

Discovering and Learning Preferred Operators for Classical Planning with Neural Networks

Pedro Probst Minini

Orientador: Marcus Ritt

Coorientador: André G. Pereira

Instituto de Informática — UFRGS

3 de julho de 2023

Agenda

Problema

Introdução

- Planejamento

- Busca Heurística

- Operadores Preferidos

- Aprendizado de Funções Heurísticas

Abordagem Proposta

- Operadores Preferidos Ideais

- Operadores Preferidos Descobertos

- Gerando Amostras com XRS

Experimentos

- Preliminares

- Aprendendo Operadores Preferidos

- Comparando XRS e FSM

- Operadores Preferidos com Funções Heurísticas Lógicas

Conclusão

Problema

O Que Queremos Fazer?

Planejadores que usam operadores preferidos para **priorizar a expansão de estados mais promissores** foram os vencedores da trilha *satisficing* da *International Planning Competition* (IPC) em **2004** (Helmert 2006), **2008** (Richter et al. 2010), **2011** (Richter et al. 2011), and **2018** (Seipp et al. 2018).

O Que Queremos Fazer?

Planejadores que usam operadores preferidos para **priorizar a expansão de estados mais promissores** foram os vencedores da trilha *satisficing* da *International Planning Competition* (IPC) em **2004** (Helmert 2006), **2008** (Richter et al. 2010), **2011** (Richter et al. 2011), and **2018** (Seipp et al. 2018).

- ▶ Abordagens existentes para a derivação de operadores preferidos são baseadas em lógica (e.g. operadores preferidos do FF).

O Que Queremos Fazer?

Planejadores que usam operadores preferidos para **priorizar a expansão de estados mais promissores** foram os vencedores da trilha *satisficing* da *International Planning Competition* (IPC) em **2004** (Helmert 2006), **2008** (Richter et al. 2010), **2011** (Richter et al. 2011), and **2018** (Seipp et al. 2018).

- ▶ Abordagens existentes para a derivação de operadores preferidos são baseadas em lógica (e.g. operadores preferidos do FF).
- ▶ Há vários métodos para o **aprendizado de funções heurísticas** usando redes neurais.
 - ▶ Mas nada sobre aprender operadores preferidos.

O Que Queremos Fazer?

- ▶ Propomos um método para gerar amostras de estados, onde cada estado tem um conjunto de operadores preferidos.

O Que Queremos Fazer?

- ▶ Propomos um método para gerar amostras de estados, onde **cada estado tem um conjunto de operadores preferidos**.
- ▶ Usamos esses conjuntos de amostras gerados para **treinar redes neurais** que, durante a resolução de um problema de planejamento, predizem operadores preferidos para o estado atual.

Introdução

Introdução

Problema

Introdução

Planejamento

Busca Heurística

Operadores Preferidos

Aprendizado de Funções Heurísticas

Abordagem Proposta

Operadores Preferidos Ideais

Operadores Preferidos Descobertos

Gerando Amostras com XRS

Experimentos

Preliminares

Aprendendo Operadores Preferidos

Comparando XRS e FSM

Operadores Preferidos com Funções Heurísticas Lógicas

Conclusão

Planejamento Clássico

Planejamento tem o objetivo de encontrar uma **sequência de ações** a partir de um **estado inicial** que satisfaça as **condições objetivo**.

Exemplo: Planejamento Clássico

ambiente

- ▶ **estático** vs. dinâmico
- ▶ **determinístico** vs. não determinístico vs. estocástico
- ▶ **observável** vs. parcialmente observável vs. não observável
- ▶ **discreto** vs. contínuo
- ▶ **agente único** vs. agentes múltiplos

STRIPS

Definição (Um problema de planejamento em STRIPS)

$\Pi = \langle F, O, s_0, s^* \rangle$ onde

- ▶ F é um conjunto de variáveis booleanas (*fat*os ou *proposições*),
- ▶ O é um conjunto de *operadores* ou ações sobre F , onde $\langle \text{pre}(o), \text{add}(o), \text{del}(o) \rangle \subseteq F$,
- ▶ $s_0 \subseteq F$ é o conjunto de fatos que representa o *estado inicial*, e
- ▶ $s^* \subseteq F$ é o conjunto de fatos que devem ser satisfeitos ("*gol*").

Definição (Um problema de planejamento em STRIPS)

$\Pi = \langle F, O, s_0, s^* \rangle$ onde

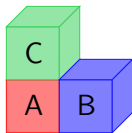
- ▶ F é um conjunto de variáveis booleanas (*fatos* ou *proposições*),
- ▶ O é um conjunto de *operadores* ou ações sobre F , onde $\langle \text{pre}(o), \text{add}(o), \text{del}(o) \rangle \subseteq F$,
- ▶ $s_0 \subseteq F$ é o conjunto de fatos que representa o *estado inicial* , e
- ▶ $s^* \subseteq F$ é o conjunto de fatos que devem ser satisfeitos (" *gol* ").

As ações $O(s)$ são *aplicáveis* em s se elas satisfizerem $\text{pre}(o)$.

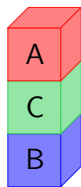
Avançamos de um estado s com o operador o definindo os add-effects como *verdadeiros* e os del-effects como *falsos* .

Exemplo: Blocks World

Figura: Configuração inicial e objetivo de uma tarefa Blocks World.



(Estado inicial)



(Estado final)

Exemplo: Blocks World

Exemplo: Blocks World

$$\Pi = \langle F, O, s_0, s^* \rangle$$

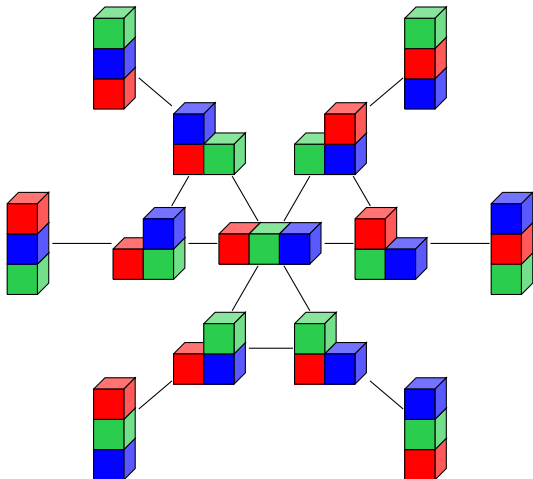
- ▶ $F = \{\text{on}(a,b), \text{on}(a,c), \text{on}(b,a), \text{on}(b,c), \text{on}(c,a), \text{on}(c,b), \text{on-table}(a), \text{on-table}(b), \text{on-table}(c), \text{clear}(a), \text{clear}(b), \text{clear}(c)\}$
- ▶ $O = \{\text{move}(a,b,c), \text{move}(a,c,b), \text{move}(b,a,c), \text{move}(b,c,a), \text{move}(c,a,b), \text{move}(c,b,a), \text{to-table}(a,b), \text{to-table}(a,c), \text{to-table}(b,a), \text{to-table}(b,c), \text{to-table}(c,a), \text{to-table}(c,b), \text{from-table}(a,b), \text{from-table}(a,c), \text{from-table}(b,a), \text{from-table}(b,c), \text{from-table}(c,a), \text{from-table}(c,b)\}$
- ▶ $s_0 = \{\text{on}(c,a), \text{on-table}(a), \text{on-table}(b), \text{clear}(c), \text{clear}(b)\}$
- ▶ $s^* = \{\text{on}(c,b), \text{on}(a,c)\}$

Exemplo: Blocks World

Exemplo: Blocks World

- ▶ $move(x,y,z)$: mover bloco x sobre y para cima de z.
 - ▶ $Pre(move(a,b,c)) = \{on(a,b), clear(a), clear(c)\}$
 - ▶ $Add(move(a,b,c)) = \{on(a,c), clear(b)\}$
 - ▶ $Del(move(a,b,c)) = \{on(a,b), clear(c)\}$
- ▶ $to-table(x,y)$: mover bloco x sobre y para a mesa.
- ▶ $from-table(x,y)$: mover bloco x da mesa para cima de y.

Espaço de Estados do Blocks World com 3 Blocos



Introdução

Problema

Introdução

- Planejamento

- Busca Heurística**

- Operadores Preferidos

- Aprendizado de Funções Heurísticas

Abordagem Proposta

- Operadores Preferidos Ideais

- Operadores Preferidos Descobertos

- Gerando Amostras com XRS

Experimentos

- Preliminares

- Aprendendo Operadores Preferidos

- Comparando XRS e FSM

- Operadores Preferidos com Funções Heurísticas Lógicas

Conclusão

Busca Heurística

- ▶ Num problema de planejamento, um estado s tem uma função heurística $h(s)$, dado por uma função heurística h , que estima o custo de atingir o gol s^* a partir de s .
 - ▶ A heurística perfeita $h^*(s)$ é a distância real de s até s^* .

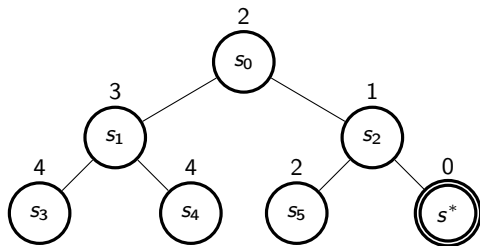
Busca Heurística

- ▶ Num problema de planejamento, um estado s tem uma função heurística $h(s)$, dado por uma função heurística h , que estima o custo de atingir o gol s^* a partir de s .
 - ▶ A heurística perfeita $h^*(s)$ é a distância real de s até s^* .
- ▶ Planejadores tipicamente buscam satisfazer o gol priorizando estados no espaço de estados com menor função heurística.
 - ▶ Idealmente tentando preservar o ranking de expansões de h^* .

Busca Heurística

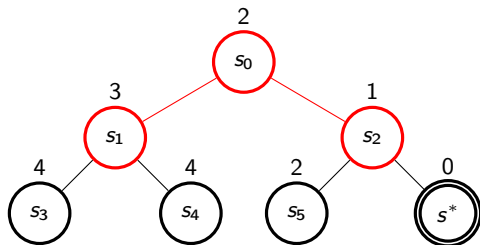
- ▶ Num problema de planejamento, um estado s tem uma função heurística $h(s)$, dado por uma função heurística h , que estima o **custo de atingir o gol s^*** a partir de s .
 - ▶ A heurística perfeita $h^*(s)$ é a **distância real** de s até s^* .
- ▶ Planejadores tipicamente buscam satisfazer o gol **priorizando** estados no espaço de estados com **menor função heurística**.
 - ▶ Idealmente tentando preservar o ranking de expansões de h^* .
- ▶ Algoritmos “best-first search”, e.g., greedy best-first search (GBFS), A^* .

Busca Heurística: Exemplo com GBFS



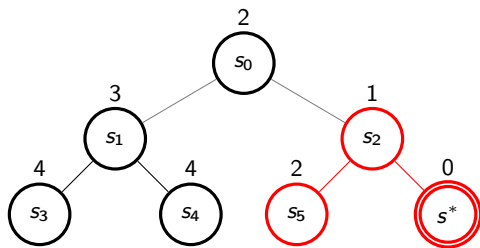
Iteração	Fila de Prioridades
0	$(s_0, 2)$

Busca Heurística: Exemplo com GBFS



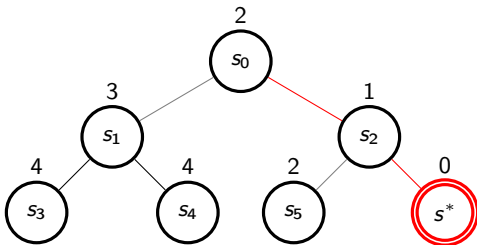
Iteração	Fila de Prioridades
0	(s₀, 2)
1	(s ₂ , 1) (s ₁ , 3)

Busca Heurística: Exemplo com GBFS



Iteração	Fila de Prioridades
0	$(s_0, 2)$
1	$(s_2, 1)$ $(s_1, 3)$
2	$(s^*, 0)$ $(s_5, 2)$ $(s_1, 3)$

Busca Heurística: Exemplo com GBFS



Iteração	Fila de Prioridades
0	$(s_0, 2)$
1	$(s_2, 1)$ $(s_1, 3)$
2	$(s^*, 0)$ $(s_5, 2)$ $(s_1, 3)$
3	$(s_5, 2)$ $(s_1, 3)$

Introdução

Problema

Introdução

- Planejamento

- Busca Heurística

- Operadores Preferidos**

- Aprendizado de Funções Heurísticas

Abordagem Proposta

- Operadores Preferidos Ideais

- Operadores Preferidos Descobertos

- Gerando Amostras com XRS

Experimentos

- Preliminares

- Aprendendo Operadores Preferidos

- Comparando XRS e FSM

- Operadores Preferidos com Funções Heurísticas Lógicas

Conclusão

Operadores Preferidos (POs)

- ▶ Operadores considerados promissores “por alguma razão”.

Operadores Preferidos (POs)

- ▶ Operadores considerados promissores “por alguma razão”.
 - ▶ A definição não é fechada, mas tipicamente isso quer dizer “operadores que levam a algum estado mais provável de satisfazer o gol” (Helmert 2006; Richter et al. 2009).

Operadores Preferidos (POs)

- ▶ Operadores considerados promissores “por alguma razão”.
 - ▶ A definição não é fechada, mas tipicamente isso quer dizer “operadores que levam a algum estado mais provável de satisfazer o gol” (Helmert 2006; Richter et al. 2009).
- ▶ Usados em conjunto com funções heurísticas.
 - ▶ Sozinhos, são equivalentes a políticas (policies).

Operadores Preferidos (POs)

- ▶ Operadores considerados promissores “por alguma razão”.
 - ▶ A definição não é fechada, mas tipicamente isso quer dizer “operadores que levam a algum estado mais provável de satisfazer o gol” (Helmert 2006; Richter et al. 2009).
- ▶ Usados em conjunto com funções heurísticas.
 - ▶ Sozinhos, são equivalentes a políticas (policies).
- ▶ No Fast Downward: usados em esquema “dual-queue”.
 - ▶ Uma fila para todos os estados gerados, outra para estados gerados exclusivamente via POs.
 - ▶ Expansão alternada, ou não (boosting).

Operadores Preferidos (POs)

- ▶ Com um valor de **boosting n** , quando progredimos a busca (encontramos um estado com valor- h menor do que todos os outros), então a fila de estados gerados por operadores preferidos é usada nas **próximas n expansões**.
- ▶ Boosting é **cumulativo**.

Operadores Preferidos do FF (Hoffmann et al. 2001)

- ▶ Para um estado s , faz um grafo da tarefa de planejamento **relaxada** (ignorando del-effects) e calcula a solução dessa tarefa.

Operadores Preferidos do FF (Hoffmann et al. 2001)

- ▶ Para um estado s , faz um grafo da tarefa de planejamento **relaxada** (ignorando del-effects) e calcula a solução dessa tarefa.
- ▶ Os operadores preferidos do estado s é dado por

$$O_{\text{pref}}(s) = \{o \mid \text{pre}(o) \subseteq s, \text{add}(o) \cap S_1^* \neq \emptyset\}$$

i.e., **operadores aplicáveis** em s que **adicionam pelo menos um fato do gol** no primeiro nível da solução relaxada.

Operadores Preferidos do FF (Hoffmann et al. 2001)

- ▶ Para um estado s , faz um grafo da tarefa de planejamento **relaxada** (ignorando del-effects) e calcula a solução dessa tarefa.
- ▶ Os operadores preferidos do estado s é dado por

$$O_{\text{pref}}(s) = \{o \mid \text{pre}(o) \subseteq s, \text{add}(o) \cap S_1^* \neq \emptyset\}$$

i.e., **operadores aplicáveis** em s que **adicionam pelo menos um fato do gol** no primeiro nível da solução relaxada.

- ▶ Precondições de ações que adicionam um fato do gol *também* são adicionadas em S^* como “sub-gols”.

Operadores Preferidos do FF (Hoffmann et al. 2001)

		on(c,b)		
		clear(a)		
on(c,a)		on-table(a)		
on-table(a)	move(c,a,b)	on-table(b)		
on-table(b)	to-table(c,a)	clear(c)
clear(c)	from-table(b,c)	on-table(c)		
clear(b)		on(b,c)		
		clear(b)		
		on(c,a)		
P_0 (s_0)	A_1	P_1	A_2	P_2



s_0



s^*

Operadores Preferidos do FF (Hoffmann et al. 2001)

		on(c,b)		
		clear(a)		
on(c,a)		on-table(a)		
on-table(a)	move(c,a,b)	on-table(b)		
on-table(b)	to-table(c,a)	clear(c)
clear(c)	from-table(b,c)	on-table(c)		
clear(b)		on(b,c)		
		clear(b)		
		on(c,a)		

$P_0 (s_0)$

A_1

P_1

A_2

P_2

$$O_{\text{pref}}(s_0) = \{\text{move}(c,a,b)\}$$



Introdução

Problema

Introdução

- Planejamento

- Busca Heurística

- Operadores Preferidos

- Aprendizado de Funções Heurísticas**

Abordagem Proposta

- Operadores Preferidos Ideais

- Operadores Preferidos Descobertos

- Gerando Amostras com XRS

Experimentos

- Preliminares

- Aprendendo Operadores Preferidos

- Comparando XRS e FSM

- Operadores Preferidos com Funções Heurísticas Lógicas

Conclusão

Aprendizado de Funções Heurísticas

Em particular, este trabalho segue uma **abordagem supervisionada** para o aprendizado de funções heurísticas e operadores preferidos.

Aprendizado de Funções Heurísticas

Em particular, este trabalho segue uma **abordagem supervisionada** para o aprendizado de funções heurísticas e operadores preferidos.

- ▶ Uso de lógica do domínio apenas para a derivação de mutexes e operadores aplicáveis durante o processo de amostragem.

Aprendizado de Funções Heurísticas

Em particular, este trabalho segue uma **abordagem supervisionada** para o aprendizado de funções heurísticas e operadores preferidos.

- ▶ Uso de lógica do domínio apenas para a derivação de mutexes e operadores aplicáveis durante o processo de amostragem.
- ▶ Rede neural simples, e.g., feedforward NN.

Aprendizado de Funções Heurísticas

Em particular, este trabalho segue uma **abordagem supervisionada** para o aprendizado de funções heurísticas e operadores preferidos.

- ▶ Uso de lógica do domínio apenas para a derivação de mutexes e operadores aplicáveis durante o processo de amostragem.
- ▶ Rede neural simples, e.g., feedforward NN.
- ▶ e.g. Ferber et al. 2020; Yu et al. 2020; Ferber et al. 2022; O'Toole et al. 2022; Bettker et al. 2022

Aprendizado de Funções Heurísticas

Em particular, este trabalho segue uma **abordagem supervisionada** para o aprendizado de funções heurísticas e operadores preferidos.

- ▶ Uso de lógica do domínio apenas para a derivação de mutexes e operadores aplicáveis durante o processo de amostragem.
- ▶ Rede neural simples, e.g., feedforward NN.
- ▶ e.g. Ferber et al. 2020; Yu et al. 2020; Ferber et al. 2022; O'Toole et al. 2022; Bettker et al. 2022
- ▶ Usamos a estratégia de **Bettker et al. 2022**.

Gerando Amostras com FSM de Bettker et al. 2022

- ▶ A partir do gol, faz **regressão** usando **BFS** (breadth-first search) até conseguir 10 % do número de amostras N .

Gerando Amostras com FSM de Bettker et al. 2022

- ▶ A partir do gol, faz **regressão** usando **BFS** (breadth-first search) até conseguir 10 % do número de amostras N .
- ▶ Obtém o restante das amostras fazendo rollouts de **Random Walk** a partir dos **estados-folha** do BFS.

Melhorando as Estimativas das Amostras Vizinhas

▶ *Successor Improvement (SUI)*:

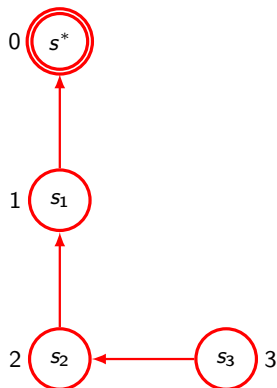
- ▶ Múltiplos rollouts podem gerar **estados vizinhos**.
 - ▶ Podemos usar essa informação para melhorar os valores- h .
- ▶ Faz um **grafo** com todas as relações de predecessores e sucessores do conjunto de amostras.
- ▶ Atualiza os valores- h dos estados a **um operador de distância**.

$$h(s) = \min(h(s), h(t) + 1)$$

Melhorando as Estimativas das Amostras Vizinhas

► Rollout #1

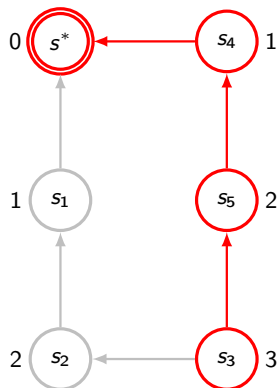
- $\langle s^*, 0 \rangle$
- $\langle s_1, 1 \rangle$
- $\langle s_2, 2 \rangle$
- $\langle s_3, 3 \rangle$



Melhorando as Estimativas das Amostras Vizinhas

► Rollout #2

- $\langle s^*, 0 \rangle$
- $\langle s_4, 1 \rangle$
- $\langle s_5, 2 \rangle$
- $\langle s_3, 3 \rangle$



Melhorando as Estimativas das Amostras Vizinhas

► Rollout #1

► $\langle s^*, 0 \rangle$

► $\langle s_1, 1 \rangle$

► $\langle s_2, 2 \rangle$

► $\langle s_3, 3 \rangle$

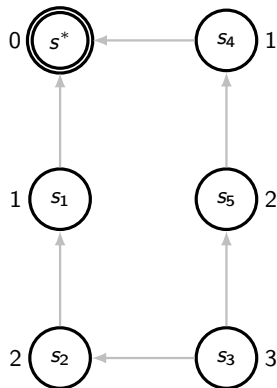
► Rollout #2

► $\langle s^*, 0 \rangle$

► $\langle s_4, 1 \rangle$

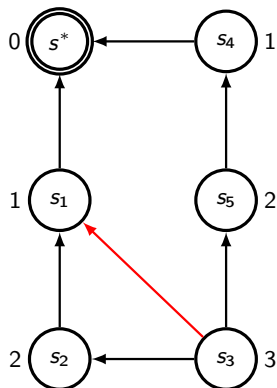
► $\langle s_5, 2 \rangle$

► $\langle s_3, 3 \rangle$



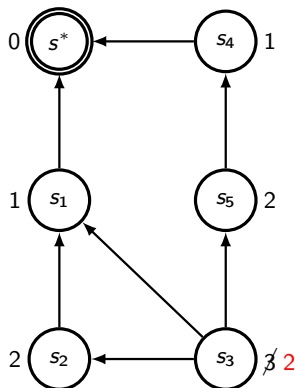
Melhorando as Estimativas das Amostras Vizinhas

- (1) Faz um grafo tal que $G = (V, A)$,
i.e., $V = \{s_i \mid i \in [N]\}$.
- (2) Para todo par $s, t \in V$,
adiciona um arco (s, t) em A
caso $\text{succ}(s, o) \subseteq t$.



Melhorando as Estimativas das Amostras Vizinhas

- (3) Atualiza as estimativas dos estados de acordo com os sucessores.



Abordagem Proposta

Como Aprender Operadores Preferidos?

- ▶ Ainda que POs sejam úteis, literatura sobre POs no geral é escassa, e sobre aprendizado de POs, inexistente.

Como Aprender Operadores Preferidos?

- ▶ Ainda que POs sejam úteis, literatura sobre POs no geral é escassa, e sobre aprendizado de POs, inexistente.
- ▶ A seguir, damos uma definição de **operadores preferidos ideais**,

Como Aprender Operadores Preferidos?

- ▶ Ainda que POs sejam úteis, literatura sobre POs no geral é escassa, e sobre aprendizado de POs, inexistente.
- ▶ A seguir, damos uma definição de **operadores preferidos ideais**,
- ▶ mostramos **como obter operadores preferidos** a partir de um conjunto de amostras, aproximando POs ideais,

Como Aprender Operadores Preferidos?

- ▶ Ainda que POs sejam úteis, literatura sobre POs no geral é escassa, e sobre aprendizado de POs, inexistente.
- ▶ A seguir, damos uma definição de **operadores preferidos ideais**,
- ▶ mostramos **como obter operadores preferidos** a partir de um conjunto de amostras, aproximando POs ideais,
- ▶ e propomos um **novo método de amostragem**.

Abordagem Proposta

Problema

Introdução

- Planejamento

- Busca Heurística

- Operadores Preferidos

- Aprendizado de Funções Heurísticas

Abordagem Proposta

- Operadores Preferidos Ideais

- Operadores Preferidos Descobertos

- Gerando Amostras com XRS

Experimentos

- Preliminares

- Aprendendo Operadores Preferidos

- Comparando XRS e FSM

- Operadores Preferidos com Funções Heurísticas Lógicas

Conclusão

Operadores Preferidos Ideais

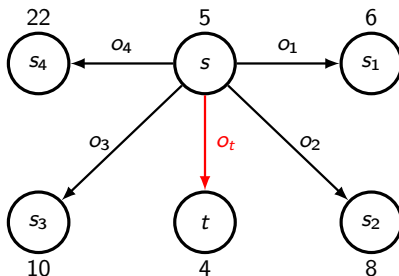
- ▶ Um operador preferido ideal gera um estado com a **menor distância até o gol** entre todos os **sucessores** de um estado s .

Operadores Preferidos Ideais

- ▶ Um operador preferido ideal gera um estado com a **menor distância até o gol** entre todos os **sucessores** de um estado s .
- ▶ Dado um estado s e um operador $o \in \mathcal{O}$ em que $\text{succ}(s, o) = t$, o é considerado um operador preferido ideal se $h^*(t) = \min_{s' \in \text{succ}(s)} h^*(s')$.
 - ▶ $h^*(t) < h^*(s)$

Operadores Preferidos Ideais

- ▶ Um operador preferido ideal gera um estado com a **menor distância até o gol** entre todos os **sucessores** de um estado s .
- ▶ Dado um estado s e um operador $o \in \mathcal{O}$ em que $\text{succ}(s, o) = t$, o é considerado um operador preferido ideal se $h^*(t) = \min_{s' \in \text{succ}(s)} h^*(s')$.
 - ▶ $h^*(t) < h^*(s)$



Abordagem Proposta

Problema

Introdução

Planejamento

Busca Heurística

Operadores Preferidos

Aprendizado de Funções Heurísticas

Abordagem Proposta

Operadores Preferidos Ideais

Operadores Preferidos Descobertos

Gerando Amostras com XRS

Experimentos

Preliminares

Aprendendo Operadores Preferidos

Comparando XRS e FSM

Operadores Preferidos com Funções Heurísticas Lógicas

Conclusão

Operadores Preferidos Descobertos

- ▶ A realidade é que não temos acesso ao espaço de estados completo com h^* .

Operadores Preferidos Descobertos

- ▶ A realidade é que não temos acesso ao espaço de estados completo com h^* .
- ▶ Para treinar redes neurais, o que temos é um conjunto de amostras do espaço de estados com estimativas imperfeitas.

Operadores Preferidos Descobertos

- ▶ A realidade é que não temos acesso ao espaço de estados completo com h^* .
- ▶ Para treinar redes neurais, o que temos é um conjunto de amostras do espaço de estados com estimativas imperfeitas.
- ▶ Precisamos “descobrir” os operadores preferidos dentro desse conjunto de amostras.

Grafo de Caminhos Mais Curtos

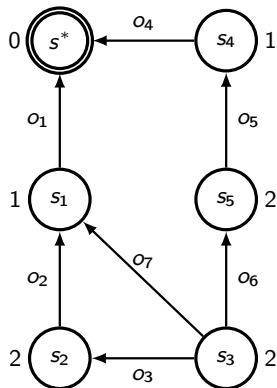
- ▶ Dado um grafo G representando um conjunto de amostras e suas relações de sucessores e predecessores, podemos calcular o(s) caminho(s) mais curto de cada estado (vértice) s até o gol s^* .
 - ▶ $\text{succ}(s, o) = t$ é representado por $s \xrightarrow{o} t$.

Grafo de Caminhos Mais Curtos

- ▶ Dado um grafo G representando um conjunto de amostras e suas relações de sucessores e predecessores, podemos calcular o(s) caminho(s) mais curto de cada estado (vértice) s até o gol s^* .
 - ▶ $\text{succ}(s, o) = t$ é representado por $s \xrightarrow{o} t$.
- ▶ Os arcos saintes de s que fazem parte de um caminho mais curto para o gol são os operadores preferidos de s .

Grafo de Caminhos Mais Curtos

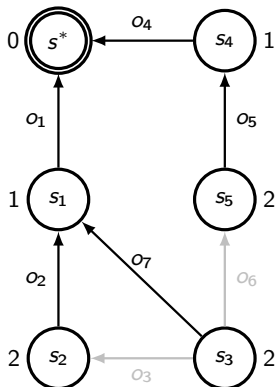
Grafo G .



Grafo de Caminhos Mais Curtos

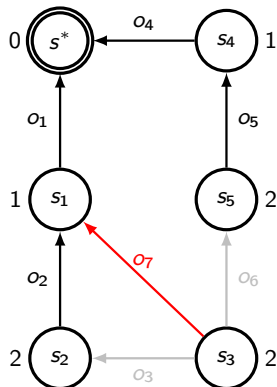
Grafo G' de caminhos mais curtos para s^* .

- Mantém só os arcos que levam ao(s) caminho(s) mais curto(s) de qualquer estado para s^* .



Grafo de Caminhos Mais Curtos

Por exemplo, para o estado s_3 ,
o operador o_7 é um **operador**
preferido.



Abordagem Proposta

Problema

Introdução

- Planejamento

- Busca Heurística

- Operadores Preferidos

- Aprendizado de Funções Heurísticas

Abordagem Proposta

- Operadores Preferidos Ideais

- Operadores Preferidos Descobertos

- Gerando Amostras com XRS**

Experimentos

- Preliminares

- Aprendendo Operadores Preferidos

- Comparando XRS e FSM

- Operadores Preferidos com Funções Heurísticas Lógicas

Conclusão

Gerando Amostras com XRS

- ▶ Método de Bettker et al. 2022 (FSM) gera muitas **amostras repetidas**.
 - ▶ Essas amostras repetidas são inúteis para a construção do grafo de caminho mais curto.
 - ▶ Logo, “perdemos a chance” de descobrir mais operadores preferidos.

Gerando Amostras com XRS

- ▶ Método de Bettker et al. 2022 (FSM) gera muitas **amostras repetidas**.
 - ▶ Essas amostras repetidas são inúteis para a construção do grafo de caminho mais curto.
 - ▶ Logo, “perdemos a chance” de descobrir mais operadores preferidos.
- ▶ Regressão via “**eXpansions from Random Successors**” (**XRS**).

Gerando Amostras com XRS

- ▶ Gera 10 % das amostras via **BFS**.
 - ▶ Isso popula a fila *open*, que contém estados gerados mas não expandidos, e conjunto *closed*, com estados já amostrados.

Gerando Amostras com XRS

- ▶ Gera 10 % das amostras via **BFS**.
 - ▶ Isso popula a fila *open*, que contém estados gerados mas não expandidos, e conjunto *closed*, com estados já amostrados.
- ▶ Gera o restante das amostras **expandindo sucessores aleatoriamente**:
 - ▶ Pega um estado $s \in open$ aleatoriamente.

Gerando Amostras com XRS

- ▶ Gera 10 % das amostras via **BFS**.
 - ▶ Isso popula a fila *open*, que contém estados gerados mas não expandidos, e conjunto *closed*, com estados já amostrados.
- ▶ Gera o restante das amostras **expandindo sucessores aleatoriamente**:
 - ▶ Pega um estado $s \in open$ aleatoriamente.
 - ▶ Expande s e coloca seus sucessores $s' \in succ(s)$ em *open* se $s' \notin closed$.

Gerando Amostras com XRS

- ▶ Gera 10 % das amostras via **BFS**.
 - ▶ Isso popula a fila *open*, que contém estados gerados mas não expandidos, e conjunto *closed*, com estados já amostrados.
- ▶ Gera o restante das amostras **expandindo sucessores aleatoriamente**:
 - ▶ Pega um estado $s \in open$ aleatoriamente.
 - ▶ Expande s e coloca seus sucessores $s' \in succ(s)$ em *open* se $s' \notin closed$.
 - ▶ Coloca s em *closed*.

Gerando Amostras com XRS

- ▶ Gera 10 % das amostras via **BFS**.
 - ▶ Isso popula a fila *open*, que contém estados gerados mas não expandidos, e conjunto *closed*, com estados já amostrados.
- ▶ Gera o restante das amostras **expandindo sucessores aleatoriamente**:
 - ▶ Pega um estado $s \in open$ aleatoriamente.
 - ▶ Expande s e coloca seus sucessores $s' \in succ(s)$ em *open* se $s' \notin closed$.
 - ▶ Coloca s em *closed*.

Depois, melhoramos as estimativas com SAI e SUI, extraímos os operadores preferidos, e completamos os estados.

Experimentos

Experimentos

Problema

Introdução

- Planejamento

- Busca Heurística

- Operadores Preferidos

- Aprendizado de Funções Heurísticas

Abordagem Proposta

- Operadores Preferidos Ideais

- Operadores Preferidos Descobertos

- Gerando Amostras com XRS

Experimentos

- Preliminares

- Aprendendo Operadores Preferidos

- Comparando XRS e FSM

- Operadores Preferidos com Funções Heurísticas Lógicas

Conclusão

Perguntas

- ▶ Conseguimos diminuir o número de estados expandidos usando a **nossa abordagem** com uma **heurística aprendida**? E como a nossa abordagem se **compara** com os operadores preferidos **ideais**?

Perguntas

- ▶ O que acontece se **aumentarmos o tamanho do nosso conjunto de amostras** para treinamento? Melhoramos? Se sim, até quanto?
 - ▶ Aliás: somos melhores que os operadores preferidos do **FF**?

Perguntas

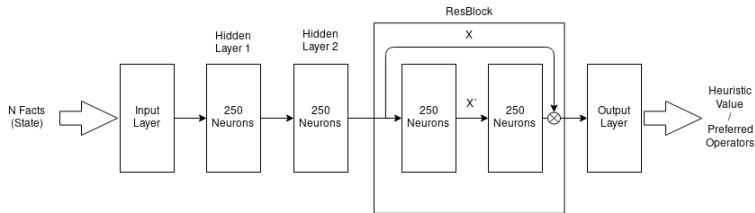
- ▶ E se usarmos o método de amostragem de Bettker et al. 2022 para descobrir operadores preferidos? É melhor ou pior do que a abordagem proposta?

Perguntas

- ▶ Nossos operadores preferidos continuam sendo úteis usando com diferentes **heurísticas lógicas**? “Ganhamos” dos operadores preferidos do FF?

Configuração

- ▶ Geração de amostras
 - ▶ Neural Fast Downward (Ferber et al. 2020).
- ▶ Redes neurais
 - ▶ PyTorch 1.9.0 (Paszke et al. 2019).
- ▶ Máquinas Ubuntu 20.04 LTS GNU/Linux
 - ▶ AMD Ryzen 9 3900X 12-core (4.2 GHz).
 - ▶ Limites de 4 GB RAM e um núcleo por processo.



Tarefas de Benchmark

Tabela: Sumário das tarefas escolhidas.

Domínio	State space	# fatos	# operadores
Blocks	65990	64	98
Grid	452353	76	252
N-Puzzle	181440	81	192
Rovers	565824	32	57
Scanalyzer	46080	42	300
Transport	637632	66	572
VisitAll	79931	31	48

Geração de Amostras (Operadores Preferidos)

- ▶ Amostragem usando **XRS** a partir do gol de uma tarefa de cada domínio.

Geração de Amostras (Operadores Preferidos)

- ▶ Amostragem usando **XRS** a partir do gol de uma tarefa de cada domínio.
- ▶ Geramos N amostras, onde N é a porcentagem do tamanho do espaço de estados de cada tarefa.
 - ▶ e.g. em Blocks, 5 % é $\lceil 65990 \times 0.05 \rceil = 3300$ amostras.

Geração de Amostras (Operadores Preferidos)

- ▶ Amostragem usando **XRS** a partir do gol de uma tarefa de cada domínio.
- ▶ Geramos N amostras, onde N é a porcentagem do tamanho do espaço de estados de cada tarefa.
 - ▶ e.g. em Blocks, 5 % é $\lceil 65990 \times 0.05 \rceil = 3300$ amostras.
- ▶ Geramos **5** conjuntos de amostras (*seeds*) para cada espaço de estados.

Geração de Amostras (Operadores Preferidos)

- ▶ Formato de cada amostra para a rede de **valores- h** (**FSM**):
 - ▶ $\langle h(s), \mathcal{F}(s) \rangle$
 - ▶ $\langle 15; 000000011110 \dots 000100000001 \rangle$
- ▶ Formato de cada amostra para a rede de **POs** (**XRS**):
 - ▶ $\langle \mathcal{F}(s), O_{pref} \subseteq O \rangle$
 - ▶ $\langle 000000011110 \dots 000100000001; 27, 26 \rangle$

Treinamento

- ▶ Duas redes treinadas separadamente: **regressão** com mean squared error (MSE) para funções heurísticas, **classificação** com binary cross-entropy (BCE) para POs.

Treinamento

- ▶ Duas redes treinadas separadamente: **regressão** com mean squared error (MSE) para funções heurísticas, **classificação** com binary cross-entropy (BCE) para POs.
- ▶ 5 net seeds
(= 5 net seeds \times 5 sample seeds = 25 seeds por tarefa).

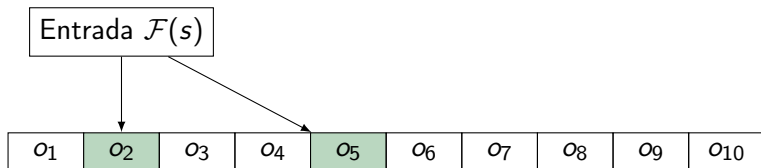
Treinamento

- ▶ Duas redes treinadas separadamente: **regressão** com mean squared error (MSE) para funções heurísticas, **classificação** com binary cross-entropy (BCE) para POs.
- ▶ 5 net seeds
(= 5 net seeds \times 5 sample seeds = 25 seeds por tarefa).
- ▶ Treino até **early-stop** de 100 épocas.

Treinamento

- ▶ Duas redes treinadas separadamente: **regressão** com mean squared error (MSE) para funções heurísticas, **classificação** com binary cross-entropy (BCE) para POs.
- ▶ 5 net seeds
(= 5 net seeds \times 5 sample seeds = 25 seeds por tarefa).
- ▶ Treino até **early-stop** de 100 épocas.

Figura: Tensor de saída de exmplo, com dois POs como “target”.



Teste (Busca)

- ▶ Para cada espaço de estados original, 50 tarefas de teste geradas via passeio aleatório a partir do estado inicial.

Teste (Busca)

- ▶ Para cada espaço de estados original, 50 tarefas de teste geradas via passeio aleatório a partir do estado inicial.
- ▶ Buscas utilizando o GBFS implementado no Fast Downward, com limite de 5 minutos por tarefa.

Teste (Busca)

- ▶ Para cada espaço de estados original, 50 tarefas de teste geradas via passeio aleatório a partir do estado inicial.
- ▶ Buscas utilizando o GBFS implementado no Fast Downward, com limite de 5 minutos por tarefa.
- ▶ Valor de boosting 1000.

Teste (Busca)

- ▶ Para cada espaço de estados original, 50 tarefas de teste geradas via passeio aleatório a partir do estado inicial.
- ▶ Buscas utilizando o GBFS implementado no Fast Downward, com limite de 5 minutos por tarefa.
- ▶ Valor de boosting 1000.
- ▶ Todas as tarefas têm cobertura total mesmo com busca cega, então o parâmetro de comparação escolhido é o número de estados expandidos.

Legenda

- ▶ h^*
 - ▶ heurística perfeita
- ▶ \hat{h}
 - ▶ heurística aprendida

- ▶ po^*
 - ▶ POs ideais (oráculo)
- ▶ \hat{po}^*
 - ▶ POs ideais (NN)

- ▶ \hat{po}^G
 - ▶ POs descobertos (NN)

Experimentos

Problema

Introdução

- Planejamento

- Busca Heurística

- Operadores Preferidos

- Aprendizado de Funções Heurísticas

Abordagem Proposta

- Operadores Preferidos Ideais

- Operadores Preferidos Descobertos

- Gerando Amostras com XRS

Experimentos

- Preliminares

- Aprendendo Operadores Preferidos**

- Comparando XRS e FSM

- Operadores Preferidos com Funções Heurísticas Lógicas

Conclusão

Conseguimos Aprender Algo?

Tabela: Expansões de GBFS guiado por h^* e \hat{h} sem/com operadores preferidos. \hat{po}^G treinado sobre 1 % do espaço de estados.

	h^*		\hat{h}		
	-	-	po^*	\hat{po}^*	\hat{po}^G
Geo. mean	14.5	35.0	15.7	17.7	30.6

Conseguimos Aprender Algo?

Tabela: Expansões de GBFS guiado por h^* e \hat{h} sem/com operadores preferidos. \hat{po}^G treinado sobre 1 % do espaço de estados.

	h^*	\hat{h}			
	-	-	po^*	\hat{po}^*	\hat{po}^G
Geo. mean	14.5	35.0	15.7	17.7	30.6

- Os POs ideais po^* e \hat{po}^* aproximam-se de h^* .

Conseguimos Aprender Algo?

Tabela: Expansões de GBFS guiado por h^* e \hat{h} sem/com operadores preferidos. \hat{po}^G treinado sobre 1 % do espaço de estados.

	h^*		\hat{h}		
	-	-	po^*	\hat{po}^*	\hat{po}^G
Geo. mean	14.5	35.0	15.7	17.7	30.6

- ▶ Os POs ideais po^* e \hat{po}^* aproximam-se de h^* .
- ▶ Os POs descobertos \hat{po}^G expandem o dobro de h^* , mas superam uma busca só com a heurística aprendida \hat{h} .

Conseguimos Aprender Algo?

Tabela: Expansões de GBFS guiado por h^* e \hat{h} sem/com operadores preferidos. \hat{po}^G treinado sobre 1 % do espaço de estados.

	h^*		\hat{h}		
	-	-	po^*	\hat{po}^*	\hat{po}^G
Geo. mean	14.5	35.0	15.7	17.7	30.6

- ▶ Os POs ideais po^* e \hat{po}^* aproximam-se de h^* .
- ▶ Os POs descobertos \hat{po}^G expandem o dobro de h^* , mas superam uma busca só com a heurística aprendida \hat{h} .
 - ▶ Então sim: POs não são um delírio, e **conseguimos aprender algo útil** mesmo treinando sobre amostras representando **1 % do espaço de estados**.

E Treinando com Mais Amostras?

Tabela: Expansões do GBFS guiadas por \hat{h} sem operadores preferidos e com po^{FF} e \hat{po}^G , treinado em diferentes tamanhos de conjunto de amostras.

		\hat{h}							
		\hat{po}^G							
	-	po^{FF}	1 %	5 %	10 %	20 %	30 %	40 %	50 %
Geo. mean	35.0	28.0	30.6	22.4	21.0	19.8	19.5	19.7	19.6

- Melhora, mas os ganhos são menores a partir de 20 % – 30 %.

E Treinando com Mais Amostras?

Tabela: Expansões do GBFS guiadas por \hat{h} sem operadores preferidos e com po^{FF} e \hat{po}^G , treinado em diferentes tamanhos de conjunto de amostras.

		\hat{h}							
		\hat{po}^G							
	- po^{FF}	1 %	5 %	10 %	20 %	30 %	40 %	50 %	
Geo. mean	35.0	28.0	30.6	22.4	21.0	19.8	19.5	19.7	19.6

- ▶ Melhora, mas os ganhos são menores a partir de 20 % – 30 %.
- ▶ Com um número de amostras equivalente a 5 % do espaço de estados, **superamos po^{FF}** em número de expansões.

E Treinando com Mais Amostras?

	\hat{h}								
	-	po^{FF}	\hat{po}^G						
			1 %	5 %	10 %	20 %	30 %	40 %	50 %
Blocks	57.0	41.1	43.0	26.2	29.6	32.2	34.9	38.0	38.8
Grid	66.5	32.7	67.4	53.0	46.4	27.4	23.8	23.6	22.9
N-Puzzle	80.9	100.2	53.3	34.8	31.6	28.5	28.1	27.4	27.4
Rovers	13.4	18.5	12.2	11.7	13.0	17.2	17.2	17.2	17.2
Scanalyzer	28.3	17.1	30.7	18.1	13.0	11.7	11.5	11.6	11.5
Transport	25.2	17.0	21.4	16.3	15.8	15.1	14.7	14.4	14.3
VisitAll	21.8	18.7	20.5	17.1	15.7	15.7	15.8	16.1	16.3
Geo. mean	35.0	28.0	30.6	22.4	21.0	19.8	19.5	19.7	19.6

- ▶ Com 5 % temos menos expansões em média comparado a po^{FF} , mas não em todos os domínios.
- ▶ Eventualmente superamos po^{FF} em **todos** os domínios, ainda que seja necessário mais dados para treinamento.

Experimentos

Problema

Introdução

- Planejamento

- Busca Heurística

- Operadores Preferidos

- Aprendizado de Funções Heurísticas

Abordagem Proposta

- Operadores Preferidos Ideais

- Operadores Preferidos Descobertos

- Gerando Amostras com XRS

Experimentos

- Preliminares

- Aprendendo Operadores Preferidos

- Comparando XRS e FSM**

- Operadores Preferidos com Funções Heurísticas Lógicas

Conclusão

XRS vs. FSM

Tabela: Expansões de GBFS guiado por \hat{h} com $\hat{p}o^G$ e $\hat{p}o^{FSM}$, variando o tamanho do conjunto de amostras.

	1 %		5 %		10 %		20 %		50 %	
	$\hat{p}o^G$	$\hat{p}o^{FSM}$	$\hat{p}o^G$	$\hat{p}o^{FSM}$	$\hat{p}o^G$	$\hat{p}o^{FSM}$	$\hat{p}o^G$	$\hat{p}o^{FSM}$	$\hat{p}o^G$	$\hat{p}o^{FSM}$
Geo. mean	30.6	34.2	22.4	26.4	21.0	24.3	19.8	23.8	19.6	22.5

► Em média, $\hat{p}o^G$ tem menos expansões do que $\hat{p}o^{FSM}$.

XRS vs. FSM

Tabela: Expansões de GBFS guiado por \hat{h} com $\hat{p}o^G$ e $\hat{p}o^{FSM}$, variando o tamanho do conjunto de amostras.

	1 %		5 %		10 %		20 %		50 %	
	$\hat{p}o^G$	$\hat{p}o^{FSM}$	$\hat{p}o^G$	$\hat{p}o^{FSM}$	$\hat{p}o^G$	$\hat{p}o^{FSM}$	$\hat{p}o^G$	$\hat{p}o^{FSM}$	$\hat{p}o^G$	$\hat{p}o^{FSM}$
Geo. mean	30.6	34.2	22.4	26.4	21.0	24.3	19.8	23.8	19.6	22.5

- ▶ Em média, $\hat{p}o^G$ tem menos expansões do que $\hat{p}o^{FSM}$.
- ▶ $\hat{p}o^{FSM}$ precisa de $10\times$ mais amostras para ter resultados comparáveis a $\hat{p}o^G$.
 - ▶ $\hat{p}o^{FSM}$ 50 %: 22.5.
 - ▶ $\hat{p}o^G$ 5 %: 22.4.

Experimentos

Problema

Introdução

- Planejamento

- Busca Heurística

- Operadores Preferidos

- Aprendizado de Funções Heurísticas

Abordagem Proposta

- Operadores Preferidos Ideais

- Operadores Preferidos Descobertos

- Gerando Amostras com XRS

Experimentos

- Preliminares

- Aprendendo Operadores Preferidos

- Comparando XRS e FSM

- Operadores Preferidos com Funções Heurísticas Lógicas

Conclusão

Operadores Preferidos com Heurísticas Lógicas

Tabela: Expansões de GBFS guiadas por heurísticas baseadas em lógica sem/com operadores preferidos (conj. de amostras de tamanho 1 %).

	h^{FF}			h^{add}			h^{GC}			h^{blind}		
	–	po^{FF}	\hat{po}^G	–	po^{FF}	\hat{po}^G	–	po^{FF}	\hat{po}^G	–	po^{FF}	\hat{po}^G
Geo. mean	39.0	30.0	27.0	37.1	33.2	26.0	124.9	51.3	41.4	19.7K	6.7K	244.3

► \hat{po}^G tem menos expansões do que po^{FF} em todos os casos.

Operadores Preferidos com Heurísticas Lógicas

Tabela: Expansões de GBFS guiadas por heurísticas baseadas em lógica sem/com operadores preferidos (conj. de amostras de tamanho 1 %).

	h^{FF}			h^{add}			h^{GC}			h^{blind}		
	–	po^{FF}	\hat{po}^G	–	po^{FF}	\hat{po}^G	–	po^{FF}	\hat{po}^G	–	po^{FF}	\hat{po}^G
Geo. mean	39.0	30.0	27.0	37.1	33.2	26.0	124.9	51.3	41.4	19.7K	6.7K	244.3

- ▶ \hat{po}^G tem menos expansões do que po^{FF} em todos os casos.
- ▶ Usar bons POs pode ser melhor do que trocar de função heurística.
 - ▶ $h^{GC} + \hat{po}^G$ tem expansões próximas de h^{FF} e h^{add} .

Operadores Preferidos com Heurísticas Lógicas

Tabela: Expansões de GBFS guiadas por heurísticas baseadas em lógica sem/com operadores preferidos (conj. de amostras de tamanho 1 %).

	h^{FF}			h^{add}			h^{GC}			h^{blind}		
	—	po^{FF}	\hat{po}^G	—	po^{FF}	\hat{po}^G	—	po^{FF}	\hat{po}^G	—	po^{FF}	\hat{po}^G
Geo. mean	39.0	30.0	27.0	37.1	33.2	26.0	124.9	51.3	41.4	19.7K	6.7K	244.3

- ▶ h^{blind} apenas **identifica o gol**, sendo inútil para guiar a busca.
- ▶ Neste caso, os operadores preferidos atuam como uma **política**. Os resultados mostram que po^{FF} não serve efetivamente como uma política, enquanto \hat{po}^G sim.

Conclusão

Conclusão

- ▶ É possível aprender operadores preferidos **comparáveis ao estado da arte** (po^{FF} , Hoffmann et al. 2001; Helmert 2006; Richter et al. 2009).

Conclusão

- ▶ É possível aprender operadores preferidos **comparáveis ao estado da arte** (po^{FF} , Hoffmann et al. 2001; Helmert 2006; Richter et al. 2009).
- ▶ POs aprendidos \hat{po}^G têm **menos expansões** do que po^{FF} nas tarefas de teste.

Conclusão

- ▶ É possível aprender operadores preferidos **comparáveis ao estado da arte** (po^{FF} , Hoffmann et al. 2001; Helmert 2006; Richter et al. 2009).
- ▶ POs aprendidos \hat{po}^G têm **menos expansões** do que po^{FF} nas tarefas de teste.
 - ▶ Isso é mais evidente quando a heurística usada é menos informada (h^{GC} , h^{blind}).

Conclusão

- ▶ É possível aprender operadores preferidos **comparáveis ao estado da arte** (po^{FF} , Hoffmann et al. 2001; Helmert 2006; Richter et al. 2009).
- ▶ POs aprendidos \hat{po}^G têm **menos expansões** do que po^{FF} nas tarefas de teste.
 - ▶ Isso é mais evidente quando a heurística usada é menos informada (h^{GC} , h^{blind}).
 - ▶ Mas treinar sobre 5 % do espaço de estados para superar po^{FF} ainda exige um número considerável de amostras.

Conclusão

- ▶ É possível aprender operadores preferidos **comparáveis ao estado da arte** (po^{FF} , Hoffmann et al. 2001; Helmert 2006; Richter et al. 2009).
- ▶ POs aprendidos \hat{po}^G têm **menos expansões** do que po^{FF} nas tarefas de teste.
 - ▶ Isso é mais evidente quando a heurística usada é menos informada (h^{GC} , h^{blind}).
 - ▶ Mas treinar sobre 5 % do espaço de estados para superar po^{FF} ainda exige um número considerável de amostras.
 - ▶ Aprender POs é uma tarefa **difícil**.

Conclusão

- ▶ É possível aprender operadores preferidos **comparáveis ao estado da arte** (po^{FF} , Hoffmann et al. 2001; Helmert 2006; Richter et al. 2009).
- ▶ POs aprendidos \hat{po}^G têm **menos expansões** do que po^{FF} nas tarefas de teste.
 - ▶ Isso é mais evidente quando a heurística usada é menos informada (h^{GC} , h^{blind}).
 - ▶ Mas treinar sobre 5 % do espaço de estados para superar po^{FF} ainda exige um número considerável de amostras.
 - ▶ Aprender POs é uma tarefa **difícil**.
- ▶ Dado um conjunto com número de amostras representando uma porcentagem do espaço de estados, POs aprendidos **generalizam bem** para o espaço de estados completo de uma tarefa.

Referências I

- [1] R. V. Bettker, P. P. Minini, A. G. Pereira e M. Ritt. “Understanding Sample Generation Strategies for Learning Heuristic Functions in Classical Planning”. Em: *arXiv preprint arXiv:2211.13316* (2022).
- [2] P. Ferber, F. Geißer, F. Trevizan, M. Helmert e J. Hoffmann. “Neural Network Heuristic Functions for Classical Planning: Bootstrapping and Comparison to Other Methods”. Em: *International Conference on Automated Planning and Scheduling*. AAAI Press, 2022.
- [3] P. Ferber, M. Helmert e J. Hoffmann. “Neural Network Heuristics for Classical Planning: A Study of Hyperparameter Space”. Em: *European Conference on Artificial Intelligence*. Santiago de Compostela, Spain: IOS Press, 2020, pp. 2346–2353.

Referências II

- [4] M. Helmert. “The Fast Downward Planning System”. Em: *Journal of Artificial Intelligence Research* 26 (2006), pp. 191–246.
- [5] J. Hoffmann e B. Nebel. “The FF Planning System: Fast Plan Generation Through Heuristic Search”. Em: *Journal of Artificial Intelligence Research* 14 (2001), pp. 253–302.
- [6] S. O’Toole, M. Ramirez, N. Lipovetzky e A. R. Pearce. “Sampling from Pre-Images to Learn Heuristic Functions for Classical Planning”. Em: *Symposium on Combinatorial Search*. Vol. 15. Vienna, Austria: AAAI Press, 2022, pp. 308–310.
- [7] A. Paszke et al. “PyTorch: An Imperative Style, High-Performance Deep Learning Library”. Em: *Advances In Neural Information Processing Systems* 32 (2019).

Referências III

- [8] S. Richter e M. Helmert. “Preferred Operators and Deferred Evaluation in Satisficing Planning”. Em: *International Conference on Automated Planning and Scheduling*. Vol. 19. Thessaloniki, Greece: AAAI Press, 2009, pp. 273–280.
- [9] S. Richter e M. Westphal. “The LAMA Planner: Guiding Cost-Based Anytime Planning with Landmarks”. Em: *Journal of Artificial Intelligence Research* 39 (2010), pp. 127–177.
- [10] S. Richter, M. Westphal e M. Helmert. “LAMA 2008 and 2011”. Em: *International Planning Competition* (2011), pp. 117–124.
- [11] J. Seipp e G. Röger. “Fast Downward Stone Soup 2018”. Em: *International Planning Competition* (2018), pp. 72–74.

Referências IV

- [12] L. Yu, R. Kuroiwa e A. Fukunaga. “Learning Search-Space Specific Heuristics Using Neural Network”. *Em: ICAPS Workshop on Heuristics and Search for Domain-independent Planning*. ICAPS, 2020.