Model Checking Meets Robot Planning: A Sampling-Based Framework for Large-Scale Optimal Temporal Logic Synthesis

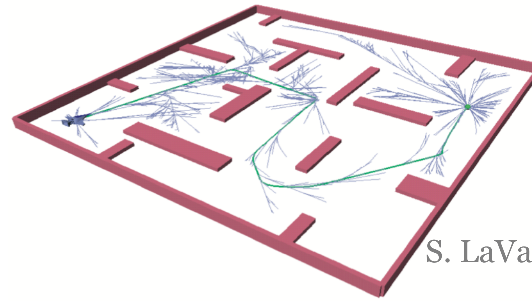
Michael M. Zavlanos
Mechanical Engineering & Materials Science
Electrical & Computer Engineering
Computer Science
Duke University

Robot Motion Planning

Point-to-point navigation tasks

- “Starting from point A, reach point B while avoiding obstacles”

L. Kavraki et al (TRA 1996), S. LaValle et al (IJJR 2001),
S. Karaman et al (IJJR 2011), L. Janson (IJRR 2015)



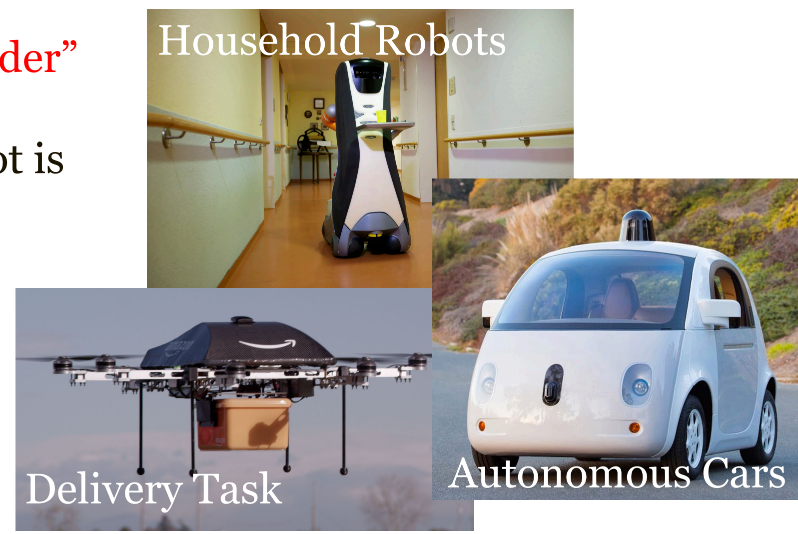
S. LaValle et al (IJJR 2001),

High-level complex tasks

- “Pick up the mail by visiting houses in a given order”
- “Next visit a delivery site”
- “Never leave the delivery site until a ground robot is present to pick up the mail”
- “Repeat this process every day”

How to express complex tasks in a formal way?
How to synthesize optimal and correct-by-construction controllers?

M. Kloetzer et al (TRO 2010), S. Smith et al (IJRR 2011)
A. Ulusoy et al (IJRR 2013), M. Guo et al (IJRR 2015)



UF UNIVERSITY of FLORIDA · Duke UNIVERSITY · TEXAS The University of Texas at Austin · UC SANTA CRUZ

Linear Temporal Logic

LTL is a formal type of logic that consists of Boolean and temporal operators defined over a set of atomic propositions/predicates.

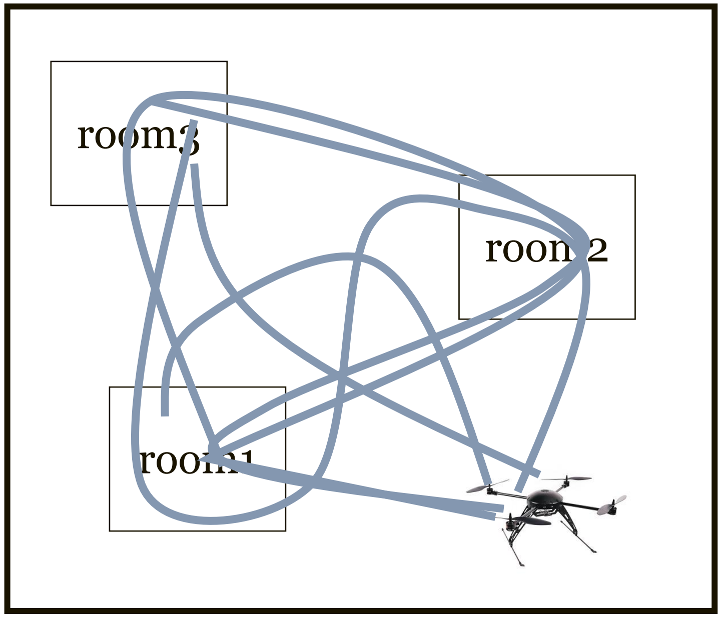
Syntax: $\phi ::= \text{true} \mid \pi \mid \phi_1 \wedge \phi_2 \mid \neg\phi \mid \bigcirc \phi \mid \phi_1 \mathcal{U} \phi_2$

Set $\mathcal{AP}$ of Atomic Propositions (Boolean variables).

Other useful temporal operators:

- Always $\square$
-
- Eventually $\diamond$
- Infinitely often $\square\diamond$

Expressing Complex Tasks using LTL



Reachability task $\Diamond \pi_i^{\texttt{room1}}$

Reachability with avoidance $\neg(\pi_i^{\texttt{room1}} \vee \pi_i^{\texttt{room2}})\mathcal{U}\pi_i^{\texttt{room3}}$

Coverage task $\Diamond \pi_i^{\texttt{room1}} \wedge \Diamond \pi_i^{\texttt{room2}} \wedge \Diamond \pi_i^{\texttt{room3}}$

Sequencing $\Diamond(\pi_i^{\texttt{room1}} \wedge (\Diamond(\pi_i^{\texttt{room2}} \wedge \Diamond \pi_i^{\texttt{room3}})))$

Recurrent sequencing $\Box\Diamond(\pi_i^{\texttt{room1}} \wedge (\Diamond(\pi_i^{\texttt{room2}} \wedge \Diamond \pi_i^{\texttt{room3}})))$

Compositional tasks: $\phi = \underbrace{\Box\Diamond(\pi_1^{\mathrm{room1}})}_{\text{Robot 1: visit room1 infinitely often}} \wedge \underbrace{(\neg\pi_1^{\mathrm{room1}}\mathcal{U}\pi_2^{\mathrm{room2}})}_{\text{Robot 1: never visit room1 until robot 2 visits room 2}} \wedge \underbrace{(\Diamond\Box(\pi_2^{\mathrm{room3}}))}_{\text{Robot 2: eventually always visit room 3}}$

Robot 1: visit room1 infinitely often

Robot 1: never visit room1 until robot 2 visits room 2

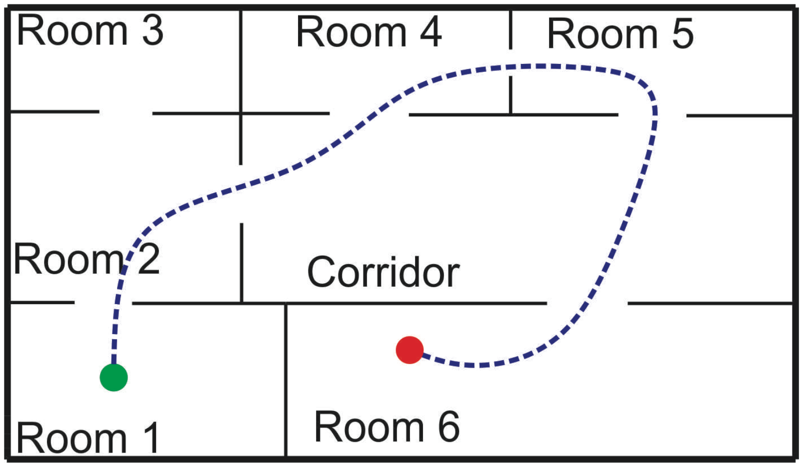
Robot 2: eventually always visit room 3

Optimal Control Synthesis

Given N robots, an abstraction of the environment and robot dynamics

$$\dot{\mathbf{x}}_i(t) = f_i(\mathbf{x}_i(t), \mathbf{u}_i(t)), \ \forall i \in \{1, 2, \ldots, N\}$$

and a collaborative task captured by a global LTL specification ϕ, synthesize a discrete motion plan τ such that $\tau \models \phi$ and a user-specified metric $J(\tau)$, such as total traveled distance, is minimized.

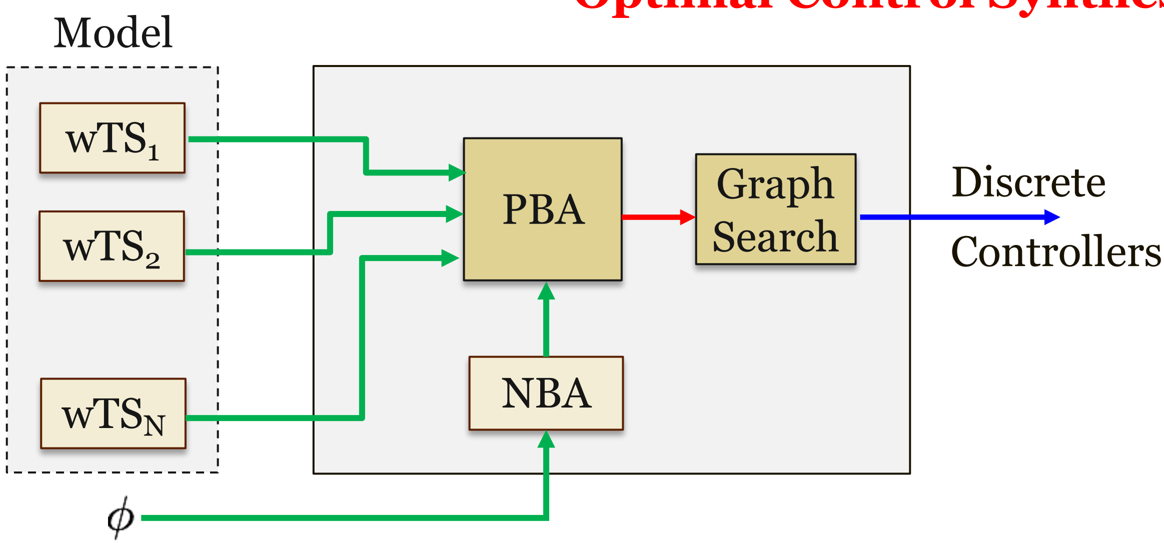


$$\phi = \Diamond(\pi_i^{\text{room2}} \wedge (\Diamond\pi_i^{\text{room4}} \wedge (\Diamond\pi_i^{\text{room5}} \wedge (\Diamond\pi_i^{\text{room6}})))) \wedge (\Diamond\Box\pi_i^{\text{room6}}) \wedge (\Box\neg\pi_i^{\text{room3}})$$

$$\tau = \text{room1,room2,corridor,room4, room5,corridor,room6, [room6]}^{\omega}$$

Challenges

Optimal Control Synthesis



M. Kloetzer (TRO 2010)
S. Smith et al (IJRR 2011)
A. Ulusoy et al (IJRR 2013)
M. Guo et al (IJRR 2015)

State explosion, Computationally expensive, Centralized (less than ~10^7 states)

Model Checking / Verification

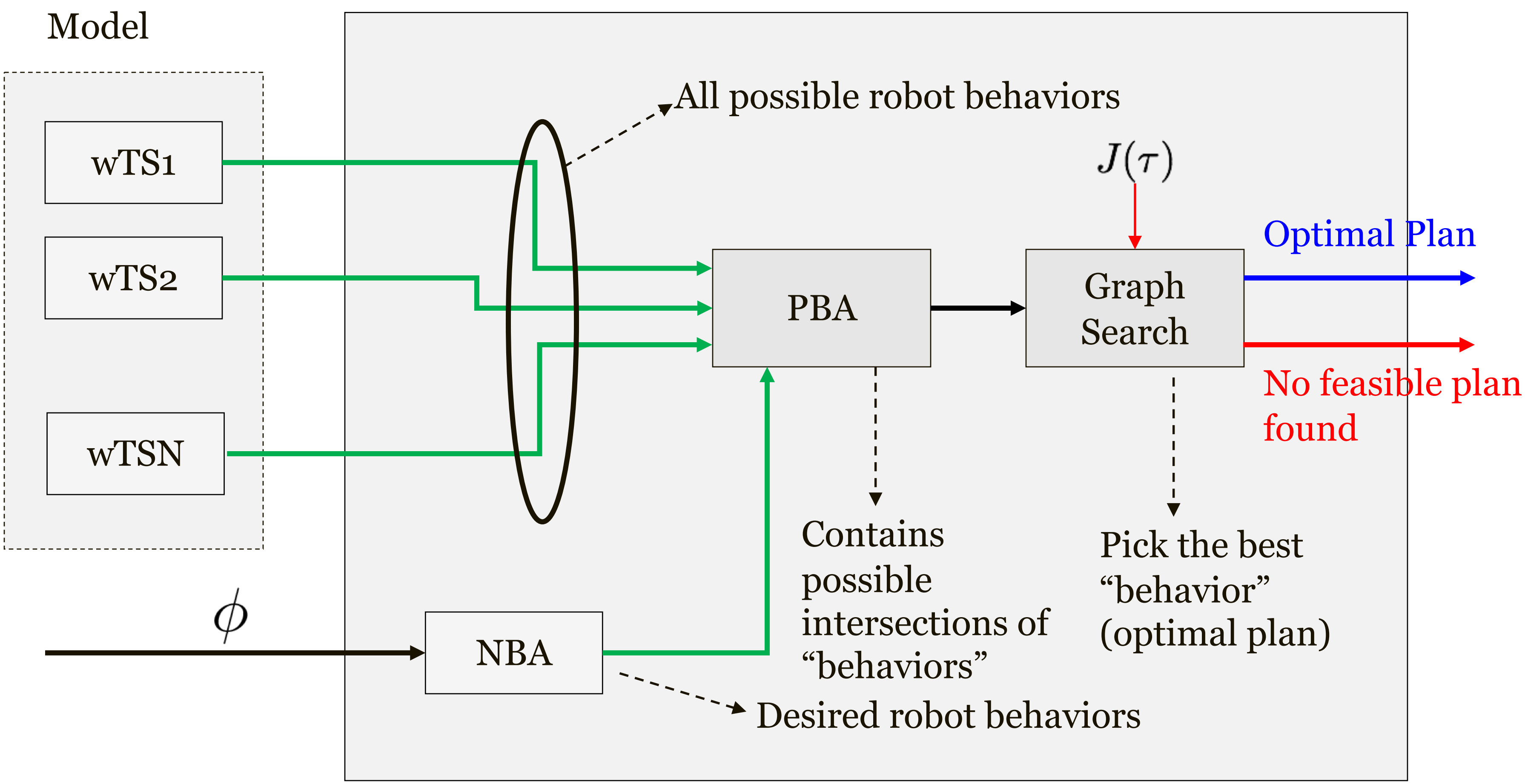
NuSMV 2, nUxmv, SPIN, SPOT

More scalable (~10^{30} states) but no optimality guarantees. Return a feasible, and not the optimal, solution.

We propose an algorithm that can solve optimally hundreds of orders of magnitude larger planning problems than state-of-the-art methods (~10^{800} states and beyond).

UF UNIVERSITY of FLORIDA · Duke UNIVERSITY · TEXAS The University of Texas at Austin · UC SANTA CRUZ

Overview


Model
wTS1
wTS2
wTSN
All possible robot behaviors
$J(\tau)$
Optimal Plan
PBA
Graph Search
No feasible plan found
Contains possible intersections of "behaviors"
Pick the best "behavior" (optimal plan)
ϕ
NBA
Desired robot behaviors

Weighted Transition Systems (wTS)

$$\dot{\mathbf{x}}_i(t) = f_i(\mathbf{x}_i(t), \dot{\mathbf{u}}_i(t))$$
$$\mathbf{x}_i(t) \in \mathcal{W}$$

Abstraction Process

$$\text{wTS}_i = (\mathcal{Q}_i, q_i^0, \rightarrow_i, w_i, \mathcal{AP}_i, L_i)$$

$\mathcal{Q}_i = \{q_i^{r_e}\}_{\forall \text{regions } r_e}$: set of states

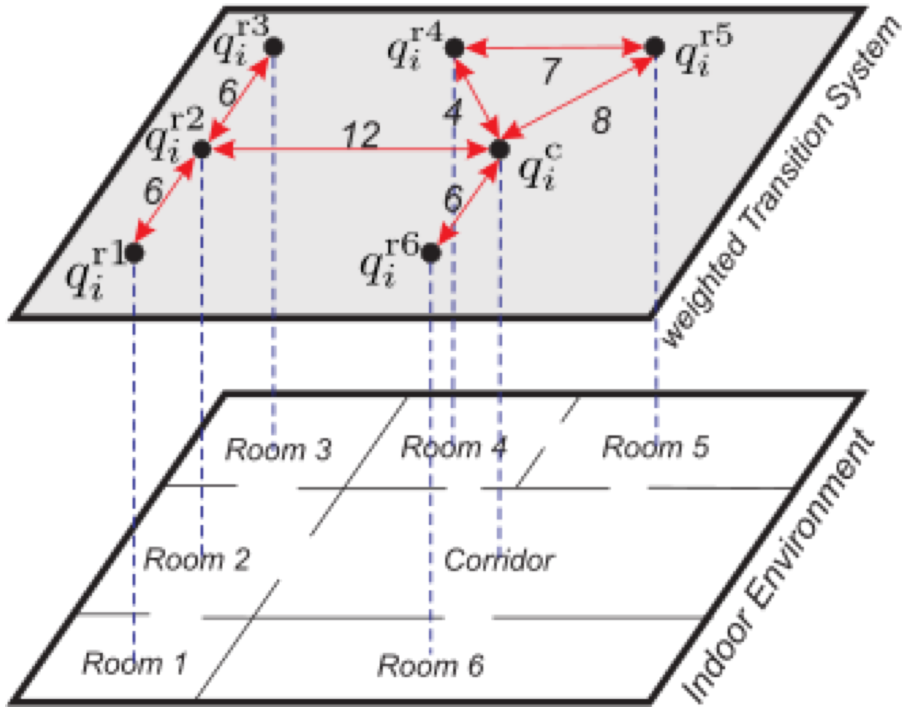
q_i^0 : initial state

$\rightarrow_i \subseteq \mathcal{Q}_i \times \mathcal{Q}_i$: transition rule

$\mathcal{AP}_i = \cup_{r_e}\{\pi_i^{r_e}\}$: set of APs

$w_i : \mathcal{Q}_i \times \mathcal{Q}_i \rightarrow R_+$: cost function

$L_i : \mathcal{Q}_i \rightarrow 2^{\mathcal{AP}_i}$: observation relation



$$L_i(q_i^{r6}) = \pi_i^{r6}$$

Non-Deterministic Buchi Automaton (NBA)

Translate the LTL formula to a NBA: $B = (\mathcal{Q}_B, \mathcal{Q}_B^0, \Sigma, \rightarrow_B, \mathcal{Q}_B^F)$

$\mathcal{Q}_B$	Set of states
$\mathcal{Q}_B^0$	Set of initial states
$\Sigma = 2^{\mathcal{AP}}$	Alphabet
$\rightarrow_B \subseteq \mathcal{Q}_B \times \Sigma \times \mathcal{Q}_B$	Transition rule
$\mathcal{Q}_B^F$	Set of final states

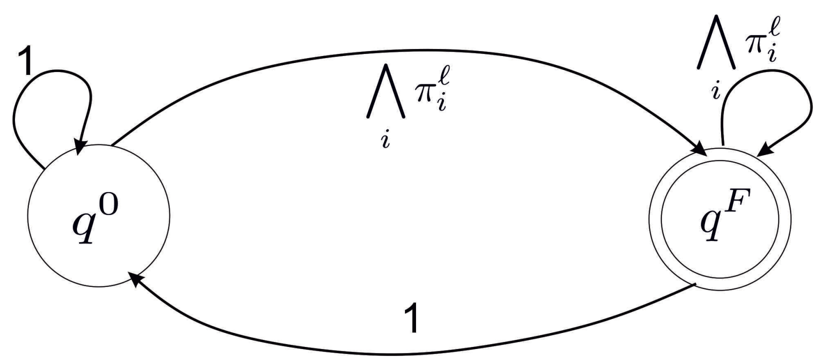
NBA transitions are activated based on the observed atomic propositions.

The LTL formula is satisfied if the set of final states is visited infinitely often.

$$\phi = \Box\Diamond \bigwedge_i \pi_i^\ell$$

Translation

D. Oddoux, P. Gastin. LTL2BA software, 2009.



Product Buchi Automaton (PBA)

Given N transition systems and a NBA, the PBA is:

PTS

$$P = \underbrace{wTS_1 \times \dots \times wTS_N}_{\text{PTS}} \times B = (\mathcal{Q}_P, \mathcal{Q}_P^0, \rightarrow_P, \mathcal{Q}_P^F)$$

$\mathcal{Q}_P = \mathcal{Q}_1 \times \cdots \times \mathcal{Q}_N \times \mathcal{Q}_B$ — Set of states

$\mathcal{Q}_P^0 = q_1^0 \times \dots q_N^0 \times \mathcal{Q}_B^0$ — Set of initial states

$\rightarrow_P : \mathcal{Q}_P \times \mathcal{Q}_P$ — Transition rule

$$\overbrace{(q_1, q_2, \dots, q_N}^{q_{\text{PTS}}}, q_B) \rightarrow_P \overbrace{(q_1', q_2', \dots, q_N'}^{q_{\text{PTS}}'}, q_B') \begin{cases} q_i \rightarrow_i q_i', \forall i \in \{1, \dots, N\} \\ q_B \xrightarrow{L_1(q_1), \dots, L_N(q_N)}_P q_B' \end{cases}$$

Feasible wTS transitions.

Feasible NBA transitions based on the observed atomic propositions.

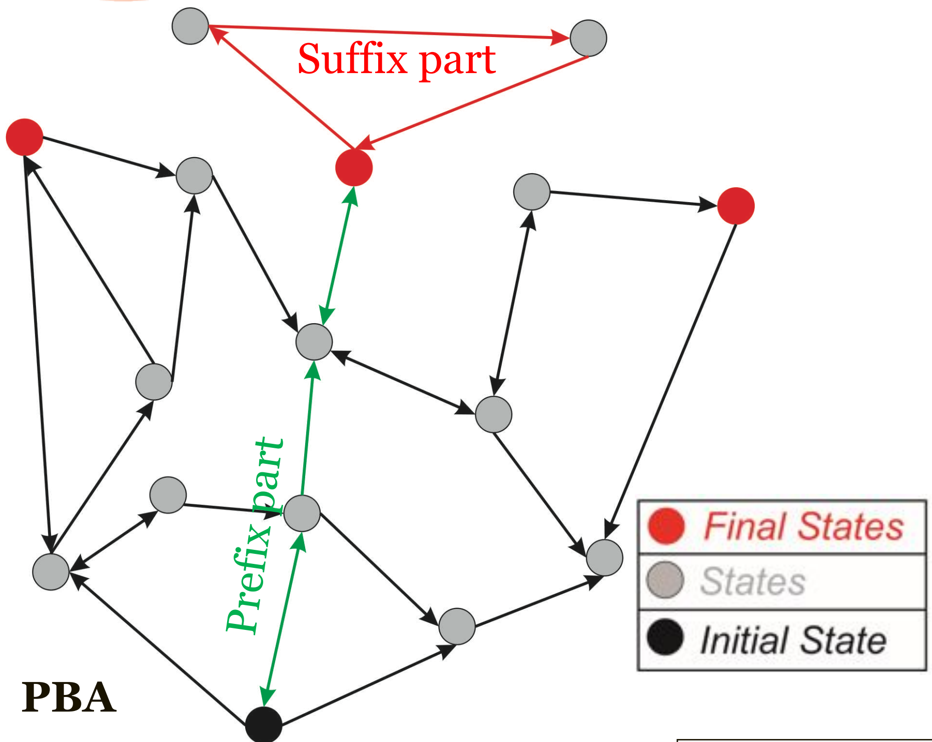
$$w_P(q_P, q_P') = \sum_{i=1}^{N} w_i(q_i, q_i')$$ — Cost function

$\mathcal{Q}_P^F = \mathcal{Q}_1 \times \cdots \times \mathcal{Q}_N \times \mathcal{Q}_B^F$ — Set of final states

The LTL formula is satisfied if the set of final states of the PBA is visited infinitely often.

UF UNIVERSITY of FLORIDA · Duke UNIVERSITY · TEXAS The University of Texas at Austin · UC SANTA CRUZ

Optimal Control Synthesis



- State-space of the PBA:

$$\mathcal{Q}_P = \mathcal{Q}_1 \times \cdots \times \mathcal{Q}_N \times \mathcal{Q}_B$$

- Find paths in the PBA from initial states to final states (prefix part) and cycles around the final states (suffix part).

- Plans in Prefix - Suffix structure:

$$\tau = \tau^{\text{pre}}[\tau^{\text{suf}}]^{\omega}$$

- Pick the prefix-suffix plan with the minimum cost.

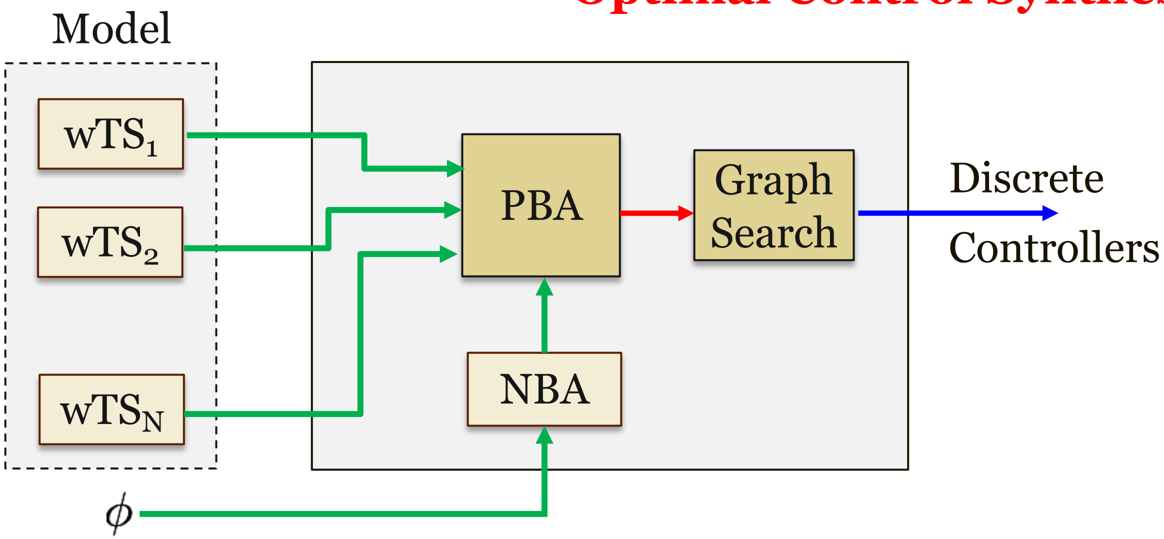
- Cost of plan:

Transition weights (e.g., distance metric)

$$J(\tau) = \underbrace{\sum_{k=1}^{|\tau^{\text{pre}}|-1} \sum_{i=1}^{N} w_i(\Pi|_{\text{wTS}_i}\tau^{\text{pre}}(k), \Pi|_{\text{wTS}_i}\tau^{\text{pre}}(k+1))}_{\text{Cost } J(\tau^{\text{pre}}) \text{ of prefix}} + \underbrace{\sum_{k=1}^{|\tau^{\text{suf}}|-1} \sum_{i=1}^{N} w_i(\Pi|_{\text{wTS}_i}\tau^{\text{suf}}(k), \Pi|_{\text{wTS}_i}\tau^{\text{suf}}(k+1))}_{\text{Cost } J(\tau^{\text{suf}}) \text{ of suffix}}$$

Limitations of Existing Methods

Optimal Control Synthesis



M. Kloetzer (TRO 2010)
S. Smith et al (IJRR 2011)
A. Ulusoy et al (IJRR 2013)
M. Guo et al (IJRR 2015)

State explosion, Computationally expensive, Centralized (less than ~10^7 states)

Model Checking / Verification

NuSMV 2, nUxmv, SPIN, SPOT

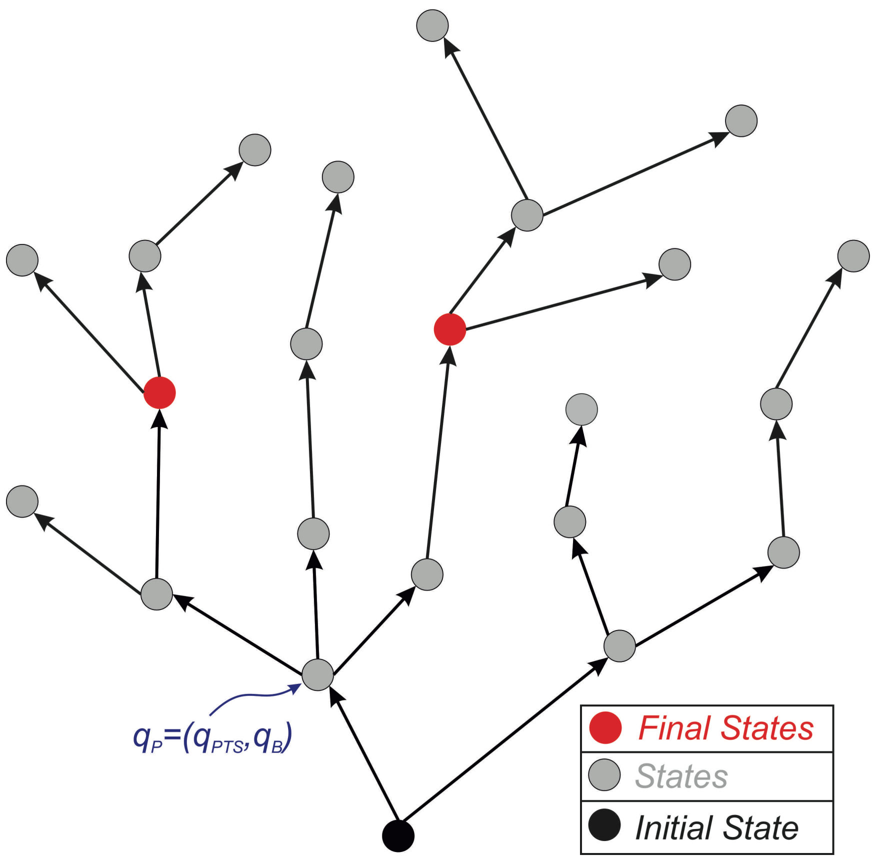
More scalable (~10^{30} states) but no optimality guarantees. Return a feasible, and not the optimal, solution.

We propose an algorithm that can solve optimally hundreds of orders of magnitude larger planning problems than state-of-the-art methods (~10^{800} states and beyond).

University of Florida · Duke University · TEXAS The University of Texas at Austin · UC Santa Cruz

Sampling-Based Optimal Control Synthesis

- Completely avoid taking the product among wTSs and NBA.
- Approximate representation of PBA by a tree $\mathcal{G}_T = \{\mathcal{V}_T, \mathcal{E}_T\}$



Approximate?

$$\mathcal{V}_T \subseteq \mathcal{Q}_P$$
$$\mathcal{E}_T \subseteq \rightarrow_P$$

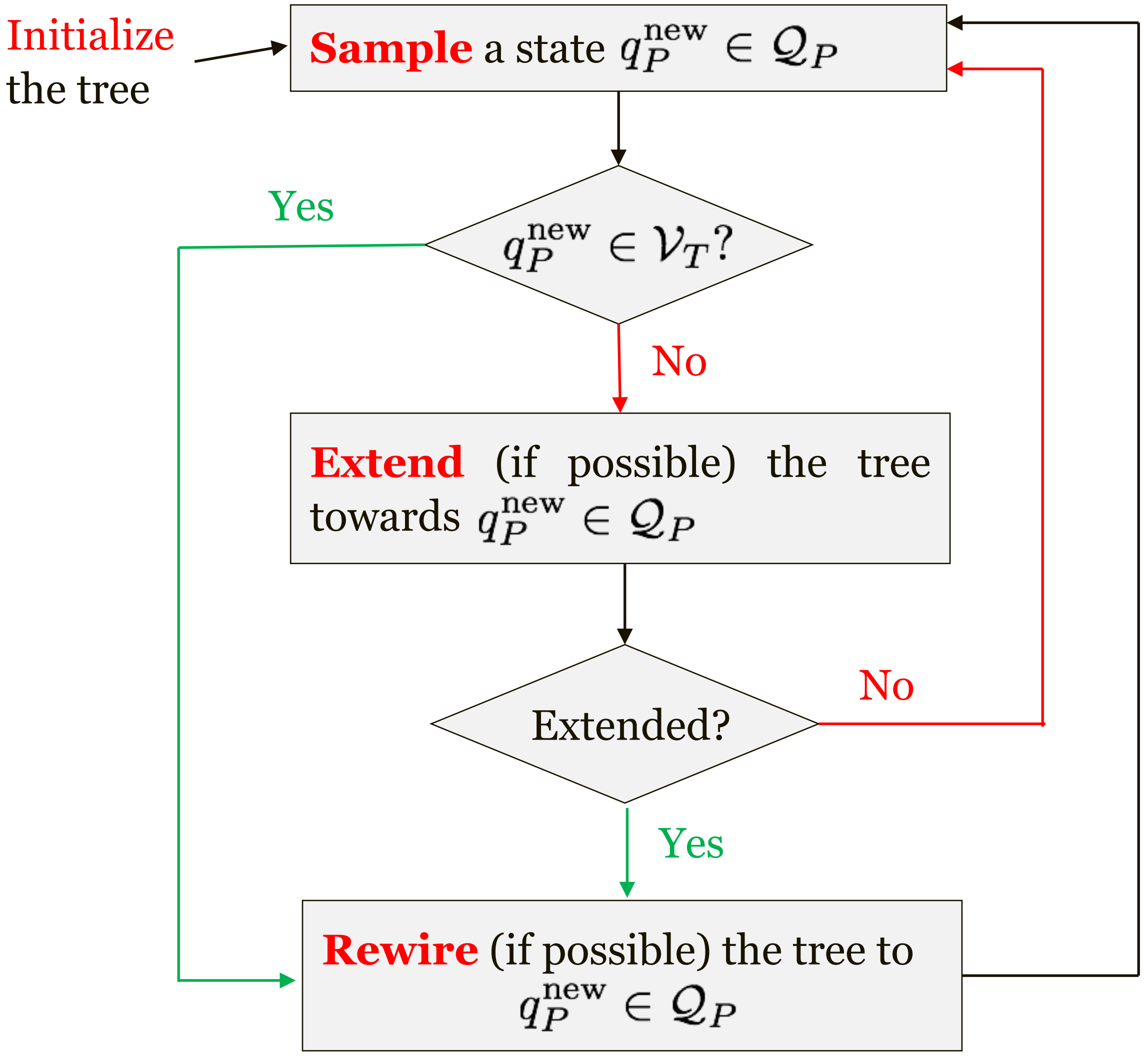
Why trees?

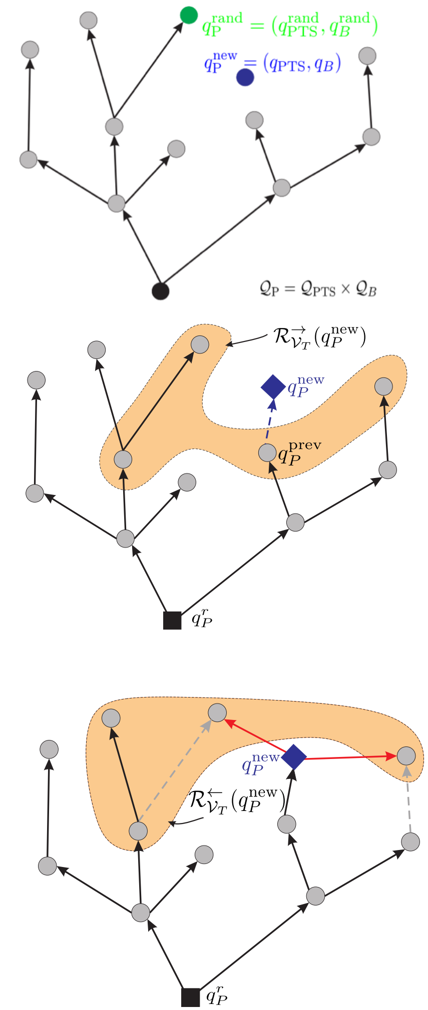
- Resource efficient (memory complexity):
  $$O(|\mathcal{E}_T|) << O(|\mathcal{V}_P| + |\mathcal{E}_P|)$$
- Computationally inexpensive graph search methods.

How to build trees?

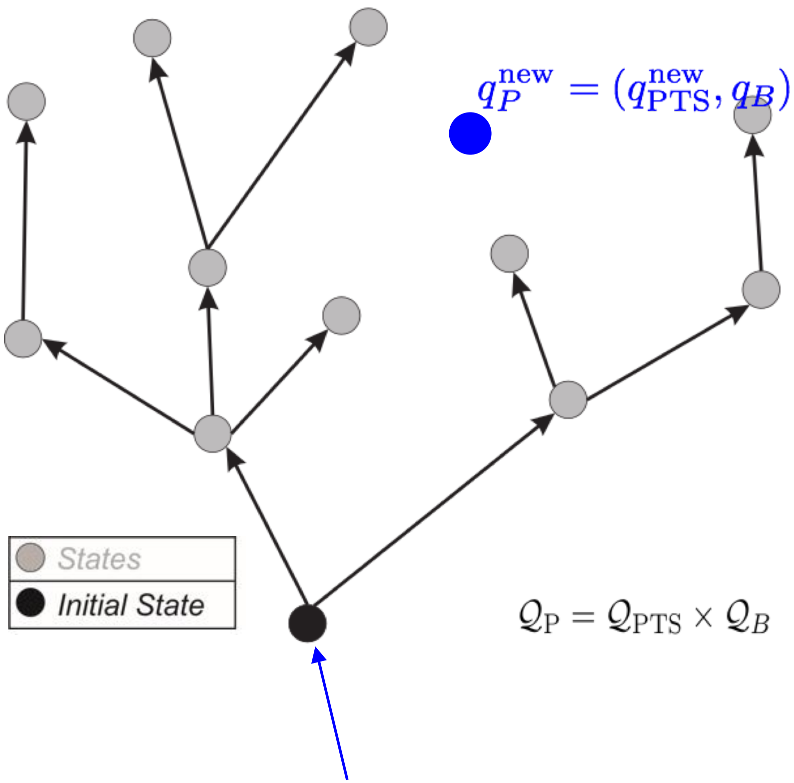
- Incrementally through sampling on the PBA state-space.
- Cycle-detection method.

Algorithm Outline





Sampling Function



The root of the tree is an initial state of the PBA.

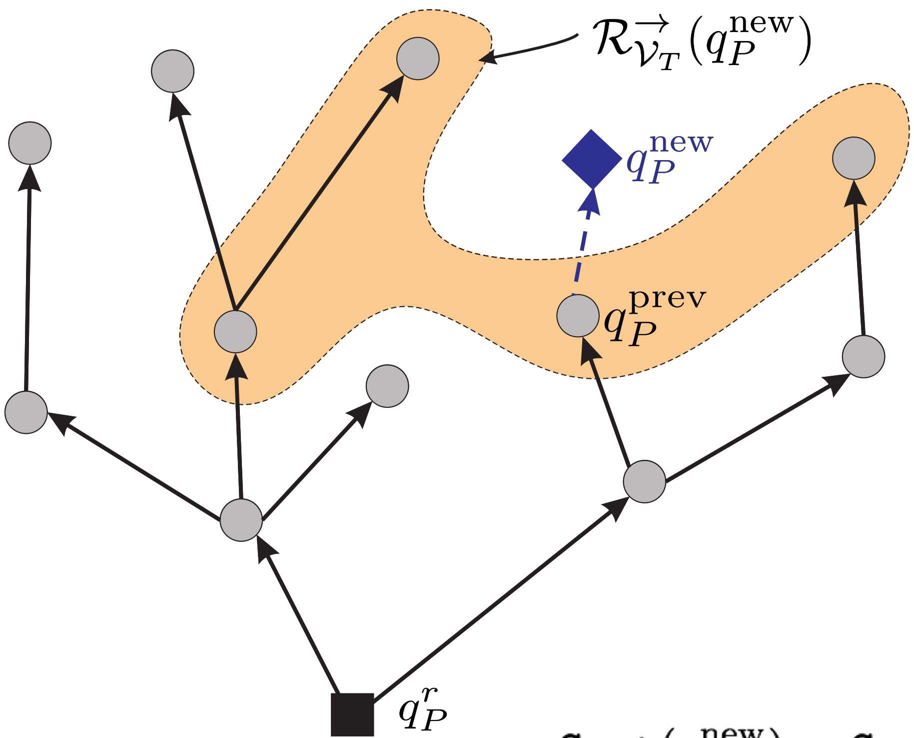
Sample a state $q_P^{\text{new}} = (q_{\text{PTS}}^{\text{new}}, q_B) \in \mathcal{Q}_P$ from the density function $f_{\text{sample}} : \mathcal{Q}_P \rightarrow [0,1]$

If $q_P^{\text{new}} = (q_{\text{PTS}}^{\text{new}}, q_B) \in \mathcal{V}_T$, then "Rewire" otherwise "Extend"

The choice of the sampling function affects the performance of the algorithm

UF UNIVERSITY of FLORIDA · Duke UNIVERSITY · TEXAS The University of Texas at Austin · UC SANTA CRUZ

Extend Function



1) Collect all states in the tree that can directly reach q_P^{new} in a set of candidate parents:

$$\mathcal{R}^{\rightarrow}_{\mathcal{V}_T}(q_P^{\text{new}}) = \{q_P \in \mathcal{V}_T \mid q_P \rightarrow_P q_P^{\text{new}}\}$$

2) Select as a parent for q_P^{new} the node

$$q_P^{\text{prev}} \in \mathcal{R}^{\rightarrow}_{\mathcal{V}_T}(q_P^{\text{new}})$$

that incurs the minimum cost from the root.

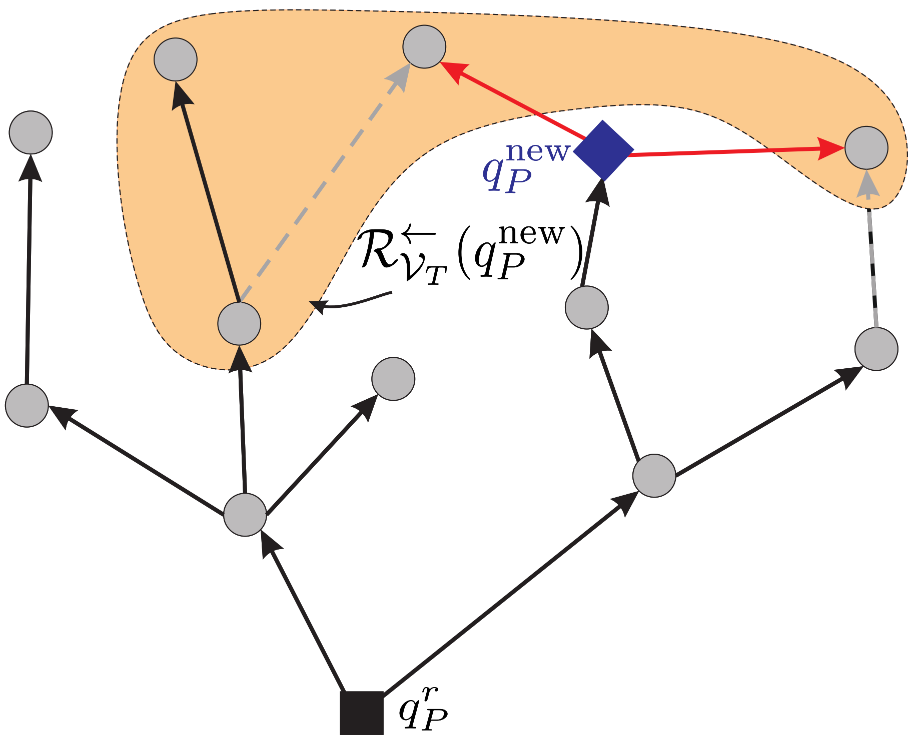
$$\texttt{Cost}(q_P^{\text{new}}) = \texttt{Cost}(q_P^{\text{prev}}) + \sum_{i=1}^{N} w_i(\Pi_{\text{wTS}_i} q_P^{\text{prev}}, \Pi_{\text{wTS}_i} q_P^{\text{new}})$$

Transition weights (e.g., distance metric)

Complexity of extending: $O\left(|\mathcal{V}_T|(N+1)\right)$

UF UNIVERSITY of FLORIDA · Duke UNIVERSITY · TEXAS The University of Texas at Austin · UC SANTA CRUZ

Rewire Function



1) Collect all states in the tree that can be directly reached by q_P^{new} in a set of possible children:

$$\mathcal{R}^{\leftarrow}_{\mathcal{V}_T}(q_P^{\text{new}}) = \{q_P \in \mathcal{V}_T \mid q_P^{\text{new}} \rightarrow_P q_P\}$$

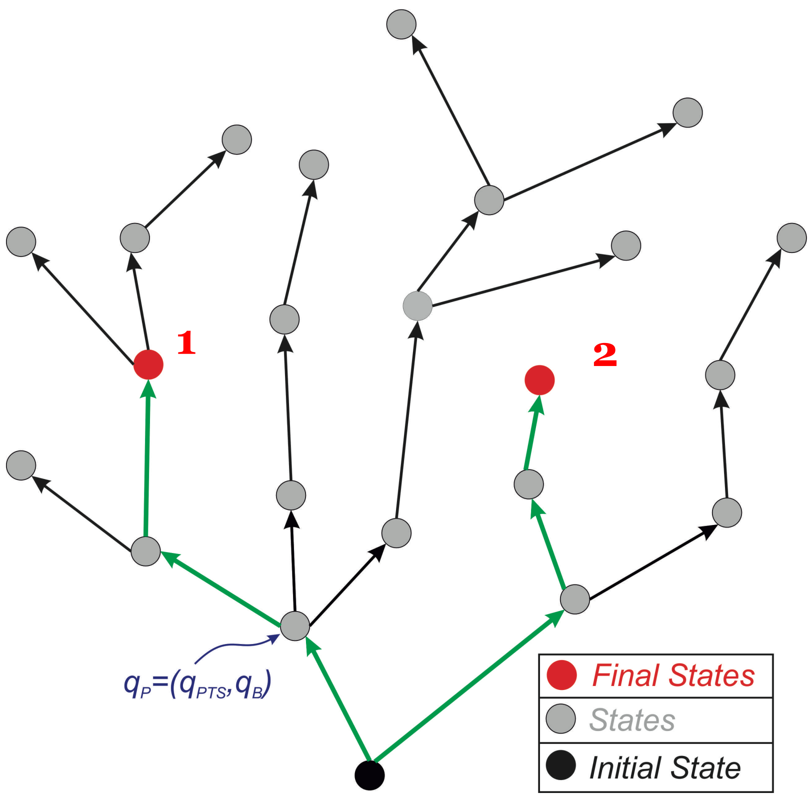
2) Rewire nodes $q_P \in \mathcal{R}^{\leftarrow}_{\mathcal{V}_T}(q_P^{\text{new}})$ to q_P^{new}

if their cost from the root can be further minimized.

Complexity of rewiring: $O\left(|\mathcal{V}_T|(N+1)\right)$

UF UNIVERSITY of FLORIDA · Duke UNIVERSITY · TEXAS The University of Texas at Austin · UC SANTA CRUZ

Construction of Prefix Plans

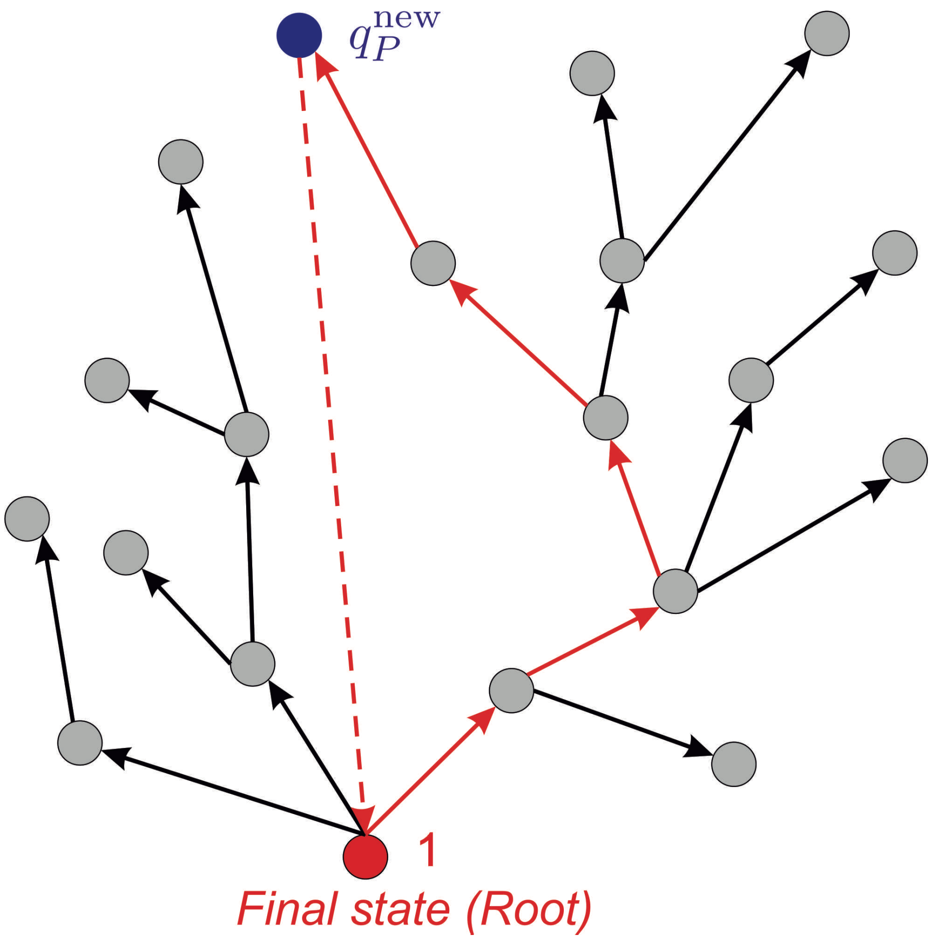


The construction of the tree is **terminated** after a user-specified maximum number of iterations.

Compute paths (prefix parts) by tracking the sequence of parent nodes from final states to the **root**.

Complexity of finding paths: $O(|\mathcal{V}_T|)$

Construction of Suffix Plans

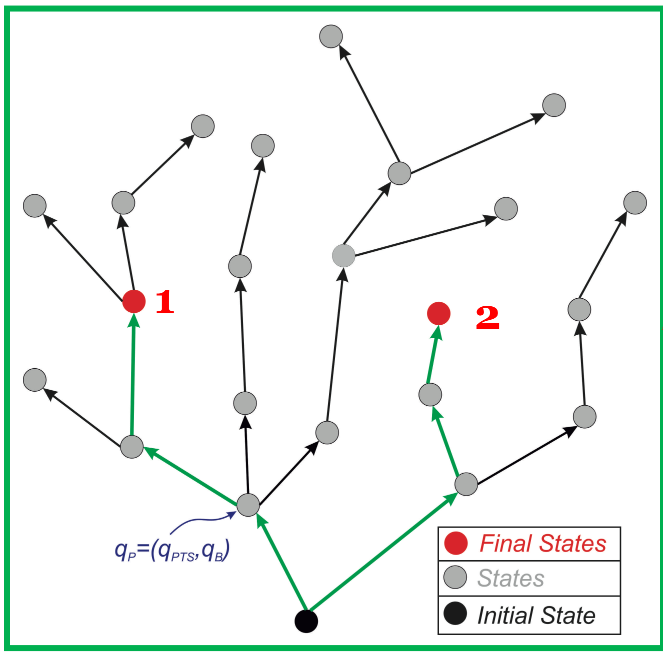


A tree graph rooted at a final state is built.

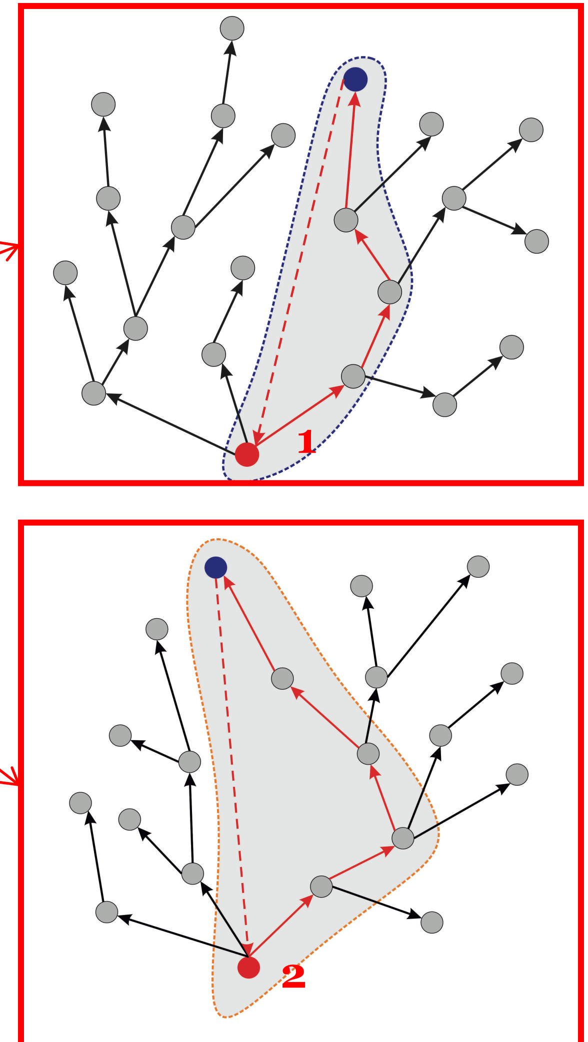
Every time a new node is added, check if a direct transition to the final state (root) is feasible, forming a suffix loop.

Optimal Discrete Plan Synthesis

Tree for prefix parts



Tree for suffix parts



Constructed motion plans: $\tau^f = \tau^{\text{pre},f}[\tau^{\text{suf},f}]^\omega; \; f = 1, 2$

Optimal motion plan: $\tau^{f^*} = \tau^{\text{pre},f^*}[\tau^{\text{suf},f^*}]^\omega$

$$f^* = \text{argmin}_f J(\tau^f)$$

Completeness and Optimality

Theorem: The proposed sampling-based algorithm is **probabilistically complete.**

Theorem: The proposed sampling-based algorithm is **asymptotically optimal**, i.e.,

$$\mathbb{P}\left(\left\{\lim_{n_{\max}^{\text{pre}}\to\infty, n_{\max}^{\text{suf}}\to\infty} J(\tau^{n_{\max}^{\text{suf}}}_{n_{\max}^{\text{pre}}}) = J^*\right\}\right) = 1$$

Convergence Rate Analysis

Theorem: Let p denote a feasible prefix or suffix path.

$$\mathtt{p} = q_P^1, q_P^2, \ldots, q_P^{K-1}, q_P^K$$

Then there exist parameters $\alpha_n(\mathtt{p}) \in (0,1]$ such that the probability $\Pi_{\mathtt{suc}}(q_P^K)$ of finding the feasible prefix/suffix path p within $n_{\mathtt{max}}$ iterations satisfies

$$1 \geq \Pi_{\mathtt{suc}}(q_P^K) \geq 1 - e^{-\frac{\sum_{n=1}^{n_{\mathtt{max}}} \alpha_n(\mathtt{p})}{2} n_{\mathtt{max}} + K}, \quad \text{if } n_{\mathtt{max}} > K$$

Depend on the selected sampling functions

Theorem: Let p* denote the optimal prefix or suffix path.

$$\mathtt{p}^* = q_P^1, q_P^2, \ldots, q_P^{K-1}, q_P^K$$

Then there exist parameters $\alpha_n(\mathtt{p}^*) \in (0,1]$ and $\gamma_n(q_P^k) \in (0,1]$ and iterations n_k for every state q_P^k in the optimal path such that the probability of finding the optimal path within $n_{\mathtt{max}} > 2K$ iterations satisfies

$$\Pi_{\mathtt{opt}}(\mathtt{p}^*) \geq \left(1 - e^{-\frac{\sum_{n=1}^{\bar{n}} \alpha_n(\mathtt{p}^*)}{2} + K}\right) \prod_{k=1}^{K-1} \left(1 - e^{-\frac{\sum_{n=n_{k-1}}^{n_{\mathtt{max}}} \gamma_n(q_P^k)}{2} + 1}\right)$$

Scalability with Uniform Sampling

Team 1, Team 2, Team 3, Team 4, Team 5, Team 6

$$\phi = [\Box\Diamond(\pi_1^{\ell_5} \wedge \pi_2^{\ell_5})] \wedge [\Box\Diamond(\pi_2^{\ell_1} \wedge \pi_3^{\ell_1} \wedge \pi_4^{\ell_1})] \wedge [\Box\Diamond(\pi_4^{\ell_7} \wedge \pi_5^{\ell_7} \wedge \pi_6^{\ell_7})] \wedge [\Box\Diamond(\pi_6^{\ell_8} \wedge \pi_7^{\ell_8})]$$
$$\wedge [\Box\Diamond(\pi_7^{\ell_4} \wedge \pi_8^{\ell_4})] \wedge [\Box\Diamond(\pi_8^{\ell_3} \wedge \pi_9^{\ell_3})] \wedge [\neg(\pi_1^{\ell_5} \wedge \pi_2^{\ell_5})\mathcal{U}\pi_1^{\ell_7}]$$

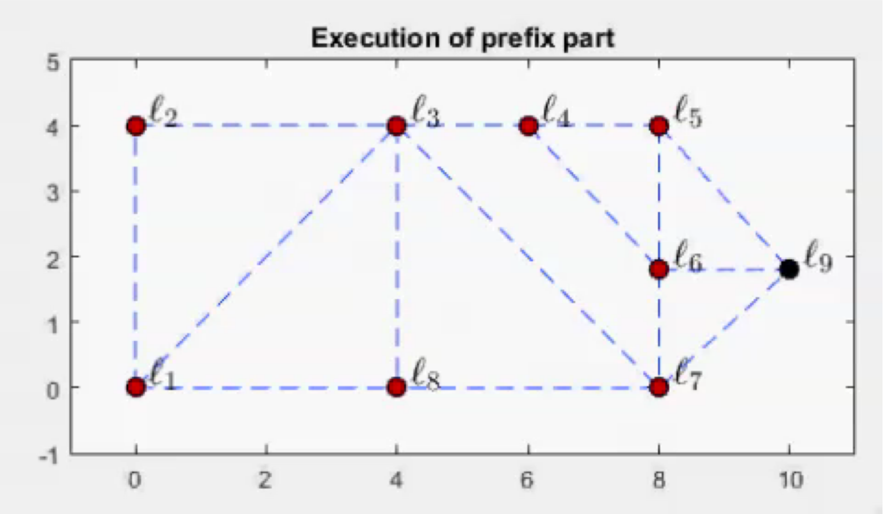
$\phi \rightarrow |\mathcal{Q}_B| = 8$

D. Oddoux, P. Gastin. LTL2BA software, 2009.

$N = 9$ robots

$|\mathcal{Q}_i| = 9, \forall i \in \{1, \ldots, 9\}$

$|\mathcal{Q}_P| = |\mathcal{Q}_i|^9 \cdot |\mathcal{Q}_B| = 3.099 \times 10^9$ states



Using **uniform sampling functions**, the proposed method detected the first feasible path in 1.6 hours with cost 568.1857 meters.

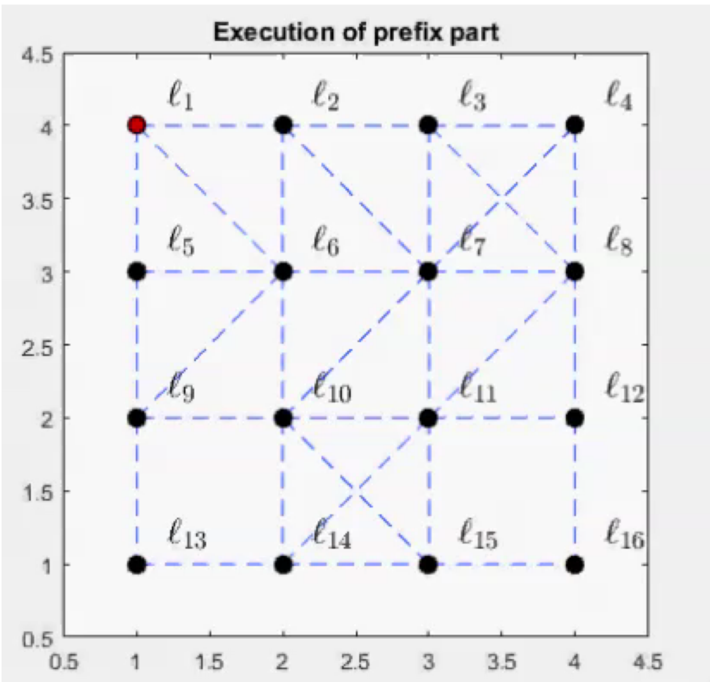
The existing optimal control synthesis method (e.g., Dijkstra, implicit graphs) failed to solve this problem (can solve problems with ~10^7 states/edges).

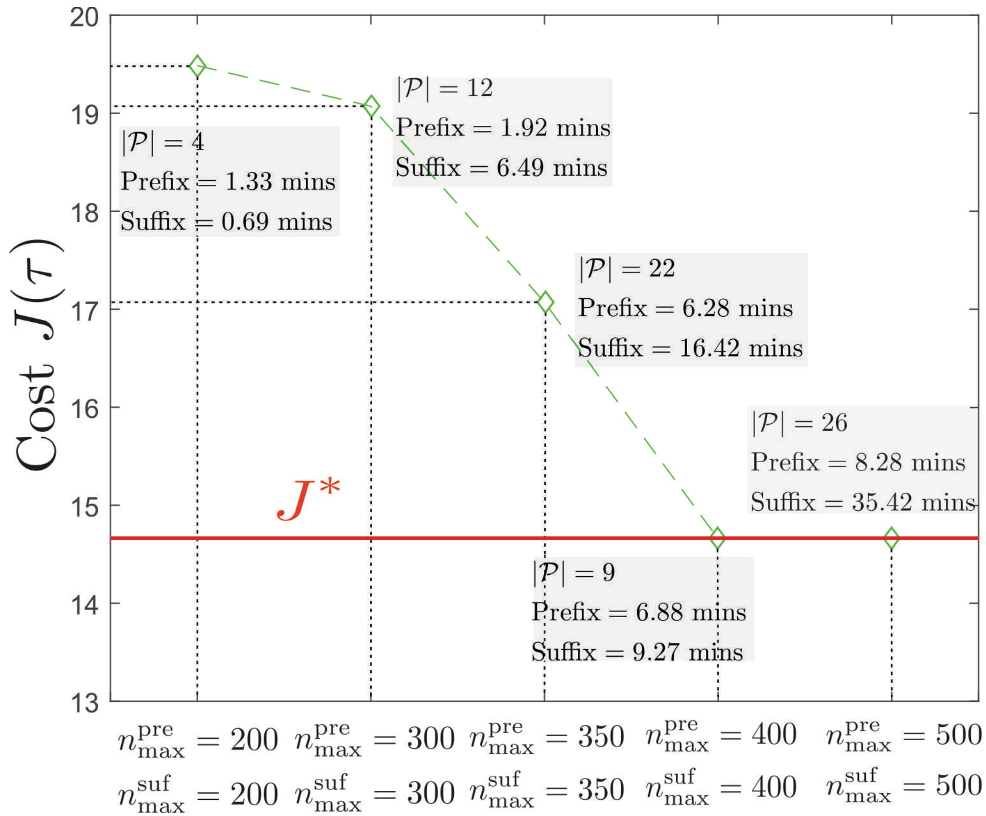
NuSMV generated a feasible plan in 2 seconds with cost 672.2431 meters.

UF UNIVERSITY of FLORIDA · Duke UNIVERSITY · TEXAS The University of Texas at Austin · UC SANTA CRUZ

Optimality

$$\phi = \Box\Diamond(\pi_1^{\ell_6} \wedge \Diamond(\pi_2^{\ell_{14}})) \wedge \Box(\neg\pi_1^{\ell_9}) \wedge \Box(\pi_2^{\ell_{14}} \rightarrow \bigcirc(\neg\pi_2^{\ell_{14}} \mathcal{U} \pi_1^{\ell_4})) \wedge (\Diamond\pi_2^{\ell_{12}}) \wedge (\Box\Diamond\pi_2^{\ell_{10}})$$





$|\mathcal{Q}_i| = 16, \ \forall i \in \{1,2\}$

$|\mathcal{Q}_B| = 24$

$|\mathcal{Q}_P| = 6,144$

$J_{\text{NuSMV}} = 30.883$ meters

Comparative Results: Small NBA

MATLAB runtimes to detect the first feasible plan

TABLE I
FEASIBILITY AND SCALABILITY ANALYSIS: $|\mathcal{Q}_B| = 21$

N	$\lvert\mathcal{Q}_i\rvert$	$\lvert\mathcal{Q}_P\rvert$	$n_{\text{Pre1}} + n_{\text{Suf1}}$	$\lvert\mathcal{V}_T^{\text{Pre1}}\rvert + \lvert\mathcal{V}_T^{\text{Suf1}}\rvert$	Pre1+Suf1	NuSMV/nuXmv
1	100	10^3	28 + 28	180 + 54	0.5+0.2 (secs)	$<$ 1 sec
1	1000	10^3	42 +31	338 + 119	0.9+0.7 (secs)	$<$ 1 sec
1	10000	10^4	71 + 43	512 + 131	19.2+11.2 (secs)	M/M
9	9	10^{10}	36 + 37	373 + 83	0.78+0.29 (secs)	$<$ 1 sec
10	100	10^{21}	31+ 31	289 + 101	0.7+0.4 (secs)	$\approx$ 1.5 secs
10	1000	10^{31}	34 + 27	309 + 82	2.1+1.7 (secs)	$\approx$50/40 secs
10	2500	10^{35}	41 + 32	367 + 142	5.94+6.4 (secs)	M/$\approx$ 31 mins
10	10000	10^{41}	40 + 23	357+123	48.44+38.1 (secs)	M/M
100	100	10^{200}	49 + 39	421 + 81	2.5+0.8 (secs)	F/F
100	1000	10^{300}	30 + 38	254 + 110	12.6+24.1 (secs)	M/M
100	10000	10^{400}	24 + 49	241 + 55	4.2+5.2 (mins)	M/M
150	10000	10^{600}	29 + 87	382+530	6.1 + 27.71 (mins)	M/M
200	10000	10^{800}	42 + 49	453 + 276	11.3+23.5 (mins)	M/M

Comparative Results: Large NBA

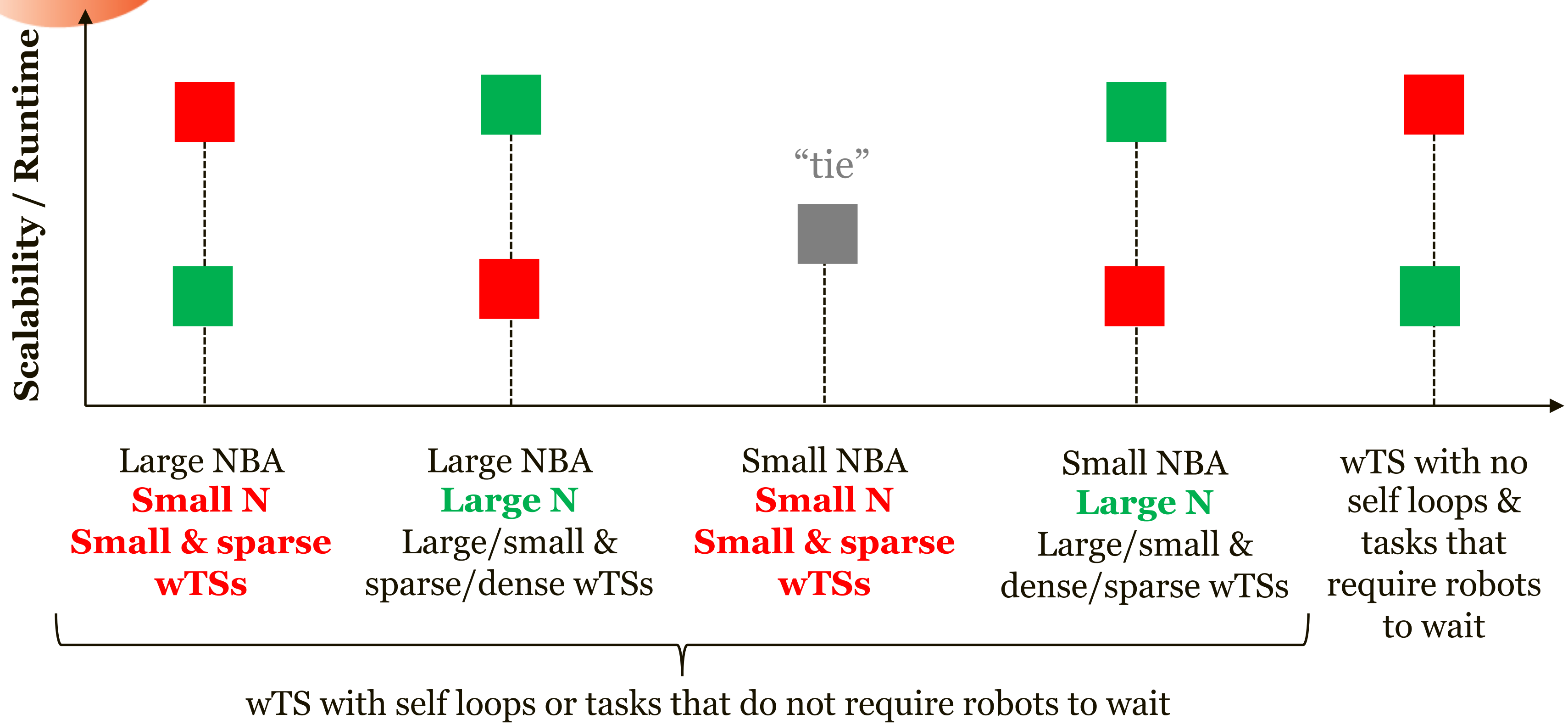
MATLAB runtimes to detect the first feasible plan

TABLE II
FEASIBILITY AND SCALABILITY ANALYSIS: $|\mathcal{Q}_B| = 59$

N	$\|\mathcal{Q}_i\|$	$\|\mathcal{Q}_P\|$	$n_{\text{Pre1}} + n_{\text{Suf1}}$	$\|\mathcal{V}_T^{\text{Pre1}}\| + \|\mathcal{V}_T^{\text{Suf1}}\|$	Pre1+Suf1	NuSMV/nuXmv
1	100	10^3	54 + 92	533 + 274	2.18+1.55 (secs)	$<$ 1 sec
1	1000	10^3	78 +51	326 + 252	1.84+1.37 (secs)	$<$ 1 sec
1	10000	10^4	150 + 107	769 + 364	19.2+11.2 (secs)	M/M
9	9	10^{10}	93 + 27	400 + 168	20.7+18.9 (secs)	$<$ 1 sec
10	100	10^{21}	51+ 39	650 + 239	2.1+0.74 (secs)	$\approx$ 3/2 secs
10	1000	10^{31}	36 + 154	450 + 404	3.9+6.1 (secs)	$\approx$ 80/65 secs
10	2500	10^{35}	61 + 98	710 + 516	10.4+11.9 (secs)	M/$\approx$ 32 mins
10	10000	10^{41}	47 + 164	722+604	56.6 + 98.1(secs)	M/M
100	100	10^{200}	21 + 117	154 + 1431	1.6+18.5 (secs)	F/F
100	1000	10^{300}	52 + 74	401 + 856	19.8+53.32 (secs)	M/M
100	10000	10^{400}	39 + 89	398 + 1621	5.1+28.3 (mins)	M/M
150	10000	10^{600}	39 + 112	526+1864	8.3 + 60.11 (mins)	M/M
200	10000	10^{800}	48 + 103	588 + 1926	11.7+65.9 (mins)	M/M

UF UNIVERSITY of FLORIDA · Duke UNIVERSITY · TEXAS The University of Texas at Austin · UC SANTA CRUZ

NuSMV vs STyLuS*

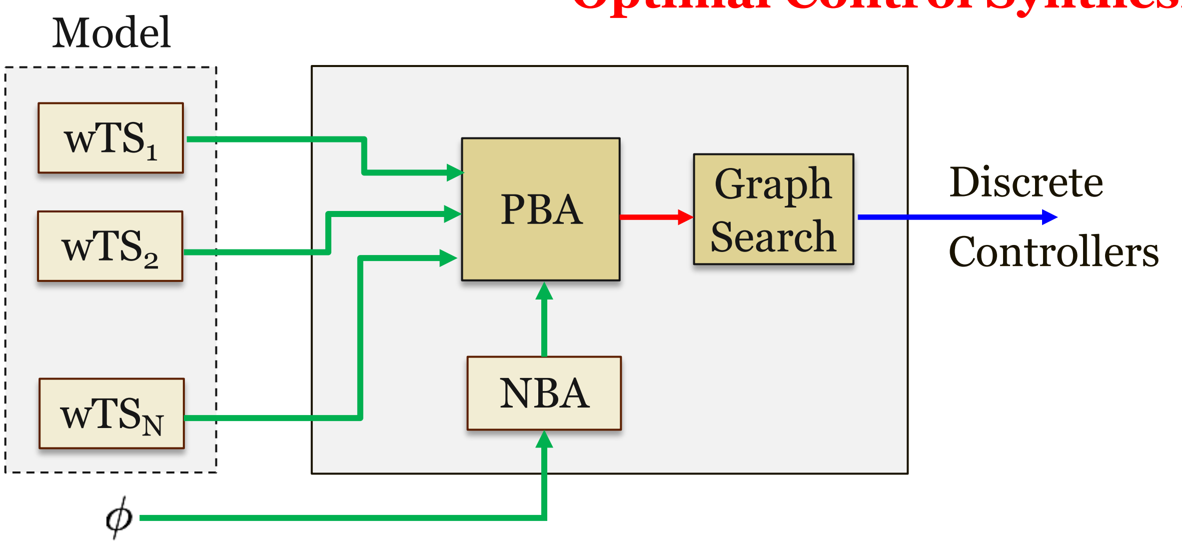


NuSMV is more scalable if the words/guards on the NBA transitions cannot be classified as feasible/infeasible.

NuSMV does not provide any optimality guarantees. STyLuS* is asymptotically optimal and can find “good” plans fast enough.

Summary

Optimal Control Synthesis



M. Kloetzer (TRO 2010)
S. Smith et al (IJRR 2011)
A. Ulusoy et al (IJRR 2013)
M. Guo et al (IJRR 2015)

State explosion, Computationally expensive, Centralized (less than ~10^7 states)

Model Checking / Verification

NuSMV 2, nUxmv, SPIN, SPOT

More scalable (~10^{30} states) but no optimality guarantees. Return a feasible, and not the optimal, solution.

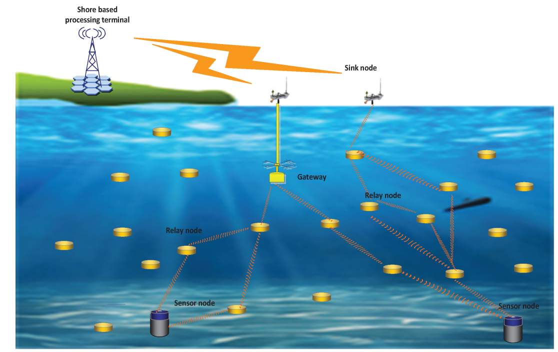
We propose an algorithm that can solve optimally hundreds of orders of magnitude larger planning problems than state-of-the-art methods (~10^{800} states and beyond).

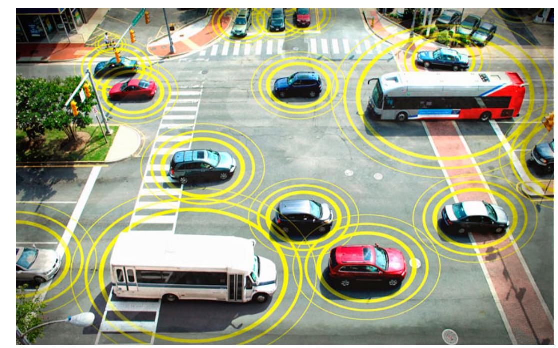
UF UNIVERSITY of FLORIDA · Duke UNIVERSITY · TEXAS The University of Texas at Austin · UC SANTA CRUZ

Open Problems

Distributed Control & Optimization for Networked Robots

- Reactive controllers for dynamic and uncertain environments
- Robust controllers (robot failures?)
- Realistic communication models (e.g., acoustic channels for underwater applications)
- Joint optimal task planning and communication control





Formal Methods/Control Synthesis for Cyber-Physical Systems

- Reactivity and learning
- Probabilistic control synthesis for large-scale networks
- Secure optimal control synthesis
- Trade-offs between security and optimality
- Model-free optimal control synthesis
- Human-in-the-loop control synthesis (e.g., human-robot collaborative tasks)

Thank You

Sampling-Based Optimal Control Synthesis

- Y. Kantaros and M. M. Zavlanos, "Sampling-Based Optimal Control Synthesis for Multi-Robot Systems under Global Temporal Tasks," IEEE Transactions on Automatic Control, 2018.
- Y. Kantaros, B. Johnson, S. Chowdhury, D. J. Cappelleri, and M. M. Zavlanos, "Control of Magnetic Microrobot Teams for Temporal Micromanipulation Tasks," IEEE Transactions on Robotics, 2018.

STyLuS*: large-Scale Temporal Logic optimal Synthesis

- Y. Kantaros and M. M. Zavlanos, "STyLuS*: A Temporal Logic Optimal Control Synthesis Algorithm for Large-Scale Multi-Robot Systems,'" International Journal of Robotics Research, under review.
- Y. Kantaros and M. M. Zavlanos, "Temporal Logic Optimal Control for Large-Scale Multi-Robot Systems: 10^{400} States and Beyond," 57th IEEE Conference on Decision and Control, 2018.

Distributed Sampling-Based Optimal Control Synthesis

- Y. Kantaros and M. M. Zavlanos, "Distributed Optimal Control Synthesis for Multi-Robot Systems under Global Temporal Tasks," 9th ACM/IEEE International Conference on Cyber- Physical Systems, 2018.