Exploiting Search Space Structure in Classical Planning: Analyses and Algorithms

Masataro Asai, Graduate School of Arts and Sciences,
The University of Tokyo
2 yrs remaining

0. Prior Work:

- 1st, 2nd paper: macro
- 3rd paper: A*, tiebreaking, plateau
- →Weak connections of topics; Requires a unified story

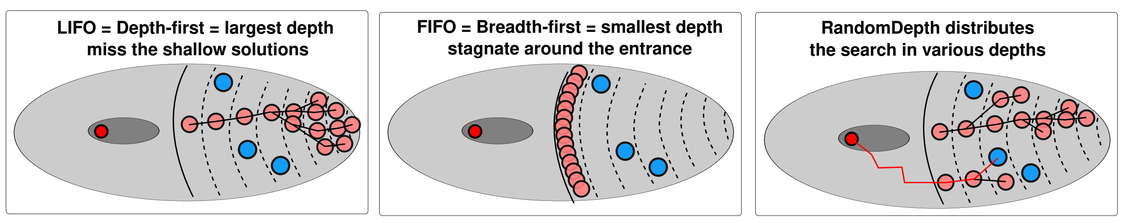
(macro ∩ plateau analysis) == search space

1. *Effective* search space (ESS)

- The portion of the search space evaluated
- Search algorithm affects the ESS
- Heuristic function, tiebreaking, macro operators

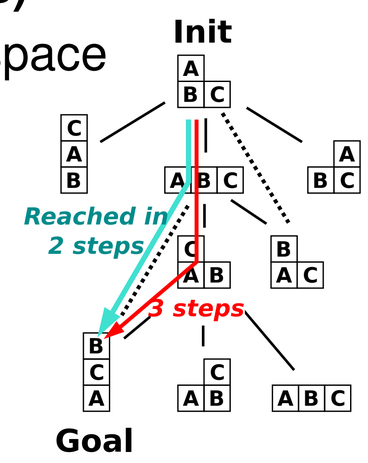
Random-Depth Tiebreaking (3rd paper)

- Search the plateau more uniformly

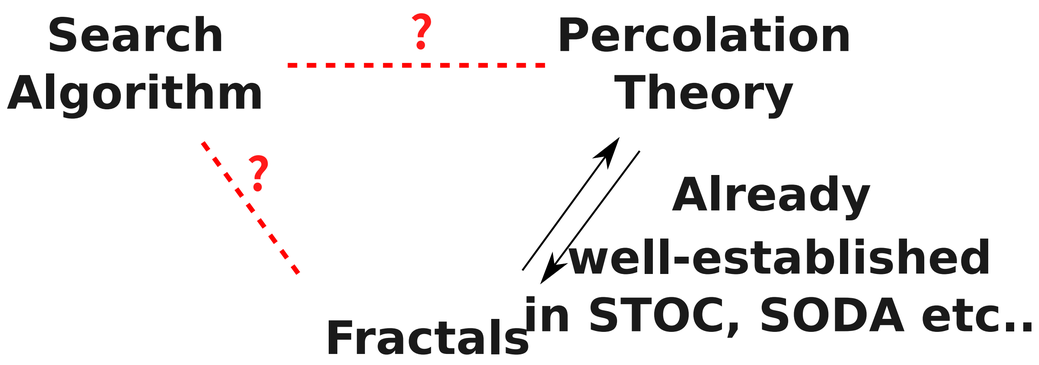


Macro operators (1st, 2nd papers)

- Creates shortcuts in the search space
- Useful when it guides the search
- Increases the branching factor in the search space
- Increases the branching factor **in the ESS → ?**

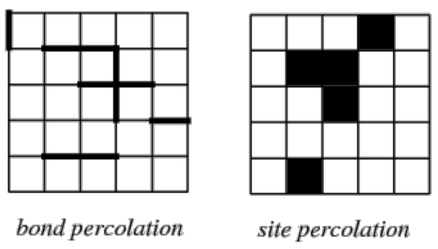


2. Framework for Analysing ESS



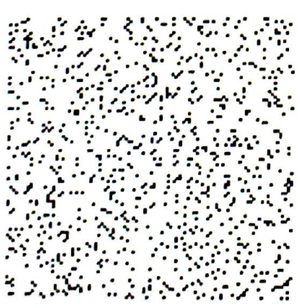
3. Percolation Theory (graph connectivity)

A node (edge) is occupied/unoccupied
Occupied node(edge): **black**
Unoccupied node(edge): white

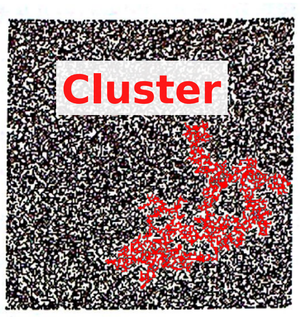


Occupation ratio ***r***
= #(occupied) / # (to

A non-percolated graph:
Obviously disconnected everywhere
r=0.1



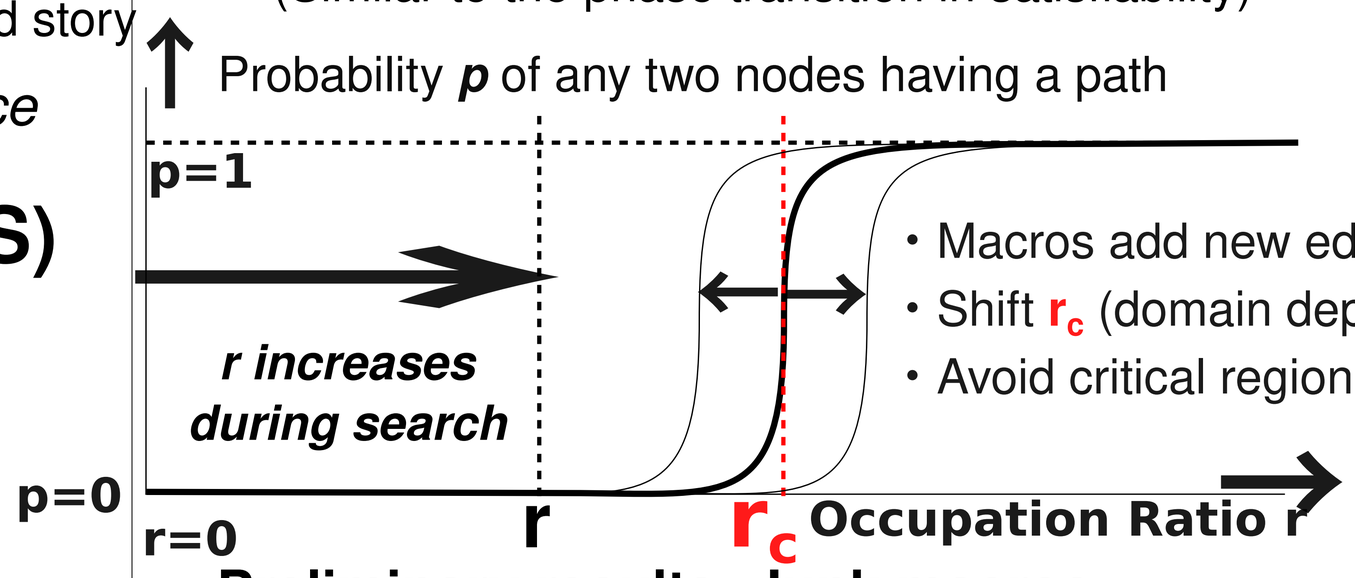
Connectivity non-trivial
$r=r_c=0.5928$ critical value



A percolated graph:
Obviously connected everywhere
r=0.9

(cont.) Percolation = Phase Transition

Around **$r=r_c$**, graph connectivity becomes increasingly difficult to answer
(Similar to the phase transition in satisfiability)



- Macros add new edges
- Shift r_c (domain dep.)
- Avoid critical region?

Preliminary results: Junk macros

(LAMA) Domain	L	Preprocess [sec]	Search [sec]	Total [sec]	Eval [node]
airport	8	*112 (1.1)*	**355 (.50)**	**467 (.57)**	**280721 (.74)**
cybersec	8	**2217 (.91)**	3	**2220 (.91)**	3309
depot	8	22 (1.3)	149 (.50)	**171 (.54)**	**190577 (.47)**
driverlog	5	*24 (1.3)*	*105 (1.6)*	*129 (1.5)*	**179752 (.88)**
hanoi	2	3 (1.0)	**287 (.79)**	**290 (.79)**	2070986 (.97)
mystery	5	*87 (1.4)*	**4 (.21)**	*91 (1.1)*	**2643 (.08)**
pipesworld-t	8	*304 (1.5)*	*893 (2.1)*	*1197 (1.9)*	**355576 (.89)**
rovers	2	*331 (1.1)*	114 (.96)	*445 (1.0)*	**87475 (.90)**
transport-sat11	2	*205 (1.3)*	*630 (2.0)*	*835 (1.8)*	**47244 (.47)**

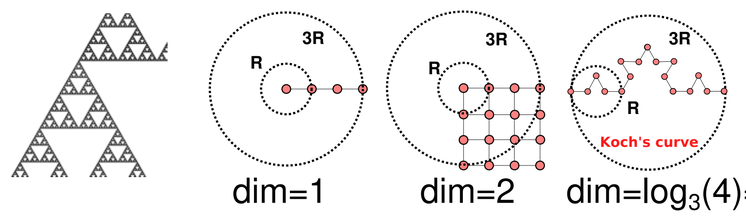
- Randomly generated junk macros
- Conventionally considered harmfu
- Actually reduces evaluations in some domains

Open Questions / new methods

- How existing macros change ESS?
- r_c of each planning domain? (e.g. Logistics=0.XX)
- When each algorithm finds a soltion? (e.g. r=0.YY)
  - e.g. Local search, lookaheads, type-GBFS
- Random restarts with randomly ignoring edges
  - Shifts r_c
  - Multiple incomplete runs
  - → Probabilistically complete algorithm

4. Analyze ESS as Fractals

Measuring the Fractal dimension of a graph

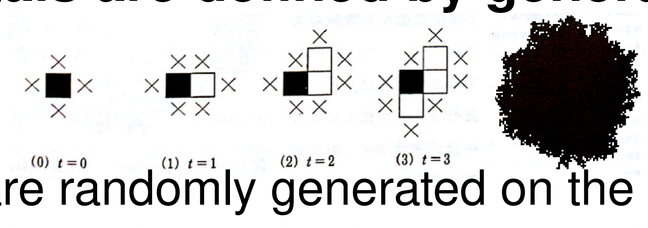


dim=1 dim=2 dim=$\log_3(4)$=1.26

e.g. Radius method
- Change the radius
- Count the nodes

Some fractals are defined by generative rules

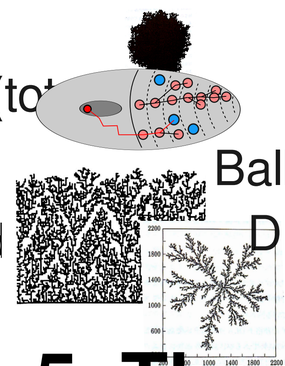
Eden Model



A model for a cell culture

Nodes are randomly generated on the surface
= ESS with random selection from the OPEN list

Fractal Models	↔	Search Algorithm
Eden	↔	Random Order
Fractal?	↔	Random Depth
Ballistic Aggregation	↔	New algorithm?
DLA (dim = 1.71)	↔	New algorithm?



5. Thesis Abstract

1. Propose the framework
2. Propose paper 1,2,3
3. Analyze the ESS of paper 1,2,3 using the framework
   → A deeper insight into why paper 1,2,3 is successfu