KeeA*: Epistemic Exploratory A* Search via Knowledge Calibration

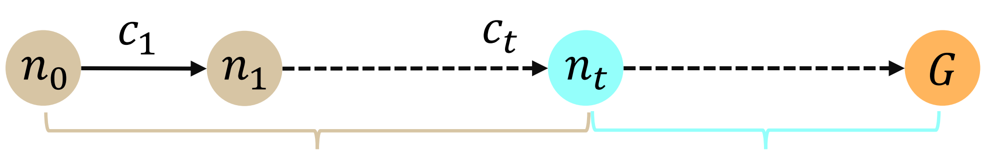
Dengwei Zhao, Shikui Tu, Yanan Sun, Lei Xu

October 2025

上海交通大學
SHANGHAI JIAO TONG UNIVERSITY

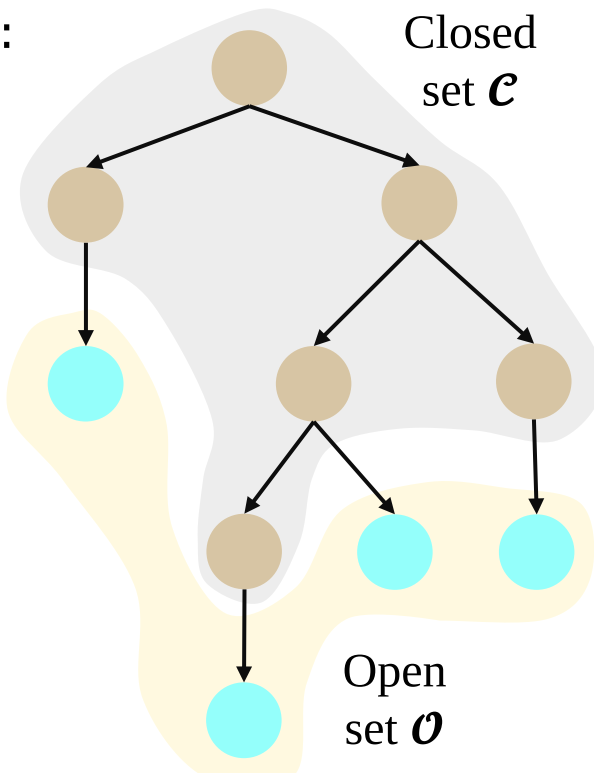
History of A* Search

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



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.

SeeA* search algorithm

- 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 $\mathcal{D}$.
- Various sampling strategies are employed.
  - Uniform sampling: K nodes are randomly selected from the open nodes as $\mathcal{D}$.
  - Clustering sampling: partition open nodes into multiple clusters and sampling nodes from each cluster evenly.
- Both theoretical analysis and experiment results have demonstrated the effectiveness of SeeA* compared to A*, when heuristic is biased.

KeeA* search

- Building on SeeA*, KeeA* is proposed:

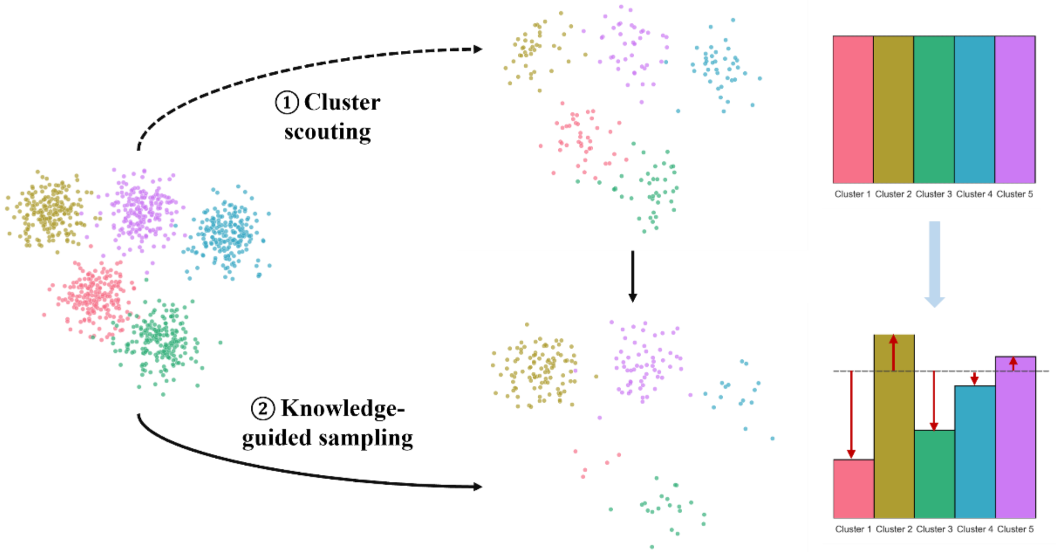
  - We theoretically demonstrate the advantage of cluster sampling in finding near-optimal solutions, compared to uniform sampling. The candidate set $\boldsymbol{\mathcal{D}}$ constructed by cluster sampling is closer to the optimal solution compared to uniform sampling, exhibiting a larger expected extreme value of f^*:

$$E_{\boldsymbol{\mathcal{D}}_{cluster}}[M] > E_{\boldsymbol{\mathcal{D}}_{uniform}}[M]$$

  - **Cluster scouting** and **path-aware sampling** are introduced by KeeA* to enhance the quality of the sampled candidate set by injecting epistemic knowledge into the sampling.

Cluster Scouting

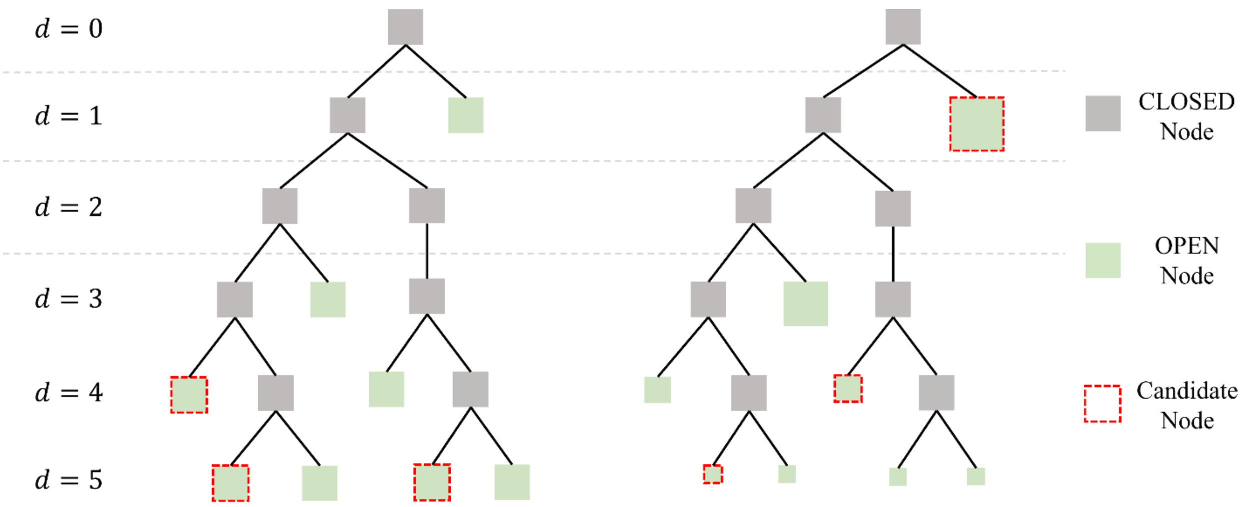
- Theoretical analysis illustrates that the probability $P(f^*(n_1) > f^*(n_2))$ is a monotonic function of the difference $f(n_1) - f(n_2)$.
- Step a: An equal number of nodes are sampled from each cluster to evaluate the quality of each cluster.
- Step b: Nodes are sampled from clusters in proportion to their quality.



Path Aware Sampling

- Node depth $d(n)$ is employed as path information to bias sampling toward shallower nodes, mitigating the concentration of samples within limited deeper branches to encourage exploration.

$$p_i(n) = \beta \times \frac{\exp d(n)}{\sum_{j=1}^{N_i} \exp d(n_j)} + (1-\beta) \times \frac{1}{N_i}$$



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.

Results on Retrosynthetic Planning

Algorithm	Solved (%) ↑	Avg Length ↓
Retro*	54.66	16.58
Retro*+	59.93	15.44
MCTS	59.20	15.91
A*	58.73	15.78
LevinTS	61.01	15.74
PHS	56.16	16.51
SeeA*(Uniform)	62.97	14.85
SeeA*(Cluster)	63.56	14.31
KeeA* w/o CS	**63.74**	**14.16**
KeeA*	**63.76**	**14.14**

Tab 1. Test results on the seven benchmark for retrosynthetic planning

Algorithm	Avg reduction ↑
DRiLLS	14.8
MCTS	18.5
ABC-RL	20.9
A*	19.5
PV-MCTS	19.5
PHS	15.9
SeeA*(Uniform)	21.6
SeeA*(Cluster)	23.5
KeeA* w/o CS	**24.0**
KeeA*	**25.2**

Tab 2. Test results on the MCNC benchmark for Logic synthesis

Summary

- In this paper, KeeA* search is proposed:
  - We first theoretically demonstrate the superiority of cluster sampling over uniform sampling within the SeeA* framework.
  - Cluster scouting and path-aware sampling is proposed by KeeA* to exploit node distributional knowledge for epistemic calibration, enhancing the mean and variance of sampled candidate nodes, respectively.
  - Extensive empirical experiments across thousands of test instances and statistical hypothesis testing results validate the effectiveness and robustness of KeeA*.

Thanks!

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