SeeA*: Efficient Exploration-Enhanced A* Search by Selective Sampling

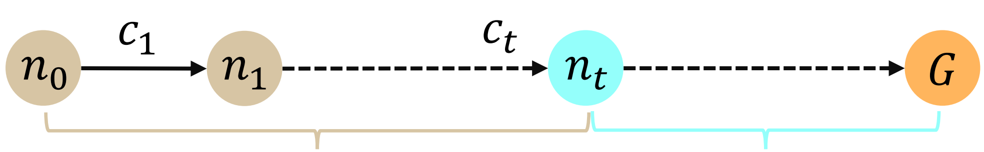
Dengwei Zhao, Shikui Tu, Lei Xu

November 2024

上海交通大學
SHANGHAI JIAO TONG UNIVERSITY

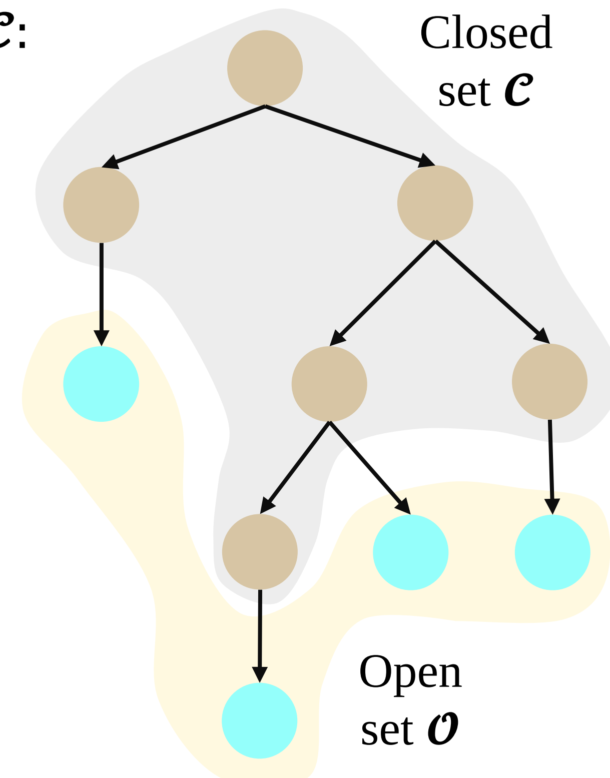
History of A* Search

- A* search algorithm is first published in 1968 by Peter Hart, Nils Nilsson and Bertram Raphael[1].
- A* search maintains an open set $\boldsymbol{\mathcal{O}}$ and a closed set $\boldsymbol{\mathcal{C}}$:
  - Select the node n with minimum f value from $\boldsymbol{\mathcal{O}}$.
  - Move n from $\boldsymbol{\mathcal{O}}$ to $\boldsymbol{\mathcal{C}}$, and put children of n into $\boldsymbol{\mathcal{O}}$.
- For a node n_t, $f(n)$ is the summation of:
  - $g(n)$: accumulated cost from n_0 to n_t.
  - $h(n)$: expected cost from n_t to the goal.



$g(n)$ computes the cost from the known searching trajectory.

$h(n)$ is a heuristic function to estimate the cost of the future path.



Renaissance of A*

- Inspired by the combination of deep neural network and Monte Carlo tree search, three possible aspects are addressed with a family of possible improvements proposed under the name of Deep IA-search[2]
  - Estimating $f(n)$ with the help of deep learning, making A^* into the era of learning aided A^*.
  - Seeking a better estimation of $f(n)$ with the help of global or future information
    - Lookahead or scouting before expanding the current node to collect future information to revise $f(n)$ of the current node.
    - Learning under path consistency condition, that is, $f(n)$ values on one optimal path should be identical.
  - Searching under inaccurate estimation of $f(n)$
    - Selecting nodes among the subset of open nodes of A^*.

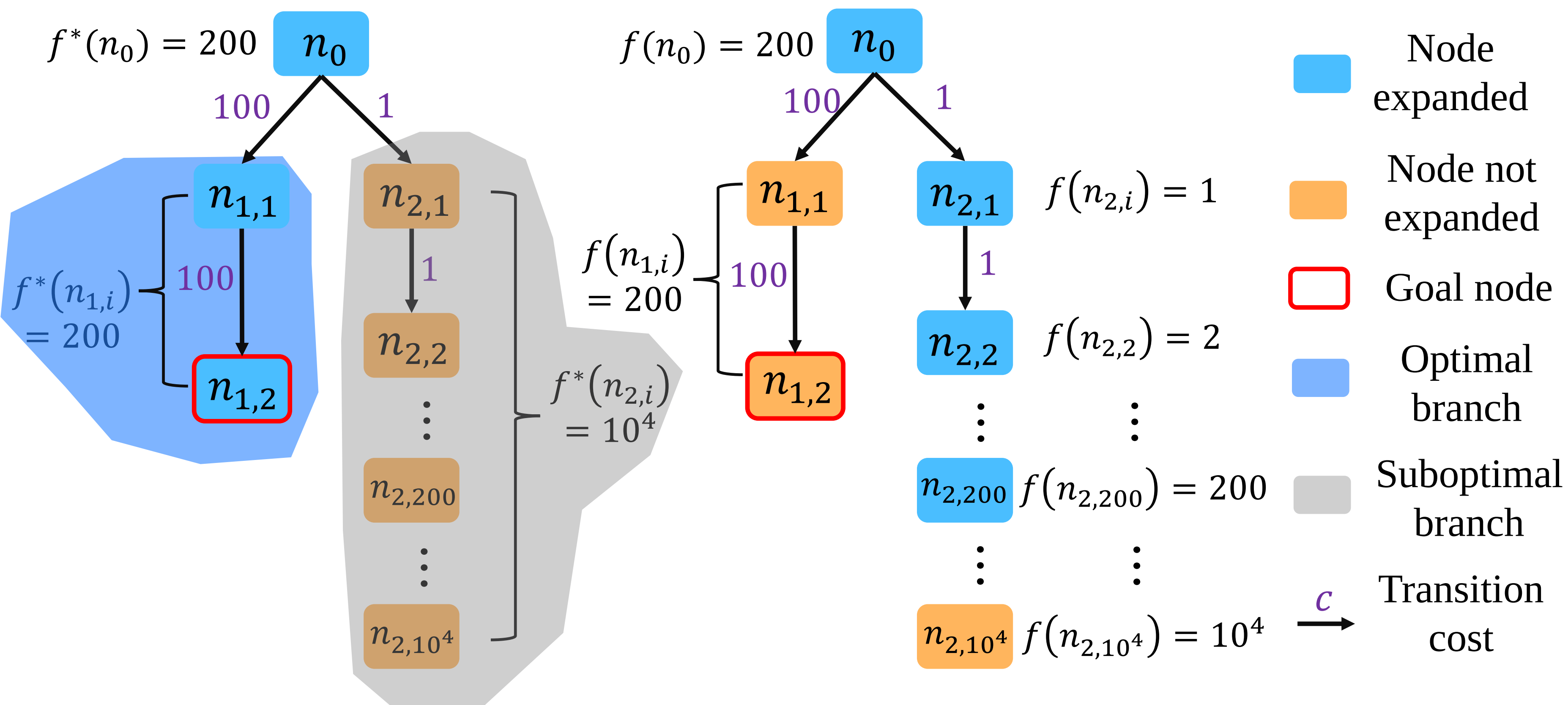
Optimality of A* Search

- $f(n) = g(n) + h(n)$ is an estimation of the real cost $f^*(n) = g^*(n) + h^*(n)$.
- $g(n) = g^*(n)$ because $g(n)$ is calculated from the known trajectory.
- Admissible assumption: heuristic function never overestimate the real cost, i.e., $h(n) \leq h^*(n) \Rightarrow$A* is guaranteed to find the optimal solution.
- The efficiency of A* search is highly influenced by the accuracy of the estimation of $h(n)$, even if the optimality is guaranteed.
- For example, heuristic function $h(n)$ satisfies the admissible assumption.

$$h(n) = \begin{cases} h^*(n), & \text{if } n \text{ is on the optimal path} \\ 0, & \text{Otherwise} \end{cases}$$

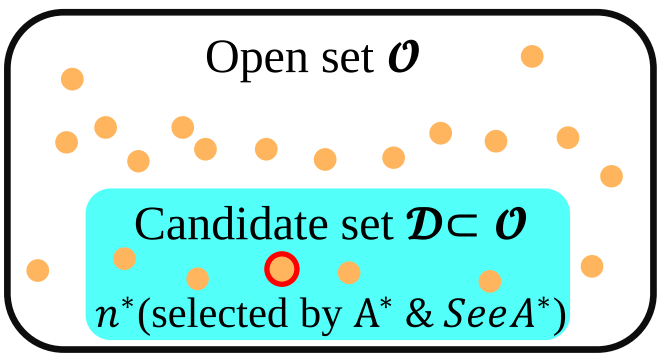
Limitation of A* Search

- The search efficiency is largely compromised due to the inaccuracy of $f(n)$ and the best-first search strategy of A*.

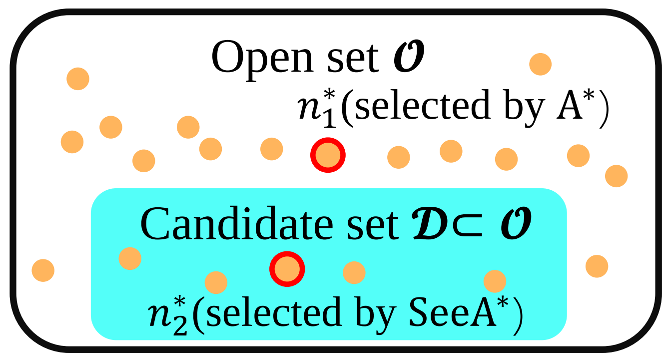


Sampling Exploration Enhanced Search

- SeeA* is proposed by introducing exploration into the best first A* search.
  - Sample a candidate subset $\mathcal{D}$ from the open set $\mathcal{O}$.
  - Select the node n with the lowest f-value from the candidate set $\mathcal{D}$.
- If the node with minimum f-value is not sampled into the candidate set $\mathcal{D}$, the node selected to be expanded later is not the same as the one by A* search.



$$n^* = \mathrm{argmin}_{n \in \mathcal{O}} f(n)$$
$$= \mathrm{argmin}_{n \in \mathcal{D}} f(n)$$

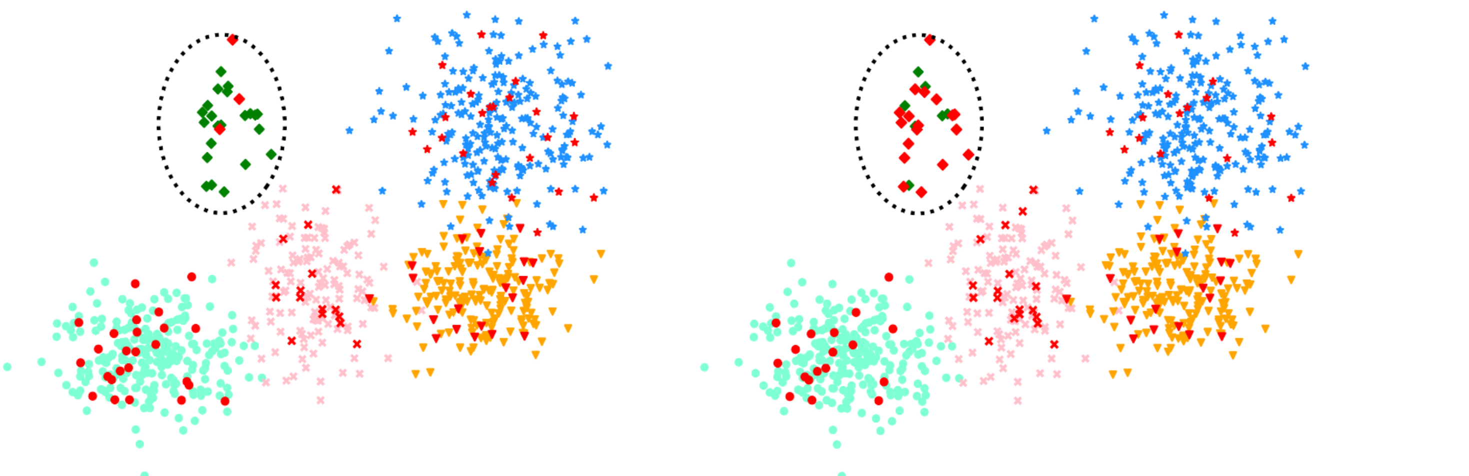


$$n_1^* = \mathrm{argmin}_{n \in \mathcal{O}} f(n)$$
$$n_2^* = \mathrm{argmin}_{n \in \mathcal{D}} f(n)$$

Exploration is enabled!

Sampling strategy

- Uniform sampling: K nodes are randomly selected from the open nodes as $\boldsymbol{\mathcal{D}}$.
- Clustering sampling: partition open nodes into multiple clusters and sampling nodes from each cluster evenly.
- UCT-like sampling: K nodes with the smallest E values are chosen
  - $E(n) = f(n) - c_b \times \frac{\sqrt{d_{max}}}{1+d(n)}$, $d(n)$ is the depth in the search tree.



Efficiency of SeeA* search

- The f^* values of all nodes on the optimal path are equal to the same cost μ_0^f and lower than the f^* value of nodes outside the optimal path, which was assumed to be sampled from a Gaussian distribution $\mathcal{G}(\mu_1^f, \sigma_s^2)$[3].
- If the estimation error of f follows an uniform distribution, we assume that: For each node n on the optimal path, $f(n) \sim \mathcal{U}(\mu_0^f - \sigma, \mu_0^f + \sigma)$. For nodes not on the optimal path, $f(n) \sim \mathcal{U}(f^*(n) - \sigma, f^*(n) + \sigma)$, and $f^*(n)$ are independently and identically sampled from $\mathcal{G}(\mu_1^f, \sigma_s^2)$.
- For n on the optimal path and n' off the optimal path, the probability

$$p_\sigma = P(f(n) \leq f(n')|\sigma)$$

- decreases as the prediction error σ increases.

Efficiency of SeeA* search

- Assume the open set $\boldsymbol{\mathcal{O}} = \{n_1, n_2, \cdots, n_{N_o}\}$, and n_1 is the optimal node.
- The probability of A* search expanding node n_1 is

$$P_A(\sigma) = P(n_1 = \mathrm{argmin}_{n \in \boldsymbol{\mathcal{O}}} f(n) | \sigma)$$

- The probability of SeeA* expanding node n_1 is

$$P_S(\sigma) = P(n_1 \in \boldsymbol{\mathcal{D}}, n_1 = \mathrm{argmin}_{n \in \boldsymbol{\mathcal{D}}} f(n) | \sigma)$$

- If the uniform sampling strategy is used, $P_S(\sigma) > P_A(\sigma)$ holds if and only if

$$p_\sigma < H(N_o), H(N_o) = \left(\frac{K}{N_o}\right)^{1/(N_o - K)} \quad and\ N_o > K \geq 1$$

- Larger P ⇒ fewer expansions to find the optimal solution ⇒ SeeA* is more efficient than A*.

Efficiency of SeeA* search

- If the estimation is quite inaccurate $\Rightarrow$ small p_σ
- $H(N_O)$ is monotonically increases with respect to N_O, and $\lim_{N_O \to \infty} H(N_O) = 1 \Rightarrow$ Complex problem with larger branching factor $\Rightarrow$ large $H(N_O)$
- Both situation makes the condition $p_\sigma < H(N_O)$ established.
- For uniform sampling, $P_S(\sigma)$ is approximately equal to $\frac{K}{N_O} p_\sigma^{K-1}$, and

$$K^* = \text{argmax} P_S(\sigma) = -1/\log p_\sigma$$

- If the estimation is quite accurate $\Rightarrow p_\sigma \to 1 \Rightarrow K^* \to \infty \Rightarrow$ SeeA* becomes A*.
- If the estimation is quite inaccurate $\Rightarrow p_\sigma \to 0 \Rightarrow K^* = 1 \Rightarrow$ SeeA* becomes random sampling, and the estimation of f provides no information.

Experiments

- Two real word applications are considered:
  - Retrosynthetic planning in organic chemistry: identify a series of chemical reactions that can utilize available molecules to generate the target molecule.
  - Logic synthesis in integrated circuit design: optimize the and-inverter logic graph to have the lowest area-delay product (ADP) through a sequence of functionality-preserving transform.
- The learned heuristic functions face a significant overfitting issue
  - State space for both problem are quite huge.
  - The training data is quite limited.

Results on Retrosynthetic Planning

- Test on seven molecule sets including the USPTO benchmark.
- SeeA* maintains its superiority over other search algorithms.
- SeeA*(Cluster) has the highest mean success rate of 63.56%.
- The clustering sampling and UCT-like sampling are better than uniform sampling in terms of the solved rate and the route length.

Algorithm	Solved ↑	Length ↓	Algorithm	Solved ↑	Length ↓
Retro*	54.66%	16.58	A*	58.73%	15.78
MCTS	59.20%	15.91	WA*	58.87%	15.66
LevinTS	61.01%	15.74	PHS	56.16%	16.51
ϵ Greedy	61.23%	19.88	SeeA*(Uniform)	62.97%	14.85
SeeA*(Cluster)	63.56%	14.31	SeeA*(UCT)	63.31%	14.33

Results on Logic Synthesis

- SeeA∗(Cluster) achieves the highest ADP reduction (i.e., 23.5%), obviously surpassing the state-of-the-art ABC-RL's 20.9%, and all other search algorithm.
- Trained on 23 chips and test on 12 chips.

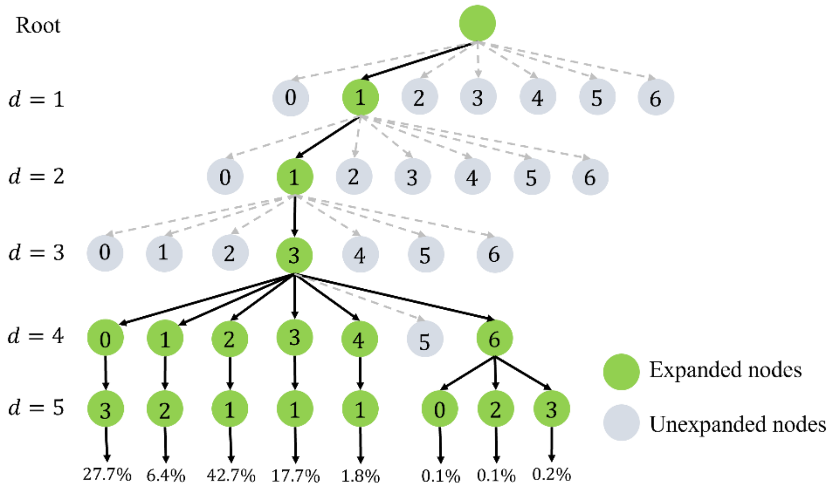
Algorithm	Mean ADP reduction ↑	Algorithm	Mean ADP reduction ↑
DRiLLS	14.8%	Online-RL	15.4%
SA+Pred.	19.0%	MCTS	18.5%
ABC-RL	20.9%	A^*	19.5%
ϵ Greedy	20.6%	PV-MCTS	19.5%
PHS	15.9%	$SeeA^*$(Uniform)	21.6%
$SeeA^*$(Cluster)	23.5%	$SeeA^*$(UCT)	22.5%

Results on Path Finding

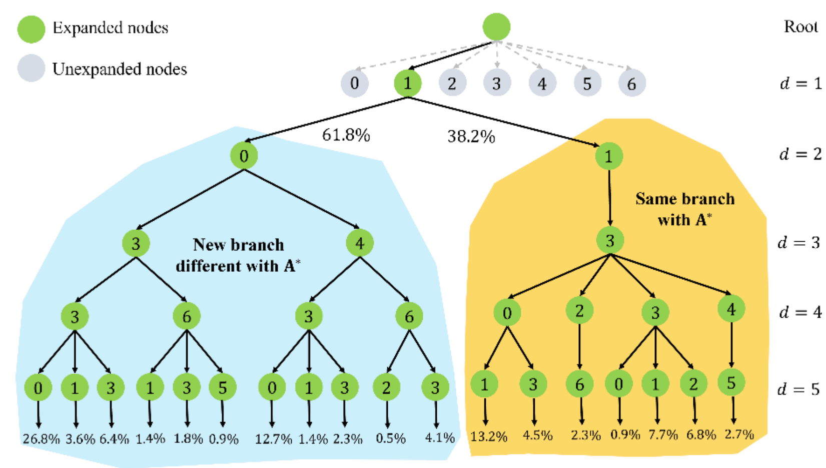
- Path finding: find the shortest path from a starting point to a destination.
- The cost for each step is 1. g is the number of steps taken to reach the current position, and h is the Euclidean distance from the current position to the target position, which is reliable enough to guide the search.
- An unreliable heuristic function $\hat{h}$ is designed, which is randomly sampled from $[0, 2 \times h]$.

	Guided by h			**Guided by $\hat{h}$**		
Algorithm	Solved	Cost	Expansions	Solved	Cost	Expansions
A*	100.0%	400	33340.52	100.0%	691.1	5028128
SeeA*	100.0%	400	33283.21	100.0%	**531.2**	54098.81

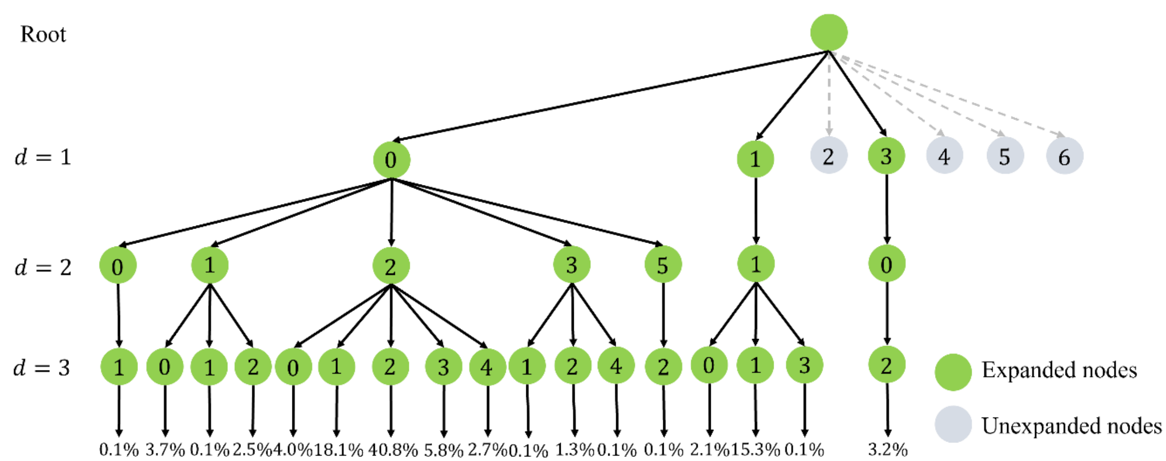
Exploration of SeeA*



Search tree of A*



Search tree of SeeA*



Search tree of MCTS

Summary

- In this paper, the SeeA* search is proposed to enhance the exploration behavior of the A* search by selecting expanded nodes from the sampled candidate nodes, rather than the entire set of open nodes.
- SeeA* is more efficient than A* from both theoretical analysis and experimental results.
- According to $P_S(\sigma) = P(n_1 \in \boldsymbol{\mathcal{D}}) \times P(n_1 = \mathrm{argmin}_{n \in \boldsymbol{\mathcal{D}}} f(n) | \sigma, n_1 \in \boldsymbol{\mathcal{D}})$:
    - Reducing the prediction error σ of the heuristic function.
    - Using a smaller number of candidate nodes K to include the optimal node in the candidate set with a greater likelihood $P(n_1 \in \boldsymbol{\mathcal{D}})$, smaller K is also helpful to select n_1 from $\boldsymbol{\mathcal{D}}$.
    - Investigations on more effective sampling strategies will be conducted in future.

Reference

- [1] Hart, Peter E., Nils J. Nilsson, and Bertram Raphael. "A formal basis for the heuristic determination of minimum cost paths." *IEEE transactions on Systems Science and Cybernetics* 4.2 (1968): 100-107.
- [2] Xu, Lei. "Deep bidirectional intelligence: AlphaZero, deep IA-search, deep IA-infer, and TPC causal learning." *Applied Informatics*. Vol. 5. No. 1. Berlin/Heidelberg: Springer Berlin Heidelberg, 2018.
- [3] Xu, Lei, Pingfan Yan, and Tong Chang. "Algorithm CNneim-A and its mean complexity." *Proc. of 2nd international conference on computers and applications. IEEE Press, Beijing*. 1987.

Thanks!

Code:
https://github.com/CMACH508/SEEA_star