University of Basel

Best first search with trial-based open list

Bachelor Thesis Presentation

Rasmus Jensen

Background

Classical Planning

- Environment is math. represented by a state space
- Initial state
- Deterministic actions
- Single agent

Forward Search

- Start at the initial state
- Search returns when goal state reached

Forward search:

1. Add initial state to the open list

repeat:

2. If open list empty return unsolvable
3. Choose node in open list
4. If node is a goal return solution
5. Expand node
6. Add successors open list

Best First search

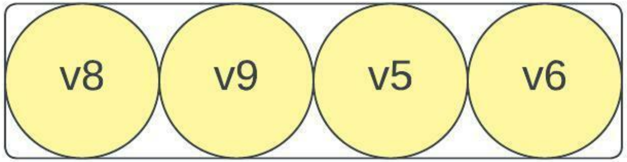
- Forward search with heuristic enhancement
- Select promising nodes first (hence Best first)
- Generally good searches due to heuristic

Open List

- Stores nodes that need to be expanded
- List order influences search process

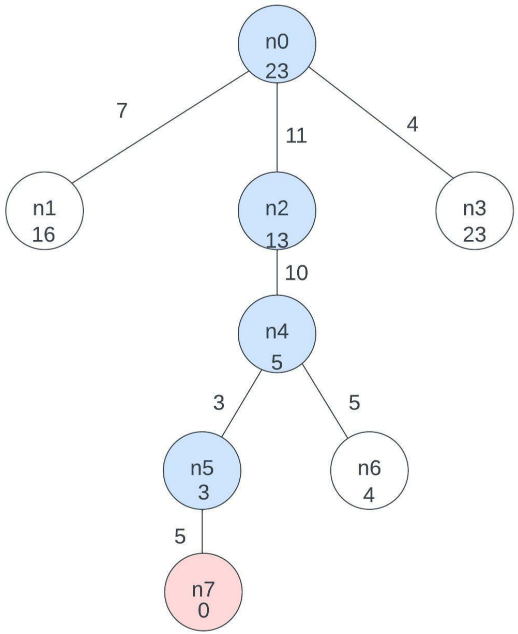
Example : Breadth first search $\longrightarrow$ FIFO queue

- Often ordered using a heuristic h
- Essential component of search



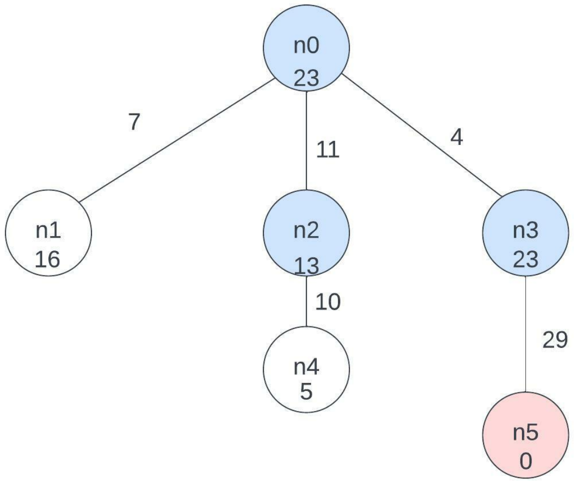
Greedy Best First search

- Expansion order completely based on heuristic
- Ignores current path cost
- Non-optimal search
- Known for fast expected search time



Epsilon-greedy Best First search

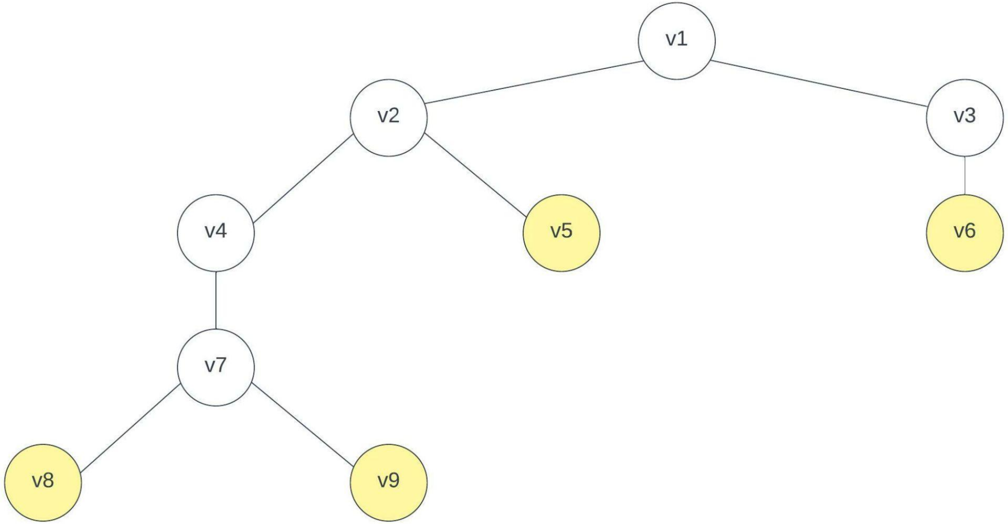
- Extension of Greedy BFS
- Probability of arbitrary node expansion



Trial-based EGBFS

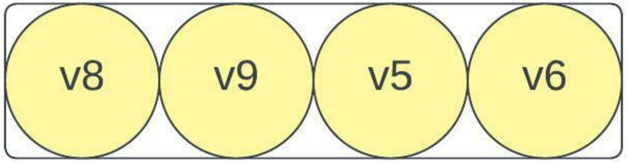
Trial-based open list

- Action selection process
- Open list is the frontier
- Maintain entire tree structure

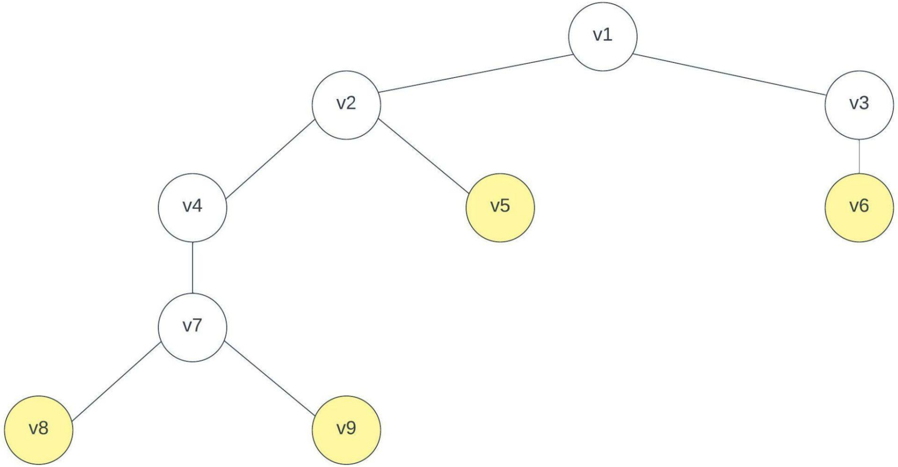


Explicit open list vs trial-based open list

- Fast selection process
- Little memory management
- Easy to check stopping criterion



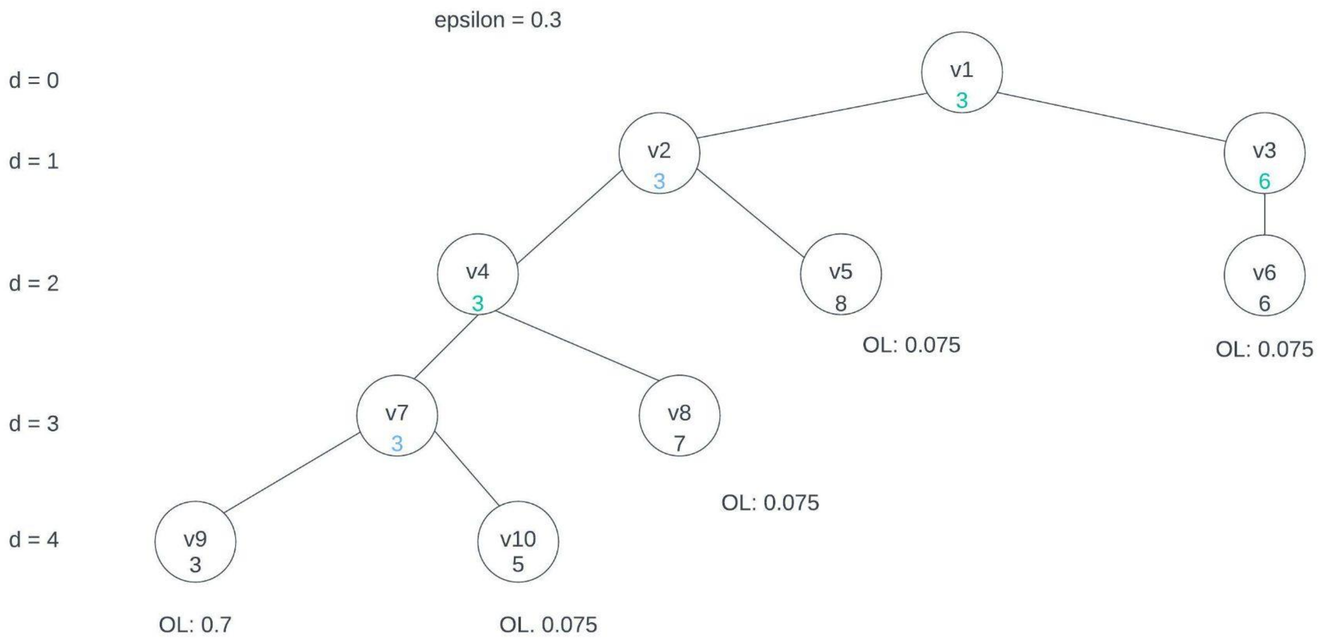
- Refined selection process
- Frontier is always updated



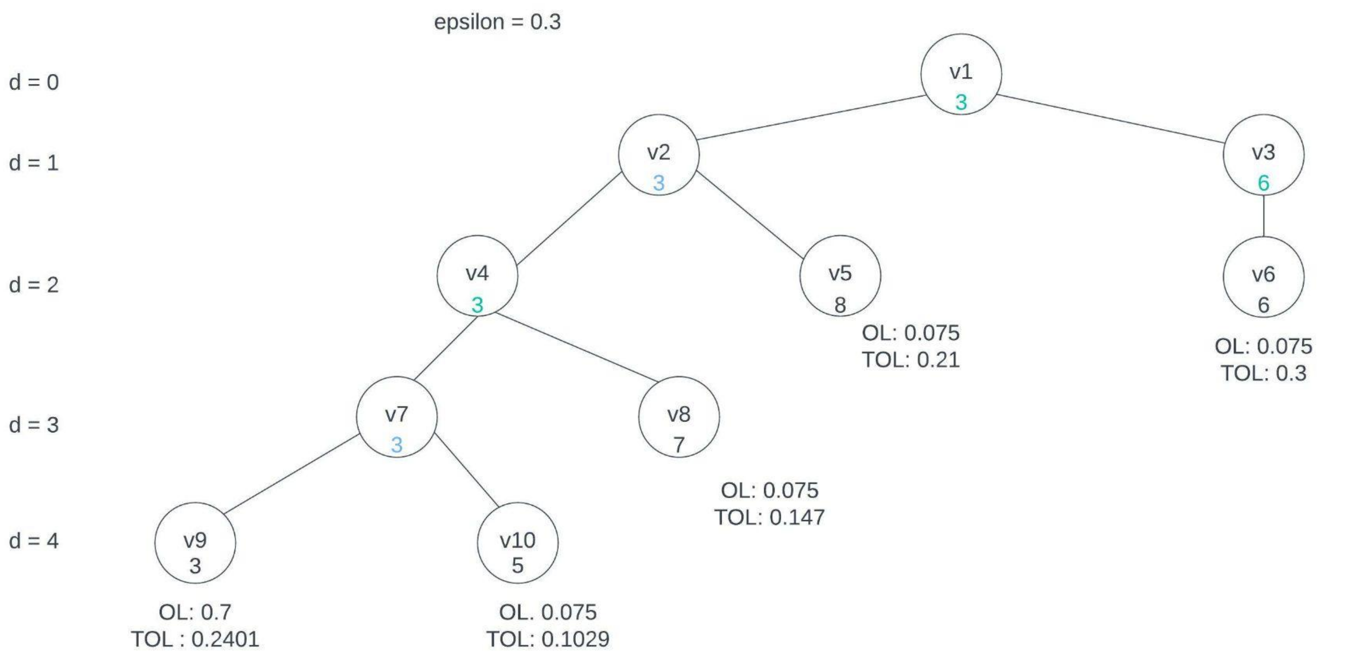
GBFS with OL vs TOL (FF heuristic)

With reopening (2742)		
Results	GBFS	TOL-0.0
Coverage	1756	1751
Memory	100810532	111492768

Exploration (OL)



Exploration (OL vs TOL)



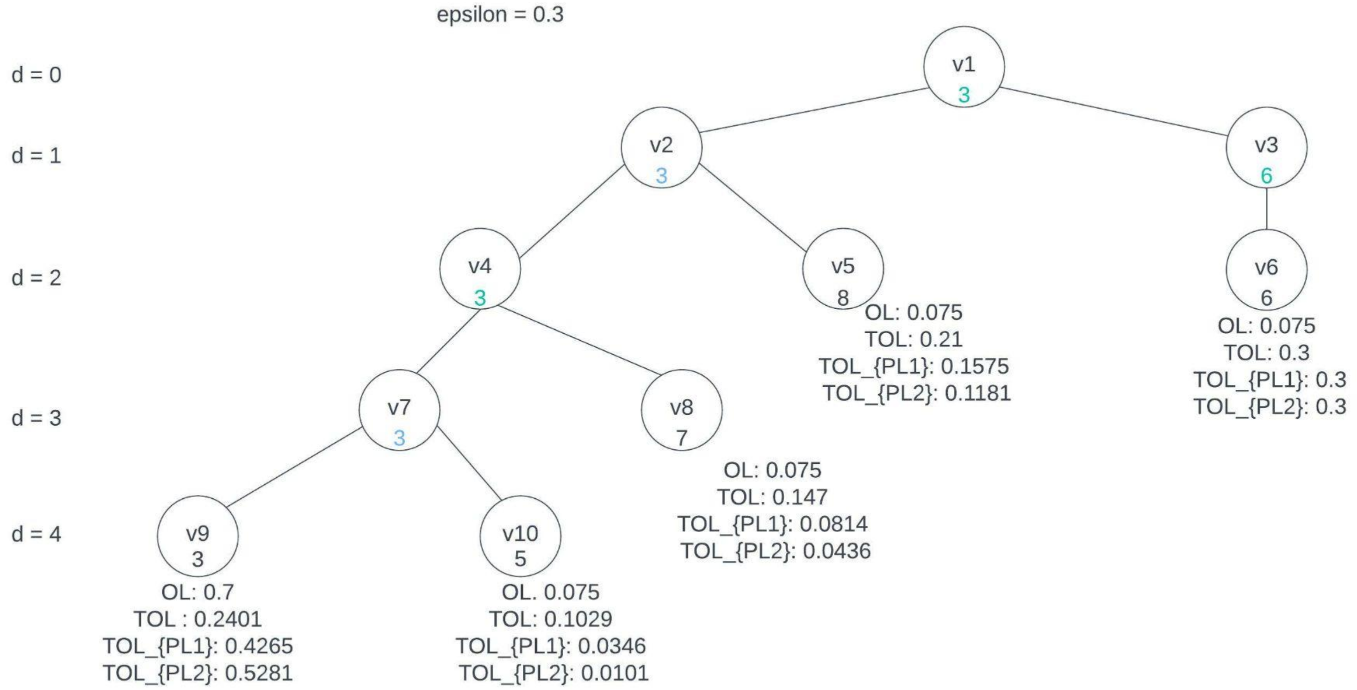
Introduction to path length

- Remedy lack of exploitation in deep tree layers
- Maintain variable with current max tree depth
- Nodes contain variable with tree depth
- Reduce exploration based on current depth and maximum depth
- Still promote early exploration
- Hone in on goal in deeper layers

$$\epsilon_{node} = \left(\frac{d_m - d_{node}}{d_m}\right)^n \epsilon$$

path length of degree n

Exploration (TOL_{PL})

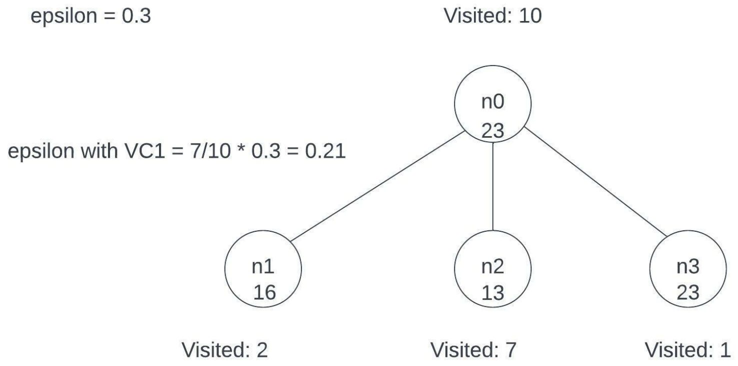


Introduction to visited count

- View parent and children (local)
- Reduce exploration based on local exploitation
- Bounding explorative variability

$$\epsilon_{node} = \left(\frac{v_{child*}}{v_{node}}\right)^n \epsilon$$

visited count of degree n



Experiments

OL vs TOL

With reopening (1747)									
Results	OL-0.3	OL-0.2	OL-0.1	OL-0.05	TOL-0.2	TOL-0.1	TOL-0.05	TOL-0.01	TOL-0.001
Coverage	1506	1498	1494	1494	1306	1444	1505	1548	1530
Cost	128504	128115	127660	129783	111300	119614	120550	131376	-
Memory	35229036	34645528	36120352	38496812	46611520	35180276	33295648	36588732	-
Search Time	0.15	0.15	0.14	0.14	0.62	0.26	0.17	0.19	-

TOL vs TOL_{PL2,VC2} (PV2)

	with reopening (1747)								
Results	TOL-0.2	TOL-0.1	TOL-0.05	TOL-0.01	PV2-0.3	PV2-0.2	PV2-0.1	PV2-0.05	PV2-0.01
Coverage	1309	1444	1504	1548	1325	1407	1477	1510	1544
Cost	111524	119798	120797	131521	113474	115080	120943	122721	129729
Memory	46482632	35343332	33600284	36634580	47783252	39184080	36043580	34637792	37309104
Search Time	0.75	0.32	0.22	0.19	0.79	0.45	0.29	0.22	0.19

Comparison

With reopening (2692)				
Results	PV2-0.01	TOL-0.01	OL-0.3	VC3-0.01
Coverage	1964	1997	1878	1976
Cost	94722	94361	99130	94670
Memory	47105748	44974748	46687472	47491860
Search Time	0.47	0.46	0.54	0.54

Conclusion

- TOL needs small fixed parameter
- TOL with exploration can surpass OL with exploration
- TOL can make better use of exploration
- Fixed exploration is not necessarily the best way to balance exploration
- Attempts to prune might be useful (in paper attempts with Median Elimination)

University of Basel

Questions?

Appendix

- Removed parcprinter domains (both in OPT and SAT tracks)
- Did not know importance of SAT until much later
- Additional method attempting to prune using PAC-bound algorithm, Median Elimination (ME)
- Further experiments made with just TOL_{VC3} (not in paper)

Pruning with PAC-bound algorithms

- Probably approximately correct algorithm

- Solutions to MAB

- Eliminate actions until one is left

- Input : error e, confidence c
- Output : one action (e-optimal with confidence c)

TOL with ME:
- Wait until all children have been visited at least so many times
- remove actions below median of expected rewards
- dont remove best actions (in terms of heuristic)

Pruning with PAC-bound algorithms

Algorithm 7 ME(error,delta) - [2]

$l \leftarrow 1, error_l \leftarrow error/4, delta_l \leftarrow delta/2$

while $|A| > 1$ **do** ▷ While there is more than one action left

Sample all actions $a \in A$ $\dfrac{1}{(error_l/2)^2 \log 3/delta_l}$-times

calculate the average of each action μ_a

if $\mu_a < Med\{\mu_a | a \in A\}$ **then**

remove a from A

end if

$l \leftarrow l + 1, error_l \leftarrow error * 3/4, delta_l \leftarrow delta/2$

end while

return best action a