

Robust Metric Hybrid Planning in Stochastic Nonlinear Domains Using Mathematical Optimization

Buser Say

Monash University
Melbourne, Australia
buser.say@monash.edu

Abstract

The deployment of automated planning in safety critical systems has resulted in the need for the development of robust automated planners that can (i) accurately model complex systems under uncertainty, and (ii) provide formal guarantees on the model they act on. In this paper, we introduce a robust automated planner that can represent such stochastic systems with metric specifications and constrained continuous-time nonlinear dynamics over mixed (i.e., real and discrete valued) concurrent action spaces. The planner uses inverse transform sampling to model uncertainty, and has the capability of performing bi-objective optimization to first enforce the constraints of the problem as best as possible, and second optimize the metric of interest. Theoretically, we show that the planner terminates in finite time and provides formal guarantees on its solution. Experimentally, we demonstrate the capability of the planner to robustly control four complex physical systems under uncertainty.

Introduction

Automated planning formally reasons about the selection, timing and duration of actions to reach desired states of the world as best as possible by representing the dynamics of the world using a model (Nau, Ghallab, and Traverso 2004). It has significantly improved the ability of autonomous systems to solve challenging tasks such as, smart grid control (Thiébaux et al. 2013), traffic control (McCluskey and Vallati 2017), Heating, Ventilation and Air Conditioning (HVAC) control (Say et al. 2017), and Unmanned Aerial Vehicles (UAV) control (Ramirez et al. 2018). The deployment of automated planning in such safety critical systems resulted in the need for the development of **robust** automated planners that can (i) accurately model complex systems under uncertainty, and (ii) provide formal guarantees on the model they act on. In this paper, we will introduce an automated planner that can robustly plan in complex domains with constrained continuous-time nonlinear dynamics that require metric optimization over mixed (i.e., real and discrete valued) concurrent action spaces under uncertainty.

We will begin with the definition of the stochastic metric hybrid planning problem by building up on the existing

deterministic metric hybrid planning problem (Say and Sanner 2018, 2019) from the literature which readily formalizes all aspects of the complex planning problems we will consider in this paper except the uncertainty. Specifically, our definition will be in the form of a **bi-objective** optimization problem; which will allow us to first enforce the constraints of the problem *as best as possible*, and second optimize the metric of interest (i.e., total reward accumulated). This is in contrast to other important risk-sensitive automated decision making systems that either model the effects of constraint violations into the reward function and view the unconstrained optimization problem from a distributional perspective (Bellemare, Dabney, and Munos 2017; Dabney et al. 2018; Yang et al. 2019; Patton et al. 2022), or assume a priori knowledge on maximum allowed constraint violation and model the problem as a single objective constrained optimization problem (Li and Williams 2008; Guo and Song 2011; Trevizan, Thiébaux, and Haslum 2017; Yang et al. 2020; Liu et al. 2021; Simão, Jansen, and Spaan 2021). Notably, multi-objective optimization has been previously studied for robust automated decision making (Lacerda, Parker, and Hawes 2015; Geisser et al. 2022) under formalisms that are built up on finite set of atomic propositions (i.e., Linear Temporal Logic (Pnueli 1977) and STRIPS planning (Fikes and Nilsson 1971)). We remark that these important works are fundamentally restricted to model problems with discrete action spaces and therefore are not suitable to model planning problems with mixed concurrent action spaces that are the focus of this paper.

We will then proceed with the introduction of an effective methodology for solving the stochastic metric hybrid planning problem. Our automated planner will perform **anticipatory planning** (Mercier and Van Hentenryck 2008) to model uncertainty (i.e., the first robustness criteria). Specifically, it will use an effective determinization procedure (Raghavan et al. 2017) based on inverse transform sampling to randomly sample multiple futures, and solve them collectively as one bi-objective optimization problem. In order to solve the resulting bi-objective optimization problem with interval constraints (i.e., used to model the constrained continuous-time dynamics of the underlying planning problem), we will use constraint generation. Namely, we will build on an existing automated planner (Say and Sanner 2019) that uses constraint generation to heuristically solve

the deterministic version of our planning problem, and modify the planner such that it provides **formal guarantees** on the model it acts on (i.e., the second robustness criteria). Specifically, we will show that our modifications guarantee that the planner terminates in finite number of constraint generation iterations and provides guarantees on its solution. Moreover we will present the results of our detailed computational experiments to test the effectiveness of using the planner to robustly control four complex physical systems under uncertainty. Namely, we will model each continuous-time control problem as a stochastic metric hybrid planning problem using the solution equations of the underlying ordinary differential equations, and solve the resulting planning problem with the planner. We will conclude our paper with a discussion of our contributions in relation to the literature for the purpose of starting new areas for future work.

Background

We begin by presenting the definition of the deterministic metric hybrid planning problem Π , an effective methodology for solving Π and a determinization procedure that will allow us to sample stochastic planning problems as parameterized Π .

Deterministic Metric Hybrid Planning Problem

A *deterministic metric hybrid planning problem* (Say and Sanner 2019) is a tuple $\Pi = \langle S, A, \Delta, C, T, V, G, R, H \rangle$ where

- $S = \{s_1, \dots, s_n\}$ is the set of factored state variables with bounded domains D_{s_1}, \dots, D_{s_n} for positive integer $n \in \mathbb{Z}^+$,
- $A = \{a_1, \dots, a_m\}$ is the set of concurrent action variables with bounded domains D_{a_1}, \dots, D_{a_m} for positive integer $m \in \mathbb{Z}^+$,
- $\Delta \in [\epsilon, M]$ is the duration of a step for positive real numbers $\epsilon \in \mathbb{R}^+$ and $M \in \mathbb{R}^+$ where $\epsilon \leq M$,
- $C : \prod_{i=1}^n D_{s_i} \times \prod_{i=1}^m D_{a_i} \times [\epsilon, M] \rightarrow \mathbb{R}$ is the deterministic temporal constraint function that is used to define the constraint $C(s_1^t, \dots, s_n^t, a_1^t, \dots, a_m^t, \Delta^t) \leq 0$ for all steps $t \in \{1, \dots, H\}$,
- $T : \prod_{i=1}^n D_{s_i} \times \prod_{i=1}^m D_{a_i} \times [\epsilon, M] \rightarrow \prod_{i=1}^n D_{s_i}$ is the deterministic state transition function,
- V is a tuple of constants $\langle V_1, \dots, V_n \rangle \in \prod_{i=1}^n D_{s_i}$ denoting the initial values of all state variables,
- $G : \prod_{i=1}^n D_{s_i} \rightarrow \mathbb{R}$ is the deterministic goal state function that is used to define the constraint $G(s_1^{H+1}, \dots, s_n^{H+1}) \leq 0$,
- $R : \prod_{i=1}^n D_{s_i} \times \prod_{i=1}^m D_{a_i} \times [\epsilon, M] \rightarrow \mathbb{R}$ is the deterministic reward function, and
- $H \in \mathbb{Z}^+$ is the planning horizon.

A *solution* to Π is a tuple of values $\langle \bar{a}_1^t, \dots, \bar{a}_m^t \rangle \in \prod_{i=1}^m D_{a_i}$ for all action variables A and a value $\bar{\Delta}^t \in [\epsilon, M]$ for all steps $t \in \{1, \dots, H\}$ (and a tuple of values $\langle \bar{s}_1^t, \dots, \bar{s}_n^t \rangle \in \prod_{i=1}^n D_{s_i}$ for all state variables S and steps $t \in \{1, \dots, H+1\}$) if and only if the following conditions hold:

1. $V_i = \bar{s}_i^1$ for all $i \in \{1, \dots, n\}$,
2. $T(\bar{s}_1^t, \dots, \bar{s}_n^t, \bar{a}_1^t, \dots, \bar{a}_m^t, \bar{\Delta}^t) = \langle \bar{s}_1^{t+1}, \dots, \bar{s}_n^{t+1} \rangle$ for steps $t \in \{1, \dots, H\}$,
3. $G(\bar{s}_1^{H+1}, \dots, \bar{s}_n^{H+1}) \leq 0$, and
4. $C(\bar{s}_1^t, \dots, \bar{s}_n^t, \bar{a}_1^t, \dots, \bar{a}_m^t, \bar{\Delta}^t) \leq 0$ for steps $t \in \{1, \dots, H\}$ and for all values of $x^t \in [0, \bar{\Delta}^t]$.

Similarly, an *optimal solution* to Π is a solution that maximizes the reward function R over the planning horizon H such that:

$$\max_{\substack{a_1^1, \dots, a_m^H \\ \Delta^1, \dots, \Delta^H}} \sum_{t=1}^H R(s_1^t, \dots, s_n^t, a_1^t, \dots, a_m^t, \Delta^t)$$

Next, we will present an effective methodology for solving Π .

Deterministic Metric Hybrid Planning with Mathematical Optimization

SCIPPlan (Say and Sanner 2019) is an automated planner that is based on mathematical optimization for solving Π . The model that describes the optimal solution of Π is provided below.

$$\max_{\substack{a_1^1, \dots, a_m^H \\ \Delta^1, \dots, \Delta^H}} \sum_{t=1}^H R(s_1^t, \dots, s_n^t, a_1^t, \dots, a_m^t, \Delta^t) \quad (1)$$

$$V_i = s_i^1 \quad \forall i \in \{1, \dots, n\} \quad (2)$$

$$s_i^{t+1} = T_i(s_1^t, \dots, \Delta^t) \quad \forall i \in \{1, \dots, n\}, t \in \{1, \dots, H\} \quad (3)$$

$$C(s_1^t, \dots, c^t \Delta^t) \leq 0 \quad \forall c^t \in [0, 1], t \in \{1, \dots, H\} \quad (4)$$

$$G(s_1^{H+1}, \dots, s_n^{H+1}) \leq 0 \quad (5)$$

$$s_i^t \in D_{s_i} \quad \forall i \in \{1, \dots, n\}, t \in \{1, \dots, H+1\} \quad (6)$$

$$a_i^t \in D_{a_i} \quad \forall i \in \{1, \dots, m\}, t \in \{1, \dots, H\} \quad (7)$$

$$\epsilon \leq \Delta^t \leq M \quad \forall t \in \{1, \dots, H\} \quad (8)$$

SCIPPlan uses a constraint generation framework that is outlined in Algorithm 1 to solve this model, since solving a model with interval constraints such as constraint (4) is theoretically complex and computationally challenging in general (Floudas and Pardalos 2009). Before the constraint generation begins, SCIPPlan relaxes constraint (4) with one constraint for each end of the interval (i.e., for $c^t \in \{0, 1\}$). Given this setup, SCIPPlan iteratively performs the following operations. SCIPPlan solves the (nonlinear) model using a spatial branch-and-bound algorithm (i.e., line 2). SCIPPlan terminates if the model is infeasible at any iteration (i.e., lines 3-4). If an optimal solution to the model is found (i.e., line 5), SCIPPlan heuristically checks¹ whether this solution also satisfies condition 4 or not. SCIPPlan terminates with a

¹SCIPPlan heuristically checks condition 4 by simulating function C with ϵ increments of duration within the interval $(0, \bar{\Delta}^t)$ for values \bar{a}_i^t, \bar{s}_i^t and $\bar{\Delta}^t$ over steps $t \in \{1, \dots, H\}$. SCIPPlan then returns the coefficient c^t that correspond to the middle point of the first detected constraint violation.

solution if condition 4 is also satisfied (i.e., line 7). Otherwise, SCIPPlan generates a new constraint that violates condition 4 by the means of adding constraint (4) with a new value of $c^t \in (0, 1)$ to the model (i.e., in line 9). This procedure is repeated until either a solution is found or the model becomes infeasible.

Algorithm 1: SCIPPlan (Say and Sanner 2019)

```

1: while True do
2:   Solve the model
3:   if the model is infeasible then
4:     return infeasibility
5:   else
6:     if values  $\bar{a}_1^t, \dots, \bar{\Delta}^H$  also satisfy condition 4 then
7:       return  $\bar{a}_1^t, \dots, \bar{\Delta}^H$  as the solution for  $\Pi$ 
8:     else
9:       Generate constraint (4) with a new  $c^t \in (0, 1)$ 

```

While SCIPPlan has been shown to work well experimentally, it remained an open research question whether SCIPPlan (i) terminates in finite number of constraint generation iterations, and/or (ii) provides any guarantees upon its termination. In this paper, we will show that under the mild assumption that function C is Lipschitz continuous, a modification of SCIPPlan that verifies condition 4 by solving a decision problem terminates in finite number of iterations, and either proves the infeasibility of Π or guarantees that the returned solution is a solution to Π within some maximum tolerance for constraint violation.

Next, we will present a determinization procedure that leverages *inverse transform sampling* to randomly sample values of random variables where the distribution of each variable is known and belongs to the location-scale family.

Determinization via Inverse Transform Sampling

In this paper, we will solve stochastic metric hybrid planning problems that will have random expressions which are functions of decision variables s_i^t, a_i^t, Δ^t and random variables \mathcal{X}_i^t such as:

$$s_1^t + a_1^t \Delta^t + \Delta^t \mathcal{X}_1^t$$

where the random variable $\mathcal{X}_1^t \sim \mathcal{Z}_1$ is assumed to come from some known distribution \mathcal{Z}_1 which belongs to the location-scale family. The determinization procedure (Raghavan et al. 2017) uses inverse transform sampling to determinize such random expressions where the resulting deterministic expression remains a function of decision variables s_i^t, a_i^t and Δ^t .

Given a stochastic expression with continuous² random variables $\mathcal{X}_i^t \sim \mathcal{Z}_i$ and known distributions \mathcal{Z}_i with location μ_i and scale $\sigma_i > 0$ parameters, the determinization procedure first samples u_i from the uniform distribution $\mathcal{U}(0, 1)$. Given u_i , it encodes the sampled expression by symbolically substituting each occurrence of \mathcal{X}_i^t with $\mu_i + \sigma_i \mathbb{F}_{\mathcal{Z}_i}^{-1}(u_i)$ where $\mathbb{F}_{\mathcal{Z}_i}^{-1}$ denotes the quantile function of the standardized form of distribution \mathcal{Z}_i . For example, the normally dis-

²A similar determinization procedure exists for discrete random variables (Raghavan et al. 2017).

tributed random variable $\mathcal{X}_1^t \sim \mathcal{N}(\mu_1, \sigma_1^2)$ would be sampled as $\mu_1 + \sigma_1 \mathbb{F}_{\mathcal{Z}_1}^{-1}(0.1)$ for $u_1 = 0.1$.

In the next two sections, we will present the first two contributions of this paper, namely: (i) the formalization of the stochastic metric hybrid planning problem Π^+ , and (ii) an effective solution methodology for the newly defined Π^+ .

Stochastic Metric Hybrid Planning Problem

A *stochastic metric hybrid planning problem* is a tuple $\Pi^+ = \langle \mathcal{X}, S, A, \Delta, C^+, T^+, V, G^+, R^+, H \rangle$ where A and S denote the sets of concurrent action and factored random state variables, Δ denotes the duration of a step, V denotes the initial values of all state variables and H denotes the planning horizon. Moreover,

- $\mathcal{X} = \{\mathcal{X}_1, \dots, \mathcal{X}_k\}$ is the set of random variables with domains $D_{\mathcal{X}_1}, \dots, D_{\mathcal{X}_k}$ (i.e., sample spaces) for positive integer $k \in \mathbb{Z}^+$ where the distribution \mathcal{Z}_i of each random variable $\mathcal{X}_i \sim \mathcal{Z}_i$ is known, and each distribution \mathcal{Z}_i belongs to the location-scale family with known location μ and scale $\sigma > 0$ parameters,
- $C^+ : \prod_{i=1}^n D_{s_i} \times \prod_{i=1}^m D_{a_i} \times [\epsilon, M] \times \prod_{i=1}^k D_{\mathcal{X}_i} \rightarrow \mathbb{R}$ is the stochastic temporal constraint function,
- $T^+ : \prod_{i=1}^n D_{s_i} \times \prod_{i=1}^m D_{a_i} \times [\epsilon, M] \times \prod_{i=1}^k D_{\mathcal{X}_i} \rightarrow \prod_{i=1}^n D_{s_i}$ is the stochastic state transition function, and $T_{pdf}^+ : \prod_{i=1}^n D_{s_i} \times \prod_{i=1}^m D_{a_i} \times [\epsilon, M] \times \prod_{i=1}^k D_{\mathcal{X}_i} \rightarrow [0, \infty)$ is the corresponding probability density function of the state transitions,
- $G^+ : \prod_{i=1}^n D_{s_i} \times \prod_{i=1}^k D_{\mathcal{X}_i} \rightarrow \mathbb{R}$ is the stochastic goal state function, and
- $R^+ : \prod_{i=1}^n D_{s_i} \times \prod_{i=1}^m D_{a_i} \times [\epsilon, M] \times \prod_{i=1}^k D_{\mathcal{X}_i} \rightarrow \mathbb{R}$ is the stochastic reward function.

In this paper, we assume that the random variables \mathcal{X} are the only source of stochasticity of functions C^+, T^+, G^+ and R^+ , and $\mathcal{X}_i^1, \dots, \mathcal{X}_i^{H+1}$ are independent and identically distributed. This assumption will allow us to effectively sample each \mathcal{X}_i from their distribution \mathcal{Z}_i such that each sampled future will remain deterministic functions of s_i^t, a_i^t and Δ^t using the previously described determinization procedure.

A *solution* to Π^+ is a tuple of values $\langle \bar{a}_1^t, \dots, \bar{a}_m^t \rangle \in \prod_{i=1}^m D_{a_i}$ for all action variables A and for all steps $t \in \{1, \dots, H\}$ and value $\bar{\Delta}^t \in [\epsilon, M]$ for all steps $t \in \{1, \dots, H\}$ that maximizes the expected solution quality $\mathbb{E}_{s_i^1=V_i, s_i^{t+1} \sim T_{pdf}^+, \mathcal{X}_i^t \sim \mathcal{Z}_i} [v]$ in which the auxiliary binary variable $v \in \{0, 1\}$ denotes the plan success such that:

$$v = \begin{cases} 1, & \text{if } G^+(s_1^{H+1}, \dots, \mathcal{X}_k^{H+1}) \leq 0 \text{ and} \\ & C^+(s_1^t, \dots, x^t, \dots, \mathcal{X}_k^t) \leq 0 \\ & \text{for all } t \in \{1, \dots, H\}, x^t \in [0, \Delta^t] \\ 0, & \text{otherwise} \end{cases}$$

Similarly, an *optimal solution* to Π^+ is a solution that maximizes the expected bi-objective solution quality such that:

$$\max_{\substack{a_1^1, \dots, a_m^H \\ \Delta^1, \dots, \Delta^H}} \mathbb{E}_{\substack{s_i^1=V_i \\ s_i^{t+1} \sim T_{pdf}^+ \\ \mathcal{X}_i^t \sim \mathcal{Z}_i}} \left[\left(v, v \sum_{t=1}^H R^+(s_1^t, \dots, \mathcal{X}_k^t) \right) \right]$$

The expected bi-objective solution quality consists of (i) the expected plan success and (ii) the expected total reward that is accumulated over the planning horizon H where the expected plan success optimization is prioritized over the expected total reward optimization in risk-sensitive settings that require *robust* decision making. Next, we will present an effective methodology for solving Π^+ .

SCIPPlan⁺: A Robust Stochastic Metric Hybrid Planner

In this section, we introduce an effective methodology for solving Π^+ , and refer to the resulting automated planner as SCIPPlan⁺. SCIPPlan⁺ will leverage the previously described determinization procedure to randomly sample $F \in \mathbb{Z}^+$ futures, and solve them collectively as one mathematical optimization problem. Given each future $f \in \{1, \dots, F\}$ will represent a version of Π that is parameterized by a set of sampled random variables, we proceed with the definition of the parameterized deterministic metric hybrid planning problem Π^p below.

Parameterized Deterministic Metric Hybrid Planning Problem

Given the set of samples $\bar{\mathcal{X}}^f$ of random variables \mathcal{X} for future $f \in \{1, \dots, F\}$, a *parameterized deterministic metric hybrid planning problem* is a tuple $\Pi^p(\bar{\mathcal{X}}^f) = \langle \bar{\mathcal{X}}^f, S^f, A^f, \Delta^f, C^p, T^p, V, G^p, R^p, H \rangle$ where A^f and S^f denote the sets of concurrent action and factored state variables, Δ^f denotes the duration of a step, V denotes the initial values of all state variables and H denotes the planning horizon. Moreover,

- $C^p : \prod_{i=1}^n D_{s_i} \times \prod_{i=1}^m D_{a_i} \times [\epsilon, M] \times \prod_{i=1}^k D_{x_i} \rightarrow \mathbb{R}$ is the parameterized deterministic temporal constraint function that is used to define the constraint $C^p(s_1^{t,f}, \dots, \bar{\mathcal{X}}_k^{t,f}) \leq 0$ for all steps $t \in \{1, \dots, H\}$,
- $T^p : \prod_{i=1}^n D_{s_i} \times \prod_{i=1}^m D_{a_i} \times [\epsilon, M] \times \prod_{i=1}^k D_{x_i} \rightarrow \prod_{i=1}^n D_{s_i}$ is the parameterized deterministic state transition function,
- $G^p : \prod_{i=1}^n D_{s_i} \times \prod_{i=1}^k D_{x_i} \rightarrow \mathbb{R}$ is the parameterized deterministic goal state function that is used to define the constraint $G^p(s_1^{H+1}, \dots, \bar{\mathcal{X}}_k^{H+1,f}) \leq 0$,
- $R^p : \prod_{i=1}^n D_{s_i} \times \prod_{i=1}^m D_{a_i} \times [\epsilon, M] \times \prod_{i=1}^k D_{x_i} \rightarrow \mathbb{R}$ is the deterministic reward function,

The deterministic functions C^p , T^p , G^p and R^p are obtained from the stochastic functions C^+ , T^+ , G^+ and R^+ respectively, following the previously described determinization procedure. The samples $\bar{\mathcal{X}}_1^1, \dots, \bar{\mathcal{X}}_k^{H+1}$ used in this determinization procedure are drawn via inverse transform sampling, and are assumed to be independent and identically distributed. As a result for example, the previously presented stochastic expression can be sampled for future f as:

$$s_1^{t,f} + a_1^{t,f} \Delta^{t,f} + \Delta^{t,f} \left(\mu + \sigma \mathbb{F}_Z^{-1}(u_1^{t,f}) \right)$$

for decision variables $s_1^{t,f}$, $a_1^{t,f}$ and $\Delta^{t,f}$, and constants $u_1^{t,f} \sim \mathcal{U}(0, 1)$, μ and $\sigma^2 > 0$. Given the definition of Π^p ,

we will define two solutions over all parameterized problems $\Pi^p(\bar{\mathcal{X}}^1), \dots, \Pi^p(\bar{\mathcal{X}}^F)$ based on the decision making setting we are in, namely: online or offline planning. In both of these definitions, we will use the binary variable $v^f \in \{0, 1\}$ to denote whether the values of state variables, action variables and durations satisfy the specifications of Π^p for future f (i.e., $v^f = 1$) or not (i.e., $v^f = 0$).

Fully Committed Plan In an offline planning setting (e.g., when we need to commit to the values of action variables and durations for all steps $t \in \{1, \dots, H\}$), we will define the solution to the parameterized problems $\Pi^p(\bar{\mathcal{X}}^1), \dots, \Pi^p(\bar{\mathcal{X}}^F)$ as a tuple of values $\langle \bar{a}_1^{t,f}, \dots, \bar{a}_m^{t,f} \rangle \in \prod_{i=1}^m D_{a_i}$ for all action variables A and a value $\bar{\Delta}^{t,f} \in [\epsilon, M]$, for all steps $t \in \{1, \dots, H\}$ and futures $f \in \{1, \dots, F\}$ if and only if the following conditions hold for all pairs of futures $f, f' \in \{1, \dots, F\}$:

1. $\bar{v}^f = 1 \rightarrow G^p(\bar{s}_1^{H+1,f}, \dots, \bar{\mathcal{X}}_k^{H+1,f}) \leq 0$,
2. $\bar{v}^f = 1 \rightarrow C^p(\bar{s}_1^{t,f}, \dots, x^{t,f}, \dots, \bar{\mathcal{X}}_k^{t,f}) \leq 0$ for steps $t \in \{1, \dots, H\}$ and for all values of $x^{t,f} \in [0, \bar{\Delta}^{t,f}]$,
3. $(\bar{v}^f = 1 \wedge \bar{v}^{f'} = 1) \rightarrow (\bar{a}_i^{t,f} = \bar{a}_i^{t,f'} \wedge \bar{\Delta}^{t,f} = \bar{\Delta}^{t,f'})$ for all $i \in \{1, \dots, m\}$ and steps $t \in \{1, \dots, H\}$,

and the solution quality $\frac{1}{F} \sum_{f=1}^F v^f$ is maximized. Similarly, an optimal solution to $\Pi^p(\bar{\mathcal{X}}^1), \dots, \Pi^p(\bar{\mathcal{X}}^F)$ is a solution that maximizes the bi-objective solution quality

$$\max_{\substack{a_1^{1,1}, \dots, a_m^{H,F} \\ \Delta^{1,1}, \dots, \Delta^{H,F} \\ v^1, \dots, v^F}} \left(\frac{1}{F} \sum_{f=1}^F v^f, \frac{1}{F} \sum_{f=1}^F v^f \sum_{t=1}^H R(s_1^{t,f}, \dots, \bar{\mathcal{X}}_k^{t,f}) \right)$$

such that the total plan success optimization (i.e., the first objective) is prioritized over the total reward optimization (i.e., the second objective).

In the definition above, the indicator constraints (i.e., denoted by the symbol \rightarrow) enforce that conditions 1-3 are satisfied for future f when $v^f = 1$. The bi-objective solution quality first aims to maximize the total number of futures for which the plan satisfies conditions 1-3, and second aims to maximize the total reward accumulated over such plans. The solution of the parameterized planning problems $\Pi^p(\bar{\mathcal{X}}^1), \dots, \Pi^p(\bar{\mathcal{X}}^F)$ over all sampled futures approximates the solution of Π^+ as F approaches infinity (i.e., the overestimation is obtained from first swapping expectation and maximization, and then taking the limit of F to infinity) (Mercier and Van Hentenryck 2008; Yoon et al. 2008).

Partially Committed Plan In an online planning setting³ (e.g., when we only need to commit to the values of action variables and duration at step $t = 1$), we will define the (optimal) solution by modifying condition 3 of fully committed plan to: $(\bar{v}^f = 1 \wedge \bar{v}^{f'} = 1) \rightarrow (\bar{a}_i^{t,f} = \bar{a}_i^{t,f'} \wedge \bar{\Delta}^{t,f} = \bar{\Delta}^{t,f'})$ for all $i \in \{1, \dots, m\}$, for all pairs of futures $f, f' \in$

³The duration of the first step $\Delta^{1,f}$ can be constrained to accommodate for the computational time required for online planning.

$\{1, \dots, F\}$ and only for step $t = 1$. Since this definition only relaxes condition 3 of the previous definition, a partially committed plan provides an upper bound on the solution quality of a fully committed plan.

Bi-Objective Mathematical Optimization for Stochastic Metric Hybrid Planning

The mathematical optimization model that SCIPPlan⁺ uses to solve Π^p over F futures is presented below.

$$\max_{\substack{a_1^{1,1}, \dots, a_m^{H,F} \\ \Delta^{1,1}, \dots, \Delta^{H,F} \\ v^1, \dots, v^F}} \frac{1}{F} \sum_{f=1}^F v^f \sum_{t=1}^H R(s_1^{t,f}, \dots, \bar{\mathcal{X}}_k^{t,f}) \quad (9)$$

$$V_i = s_i^{1,f} \quad \forall i \in \{1, \dots, n\}, f \in \{1, \dots, F\} \quad (10)$$

$$T_i^p(s_1^{t,f}, \dots, \bar{\mathcal{X}}_k^{t,f}) = s_i^{t+1,f} \quad \forall i \in \{1, \dots, n\}, t \in \{1, \dots, H\}, f \in \{1, \dots, F\} \quad (11)$$

$$v^f = 1 \rightarrow C^p(s_1^{t,f}, \dots, c^{t,f} \Delta^{t,f}, \dots, \bar{\mathcal{X}}_k^{t,f}) \leq 0 \quad \forall c^{t,f} \in [0, 1], t \in \{1, \dots, H\}, f \in \{1, \dots, F\} \quad (12)$$

$$v^f = 1 \rightarrow G^p(s_1^{H+1,f}, \dots, \bar{\mathcal{X}}_k^{H+1,f}) \leq 0 \quad \forall f \in \{1, \dots, F\} \quad (13)$$

$$s_i^{t,f} \in D_{s_i} \quad \forall i \in \{1, \dots, n\}, t \in \{1, \dots, H+1\}, f \in \{1, \dots, F\} \quad (14)$$

$$a_i^{t,f} \in D_{a_i} \quad \forall i \in \{1, \dots, m\}, t \in \{1, \dots, H\}, f \in \{1, \dots, F\} \quad (15)$$

$$\epsilon \leq \Delta^{t,f} \leq M \quad \forall t \in \{1, \dots, H\}, f \in \{1, \dots, F\} \quad (16)$$

$$\sum_{f=1}^F v^f = F' \quad (17)$$

$$(v^f = 1 \wedge v^{f'} = 1) \rightarrow a_i^{t,f} = a_i^{t,f'} \quad \forall i \in \{1, \dots, m\}, t \in \{1, \dots, H'\}, f, f' \in \{1, \dots, F\} \quad (18)$$

$$(v^f = 1 \wedge v^{f'} = 1) \rightarrow \Delta^{t,f} = \Delta^{t,f'} \quad \forall t \in \{1, \dots, H'\}, f, f' \in \{1, \dots, F\} \quad (19)$$

where the hyperparameter $F' \in \{1, \dots, F\}$ denotes the total number of plans that must satisfy conditions 1-3, and the hyperparameter H' is set equal to the planning horizon H for the computation of a fully committed plan or set to 1 for the computation of a partially committed plan. Given this mathematical optimization model, next we present the algorithmic description of SCIPPlan⁺ (i.e., Algorithm 2).

Algorithm 2: SCIPPlan⁺

- 1: Planning Setting: $H' = H$ for FC plan or $H' = 1$ for PC plan
 - 2: Number of sampled futures: $F' = F$
 - 3: **while** $F' > 0$ **do**
 - 4: Solve the model with H' and F' .
 - 5: **if** A solution is found **then**
 - 6: **return** the solution.
 - 7: $F' \leftarrow F' - 1$
 - 8: **return** infeasibility.
-

For a given planning setting (i.e., either FC plan or PC plan is set in line 1), SCIPPlan⁺ solves Π^p over F futures by

solving the mathematical optimization model initially with the value $F' = F$ (i.e., lines 2-4). If the model is feasible at any iteration (i.e., line 5), SCIPPlan⁺ terminates with a solution (i.e., line 6). If the model is infeasible, F' is decremented by 1 (i.e., line 7) and the model is solved again (i.e., line 4). This procedure is repeated (i.e., line 3) until either the model is feasible and SCIPPlan⁺ returns a solution, or the model is infeasible when F' is equal to 1 and SCIPPlan⁺ terminates (i.e., line 8).

Next, we will present the remaining two contributions of this paper, namely: (iii) the theoretical results on the finiteness and correctness of both SCIPPlan and SCIPPlan⁺, and (iv) the experimental results on the computational performance of SCIPPlan⁺.

Theoretical Results

In this section, we present our theoretical results on the finiteness and correctness of a modification of SCIPPlan for solving Π , and consequently the finiteness and correctness of SCIPPlan⁺ for solving Π^p . Namely, SCIPPlan can be modified to verify condition 4 given \bar{a}_i^t, \bar{s}_i^t and $\bar{\Delta}^t$ by solving the following decision problem (i.e., line 6 of Algorithm 1):

$$C(\bar{s}_1^t, \dots, \bar{s}_n^t, \bar{a}_1^t, \dots, \bar{a}_m^t, c^t \bar{\Delta}^t) \geq \gamma$$

for decision variable $c^t \in (0, 1)$ within some maximum tolerance for constraint violation $\gamma > 0$. This modification allows SCIPPlan to either verify the values \bar{a}_i^t, \bar{s}_i^t and $\bar{\Delta}^t$ to be a solution to Π within γ tolerance, or generate constraint (4) with a correct value of coefficient $c^t \in (0, 1)$ in any iteration of Algorithm 1. Now we can show that SCIPPlan correctly terminates in finite number of iterations if function C is assumed to be Lipschitz continuous.

Theorem 1 (Finiteness of SCIPPlan). *SCIPPlan terminates in finite number of constraint generation iterations if function C is Lipschitz continuous.*

Proof. The definition of Lipschitz continuity of function C allows us to write the following inequality:

$$|C(s_1^t, \dots, x_1^t) - C(s_1^t, \dots, x_2^t)| \leq \kappa |x_1^t - x_2^t|$$

for any two durations $x_1^t, x_2^t \in [0, \Delta^t]$, constant κ and for all steps $t \in \{1, \dots, H\}$. Assume SCIPPlan does not terminate which means Algorithm 1 generates infinite number of constraints (i.e., line 9 of Algorithm 1 is executed infinite number of times). For any iteration of Algorithm 1, we can let $c_1^t \in (0, 1)$ denote the newly generated coefficient of constraint (4), and $c_2^t \in [0, 1]$ denote either a previously generated coefficient (i.e., $c_2^t \in (0, 1)$) or one of the initial coefficients (i.e., $c_2^t \in \{0, 1\}$) of constraint (4). Given this setup, we have:

$$|C(s_1^t, \dots, c_1^t \Delta^t) - C(s_1^t, \dots, c_2^t \Delta^t)| \leq \kappa |c_1^t \Delta^t - c_2^t \Delta^t|$$

Let constant $\gamma > 0$ denote the maximum tolerance for constraint violation. Since constraint (4) is violated at duration $c_1^t \Delta^t$ and not violated at duration $c_2^t \Delta^t$, we also have:

$$|\gamma - 0| \leq |C(s_1^t, \dots, c_1^t \Delta^t) - C(s_1^t, \dots, c_2^t \Delta^t)|$$

From combining these two inequalities, restricting $\kappa > 0$ and rearranging the terms, we can obtain the following constant bound for the separation of coefficients c_1^t and c_2^t :

$$\frac{\gamma}{\kappa M} \leq \frac{\gamma}{\kappa \Delta^t} \leq |c_1^t - c_2^t|$$

which concludes our proof since every newly generated $c_1^t \in (0, 1)$ must be separated from the existing $c_2^t \in [0, 1]$ minimally by the constant $\frac{\gamma}{\kappa M}$, and therefore SCIPPlan must be bounded by $\frac{\kappa M}{\gamma} H$ constraint generation iterations. \square

Lemma 1 (Correctness of SCIPPlan). *SCIPPlan either returns a solution to Π within $\gamma > 0$ tolerance or proves the infeasibility of Π .*

Proof. Case 1: Assume SCIPPlan terminates with the values \bar{a}_i^t , \bar{s}_i^t and $\bar{\Delta}^t$ which violate either conditions 1-3 or condition 4 by more than γ . We know that conditions 1-3 must be feasible for \bar{a}_i^t , \bar{s}_i^t and $\bar{\Delta}^t$ since the model is solved by a complete and sound algorithm. It also means there exists a coefficient $c^t \in (0, 1)$ that violates condition 4 by more than γ . This creates a contradiction since the problem of finding the value of constant c^t is solved exactly and therefore SCIPPlan would not have terminated. Case 2: Assume SCIPPlan terminates with infeasibility and Π is feasible. This means that the underlying mathematical optimization model is infeasible which is a contradiction since the model is solved by a complete and sound algorithm. \square

Lemma 2 (Finiteness of SCIPPlan⁺). *SCIPPlan⁺ terminates in finite number of constraint generation iterations if function C is Lipschitz continuous.*

Proof. SCIPPlan⁺ is bounded by $\frac{\kappa M}{\gamma} H F^2$ constraint generation iterations since each iteration of Algorithm 2 can generate at most $\frac{\kappa M}{\gamma} H F$ constraints over F futures (i.e., from Theorem 1) and the value of F' can be decremented by 1 at most F times. \square

Lemma 3 (Correctness of SCIPPlan⁺). *SCIPPlan⁺ either returns a solution to Π^P within $\gamma > 0$ tolerance or proves the infeasibility of Π^P (i.e., $\sum_{f=1}^F v^f = 0$).*

Proof. Every iteration of Algorithm 2 is correctly solved within γ tolerance (i.e., from Lemma 1), and Algorithm 2 maximizes the bi-objective solution quality given F' is initially set to F , and is decremented by 1 if and only if the model is infeasible in each iteration of Algorithm 2. \square

Experimental Results

In this section, we present the results of our detailed computational experiments for testing the effectiveness of using SCIPPlan⁺ to approximately solve Π^+ by computing both fully committed (FC) plans and partially committed (PC) plans on Π^P over F futures.

Domain Descriptions

Four challenging domains are selected from the literature to be used in our experiments, namely: two versions of Navigation (Faulwasser and Findeisen 2009), Reservoir Control (Yeh 1985) and Heating, Ventilation and Air Conditioning (HVAC) (Agarwal et al. 2010). Each domain is formalized as Π^+ using the solution equations of the underlying physics ordinary differential equations. We provide a brief summary of the domains in Table 1.

Domain	n, m	Brief Description
Navigation 1	2,2	Continuous control of an agent in a two dimensional maze with obstacle(s). The movement of the agent is based on the solution of the first-order ordinary differential equation: $\frac{ds_i^t}{dx^t} = a_i^t$ T and C are stochastic functions, and A and S have real domains.
Navigation 2	4,2	Similar to Navigation 1 except the movement of the agent is based on the solution of the second-order ordinary differential equation: $\frac{d^2 s_i^t}{dx^{t2}} = a_i^t$
Reservoir Control	5,5	Continuous control of n connected reservoirs that are about to overflow back to safe water levels. The water levels are based on the solution of the first-order ordinary differential equation: $\frac{ds_i^t}{dx^t} = \sum_{j \in J(i)} a_j^t - a_i^t$ where $J(i)$ is the set of reservoirs flowing into reservoir i . Reservoirs are required to maintain water within the safety limits at all times. T and C are stochastic functions, and A and S have real domains.
HVAC	6,5	Continuous HVAC control of a building with n rooms. Temperature of each room is based on the (approximate) solution of the ordinary differential equation: $\frac{ds_i^t}{dx^t} = c_1 \sum_{j \in J(i)} s_j^t + c_2 s_i^t + c_3 a_i^t$ where $J(i)$ is the set of rooms adjacent to room i , and c_i are constants. Rooms are required to maintain temperature within the desired limits at all times. T and C are stochastic functions, and A and S have real domains.

Table 1: Summary of the experimental domains where n and m are the number of state and action variables, respectively.

Experimental Setup

All experiments were run on the CPU of a MacBookPro with 2.8 GHz Intel Core i716GB memory, using a single thread with one hour total time limit per instance. We used SCIP (Vigerske and Gleixner 2018) as the spatial branch-and-bound solver.

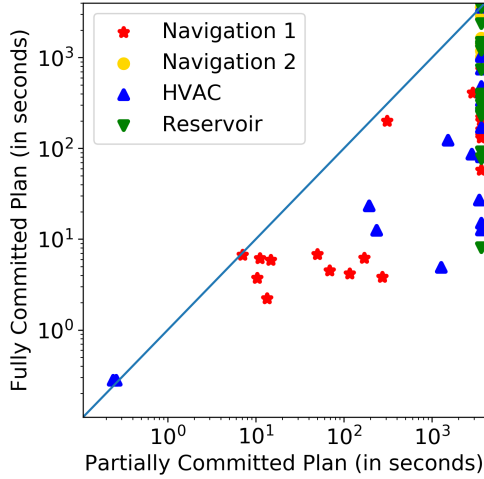


Figure 1: Visualization of runtime comparison between the computation of FC plans and PC plans using SCIPPlan⁺ over the four domains.

Implementation Details

In our experiments, we fixed the values of the sampled random variables between the computation of PC and FC plans per instance. In the first iteration of Algorithm 2, we have reduced all indicator constraints to their right hand side, removed constraint (17) and removed the expression $\sum_{f=1}^F v^f$ from the objective function. For the remaining iterations, we first heuristically computed feasible solutions by removing futures based on the order statistics of expressions $|u_i^f - 0.5|$.

Comparison of Runtime Performance

Figure 1 visualizes the comparison of the computational efforts spent for finding FC plans over PC plans in terms of logarithmic runtime where each data point represents an instance that is labelled uniquely by their domain. The inspection of Figure 1 clearly highlights the effectiveness of computing FC plans over PC plans. Overall we found that 79% of instances were solved within the one hour time limit by the computation of FC plans whereas this percentage was dropped to 25% for PC plans. Interestingly this wide gap in runtime performance resulted in more instances to be covered by FC plans with higher solution qualities than PC plans, since most PC plans found within the time limit were suboptimal. Next, we explore the effects of runtime performance on the computation of FC and PC plans in terms of coverage and solution quality.

Comparison of Coverage and Solution Quality

Table 2 summarizes the comparison between the computation of FC plans and PC plans in terms of coverage (i.e., measured by the total number of problem instances solved) and normalized solution quality (i.e., reported as a number between 0 and 1 measuring the optimization of the bi-objective solution quality together with the respective standard deviation) per domain within the time limit. The in-

spection of Table 2 highlights that FC plans cover 31% more instances compared to PC plans due to the relative runtime performance of their computation as previously shown in Figure 1. The inspection of solution qualities demonstrate the theoretical benefit of computing PC plans over FC plans in domains that can be solved within the time limit. Namely in Navigation 1, PC plans on average have 32% higher normalized solution quality compared to FC plans with lower standard deviations.

Domain	Coverage		Solution Quality	
	FC	PC	FC	PC
Navigation 1	20/20	13/20	0.50 ± 0.36	0.82 ± 0.15
Navigation 2	16/20	0/20	0.58 ± 0.47	-
HVAC	20/20	20/20	0.68 ± 0.26	0.64 ± 0.28
Reservoir	20/20	18/20	0.42 ± 0.25	0.46 ± 0.33

Table 2: Summary of problem coverage and solution quality comparison between the computation of FC plans and PC plans using SCIPPlan⁺ over the four domains.

Discussion and Related Work

In this section, we discuss the importance of our theoretical and computational results in relation to the literature for the purpose of opening new areas for future work.

We have shown important theoretical results on finiteness and correctness of SCIPPlan and SCIPPlan⁺. Our theoretical results sit at the core of robust decision making with SCIPPlan⁺ which provide theoretical guarantees on the performance of SCIPPlan⁺. We have made minimal assumptions (i.e., the Lipschitz continuity of function C) in achieving our theoretical results. This is in contrast to other related decision making systems that are built on top of additional restrictive assumptions (e.g., the assumption that the state transition function T is piecewise linear (Shin and Davis 2005), polynomial (Cashmore et al. 2016) etc.). Our results suggest similar constraint generation approaches can be utilized for effective planning under other planning formalisms that support continuous-time planning (Fox and Long 2003, 2006; Benton, Coles, and Coles 2012; Scala et al. 2016; Micheli and Scala 2019; Denenberg and Coles 2021; Percassi, Scala, and Vallati 2021). Moreover, our theoretical results provide us with some practical insights. For example, a direct implication of Lemma 1 is that the constant γ can be used to offset each generated constraint (4), meaning if SCIPPlan terminates with feasibility, it must satisfy conditions 1-4 exactly. We highlight this important feature of SCIPPlan⁺ especially for safety critical settings that do not allow γ tolerance for constraint violation.

We have shown the experimental performance of SCIPPlan⁺ to solve stochastic, continuous-time and constrained domains with concurrent action spaces. In particular, we found that computing FC plans was easier in comparison to PC plans even though a PC plan provides an upper bound on a FC plan if it can be computed optimally. An important area of future work here is to study how to improve the computational performance of SCIPPlan⁺ through (i) the coupling of SCIPPlan⁺ with gradient descent based planning (Wu, Say, and Sanner 2017, 2020; Patton et al.

2022; Jin et al. 2022) inside a unified framework for planning (Say 2021), and (ii) the exploitation of model symmetries across different steps and futures inside an effective constraint generation framework (Say et al. 2020).

Conclusion

In this paper, we have (i) formalized the stochastic metric hybrid planning problem Π^+ , (ii) presented an effective solution methodology (i.e., SCIPPlan⁺) for solving Π^+ using bi-objective mathematical optimization, and presented both (iii) theoretical results and (iv) experimental results on the performance of SCIPPlan⁺. Overall, we have introduced a robust automated planner for risk-sensitive decision making in stochastic, continuous-time and constrained domains with mixed concurrent action spaces.

References

- Agarwal, Y.; Balaji, B.; Gupta, R.; Lyles, J.; Wei, M.; and Weng, T. 2010. Occupancy-driven energy management for smart building automation. In *ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Building*, 1–6.
- Bellemare, M. G.; Dabney, W.; and Munos, R. 2017. A Distributional Perspective on Reinforcement Learning. In Precup, D.; and Teh, Y. W., eds., *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, 449–458. PMLR.
- Benton, J.; Coles, A.; and Coles, A. 2012. Temporal Planning with Preferences and Time-Dependent Continuous Costs. In *Proceedings of the Twenty-Second International Conference on Automated Planning and Scheduling*, volume 22, 2–10. AAAI Press.
- Cashmore, M.; Fox, M.; Long, D.; and Magazzeni, D. 2016. A Compilation of the Full PDDL+ Language into SMT. In *ICAPS*, 79–87.
- Dabney, W.; Rowland, M.; Bellemare, M.; and Munos, R. 2018. Distributional Reinforcement Learning With Quantile Regression. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*.
- Denenberg, E.; and Coles, A. 2021. Mixed Discrete Continuous Non-Linear Planning through Piecewise Linear Approximation. In *Proceedings of the Twenty-Ninth International Conference on Automated Planning and Scheduling*, 137–145. AAAI Press.
- Faulwasser, T.; and Findeisen, R. 2009. Nonlinear Model Predictive Path-Following Control. In *Nonlinear Model Predictive Control - Towards New Challenging Applications*, 335–343. Berlin, Heidelberg: Springer.
- Fikes, R. E.; and Nilsson, N. J. 1971. STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving. In *Proceedings of the Second International Joint Conference on Artificial Intelligence*, IJCAI, 608–620. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.
- Floudas, C. A.; and Pardalos, P. M., eds. 2009. *Encyclopedia of Optimization, Second Edition*. Springer.
- Fox, M.; and Long, D. 2003. PDDL2.1: An Extension to PDDL for Expressing Temporal Planning Domains. *Journal of Artificial Intelligence Research*, 20(1): 61–124.
- Fox, M.; and Long, D. 2006. Modelling Mixed Discrete-continuous Domains for Planning. *Journal of Artificial Intelligence Research*, 27(1): 235–297.
- Geisser, F.; Haslum, P.; Thiébaux, S.; and Trevizan, F. 2022. Admissible Heuristics for Multi-Objective Planning. In *Proceedings of Thirty-Second International Conference on Automated Planning and Scheduling*, ICAPS.
- Guo, X.; and Song, X. 2011. Discounted continuous-time constrained Markov decision processes in Polish spaces. In *The Annals of Applied Probability*, volume 21, 2016 – 2049. Institute of Mathematical Statistics.
- Jin, K.; Zhuo, H. H.; Xiao, Z.; Wan, H.; and Kambhampati, S. 2022. Gradient-based mixed planning with symbolic and numeric action parameters. *Artificial Intelligence*, 313: 103789.
- Lacerda, B.; Parker, D.; and Hawes, N. 2015. Optimal Policy Generation for Partially Satisfiable Co-Safe LTL Specifications. In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI, 1587–1593. AAAI Press.
- Li, H. X.; and Williams, B. C. 2008. Generative Planning for Hybrid Systems Based on Flow Tubes. In *Proceedings of the Eighteenth International Conference on International Conference on Automated Planning and Scheduling*, ICAPS, 206–213. AAAI Press.
- Liu, T.; Zhou, R.; Kalathil, D.; Kumar, P. R.; and Tian, C. 2021. Learning Policies with Zero or Bounded Constraint Violation for Constrained MDPs. In *Proceedings of the Thirty-Fifth International Conference on Neural Information Processing Systems*.
- McCluskey, T.; and Vallati, M. 2017. Embedding Automated Planning within Urban Traffic Management Operations. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 27, 391–399.
- Mercier, L.; and Van Hentenryck, P. 2008. Amsaa: A Multistep Anticipatory Algorithm for Online Stochastic Combinatorial Optimization. In Perron, L.; and Trick, M. A., eds., *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, 173–187. Berlin, Heidelberg: Springer Berlin Heidelberg.
- Micheli, A.; and Scala, E. 2019. Temporal Planning with Temporal Metric Trajectory Constraints. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence*, 7675–7682.
- Nau, D.; Ghallab, M.; and Traverso, P. 2004. *Automated Planning: Theory & Practice*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.
- Patton, N.; Jeong, J.; Gimelfarb, M.; and Sanner, S. 2022. A Distributional Framework for Risk-Sensitive End-to-End Planning in Continuous MDPs. In *Proceedings of the Thirty-Sixth AAAI Conference on Artificial Intelligence*.
- Percassi, F.; Scala, E.; and Vallati, M. 2021. Translations from Discretised PDDL+ to Numeric Planning. In *Proceedings of the Thirty-First International Conference on*

- Automated Planning and Scheduling*, volume 31, 252–261. AAAI Press.
- Pnueli, A. 1977. The temporal logic of programs. In *Eighteenth Annual Symposium on Foundations of Computer Science*, 46–57.
- Raghavan, A.; Sanner, S.; Tadepalli, P.; Fern, A.; and Khardon, R. 2017. Hindsight Optimization for Hybrid State and Action MDPs. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI. San Francisco, USA.
- Ramirez, M.; Papisimeon, M.; Lipovetzky, N.; Benke, L.; Miller, T.; Pearce, A. R.; Scala, E.; and Zamani, M. 2018. Integrated Hybrid Planning and Programmed Control for Real Time UAV Maneuvering. In *Proceedings of the 17th International Conference on Autonomous Agents and Multi-Agent Systems*, AAMAS, 1318–1326. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems.
- Say, B. 2021. A Unified Framework for Planning with Learned Neural Network Transition Models. In *Proceedings of the Thirty-Fifth AAAI Conference on Artificial Intelligence*, 5016–5024.
- Say, B.; Devriendt, J.; Nordström, J.; and Stuckey, P. 2020. Theoretical and Experimental Results for Planning with Learned Binarized Neural Network Transition Models. In Simonis, H., ed., *Proceedings of the Twenty-Sixth International Conference on Principles and Practice of Constraint Programming*, 917–934. Cham: Springer International Publishing.
- Say, B.; and Sanner, S. 2018. Metric Nonlinear Hybrid Planning with Constraint Generation. In *PlanSOpt 2018*, 19–25. 28th ICAPS Workshop on Planning, Search and Optimization (PlanSOpt).
- Say, B.; and Sanner, S. 2019. Metric Hybrid Factored Planning in Nonlinear Domains with Constraint Generation. In *Proceedings of the Sixteenth International Conference on the Integration of Constraint Programming, Artificial Intelligence, and Operations Research*, CPAIOR, 502–518.
- Say, B.; Wu, G.; Zhou, Y. Q.; and Sanner, S. 2017. Nonlinear Hybrid Planning with Deep Net Learned Transition Models and Mixed-integer Linear Programming. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, IJCAI, 750–756.
- Scala, E.; Haslum, P.; Thiébaux, S.; and Ramírez, M. 2016. Interval-Based Relaxation for General Numeric Planning. In *ECAI*, 655–663.
- Shin, J.-A.; and Davis, E. 2005. Processes and Continuous Change in a SAT-based Planner. *Artificial Intelligence*, 166(1-2): 194–253.
- Simão, T. D.; Jansen, N.; and Spaan, M. T. J. 2021. AlwaysSafe: Reinforcement Learning without Safety Constraint Violations during Training. In *Proceedings of the Twentieth International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS, 1226–1235. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems.
- Thiébaux, S.; Coffrin, C.; Hijazi, H.; and Slaney, J. 2013. Planning with MIP for Supply Restoration in Power Distribution Systems. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, 2900–2907.
- Trevizan, F.; Thiébaux, S.; and Haslum, P. 2017. Occupation Measure Heuristics for Probabilistic Planning. In *Proceedings of Twenty-Seventh International Conference on Automated Planning and Scheduling*, ICAPS.
- Vigerske, S.; and Gleixner, A. 2018. SCIP: Global Optimization of Mixed-Integer Nonlinear Programs in a Branch-and-cut Framework. *Optimization Methods and Software*, 33(3): 563–593.
- Wu, G.; Say, B.; and Sanner, S. 2017. Scalable Planning with Tensorflow for Hybrid Nonlinear Domains. In *Proceedings of the Thirty-First International Conference on Neural Information Processing Systems*, NIPS, 6276–6286. USA: Curran Associates Inc.
- Wu, G.; Say, B.; and Sanner, S. 2020. Scalable Planning with Deep Neural Network Learned Transition Models. *Journal of Artificial Intelligence Research*, 68: 571–606.
- Yang, D.; Zhao, L.; Lin, Z.; Qin, T.; Bian, J.; and Liu, T. 2019. Fully Parameterized Quantile Function for Distributional Reinforcement Learning. In *Proceedings of the Thirty-Second International Conference on Neural Information Processing Systems*. Red Hook, NY, USA: Curran Associates Inc.
- Yang, T.-Y.; Rosca, J.; Narasimhan, K.; and Ramadge, P. J. 2020. Projection-Based Constrained Policy Optimization. In *Proceedings of the Eighth International Conference on Learning Representations*, ICLR.
- Yeh, W. G. 1985. Reservoir Management and Operations Models: A State-of-the-art Review. In *Water Resources research*, volume 21,12, 1797–1818.
- Yoon, S.; Fern, A.; Givan, R.; and Kambhampati, S. 2008. Probabilistic Planning via Determinization in Hindsight. In *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence*, 1010—1016.