	0	1612.9	6585	0
3	27.9	2268	0	260.0	14516	0	22.6	2268	0	224.8	14516	0	timeout	-	-	-	-	-	timeout	-	-	-	-	-
4	48.5	2508	0	1271.0	41283	0	35.2	2508	0	1062.0	41283	0	141.2*	-	-	-	-	-	140.1*	-	-	-	-	-
5	345.1	7692	0	2561.5*	-	-	264.7	7692	0	2987.5*	-	-	485.3*	-	-	-	-	-	458.5*	-	-	-	-	-
Density 1																								
1	0.06	8	0	2002.2	90623	3	0.06	8	0	1775.7	183257	3	0.20	8	0	2936.0*	-	-	0.14	8	0	1478.0	167930	4
2	0.24	26	0	190.0	22792	0	0.24	26	0	179.3	22792	0	254.3	26	0	2761.1	11713	0	128.5	26	0	2505.7	11713	0
3	26.7	2115	0	1163.6	59770	0	21.8	2115	0	1040.1	59770	0	timeout	-	-	-	-	-	timeout	-	-	-	-	-
4	34.4	1882	0	2027.2*	-	-	24.5	1882	0	2358.0*	-	-	140.5*	-	-	-	-	-	140.1*	-	-	-	-	-
5	158.8	4062	0	2593.4*	-	-	118.2	4062	0	2995.6*	-	-	487.3*	-	-	-	-	-	463.2*	-	-	-	-	-
Density 2																								
1	0.06	8	0	timeout	-	-	0.06	8	0	1269.6	154476	3	0.18	8	0	3529.2*	-	-	0.14	8	0	1250.8	139669	4
2	0.24	26	0	1209.9	151592	0	0.22	26	0	1115.7	151592	0	256.1	26	0	2879.9	11713	0	136.8	26	0	2657.7	11713	0
3	26.2	2129	0	1604.2	86052	0	21.3	2129	0	1403.1	86052	0	timeout	-	-	-	-	-	timeout	-	-	-	-	-
4	58.7	2797	0	1981.9*	-	-	44.0	2797	0	2287.8*	-	-	141.8*	-	-	-	-	-	139.7*	-	-	-	-	-
5	316.6	6977	0	2587.9*	-	-	234.8	6977	0	2955.9*	-	-	487.4*	-	-	-	-	-	464.2*	-	-	-	-	-
Density 3																								
1	0.06	8	0	timeout	-	-	0.06	8	0	1945.2	202444	4	0.20	8	0	timeout	-	-	0.14	8	0	2764.9	281129	4
2	0.26	26	0	1272.9	151592	0	0.22	26	0	1133.7	151592	0	354.0	26	0	2993.6	11713	0	207.3	26	0	2811.5	11713	0
3	26.8	2075	0	965.9	50272	0	20.7	2075	0	1609.5	100868	0	timeout	-	-	-	-	-	timeout	-	-	-	-	-
4	63.5	2959	0	1990.5*	-	-	48.5	2959	0	2292.8*	-	-	140.7*	-	-	-	-	-	140.3*	-	-	-	-	-
5	331.7	7338	0	2541.6*	-	-	253.8	7338	0	2967.7*	-	-	485.3*	-	-	-	-	-	468.1*	-	-	-	-	-
Density 4																								
1	0.06	8	0	timeout	-	-	0.06	8	0	1974.3	222945	3	0.22	8	0	1037.5	90968	3	0.12	8	0	1315.0	172596	4
2	0.24	26	0	1115.1	143131	0	0.22	26	0	1031.5	143131	0	807.0	26	0	2707.3	11360	0	581.6	26	0	timeout	-	-
3	24.6	2075	0	898.7	50272	0	20.1	2075	0	1559.2	100868	0	timeout	-	-	-	-	-	timeout	-	-	-	-	-
4	66.4	3047	0	1537.6	50611	0	51.9	3047	0	1311.3	50611	0	141.2*	-	-	-	-	-	139.8*	-	-	-	-	-
5	461.8	9850	0	1776.2	32746	0	346.8	9850	0	1377.4	32746	0	486.0*	-	-	-	-	-	461.5*	-	-	-	-	-

Table 5. Runtime (s) and number of evaluated states for the planner OPTIC to find a solution with a Bingo game in the *Scenario 1*. The '-' indicates that the planner could not find a solution with a Bingo game.

Density	**Models**							
	RP		*RN*		*NP*		*NN*	
	runtime (s)	states	runtime (s)	states	runtime (s)	states	runtime (s)	states
0	389.1	67419	305.3	67547	1430.5	84353	338.9	56550
1	562.7	73856	271.7	67234	-	-	314.2	56316
2	-	-	269.7	66577	-	-	310.8	55659
3	-	-	263.5	64858	-	-	304.7	54286
4	-	-	296.2	71191	1037.5	90968	284.1	54212