Mixed-Integer and Constraint Programming Techniques for Mobile Robot Task Planning (Extended Abstract)*

Kyle E. C. Booth, Tony T. Tran, Goldie Nejat, and **J. Christopher Beck**
Department of Mechanical & Industrial Engineering
University of Toronto, Toronto, Ontario, Canada
{kbooth, tran, nejat, jcb}@mie.utoronto.ca

Mobile Robot Task Planning

Driven by the increased use of mobile robotics for everyday applications, there has been a flurry of research activity in the pursuit of computationally efficient techniques for autonomous decision making (Gerkey and Matarić 2004). The automated planning and scheduling of tasks is of particular interest to the artificial intelligence (AI) and robotics communities, and considered a core competency of intelligent behavior. As such, the development and integration of solution techniques for such reasoning is fundamental to the successful design of autonomous mobile robots (Ghallab, Nau, and Traverso 2004).

Automated task planning and scheduling has been previously studied in mobile robotics applications such as warehouse management (Kim et al. 2003), hospital assistance, and human care (Cesta et al. 2011). There are a variety of existing solution methods, including those using mathematical programming techniques (Coltin, Veloso, and Ventura 2011), customized interval-algebra algorithms (Mudrova and Hawes 2015), and forward-chaining temporal planners (Louie et al. 2014).

In this work we investigate the application of optimization-based scheduling technologies to such robot task planning problems. Namely, we develop and apply *mixed-integer programming* (MIP) and *constraint programming* (CP) methods to solve two mobile robot task planning problems from the literature. Furthermore, for the second robot task planning problem, we integrate our CP task planning approach on the mobile social robot, Tangy.

In the first problem, a robot plans a set of tasks each with different temporal constraints dictating when a task is available for execution and when task execution must be completed. For this particular problem, the task planner must determine a feasible plan that minimizes the sum of task completion times. In the second mobile robot task planning problem, a socially-interacting robot must generate feasible task plans while adhering to a number of restrictions, including temporal constraints, the timetables of human users, and robot energy levels. We model and solve each of these problems with MIP and CP to find high-quality task plans. For the second problem, we demonstrate the physicaly utility of our methods by integrating our CP approach into a real robot architecture. Eliminating the need for algorithmic development, our model-and-solve techniques exploit ongoing advances within MIP and CP and our experimental results illustrate the promising nature of these general approaches for mobile robot task planning problems.

Optimization Technologies

Combinatorial optimization problems have been historically approached with a wide-range of methods including MIP and CP. MIP is a mathematical programming approach that models problems with continuous or integer variables whose values are restricted by linear constraints and contribute towards a global linear objective function. The approach commonly employs *branch-and-bound* tree search (Land and Doig 1960) and often avoids worst-case exponential search by solving the associated *linear programming* (LP) *relaxation* at each node to attain a bound on the objective and systematically prune subtrees. More sophisticated algorithmic developments have been proposed over the years, resulting in significant machine-independent speedups from the early 1990s to 2012 (Bixby 2012).

Conversely, CP is a rich approach that eschews structural restrictions and is capable of modeling constraints and variables of a variety of forms. Developed primarily within the AI community, CP focuses on the notation of *global constraints* to encapsulate frequently recurring combinatorial substructure. Such global constraints are combined in CP modeling and search effort is reduced through logical *inference* (Jaffar and Maher 1994) where each constraint has an associated algorithm that performs *domain filtering*. Such filtering removes values from variable domains that cannot participate in global solutions, and is performed at each node within the search. CP has also seen significant improvement in recent decades and has established itself as a viable alternative to mathematical programming-based approaches.

*The work in this extended abstract is presented in detail in a 2016 IEEE Journal article (Booth et al. 2016). It accompanies an invited talk given at the *Workshop on Constraint Satisfaction Techniques for Planning and Scheduling* (COPLAS2016) as part of the *26th International Conference on Planning and Scheduling* (ICAPS2016) on June 14, 2016.

This research has been funded by the Natural Sciences and Engineering Council of Canada (NSERC), Dr. Robot Inc., and the Canada Research Chairs (CRC) Program.

Robot Task Planning Problems

We study two mobile robot task planning problems, each requiring the autonomous assignment of start times to a set of tasks while adhering to problem constraints.

Task Planning Problem #1 Given a set of n tasks, $j \in J$, each with a release time, r_j, deadline time, d_j, and processing time, p_j, the robot must find a feasible task plan, or determine that none exist, over a planning horizon, H. Using standard scheduling terminology, this problem can be represented as $1|r_j, d_j, \delta_{jk}|\sum_j C_j$, where 1 represents the single robot, δ_{jk} defines the robot travel time between tasks j and k, and $\sum_j C_j$ is the objective function which minimizes the sum of task completion times. Robot travel times are asymmetric such that $\delta_{jk} \neq \delta_{kj}$ may hold, and follow the triangle inequality, namely $\delta_{jl} + \delta_{lk} \geq \delta_{jk}$. A solution task plan is a set of start times for each task, $\{s_1, s_2, ..., s_n\}$, such that these times adhere to the temporal constraints of each task (i.e. $s_j \in [r_j, d_j - p_j], \forall j \in J$), travel times are satisfied, and the objective is minimized.

We propose both MIP and CP models for this problem. Our disjunctive MIP model is defined by Eqs. (1) through (6), and uses decision variable $x_{jk} :=\{1$ if task j precedes task k, and 0 otherwise$\}$. In this model, Eqn. (1) is the minimization objective function, (2) defines task completion time, Eqs. (3) and (4) ensure a disjunctive relationship between all pairs of tasks, such that they do not conflict temporally, and the remainder of the model identifies variable domains.

$$\min \quad \textstyle\sum_j C_j \tag{1}$$
$$\text{s.t.} \quad C_j = s_j + p_j, \quad \forall j \tag{2}$$
$$C_j + \delta_{jk} \leq s_k + (H + \delta_{jk})(1 - x_{jk}), \quad \forall j, k \tag{3}$$
$$C_k + \delta_{kj} \leq s_j + (H + \delta_{kj})(x_{jk}), \quad \forall j, k \tag{4}$$
$$x_{jk} \in \{0, 1\}, \quad \forall j, k \tag{5}$$
$$s_j \in [r_j, d_j - p_j] \quad \forall j \tag{6}$$

Our CP model is defined by Eqns. (7) through (10), making use of the $NoOverlap$ global constraint (Laborie 2009) in Eqn. (9) to prevent tasks from conflicting temporally including travel times, where Δ is the matrix of travel times between all pairs of tasks, δ_{jk}. Eqn. (7) defines the objective function, Eqn. (8) completion time, and the remainder identify varible domains.

$$\min \quad \textstyle\sum_j C_j \tag{7}$$
$$\text{s.t.} \quad C_j = s_j + p_j, \quad \forall j \tag{8}$$
$$NoOverlap(\{s_1, .., s_n\}, \{p_1, .., p_n\}, \Delta), \tag{9}$$
$$s_j \in [r_j, d_j - p_j] \quad \forall j \tag{10}$$

There have been previously proposed methods for solving this problem within the literature. Specifically, *dynamic user task scheduling* (DUTS) (Coltin, Veloso, and Ventura 2011) introduces a pre-processing step that determines pairs of tasks with overlapping time windows and adds constraints similar to Eqs. (3) and (4) to a mathematical model before assigning start times via a MIP solver. An alternative method uses *task scheduling with interval algebra* (TSIA) (Mudrova and Hawes 2015) to heuristically order all pairs of tasks before using also using MIP to solve the problem. We note that each of these proposed methods are incomplete and not guaranteed to find a feasible solution if such a task plan exists. For larger optimization problems, global optimality may be unachievable within reasonable time, and thus heuristic methods may be preferred. As such, within our experimental analysis, we evaluate the solution-quality vs. run-time tradeoff of the different methods.

Task Planning Problem #2 Given a single-day planning horizon from 8:00AM to 7:00PM, the social robot Tangy must plan and facilitate a set of activities (tasks) involving human users while reasoning about temporal constraints, user timetables, and robot energy levels (Louie et al. 2014). The activities consist of *bingo games* (involving multiple users), *bingo game reminders* (involving a single user), and *robot recharge* tasks. The participants, location, and processing time of each task are known a priori, and the problem requires the robot to autonomously determine task start times and, in the case of optional robot recharge tasks, task presence and duration.

Each user has a timetable dictating when he/she is available, including mandatory breaks for meals from 8:00-9:00AM, 12:00-1:00PM, and 5:00-6:00PM. The set of bingo games and participants are parameters to the problem, and the robot must perform a reminder task with each user prior to his/her game. Robot travel times between any two locations are known, and a feasible task plan must account for these required transitions. Instantaneous battery level for the robot is available and must stay within pre-specified bounds. Each task type has a unique energy consumption rate, and *optional* robot recharge tasks allow for energy replenishment; an upper bound of these is supplied to the model, and they do not need to be utilized.

We solve this problem using both MIP and CP technologies, making use of continuous, integer, and binary decision variables within MIP and *optional interval variables* (Laborie 2009) within CP to properly model task optionality, in addition to a number of global constraints. Due to space limitations, these models, as well as a more comprehensive problem description, are detailed elsewhere (Booth et al. 2016). Prior to this work we proposed an approach for solving this problem using a forward chaining temporal planner (Louie et al. 2014), and we compare the results of our proposed models to this temporal planner.

Implementation & Experimental Analysis

Due to the application-driven focus on quickly finding feasible, high-quality task plans, we define algorithm performance based on run-time and optimality gap (%). Our methods are implemented in C++ on a hexacore machine with a Xeon processor and 12GB of RAM running Linux Ubuntu 14.04. We use the IBM ILOG CPLEX V12.6.2 Optimization Studio, which includes both MIP and CP solvers.

Benchmark problem sets are generated as identified in the

journal version of this work (Booth et al. 2016), and the task plan solutions for the second problem are simulated using the Robot Operating System (ROS) (Quigley et al. 2009) on custom-developed visualization software. To validate the physical utility of our methods, the CP approach (best performing) is implemented within a ROS-based architecture on the mobile robot Tangy, using the GMapping technique in OpenSlam (openslam.org) to create an environment map via simultaneous localization and mapping.

Table 1 illustrates the MRE of the various approaches over time for Problem #1. These values are calculated according to the following expression: $MRE_{(CP,P40,0.1)} = \frac{1}{|\bar{P40}|} \sum_{p \in \bar{P40}} \frac{c(CP,p,0.1) - c^*(p)}{c^*(p)} \times 100$, which would yield the average MRE for the CP approach for all five problems with 40 tasks, $P40$, at a run-time duration of 0.1 seconds. In this expression $p \in \bar{P40}$ is the set of 40 task instances where feasible solutions were found at 0.1 seconds using CP. The value $c(CP, p, 0.1)$ is the best solution found by CP at this run-time for problem instance p, and $c^*(p)$ is the optimal solution, if known, or best known bound attained by running the MIP model for 18,000 seconds. If an approach failed to find any feasible plans at a specified run-time, a value of '-' is used. Values with a '†' indicate that MRE was calculated from the subset of instances for which the method found a feasible plan at the associated run-time. '# Inf.' identifies, for a technique, the number of instances for which no feasible plan was found after 100 seconds.

The proposed CP approach is able to find better solutions in shorter run-times than all other methods at nearly all time points, and our proposed MIP model generally outperforms existing MIP-based approaches. Furthermore, our methods do not sacrifice algorithmic completeness like the DUTS and TSIA methods, in part illustrated by the inability for the TSIA method to improve upon its initial heuristic solution and, in some cases, inability to find any feasible solutions.

Table 1: Problem #1: Mean relative error (%) over time

		Run-time (s)				
# Tasks	Technique	0.1	1	10	100	# Inf.
40	**CP**	**0.08**	**0.00**	**0.00**	**0.00**	0
	MIP	7.93	0.13	**0.00**	**0.00**	0
	DUTS	13.10	0.06	0.02	0.02	0
	TSIA	0.98	0.98	0.98	0.98	0
80	**CP**	**0.32**	**0.15**	**0.10**	**0.10**	0
	MIP	9.02	1.38	0.11	0.11	0
	DUTS	10.23	4.49	0.15	0.12	0
	TSIA	0.45†	0.45†	0.45†	0.45†	**2**
120	**CP**	**0.37**	**0.34**	**0.25**	**0.24**	0
	MIP	6.60†	3.67	**0.25**	0.25	0
	DUTS	7.06†	4.48	0.28	0.25	0
	TSIA	0.40†	0.40†	0.40†	0.40†	**4**
160	**CP**	**0.33**	**0.30**	**0.23**	**0.22**	0
	MIP	-	4.07	1.13	0.23	0
	DUTS	4.74†	3.08	0.85	0.23	0
	TSIA	0.33†	0.33†	0.33†	0.33†	**4**
200	**CP**	**0.26**	**0.25**	**0.20**	**0.18**	0
	MIP	-	3.56	1.63	**0.18**	0
	DUTS	4.77	3.83	1.93	**0.18**	0
	TSIA	-	-	-	-	**5**

Experimental results for the simulation of our proposed methods for the second problem are illustrated in Table 2. Again, CP is the dominant performing algorithm, finding feasible solutions much faster than the alternate methods. We note that though CP is by far the best approach for this problem, both the CP and MIP optimization-based technologies outperform the previously proposed forward-chaining temporal planning approach that uses OPTIC (Benton, Coles, and Coles 2012), even though the feasibility-focus of the problem favours the planning method over its optimization-based counterparts.

Table 2: Problem #2: Time to first feasible plan

Scenario		Technique		
Users	Bingo Games	**CP**	MIP	OPTIC
4	1	< **0.01**	0.01	0.54
8	2	< **0.01**	0.36	9.13
12	3	**0.04**	1.30	13.09
16	4	**0.01**	-	-
20	5	**0.08**	-	-

As a proof of concept, we implement our CP-approach in a real-world environment on the social robot, Tangy. We used the first scenario for this physical implementation, consisting of four users, one bingo game activity, and the associated reminder tasks. The results of this physical implementation are detailed within (Booth et al. 2016). This real-world experimentation is significant as it validates the physical utility of our task planning methods in realistic environments.

Conclusions & Future Work

We explored the modeling and solving of two robot task planning problems using optimization-based formalisms mixed-integer programming (MIP) and constraint programming (CP). The first problem involved the automated generation of feasible task plans that adhere to temporal constraints surrounding task release and deadline times. The second problem required reasoning about task precedence relationships, human user timetables, and robot energy consumption and replenishment. We implemented our models within simulated and real environments, comparing them with previous methods and concluding that, for the problems studied, the inference-based search of CP is the superior approach. Additionally, we implemented our CP approach for the second problem on the social robot Tangy to validate the physical utility of our methods.

Overall, our results indicate that these optimization-based techniques are promising for solving mobile robot task planning problems, and a main direction for our future research involves exploring the role of these methods for the development of re-planning and plan repair techniques. We also plan to further investigate robot task planning problems in order to understand the point at which such problems will require more sophisticated methods, including problem-based search manipulations and decompositions.

Acknowledgment

We would like to thank M. Schwenk for the design of the simulation environment and S. Mohamed for the robot navigation and mapping modules utilized in the experiments.

References

Benton, J.; Coles, A. J.; and Coles, A. 2012. Temporal planning with preferences and time-dependent continuous costs. In *ICAPS*, volume 77, 78.

Bixby, R. E. 2012. A brief history of linear and mixed-integer programming computation. *Documenta Mathematica, Extra Volume: Optimization Stories* 107–121.

Booth, K. E.; Tran, T. T.; Nejat, G.; and Beck, J. C. 2016. Mixed-integer and constraint programming techniques for mobile robot task planning. *Robotics and Automation Letters, IEEE* 1(1):500–507.

Cesta, A.; Cortellessa, G.; Rasconi, R.; Pecora, F.; Scopelliti, M.; and Tiberio, L. 2011. Monitoring elderly people with the robocare domestic environment: Interaction synthesis and user evaluation. *Computational Intelligence* 27(1):60–82.

Coltin, B.; Veloso, M. M.; and Ventura, R. 2011. Dynamic user task scheduling for mobile robots. In *Automated Action Planning for Autonomous Mobile Robots, AAAI Workshops*, volume WS-11-09.

Gerkey, B. P., and Matarić, M. J. 2004. A formal analysis and taxonomy of task allocation in multi-robot systems. *The International Journal of Robotics Research* 23(9):939–954.

Ghallab, M.; Nau, D.; and Traverso, P. 2004. *Automated planning: theory & practice*. Elsevier.

Jaffar, J., and Maher, M. J. 1994. Constraint logic programming: A survey. *The journal of logic programming* 19:503–581.

Kim, B.-I.; Heragu, S. S.; Graves, R. J.; and Onge, A. S. 2003. A hybrid scheduling and control system architecture for warehouse management. *Robotics and Automation, IEEE Transactions on* 19(6):991–1001.

Laborie, P. 2009. IBM ILOG CP Optimizer for detailed scheduling illustrated on three problems. In *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*. Springer. 148–162.

Land, A. H., and Doig, A. G. 1960. An automatic method of solving discrete programming problems. *Econometrica: Journal of the Econometric Society* 497–520.

Louie, W.-Y. G.; Vaquero, T.; Nejat, G.; and Beck, J. C. 2014. An autonomous assistive robot for planning, scheduling and facilitating multi-user activities. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, 5292–5298.

Mudrova, L., and Hawes, N. 2015. Task scheduling for mobile robots using interval algebra. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, 383–388.

Quigley, M.; Conley, K.; Gerkey, B.; Faust, J.; Foote, T.; Leibs, J.; Wheeler, R.; and Ng, A. Y. 2009. Ros: an open-source robot operating system. In *ICRA workshop on open source software*, volume 3, 5.