Optimal control with weighted average costs and temporal logic specifications

Eric M. Wolff, Ufuk Topcu, and Richard M. Murray

California Institute of Technology

July 12, 2012

Motivation

Goal:

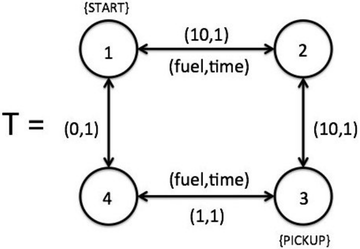
- Optimal control for an autonomous system doing a complex task

Challenges:

- Reasoning about how system properties change over time
- Specifying properties like safety, response, priority, liveness, and persistence
- Optimizing the system trajectory to conserve fuel or minimize time

Problem Description

Simple example:



- Task: repeatedly visit PICKUP

Given:

- System model: transition system $\mathcal{T}$ with costs and weights
- Task specification: linear temporal logic (LTL) formula φ

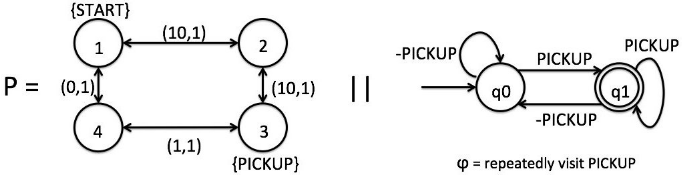
Problem: Minimize

$$J(\sigma) := \limsup_{n \to \infty} \frac{\sum_{i=0}^{n} c(\sigma_i, \sigma_{i+1})}{\sum_{i=0}^{n} w(\sigma_i, \sigma_{i+1})}$$

over all system trajectories σ that satisfy the LTL specification φ.

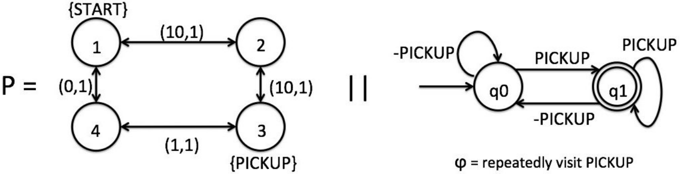
Solution overview

1. Construct the product automaton $\mathcal{P} = \mathcal{T} \times \mathcal{A}_{\varphi}$



Solution overview

1. Construct the product automaton $\mathcal{P} = \mathcal{T} \times \mathcal{A}_\varphi$

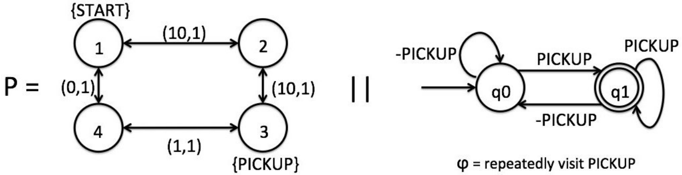


2. Show that problem is equivalent to finding an optimal cycle in $\mathcal{P}$

- $\sigma = \sigma_{\text{pre}}\sigma_{\text{suf}}^{\omega}$
- $J(\sigma_{\text{pre}}\sigma_{\text{suf}}^{\omega}) = J(\sigma_{\text{suf}}^{\omega})$
- σ_{suf} is a cycle including an accepting state in $\mathcal{P}$

Solution overview

1. Construct the product automaton $\mathcal{P} = \mathcal{T} \times \mathcal{A}_\varphi$



2. Show that problem is equivalent to finding an optimal cycle in $\mathcal{P}$

- $\sigma = \sigma_{\text{pre}}\sigma_{\text{suf}}^{\omega}$
- $J(\sigma_{\text{pre}}\sigma_{\text{suf}}^{\omega}) = J(\sigma_{\text{suf}}^{\omega})$
- σ_{suf} is a cycle including an accepting state in $\mathcal{P}$

3. Compute optimal cycle using dynamic programming

Example—autonomous driving

Task: repeatedly visit *a*, *b*, and *c* and avoid obstacles *x*

LTL spec: $\varphi = \Box\Diamond a \wedge \Box\Diamond b \wedge \Box\Diamond c \wedge \Box\neg x$

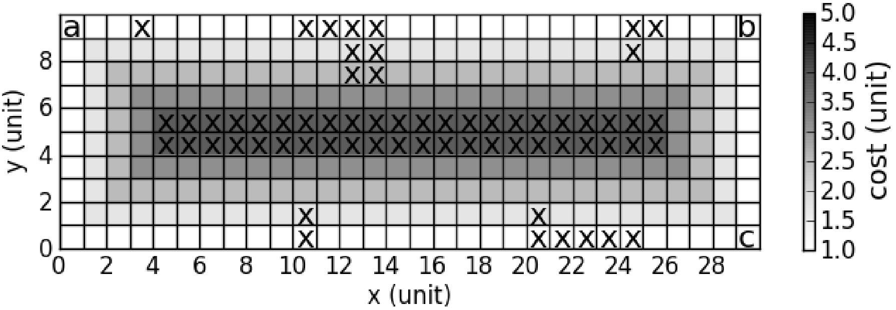


Figure: Driving task, with optimal run (blue) and feasible run (red).

Example—autonomous driving

Task: repeatedly visit a, b, and c and avoid obstacles x

LTL spec: $\varphi = \Box \Diamond a \wedge \Box \Diamond b \wedge \Box \Diamond c \wedge \Box \neg x$

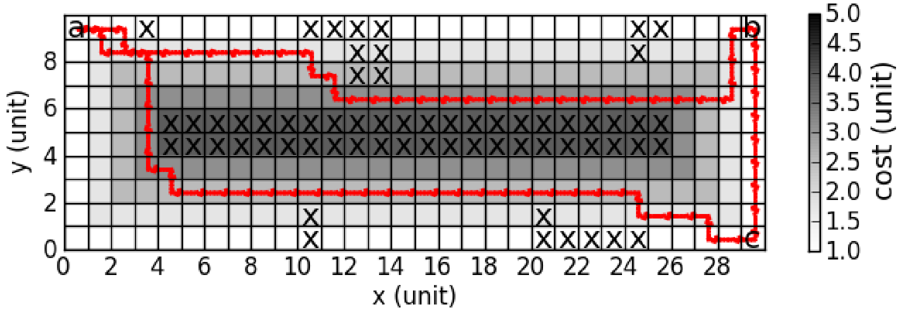


Figure: Driving task, with optimal run (blue) and feasible run (red).

Example—autonomous driving

Task: repeatedly visit a, b, and c and avoid obstacles x

LTL spec: $\varphi = \square \lozenge a \ \wedge \ \square \lozenge b \ \wedge \ \square \lozenge c \ \wedge \ \square \neg x$

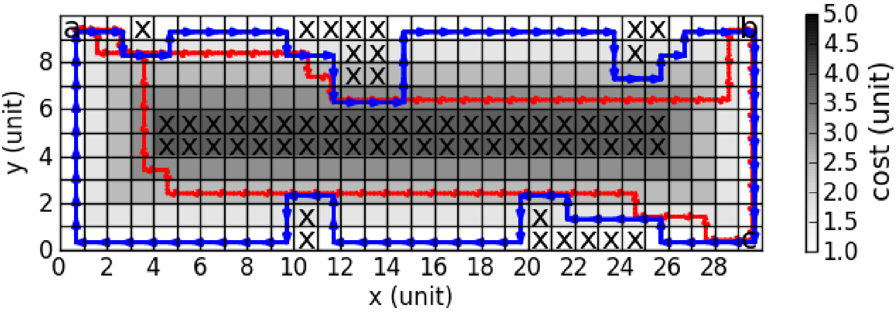


Figure: Driving task, with optimal run (blue) and feasible run (red).

Example—autonomous driving

Task: repeatedly visit a, b, and c and avoid obstacles x

LTL spec: $\varphi = \square\lozenge a \wedge \square\lozenge b \wedge \square\lozenge c \wedge \square\neg x$

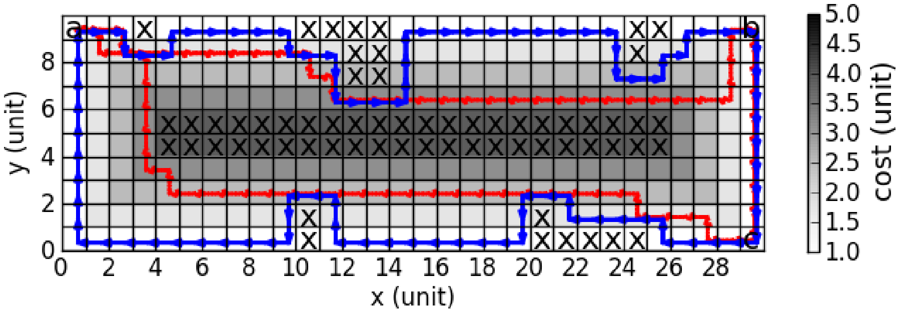


Figure: Driving task, with optimal run (blue) and feasible run (red).

Cost: $J_{\text{opt}} = 49$ and $J_{\text{feas}} = 71$ (units)

CPU time: $t_{\text{opt}} = 2.5$ and $t_{\text{feas}} = 0.68$ (sec)