Experimental design for expensive path planning simulators via integer programming

Yen-Chun Liu, Simon Mak
Department of Statistical Science
Duke University

2024 JRC at University of Waterloo

Supported by NSF CSSI Frameworks 2004571, NSF DMS 2210729, 2316012 and DE-SC0024477

What are path planning problems?

Duke

What are path planning problems?

Goal

*Finding an **optimal navigation path** that minimizes a cost function while meeting specified constraints*

e.g. warehouse robotics, rover navigation tasks

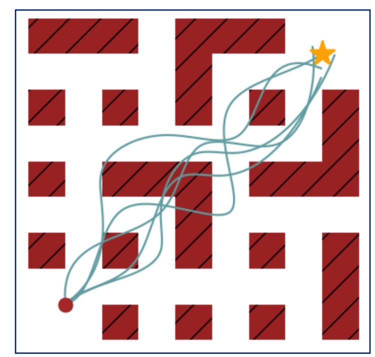
What are path planning problems?

Goal

*Finding an **optimal navigation path** that minimizes a cost function while meeting specified constraints*

e.g. warehouse robotics, rover navigation tasks

Amazon robots *(source: amazon.com)*



A rover navigation task (obstacles are swamps!) *(Wang et al. 2018)*

What are path planning problems?

Goal

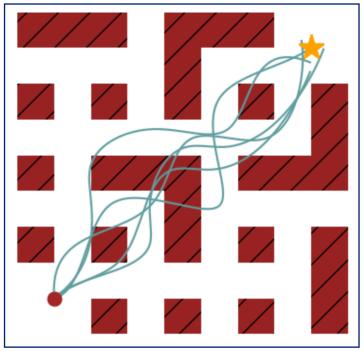
*Finding an **optimal navigation path** that minimizes a cost function while meeting specified constraints*

e.g. warehouse robotics, rover navigation tasks

Amazon robots *(source: amazon.com)*

Discrete path planning problems

- Action spaces are **discrete**
- More economical; aligns with many robotics tasks
- e.g. maze-solving problems (up, down, right, left)



A rover navigation task (obstacles are swamps!) *(Wang et al. 2018)*

Traditional methods

Traditional methods

- Dijkstra's algorithm and its variants *(Dijkstra, 1976)*
- Particle swarm optimization *(Kennedy and Eberhart, 1995)*
- Rapidly exploring random trees *(LaValle and Kuffner, 2001)*

Traditional methods

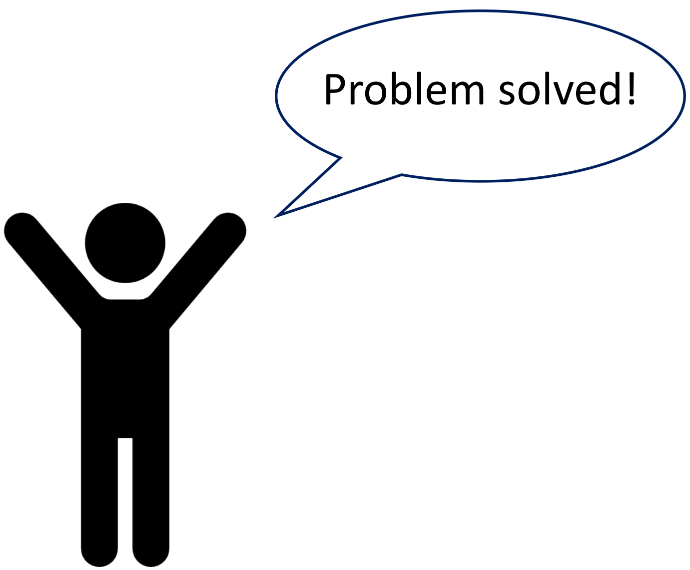
- Dijkstra’s algorithm and its variants *(Dijkstra, 1976)*
- Particle swarm optimization *(Kennedy and Eberhart, 1995)*
- Rapidly exploring random trees *(LaValle and Kuffner, 2001)*

… and so on

Traditional methods

- Dijkstra's algorithm and its variants *(Dijkstra, 1976)*
- Particle swarm optimization *(Kennedy and Eberhart, 1995)*
- Rapidly exploring random trees *(LaValle and Kuffner, 2001)*

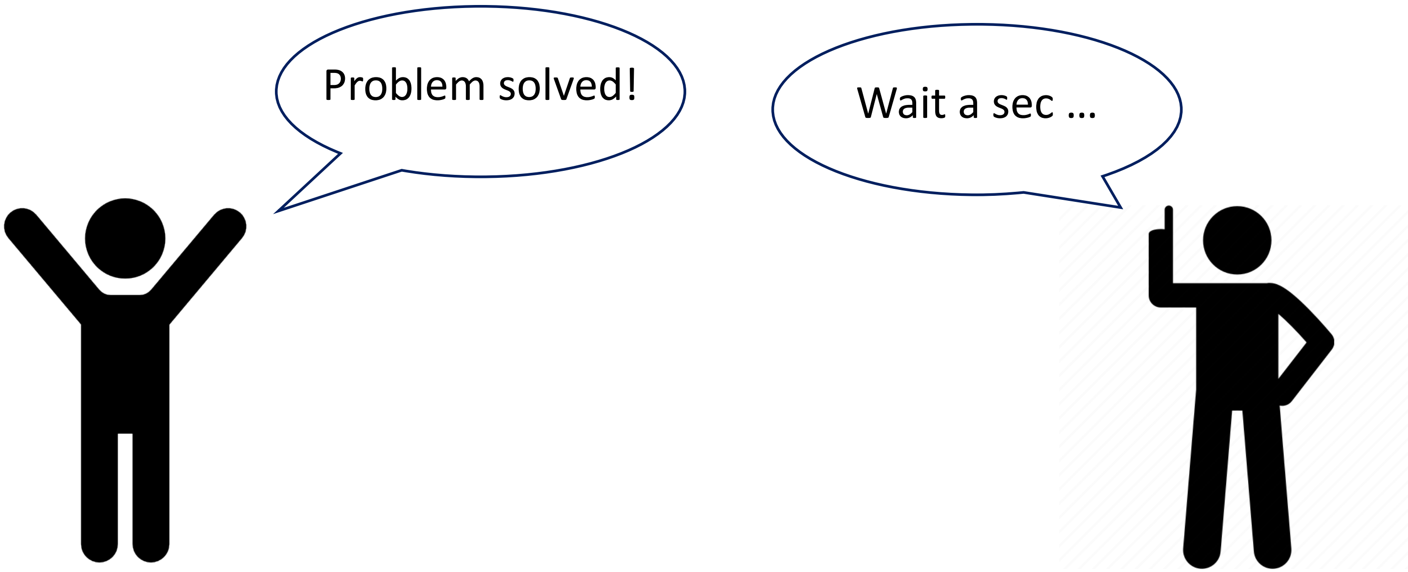
… and so on



Traditional methods

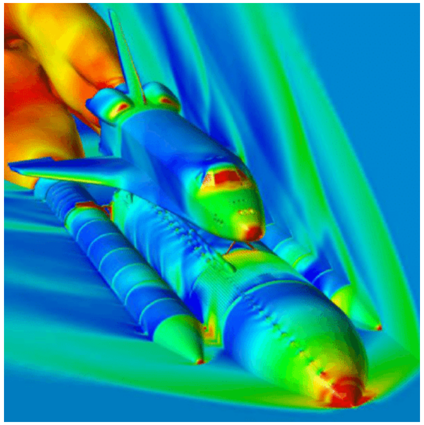
- Dijkstra's algorithm and its variants *(Dijkstra, 1976)*
- Particle swarm optimization *(Kennedy and Eberhart, 1995)*
- Rapidly exploring random trees *(LaValle and Kuffner, 2001)*

… and so on



Experiments can be expensive!

- **Path planning** is now needed for **highly sophisticated** systems, e.g., space rover, unmanned aerial vehicle, … etc.
- High-fidelity simulators can be **computationally expensive (days)**!



Rocket propulsion

unmanned aerial vehicle
(Narayanan et al. 2024)

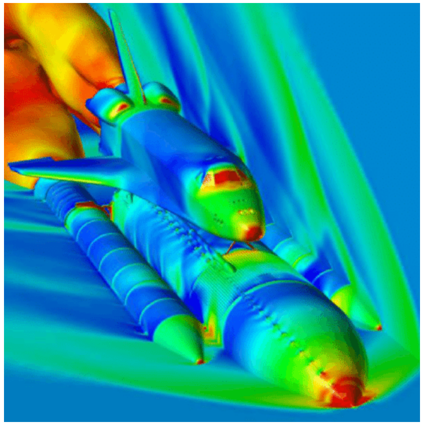
Duke

Experiments can be expensive!

- **Path planning** is now needed for **highly sophisticated** systems, e.g., space rover, unmanned aerial vehicle, … etc.
- High-fidelity simulators can be **computationally expensive (days)**!

We should be more careful when ***designing initial runs*** and ***selecting sequential runs***.

Especially when the problem is **high-dimensional!**



Rocket propulsion

unmanned aerial vehicle (Narayanan et al. 2024)

In this talk

Goals:

- Construct **optimal space-filling initial designs** via integer programming (IP)
- Propose simple, straightforward IP-based **sequential designs for prediction and optimization** that **avoids exhaustive search**

Action space: q descisions to make; m different choices of actions

$$\mathscr{A} = \{x; x \in \{a_1, \cdots, a_m\}^q\}$$

In this talk

Goals:

- Construct **optimal space-filling initial designs** via integer programming (IP)
- Propose simple, straightforward IP-based **sequential designs for prediction and optimization** that **avoids exhaustive search**

Action space: q descisions to make; m different choices of actions

$$\mathscr{A} = \{x; x \in \{a_1, \cdots, a_m\}^q\}$$

both q and m can be high-dimensional!

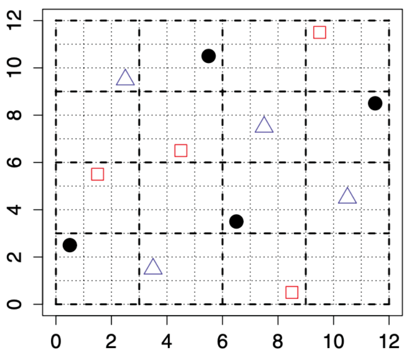
Literature Review

Popular experimental designs

- **Orthogonal Arrays (restricted run sizes)**
- **Latin hypercube designs**
  - Sliced-LHD (exchangeable algorithms) *(Ba et al., 2015)*
- **Maximin designs**

$$\max_{\mathcal{X}_n} \min_{1 \leq i < j \leq n} d(x_i, x_j)$$

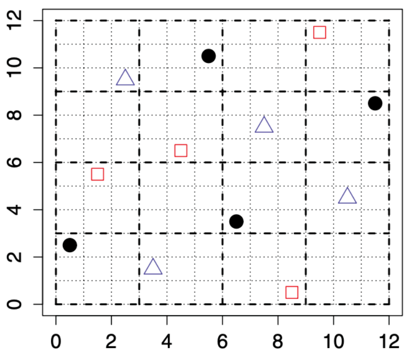
  - Metaheuristic algortihms *(Stokes et al., 2023)*



Example of sliced-LHD *(Ba et al., 2015)*

Popular experimental designs

- **Orthogonal Arrays (restricted run sizes)**
- **Latin hypercube designs**
  - Sliced-LHD (exchangeable algorithms) *(Ba et al., 2015)*



Example of sliced-LHD *(Ba et al., 2015)*

- **Maximin designs**

$$\max_{\mathcal{X}_n} \min_{1 \le i < j \le n} d(x_i, x_j)$$

  - Metaheuristic algortihms *(Stokes et al., 2023)*

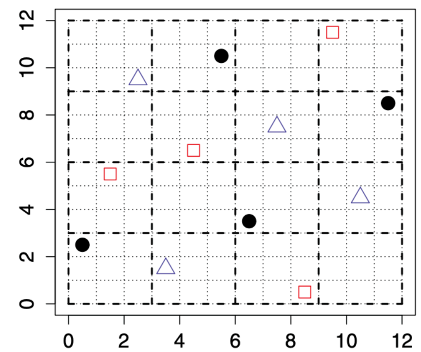
Simple, fast, but do not guarantee convergence to global optimum (especially high-dim!!)

Popular experimental designs

- **Orthogonal Arrays (restricted run sizes)**
- **Latin hypercube designs**
  - Sliced-LHD (exchangeable algorithms) *(Ba et al., 2015)*
- **Maximin designs**

$$\max_{\mathcal{X}_n} \min_{1 \leq i < j \leq n} d(x_i, x_j)$$

  - Metaheuristic algortihms *(Stokes et al., 2023)*



Example of sliced-LHD *(Ba et al., 2015)*

Simple, fast, but do not guarantee convergence to global optimum (especially high-dim!!)

Qualitative Gaussian Processes

Hamming-distance based kernel (Qian et al. 2008)

- Hamming distance

$$d(x_i, x_j) = \sum_{k=1}^{q} \mathbf{1}(x_{ik} \neq x_{jk})$$

- Exchangeable kernel

$$k_\theta\left(x_i, x_j\right) = \tau^2 \exp\left\{-\sum_{l=1}^{q} \theta_l \mathbf{1}(x_{il} \neq x_{jl})\right\}$$

Qualitative Gaussian Processes

Hamming-distance based kernel (Qian et al. 2008)

- Hamming distance

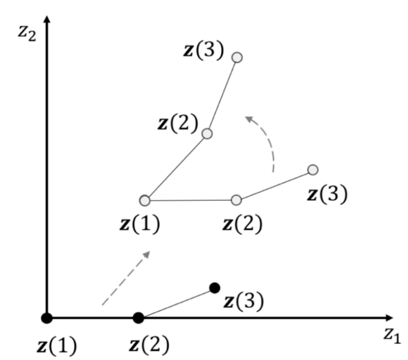
$$d(x_i, x_j) = \sum_{k=1}^{q} \mathbf{1}(x_{ik} \neq x_{jk})$$

- Exchangeable kernel

$$k_\theta\left(x_i, x_j\right) = \tau^2 \exp\left\{-\sum_{l=1}^{q} \theta_l \mathbf{1}(x_{il} \neq x_{jl})\right\}$$

Latent variable kernel (Zhang et al. 2020)

- **Idea**: mapping qualitative variables to ***latent continuous spaces*** and applying continuous kernels
- **Assumption**: there ***exists*** latent continuous spaces for qualitative variables
- **Applications**: material science, physics-based simulation



LVGP *(Zhang et al. 2020)*

Duke

Sequential design for qualitative variables

Exhaustive search

- **Pros**: ensured to achieve global optimum
- **Concerns**: infeasible in high-dim cases or when the decision space is large

Sequential design for qualitative variables

Exhaustive search

- **Pros**: ensured to achieve global optimum
- **Concerns**: infeasible in high-dim cases or when the decision space is large

Treating dummy variables as continuous variables

- **Pros**: straightforward, fast, easy to implement
- **Concerns**: dummy variables are not really quantitative variables; can be problematic

Sequential design for qualitative variables

Exhaustive search

- **Pros**: ensured to achieve global optimum
- **Concerns**: infeasible in high-dim cases or when the decision space is large

Treating dummy variables as continuous variables

- **Pros**: straightforward, fast, easy to implement
- **Concerns**: dummy variables are not really quantitative variables; can be problematic

Latent variable Bayesian optimization (Zhang et al. 2020b)

- **Pros**: flexible, efficient, nice UQ
- **Concerns**: can be slow in high-dim cases; need sufficient training samples, latent space assumption might not hold

Proposed methods

Experimental designs via IP

- **Convert inputs to dummy variables**
  e.g. Suppsoe action set = {a, b, c, d} and design = (b,c,a,d,a). Then

$$I(b,c,a,d,a) = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix}.$$

- Each I*(x)* must satisfy $\sum_{j=1}^{m} I(x)_{i,j} = 1$ for all *i*.
- Hamming distance between designs can be represented as

$$d(x_i, x_j) = q - \mathrm{tr}\{I(x_i)I(x_j)^\top\}\ .$$

Experimental designs via IP

Solving a **maximin design** with minimal Hamming distance $\geq C$ is the same as solving:

$$\max_{\mathcal{X}_n} 1, \quad \text{subject to} \quad q - \sum_{k=1}^{q} \left(I_i I_{i'}^\top\right)_{kk} \geq C, \quad \forall 1 \leq i < i' \leq n$$

$$I_i \mathbf{1}_m - \mathbf{1}_q = \mathbf{0}_q, \quad \forall 1 \leq i \leq n, \quad I_i \in \{0,1\}^q \times \{0,1\}^m \quad \forall 1 \leq i \leq n\,.$$

Experimental designs via IP

Solving a **maximin design** with minimal Hamming distance $\geq C$ is the same as solving:

$$\max_{\mathscr{X}_n} 1, \quad \text{subject to} \quad q - \sum_{k=1}^{q} \left(I_i I_{i'}^\top\right)_{kk} \geq C, \quad \forall 1 \leq i < i' \leq n$$

$$I_i \mathbf{1}_m - \mathbf{1}_q = \mathbf{0}_q, \quad \forall 1 \leq i \leq n, \quad I_i \in \{0,1\}^q \times \{0,1\}^m \quad \forall 1 \leq i \leq n\,.$$

- A **non-convex integer programming** (IP) problem

Experimental designs via IP

Solving a **maximin design** with minimal Hamming distance $\geq C$ is the same as solving:

$$\max_{\mathcal{X}_n} 1, \quad \text{subject to} \quad q - \sum_{k=1}^{q} \left(I_i I_{i'}^{\top}\right)_{kk} \geq C, \quad \forall 1 \leq i < i' \leq n$$

$$I_i \mathbf{1}_m - \mathbf{1}_q = \mathbf{0}_q, \quad \forall 1 \leq i \leq n, \quad I_i \in \{0,1\}^q \times \{0,1\}^m \quad \forall 1 \leq i \leq n\,.$$

- A **non-convex integer programming** (IP) problem
- That can be viewed as **a classic assignment problem (AP)** from operational research!!

Duke

Experimental designs via IP

Solving a **maximin design** with minimal Hamming distance $\geq C$ is the same as solving:

$$\max_{\mathscr{X}_n} 1, \quad \text{subject to} \quad q - \sum_{k=1}^{q} \left(I_i I_{i'}^\top\right)_{kk} \geq C, \quad \forall 1 \leq i < i' \leq n$$

$$I_i \mathbf{1}_m - \mathbf{1}_q = \mathbf{0}_q, \quad \forall 1 \leq i \leq n, \quad I_i \in \{0,1\}^q \times \{0,1\}^m \quad \forall 1 \leq i \leq n\,.$$

- A **non-convex integer programming** (IP) problem
- That can be viewed as **a classic assignment problem (AP)** from operational research!!

Advantages

1. Guaranteed **convergence to global optimum** (IP+AP)
2. Can be solved by **state-of-art IP optimizers** (Gurobi)
3. Providing **optimization gaps** (not available in heuristic algorithms)

How to select initial threshold *C*?

The Gilbert–Varshamov bound (Gilbert, 1952; Varshamov, 1957)

Let $N_m(q, s)$ denote the maximum size of path planning designs wtih action space (m, q) and minimum pairwise Hamming distance s. Then,

$$N_m(q, s) \geqslant \frac{m^q}{\sum_{j=0}^{s-1} \binom{q}{j} (q-1)^j}.$$

Proposition (Initial threshold C)

By Gilbert-Varshamov bound, a lower bound threshold $S(m, q, n)$ of DPP designs of size n and action space (m, q) can be estimated by

$$S(m, q, n) = \arg \max_{s \in \{1, 2, \cdots, q\}} \left\{ m^q / \sum_{j=0}^{s-1} \binom{q}{j} (q-1)^j \geq n \right\}.$$

Duke

Constructing initial designs $D(m, q, n)$

Use existing maximin design $D(m_0, q_0, n_0)$, where
$m_0 \leq m, q_0 \leq q, n_0 \leq n$ as warmstarts

optional

Duke

Constructing initial designs $D(m, q, n)$

Use existing maximin design $D(m_0, q_0, n_0)$, where $m_0 \leq m, q_0 \leq q, n_0 \leq n$ as warmstarts

↓ **optional**

Search for a design with minimal pairwise distance C = S(m,q,n)

Duke

Constructing initial designs $D(m, q, n)$

Use existing maximin design $D(m_0, q_0, n_0)$, where
$m_0 \leq m, q_0 \leq q, n_0 \leq n$ as warmstarts

optional

Search for a design with minimal pairwise distance C = S(m,q,n)

C = C+1

Search for a design with minimal pairwise distance = C

Constructing initial designs $D(m, q, n)$

Use existing maximin design $D(m_0, q_0, n_0)$, where $m_0 \leq m, q_0 \leq q, n_0 \leq n$ as warmstarts

optional

Search for a design with minimal pairwise distance C = S(m,q,n)

C = C+1

If design exists

Search for a design with minimal pairwise distance = C

Constructing initial designs $D(m, q, n)$

Use existing maximin design $D(m_0, q_0, n_0)$, where $m_0 \leq m, q_0 \leq q, n_0 \leq n$ as warmstarts

optional

Search for a design with minimal pairwise distance C = S(m,q,n)

C = C+1

Search for a design with minimal pairwise distance = C

If design exists

if design **not** exists or C+1 = q-1

Return design

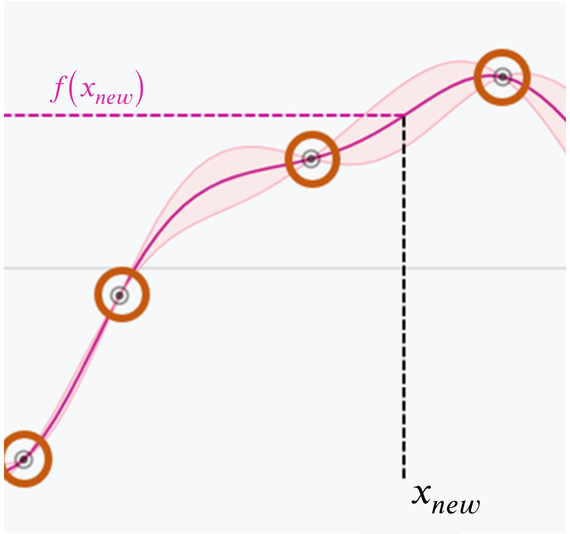
Sequential Prediction

Goal: optimizing prediction performance

Active learning Mackay (ALM; MacKay, 1992) for prediction

Explore the unknown!

$$x_{new} = \arg\max_{x} \sigma_n^2(x)$$



IP-based ALM for qualitative kernel

Let $\mathscr{X}_n = (X_1, X_2, \cdots, X_n)$ be the observed data and $\{\theta_k\}_{k=1}^q$ be the set of lengthscale parameters.

$$\max_x \sigma_n^2(x) = \max_x \tau^2\{K(x,x) - K(x,\mathscr{X}_n)K_{nn}^{-1}K(\mathscr{X}_n,x)\}$$
$$= \min_{I(x)} K(x,\mathscr{X}_n)K_{nn}^{-1}K(\mathscr{X}_n,x)$$
$$= \min_{I(x)} \sum_{i=1}^n \sum_{j=1}^n \exp\left[-\sum_{k=1}^q \{I(x)I(X_i)^\top\}_{kk}\,\theta_k\right](K_{nn}^{-1})_{ij}\exp\left[-\sum_{k=1}^q \left\{I(x)I(X_j)^\top\right\}_{kk}\theta_k\right]$$
$$\boxed{= \min_{I(x)} \sum_{i=1}^n \sum_{j=1}^n \exp\left(-\sum_{k=1}^q \left[I(x)\left\{I(X_i) + I(X_j)\right\}^\top\right]_{kk}\theta_k\right)(K_{nn}^{-1})_{ij}}$$

subject to

$$I(x)\mathbf{1}_m - \mathbf{1}_q = \mathbf{0}_q \text{ and } I(x) \in \{0,1\}^q \times \{0,1\}^m.$$

Duke

IP-based ALM for qualitative kernel

Let $\mathcal{X}_n = (X_1, X_2, \cdots, X_n)$ be the observed data and $\{\theta_k\}_{k=1}^q$ be the set of lengthscale parameters.

$$
\begin{aligned}
\max_x \sigma_n^2(x) &= \max_x \tau^2\{K(x,x) - K(x,\mathcal{X}_n)K_{nn}^{-1}K(\mathcal{X}_n,x)\} \\
&= \min_{I(x)} K(x,\mathcal{X}_n)K_{nn}^{-1}K(\mathcal{X}_n,x) \\
&= \min_{I(x)} \sum_{i=1}^n \sum_{j=1}^n \exp\left[-\sum_{k=1}^q \left\{I(x)I(X_i)^\top\right\}_{kk}\theta_k\right](K_{nn}^{-1})_{ij}\exp\left[-\sum_{k=1}^q \left\{I(x)I(X_j)^\top\right\}_{kk}\theta_k\right] \\
&\boxed{= \min_{I(x)} \sum_{i=1}^n \sum_{j=1}^n \exp\left(-\sum_{k=1}^q \left[I(x)\left\{I(X_i)+I(X_j)\right\}^\top\right]_{kk}\theta_k\right)(K_{nn}^{-1})_{ij}}
\end{aligned}
$$

subject to

$$
I(x)\mathbf{1}_m - \mathbf{1}_q = \mathbf{0}_q \text{ and } I(x) \in \{0,1\}^q \times \{0,1\}^m .
$$

A mixed integer programming problem (MIP)!

Able to achieve global optimum without exhaustive search!

Sequential Optimization

Goal: finding $x^* = \arg\max f(x)$

Upper confidence bound (UCB)

Exploration - exploitation

$$\max_{I(x)} \mu_n\{I(x)\} + \lambda \sigma_n\{I(x)\}$$

Posterior mean:
$$\mu_n\{I(x)\} = \hat{\mu}_n + \sum_{i=1}^{n} \sum_{j-1}^{n} \exp\left[-\sum_{k=1}^{q} \left\{ I(x) I(X_i)^\top \right\}_{kk} \theta_k \right] (K_{nn}^{-1})_{ij} (y_{n,j} - \hat{\mu}_n)$$

Posterior variance:
$$\sigma_n^2\{I(x)\} = \hat{\tau}^2 \left\{ 1 - \exp\left(-\sum_{k=1}^{q} \left[I(x) \left\{ I(X_i) + I(X_j) \right\}^\top \right]_{kk} \theta_k \right) (K_{nn}^{-1})_{ij} \right\}$$

Sequential Optimization

Goal: finding $x^* = \arg\max f(x)$

Upper confidence bound (UCB)

Exploration - exploitation

$$\max_{I(x)} \mu_n\{I(x)\} + \lambda \sigma_n\{I(x)\}$$

Posterior mean: $$\mu_n\{I(x)\} = \hat{\mu}_n + \sum_{i=1}^{n} \sum_{j-1}^{n} \exp\left[- \sum_{k=1}^{q} \left\{ I(x) I(X_i)^\top \right\}_{kk} \theta_k \right] (K_{nn}^{-1})_{ij} (y_{n,j} - \hat{\mu}_n)$$

Posterior variance: $$\sigma_n^2\{I(x)\} = \hat{\tau}^2 \left\{ 1 - \exp\left(- \sum_{k=1}^{q} \left[I(x) \left\{ I(X_i) + I(X_j) \right\}^\top \right]_{kk} \theta_k \right) (K_{nn}^{-1})_{ij} \right\}$$

A mixed integer programming problem (MIP)!

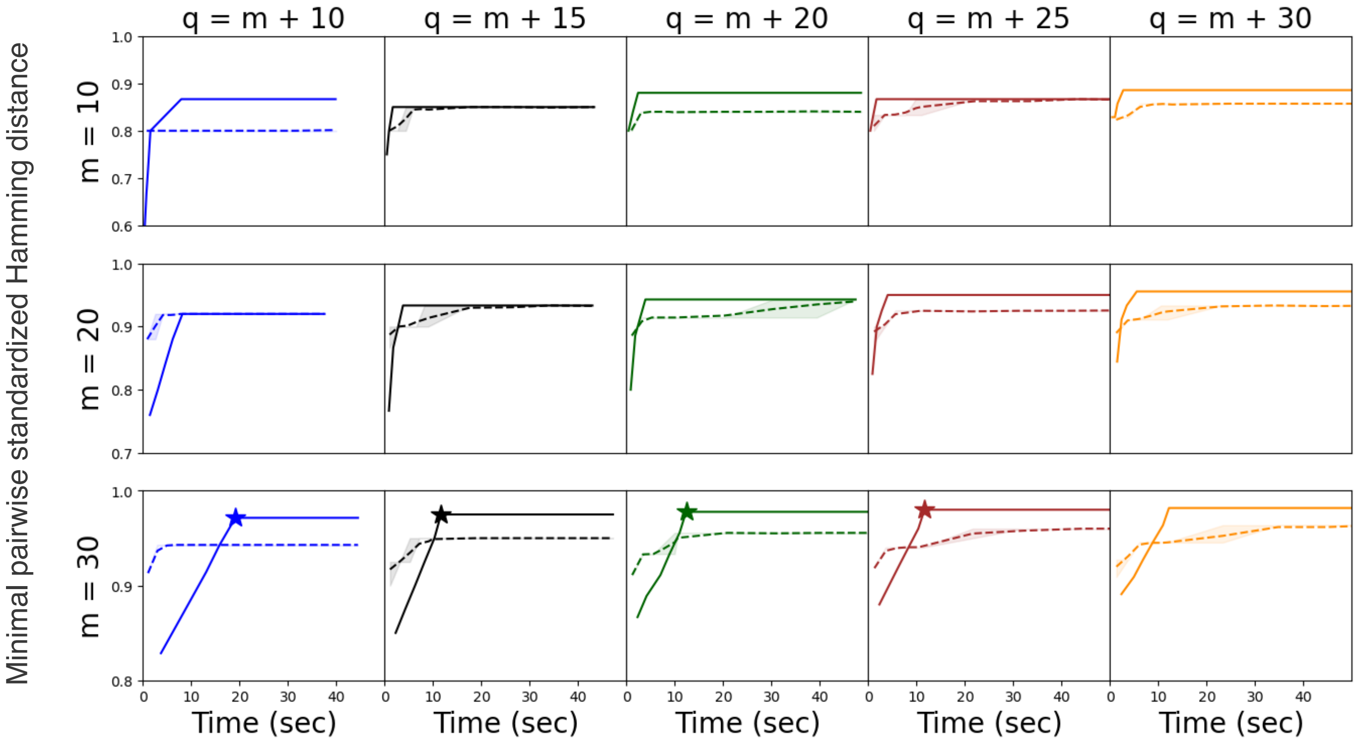
Able to achieve global optimum without exhaustive search!

Duke

Numerical Results

Maximin designs

Maximin distance achieved by **proposed IP optimizer (Gurobi)** v.s. **metaheuristics algorithms** (Stokes et al. ,2023) when design size is 50.



solid lines: IP optimizer (Gurobi)
dashed lines: metaheuristic algorithm (differential evolution) by Stokes et al. (2023)

Duke

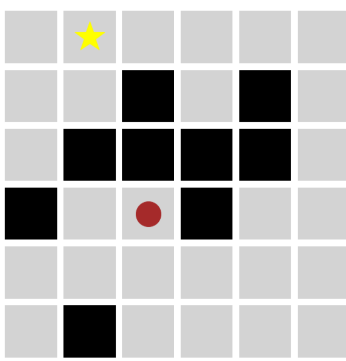
Simulation - Maze problems

Goal:

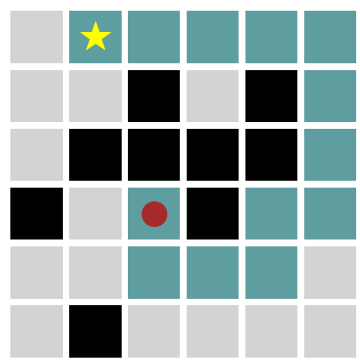
Predict the number of steps needed to reach the goal after taking *q* steps.

Action space $(m^q = 5^7, 5^{12})$:

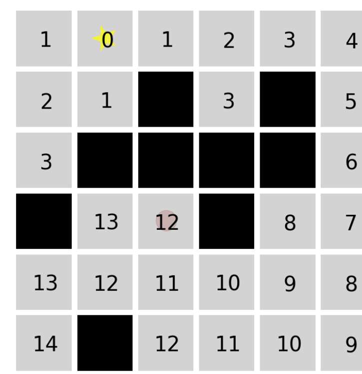
- Number of actions (m) = 5 (left, right, up, down, stay)
- Step lengths (q) = 7 or 12



A 6x6 maze

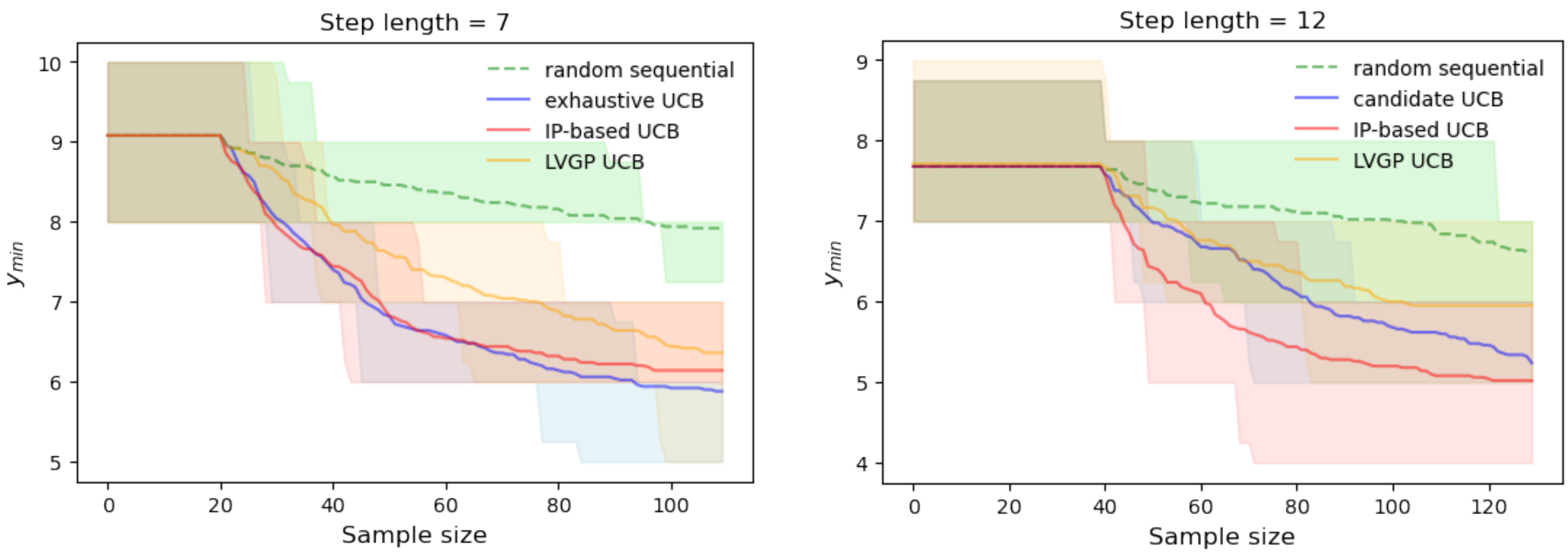


optimal path



Cost values at each grid

Sequential optimization



Maximin initial designs were obtained by IP optimizers

Sequential prediction

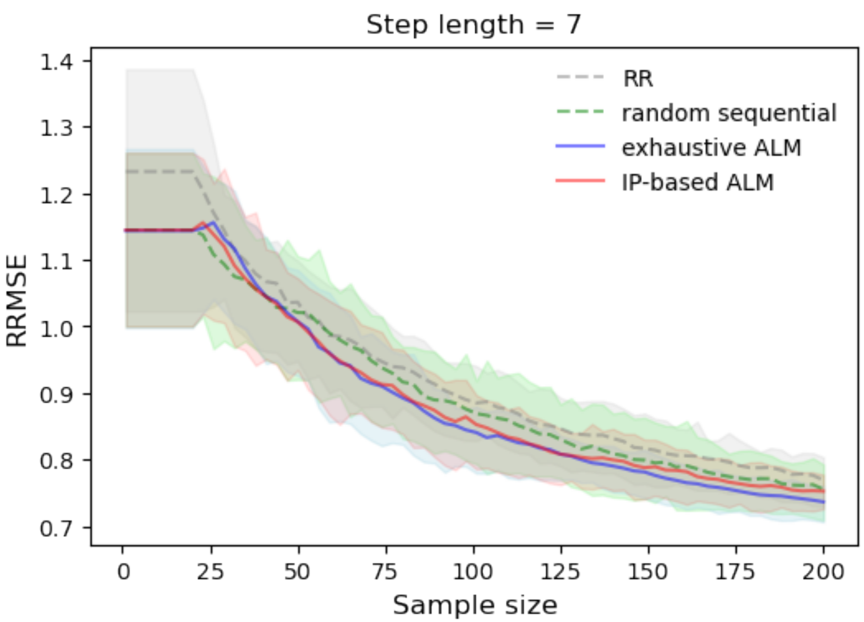
Evaluation criteria: RRMSE

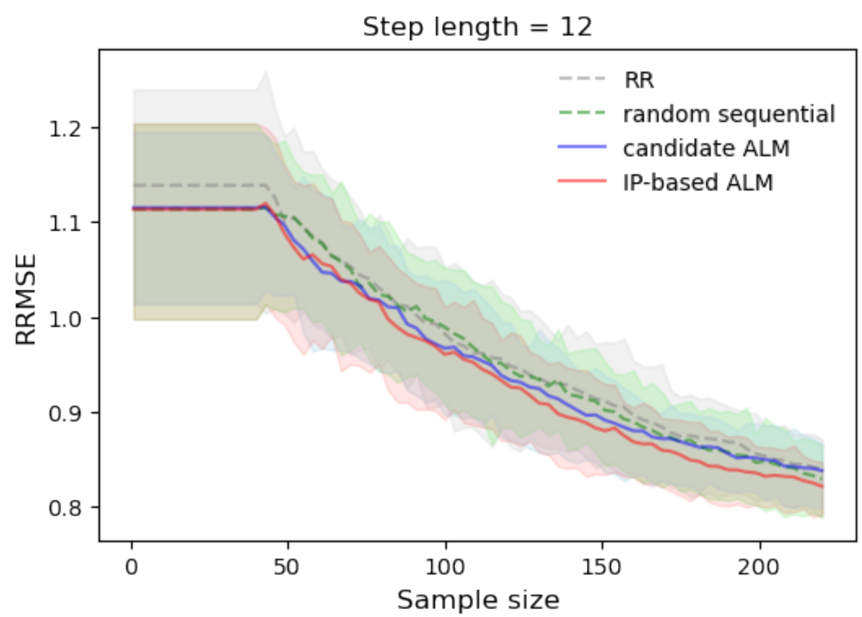
$$\mathrm{RRMSE} = \sqrt{\frac{N^{-1}\sum_{i=n+1}^{n+N}(\hat{y}_i - y_i)}{(n-1)^{-1}\sum_{i=1}^{n}(y_i - \bar{y}_n)^2}}\,.$$

Sequential prediction

Evaluation criteria: RRMSE

$$\text{RRMSE} = \sqrt{\frac{N^{-1}\sum_{i=n+1}^{n+N}(\hat{y}_i - y_i)}{(n-1)^{-1}\sum_{i=1}^{n}(y_i - \bar{y}_n)^2}}\,.$$





- RR: random initial design + random sequential design
- Others: use maximin initial design found by IP optimizer

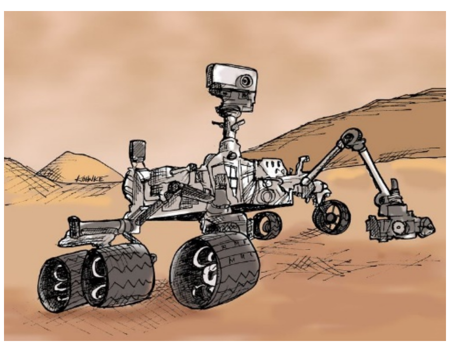
Rover navigation task

Goal:

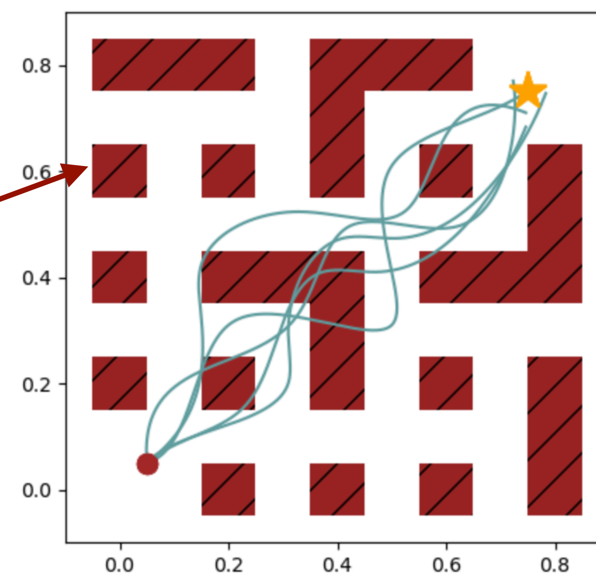
Reaching the goal in limited steps and minimizing the time trapped in the swamps

Action space $(m^q = 9^8)$:

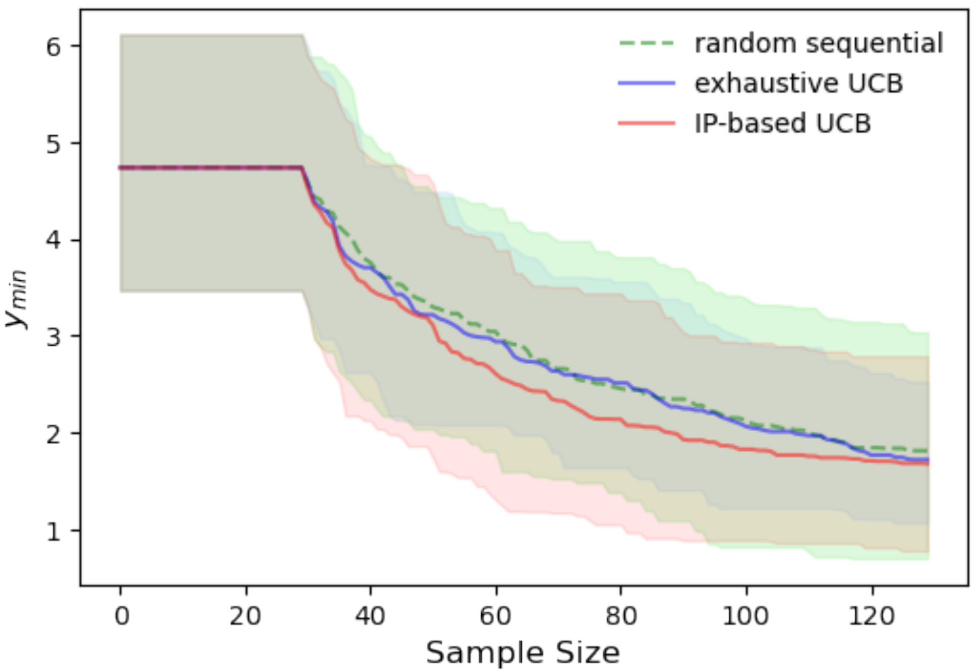
- Number of actions (m) = 9:
  4 angles $(0,\ \pi/6,\ \pi/4,\ \pi/2) \times$ 2 lengths $(0.1,\ 0.2)$ + 1 (stay)
- Step lengths (q) = 8

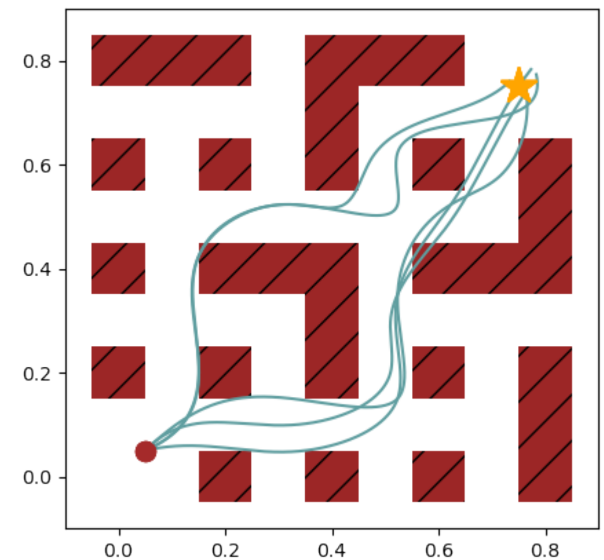


Swamps
(can pass through with greatly reduced speed)



Sequential optimization





Optimal paths found by IP-based UCB

Thank you!

Any questions?