The Grid-Based Path Planning Competition

-or-

Contractions, contractions everywhere

Nathan R. Sturtevant
University of Denver

Symposium on Combinatorial Search
June 13, 2015

UNIVERSITY of DENVER
DANIEL FELIX RITCHIE SCHOOL OF ENGINEERING & COMPUTER SCIENCE

Testing Heuristics: We Have It All Wrong

J. N. Hooker
Graduate School of Industrial Administration
Carnegie Mellon University
Pittsburgh, PA 15213 USA

May 1995

Testing Heuristics

- Suppose: bright idea for a new algorithm

Testing Heuristics

- Suppose: bright idea for a new algorithm
- Test on a standard set of benchmark problems

Testing Heuristics

- Suppose: bright idea for a new algorithm
- Test on a standard set of benchmark problems
    - If the new algorithm wins:

Testing Heuristics

- Suppose: bright idea for a new algorithm
- Test on a standard set of benchmark problems
  - If the new algorithm wins:
    - The work is submitted for publication

Testing Heuristics

- Suppose: bright idea for a new algorithm
- Test on a standard set of benchmark problems
  - If the new algorithm wins:
    - The work is submitted for publication
  - Otherwise it is written off as a failure

Testing Heuristics

- Suppose: bright idea for a new algorithm
- Test on a standard set of benchmark problems
  - If the new algorithm wins:
    - The work is submitted for publication
  - Otherwise it is written off as a failure

- Approach “spawns a host of evils”

Complaints

Complaints

- The emphasis on competition is fundamentally antiintellectual
  - Does not build the sort of insight…(for)…more effective algorithms

Complaints

- The emphasis on competition is fundamentally antiintellectual
  - Does not build the sort of insight…(for)…more effective algorithms
- Competition diverts time and resources from productive investigation
  - Hours spent crafting the fastest possible code

Example

- 2nd DIMACS challenge
  - Studied SAT problems
  - Stimulated great interest
  - Difficult to know why one approach is better

- Studies which tease out why approaches work without performance increase are hard to publish, but important intellectually

Conclusion

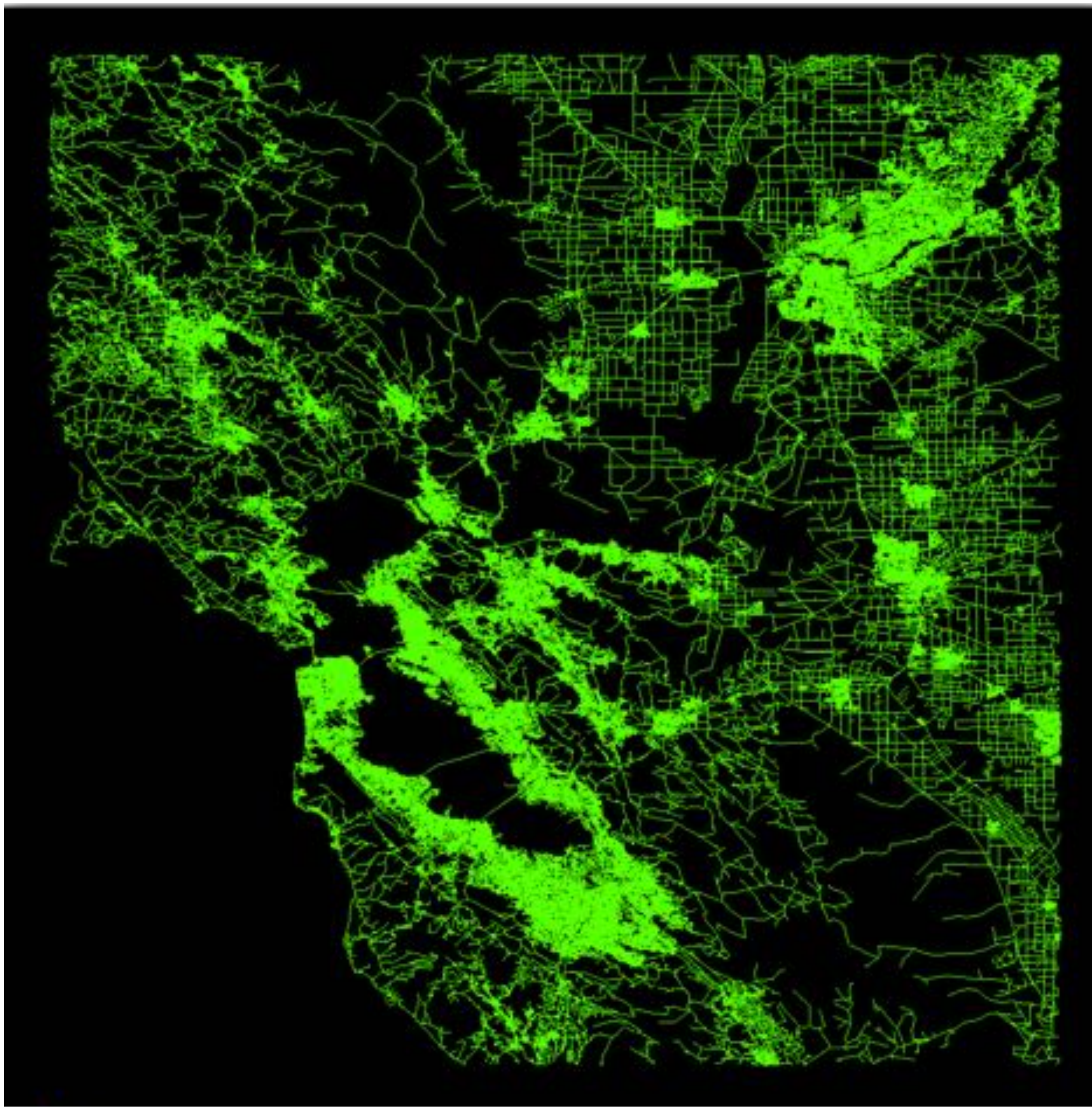
- New norms for research:
  - “That experimental results be evaluated on the basis of whether they contribute to our understanding rather than whether they show that the authors algorithm can win a race with the state of the art.”
  - Algorithm researchers shouldn’t have the “burden of exhibiting faster and better algorithms in each paper.”

Talk Goals

- Present latest GPPC results
- Present insights into commonalities in approaches
- Discuss future challenges

Recent (related) example

- 9th DIMACS challenge
  - Begun in 2005
  - Path planning on road networks



Result of Challenge

- Road networks for testing made available
  - 200k to 24 million nodes
  - 700k to 60 million edges

Result of Challenge

- **Better Heuristics:**
  - Computing the shortest path: A search meets graph theory [Goldberg & Harrelson, 2005]
- **Better Bounding:**
  - Better Landmarks Within Reach [Goldberg et. al., 2007]
- **Exploiting Structure:**
  - In Transit to Constant Time Shortest-Path Queries in Road Networks [Bast et. al. 2007]
  - Contraction Hierarchies: Faster and Simpler Hierarchical Routing in Road Networks [Geisberger et. al. 2008]
  - A Hub-Based Labeling Algorithm for Shortest Paths in Road Networks [Abraham et. al., 2011]
- **Theoretical Justification:**
  - Highway Dimension, Shortest Paths, and Provably Efficient Algorithms [Abraham et. al., 2010]

Observation

Benchmark data spurs research

Observation

Benchmark data spurs research

Imperfect benchmarks are better than no benchmarks.

Observation

Benchmark data spurs research

Imperfect benchmarks are better than no benchmarks.

You have to have (flawed) benchmarks before
you can fix/improve them.

Grid Map Benchmarks

- Between Fall 2006 - Dec. 2008 consulted with BioWare on Dragon Age: Origins

Grid Map Benchmarks

- Between Fall 2006 - Dec. 2008 consulted with BioWare on Dragon Age: Origins
  - They agreed to allow free distribution of map data!

Grid Map Benchmarks

- Between Fall 2006 - Dec. 2008 consulted with BioWare on Dragon Age: Origins
  - They agreed to allow free distribution of map data!
- Combined with other sources to form the “movingai” grid map repository

Grid Map Benchmarks

- Between Fall 2006 - Dec. 2008 consulted with BioWare on Dragon Age: Origins
  - They agreed to allow free distribution of map data!
- Combined with other sources to form the “movingai” grid map repository
  - [Sturtevant, 2012]

Grid-Based Path Planning Competition

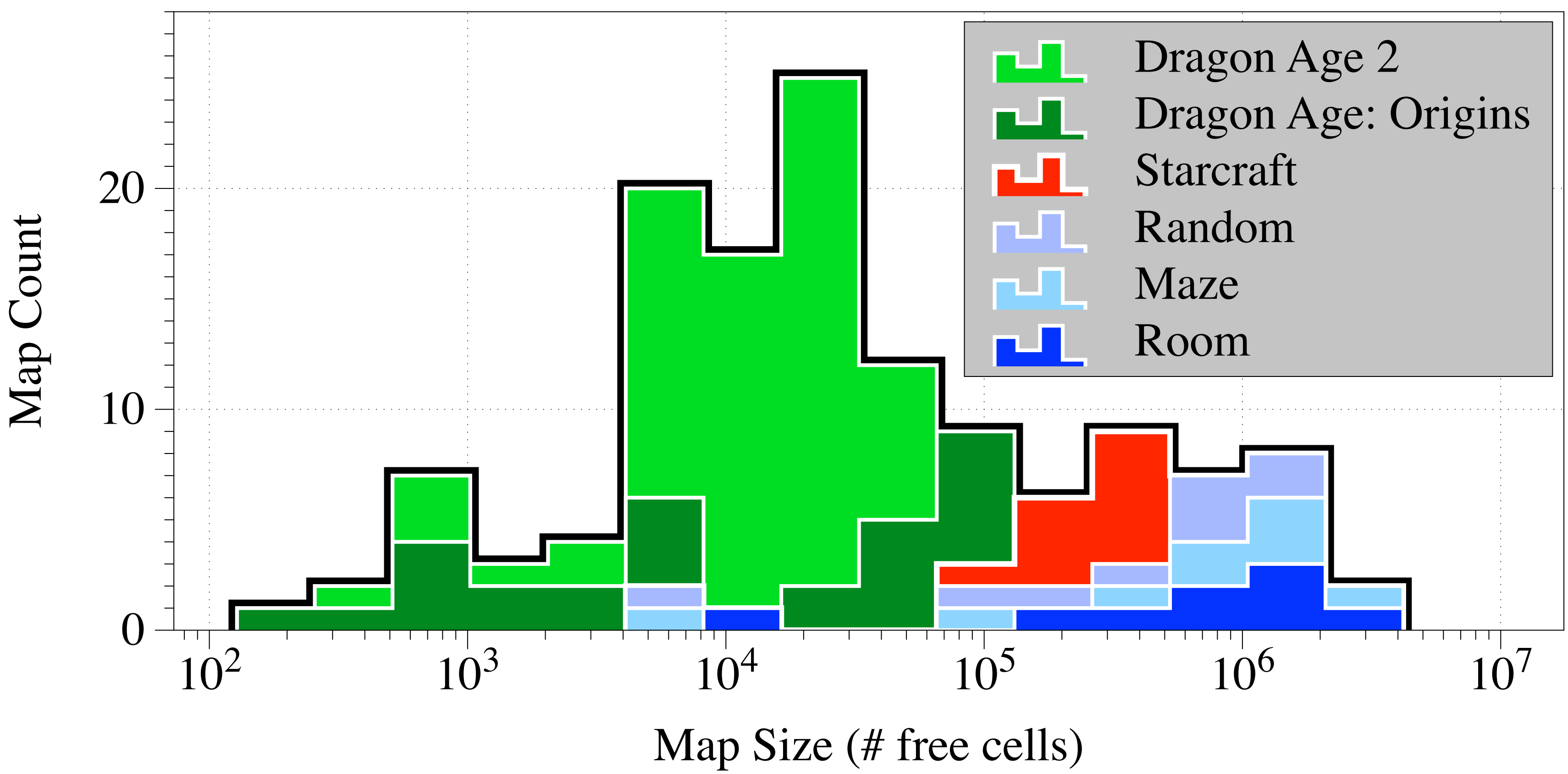
- Started in 2012; goals to:
  - Improve evaluation
  - Establish metrics
  - Improve comparisons

Entry Specification

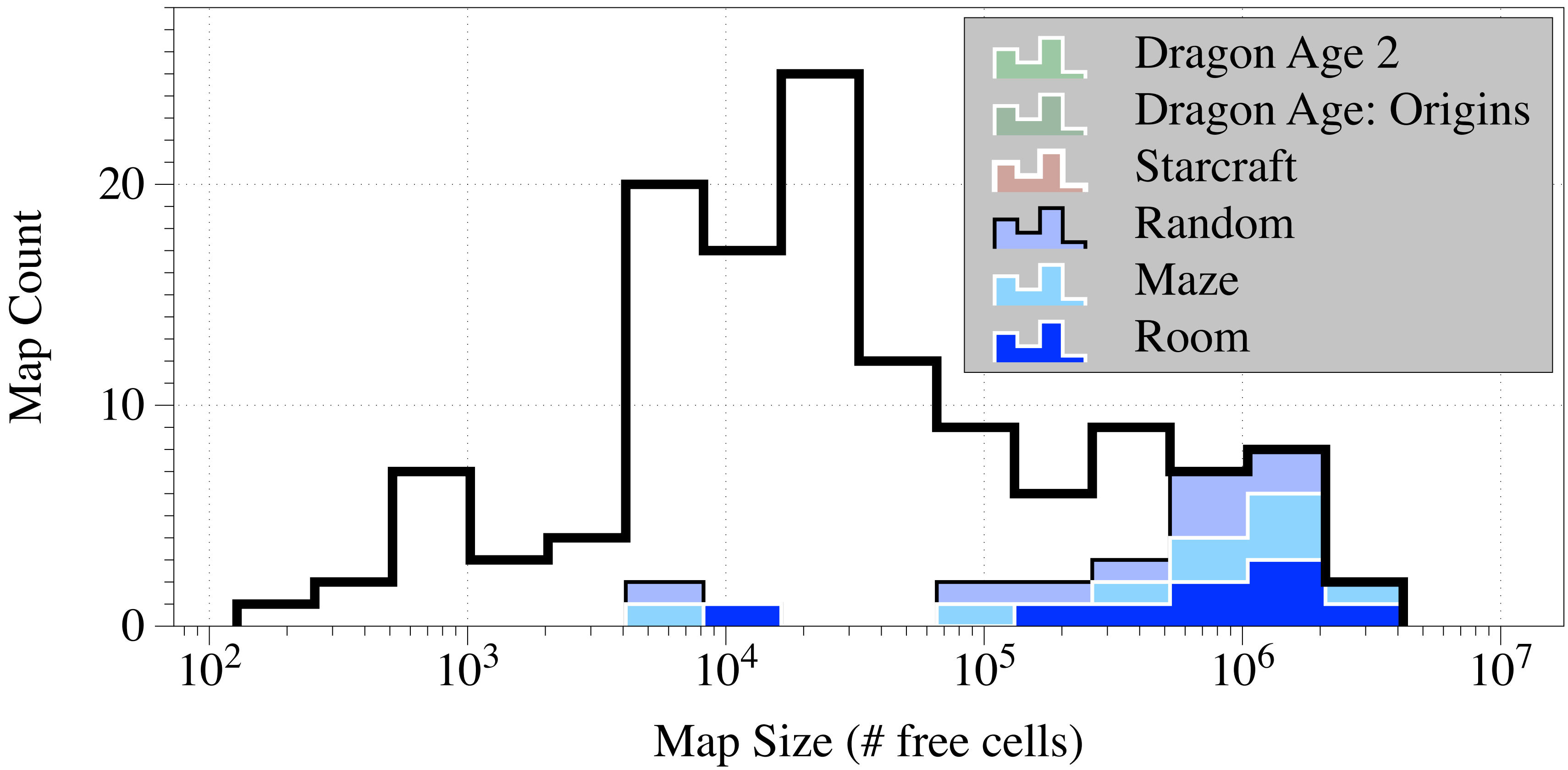
- Time allowed for pre-processing
- At runtime:
  - Load pre-processed data
  - Path query (start, goal) given to entry
    - Entry returns full or partial path
    - Query repeated until full path returned
- Run 5 times for statistical significance

Problem Setup

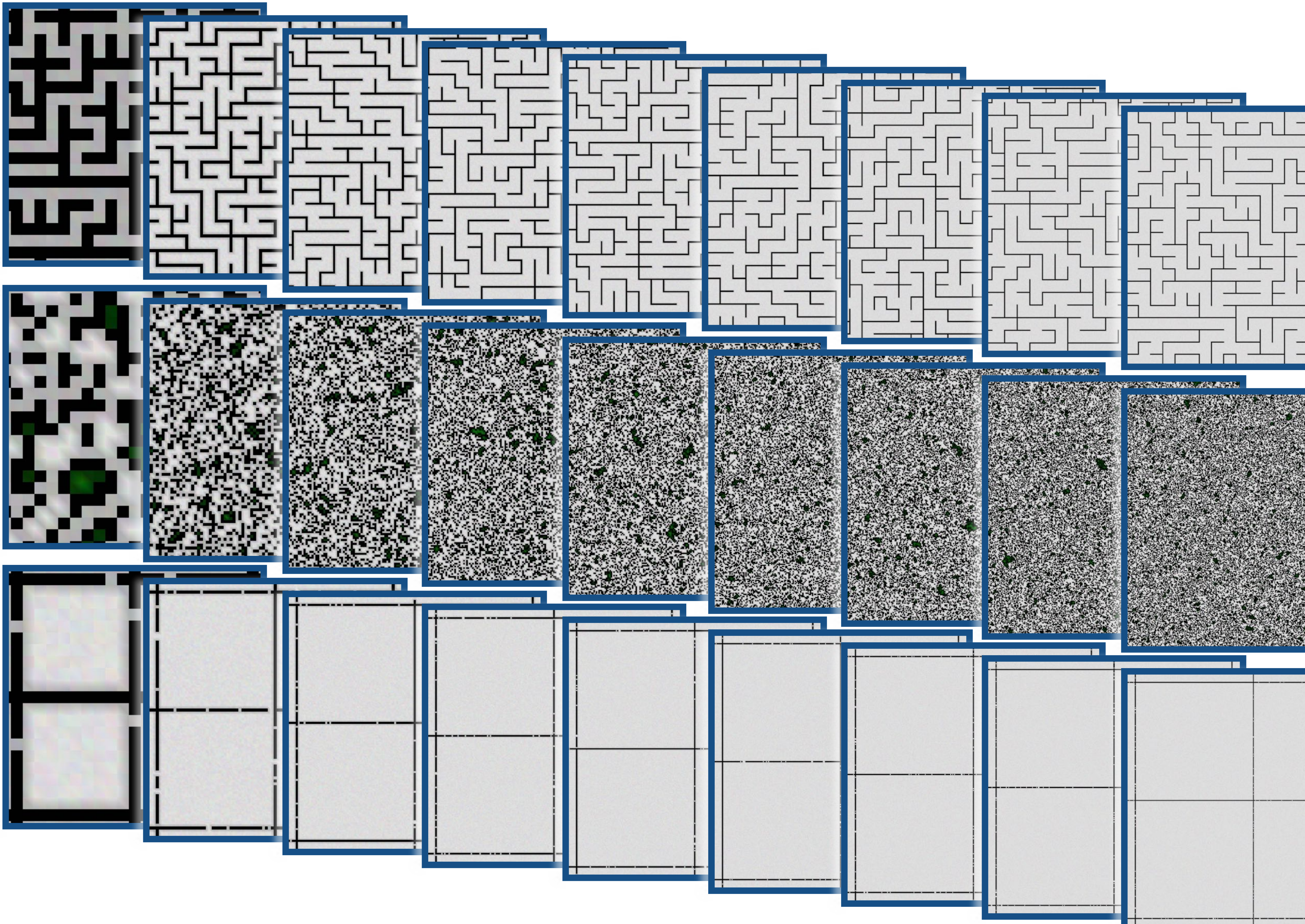
Source	# Maps	# Problems
Starcraft	11	29,970
Dragon Age: Origins	27	44,414
Dragon Age 2	57	54,360
Mazes	18	145,976
Random	18	32,228
Rooms	18	27,130
Total	132	347,868

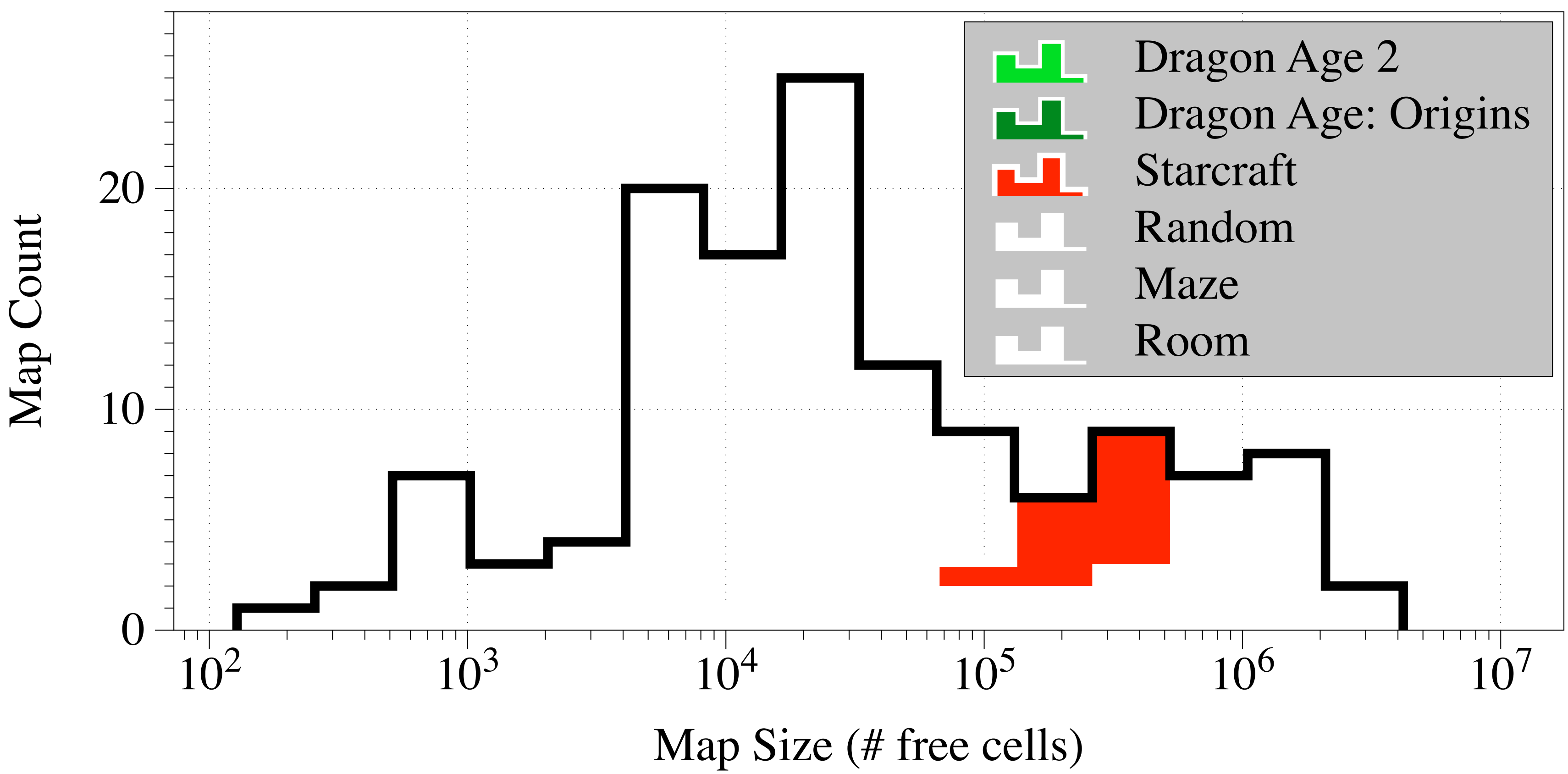


Dragon Age 2
Dragon Age: Origins
Starcraft
Random
Maze
Room
Map Count
0
10
20
10^2
10^3
10^4
10^5
10^6
10^7
Map Size (# free cells)

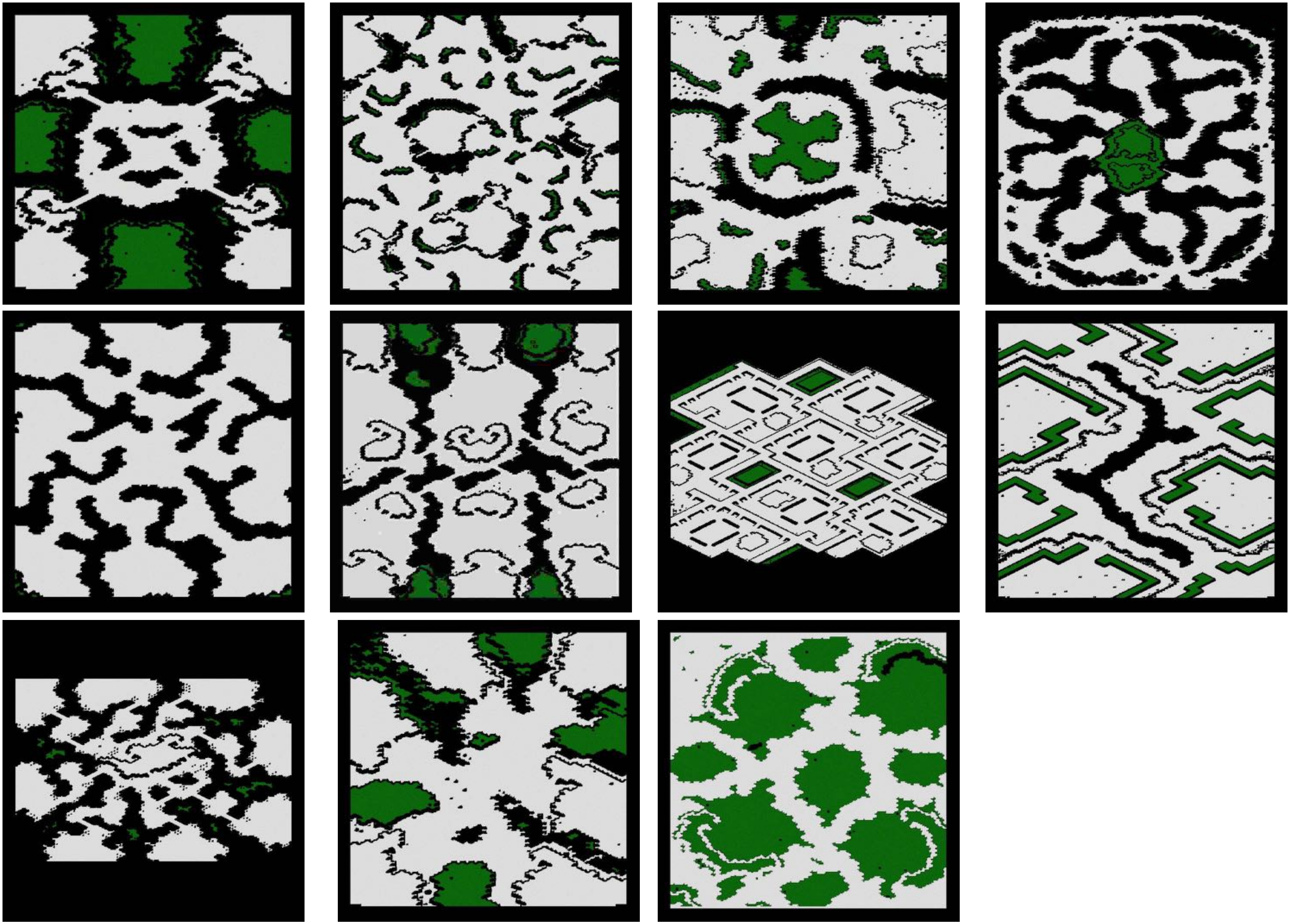


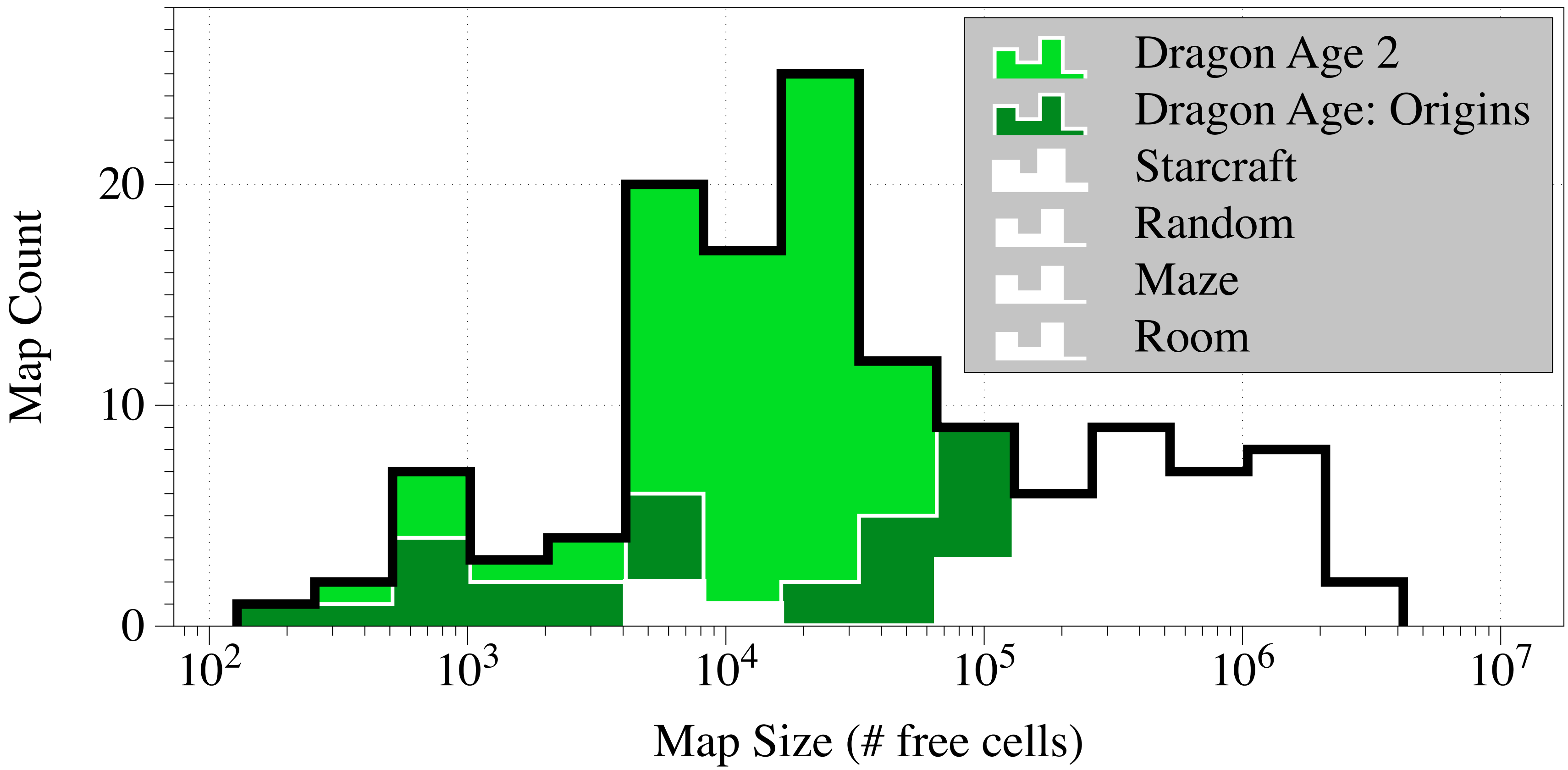
Dragon Age 2
Dragon Age: Origins
Starcraft
Random
Maze
Room
Map Count
0
10
20
10^2
10^3
10^4
10^5
10^6
10^7
Map Size (# free cells)



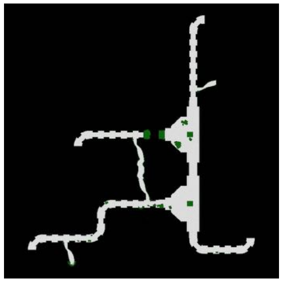
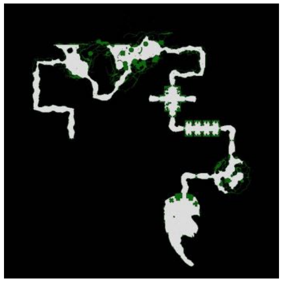
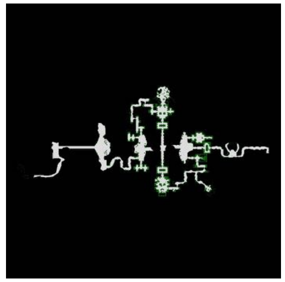
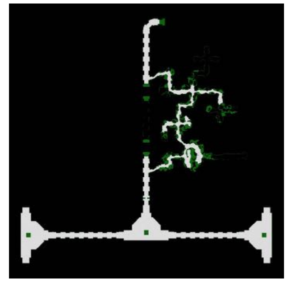
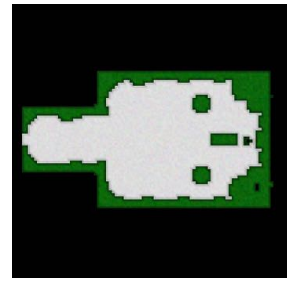
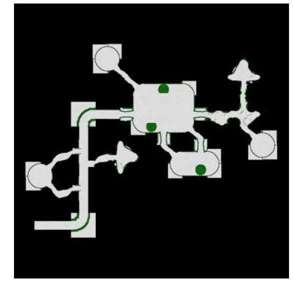
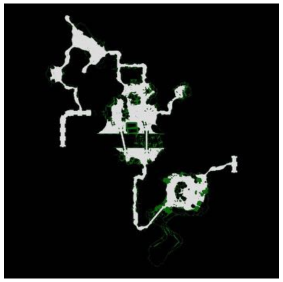
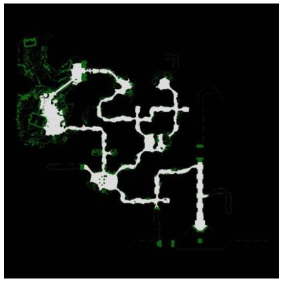
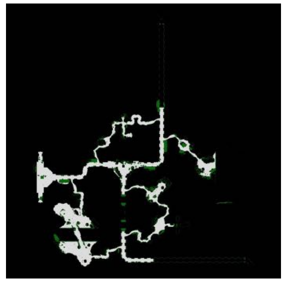
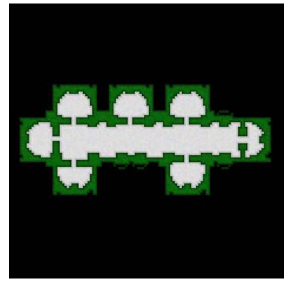
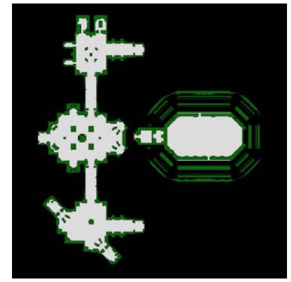
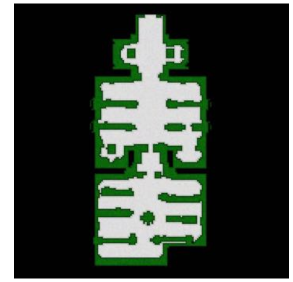
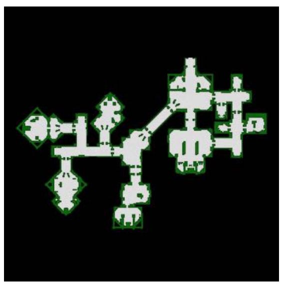
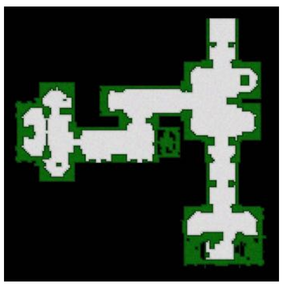
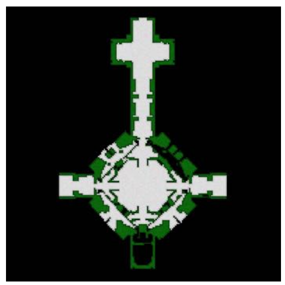
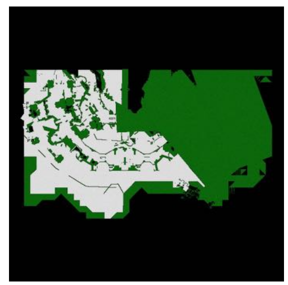
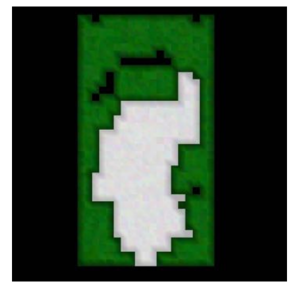
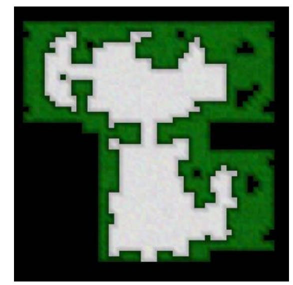
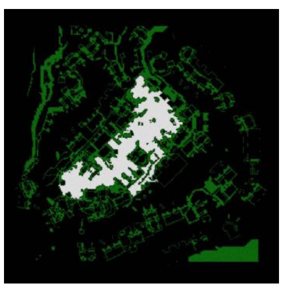
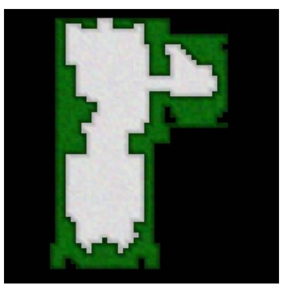
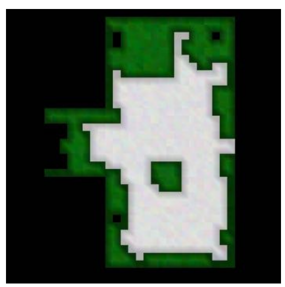
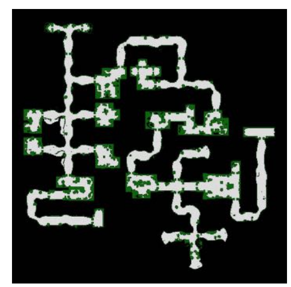
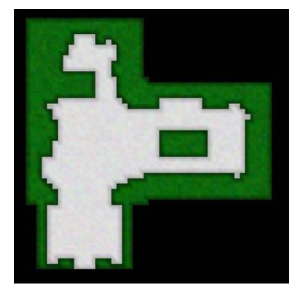
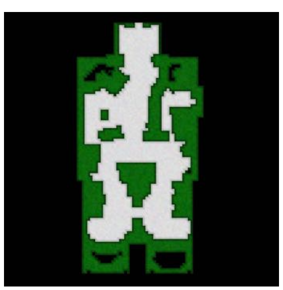
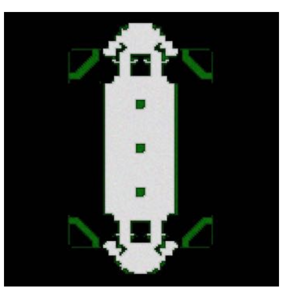


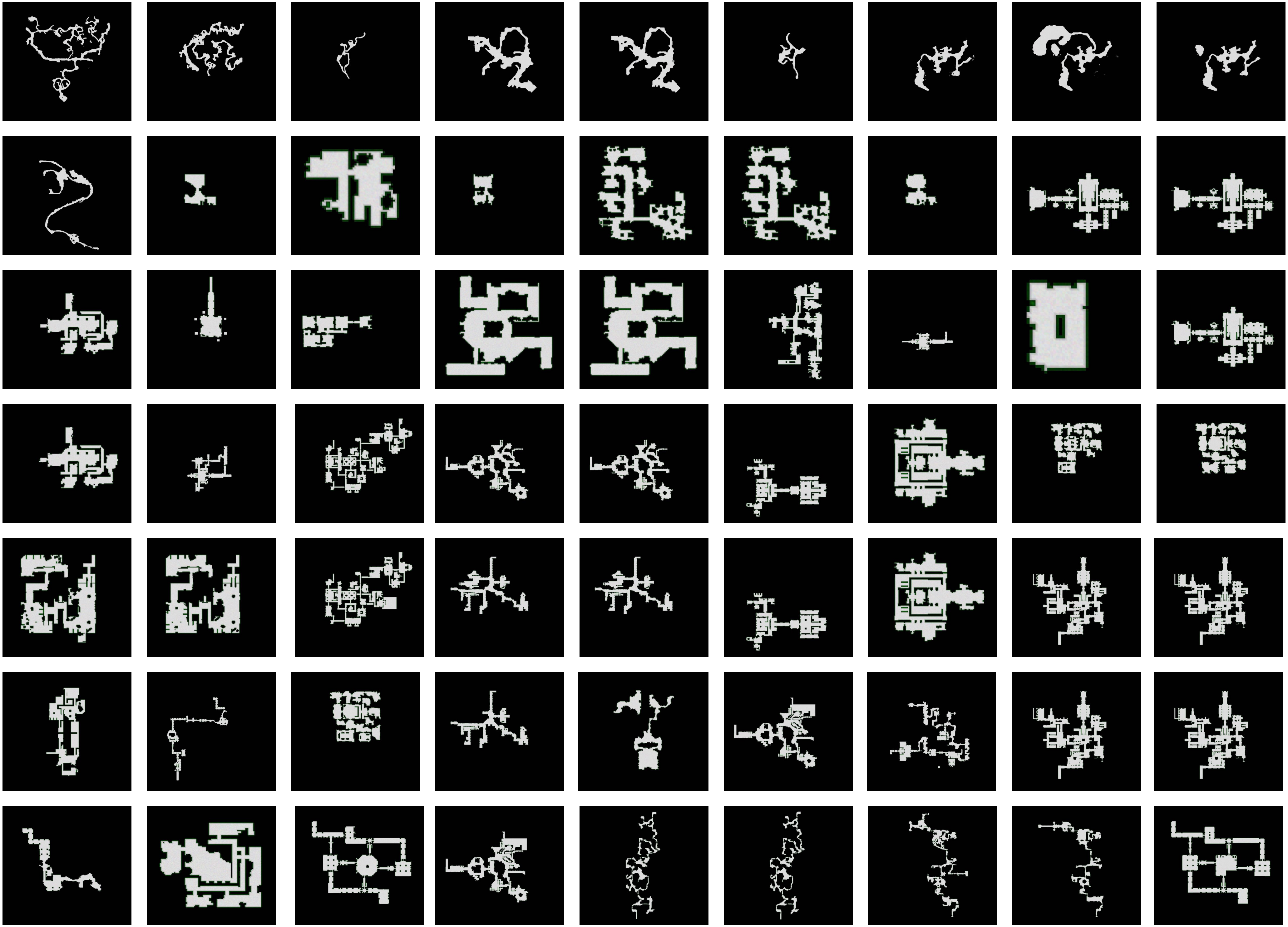
Dragon Age 2
Dragon Age: Origins
Starcraft
Random
Maze
Room
Map Count
0
10
20
10^2
10^3
10^4
10^5
10^6
10^7
Map Size (# free cells)





Dragon Age 2
Dragon Age: Origins
Starcraft
Random
Maze
Room
Map Count
0
10
20
10^2
10^3
10^4
10^5
10^6
10^7
Map Size (# free cells)



Metrics

- Total/average time to solve problem set

Metrics

- Total/average time to solve problem set
- Time for 20 steps (real-time startup)

Metrics

- Total/average time to solve problem set
- Time for 20 steps (real-time startup)
- Maximum segment time (real-time steady state)

Metrics

- Total/average time to solve problem set
- Time for 20 steps (real-time startup)
- Maximum segment time (real-time steady state)
- Suboptimality

Metrics

- Total/average time to solve problem set
- Time for 20 steps (real-time startup)
- Maximum segment time (real-time steady state)
- Suboptimality
- Correctness

Metrics

- Total/average time to solve problem set
- Time for 20 steps (real-time startup)
- Maximum segment time (real-time steady state)
- Suboptimality
- Correctness
- RAM at runtime (before/after)

Metrics

- Total/average time to solve problem set
- Time for 20 steps (real-time startup)
- Maximum segment time (real-time steady state)
- Suboptimality
- Correctness
- RAM at runtime (before/after)
- Disk storage

Metrics

- Total/average time to solve problem set
- Time for 20 steps (real-time startup)
- Maximum segment time (real-time steady state)
- Suboptimality
- Correctness
- RAM at runtime (before/after)
- Disk storage
- Pre-computation

All 2014 Approaches

- Single-Row Compression (SRC) (4 variants)

All 2014 Approaches

- Single-Row Compression (SRC) (4 variants)
- Contraction Hierarchies (CH)

All 2014 Approaches

- Single-Row Compression (SRC) (4 variants)
- Contraction Hierarchies (CH)
- N-level Subgoals

All 2014 Approaches

- Single-Row Compression (SRC) (4 variants)
- Contraction Hierarchies (CH)
- N-level Subgoals
- JPS+, JPS+ Bucket, A* Bucket

All 2014 Approaches

- Single-Row Compression (SRC) (4 variants)
- Contraction Hierarchies (CH)
- N-level Subgoals
- JPS+, JPS+ Bucket, A* Bucket
- BLJPS (2 variants), BLJPS_Subgoal

All 2014 Approaches

- Single-Row Compression (SRC) (4 variants)
- Contraction Hierarchies (CH)
- N-level Subgoals
- JPS+, JPS+ Bucket, A* Bucket
- BLJPS (2 variants), BLJPS_Subgoal
- Relaxed A*, Relaxed A* Subgoal

Single Row Compression

- Ben Strasser [KIT], Adi Botea [IBM Dublin], and Daniel Harabor [NICTA]
  - Build full all-pairs-shortest-path data
  - Run-length encoding to compress each row
  - Incremental or non-incremental path generation

- *Fast First-Move Queries through Run-Length Encoding, Strasser, Harabor and Botea (2014)*
- *Complexity Results for Compressing Optimal Paths, Botea, Strasser and Harabor (2015)*

Contraction Hierarchies

- Ben Strasser [KIT]
  - Originally developed by [Geisberger et. al. 2008]
  - Wasn't optimized for grids [Sturtevant and Geisberger, 2010]
  - Later optimization for grids [Storandt, 2013]

- *Contraction Hierarchies: Faster and Simpler Hierarchical Routing in Road Networks, Robert Geisberger, Peter Sanders, Dominik Schultes and Daniel Delling, 2008*

N-level Subgoals

- Tansel Uras and Sven Koenig [USC]
  - 3rd entry into GPPC
  - Builds a n-level hierarchy from the basic subgoal approach
  - Basic subgoals are built up from visibility graphs

- *Identifying Hierarchies for Fast Optimal Search, Uras and Koenig, 2014*

JPS+

- Steve Rabin [Digipen & Games Industry*]
  - Independently invented JPS+
  - Based on Jump Point Search

- *Improving Jump Point Search, Daniel Harabor and Alban Grastien, 2014*
- *Online Graph Pruning for Pathfinding On Grid Maps, Daniel Harabor and Alban Grastien, 2011*

BLJPS

- Jason Traish and James Tulip [Charles Sturt University]
- Boundary Lookup JPS
- Alternate optimization similar to JPS+
  - Also applied ideas to subgoals

- *Optimization using Boundary Lookup Jump Point Search, Traish and Tulip, 2015*

Relaxed A*

- http://www.iroboapp.org/
- Doesn't perform A* re-expansions & other optimizations
  - Also applied to subgoal search

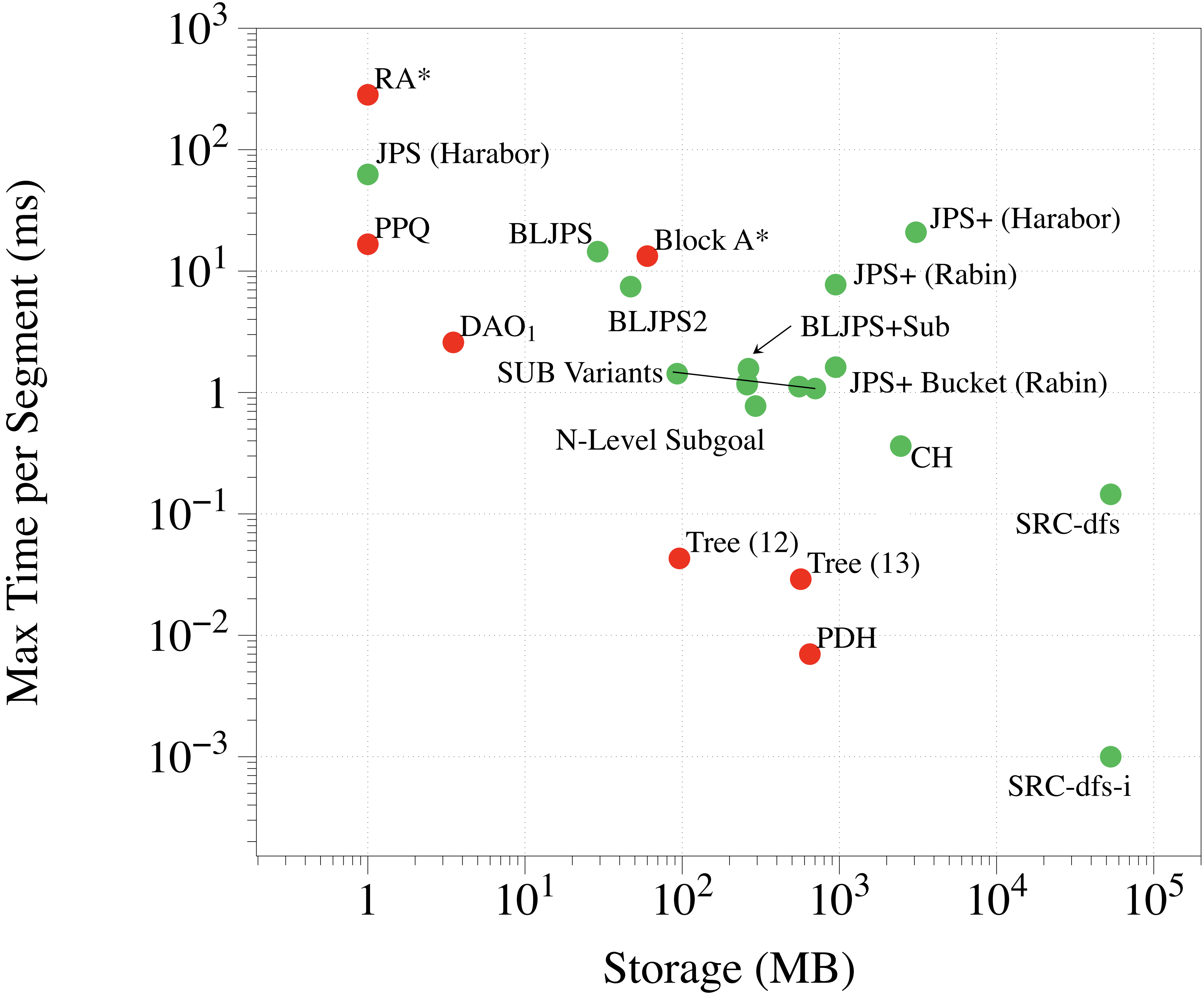
Results

Non-Dominated Approaches

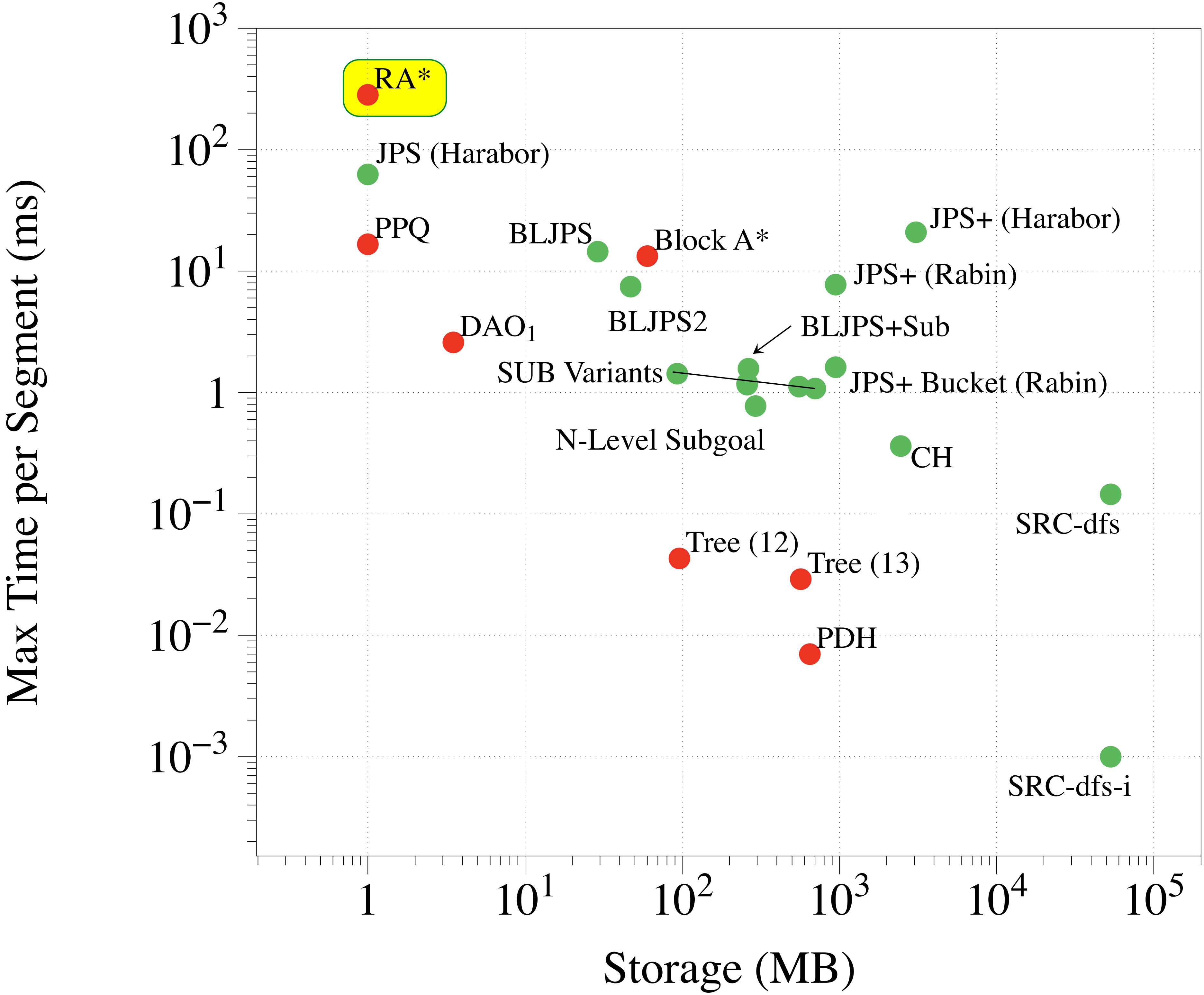
- RA*, RA* Subgoal
- BLJPS, BLJPS2
- N-level Subgoals
- Contraction Hierarchies (CH)
- Single-Row Compression (SRC) (2 variants)

Speed vs Storage (2014)

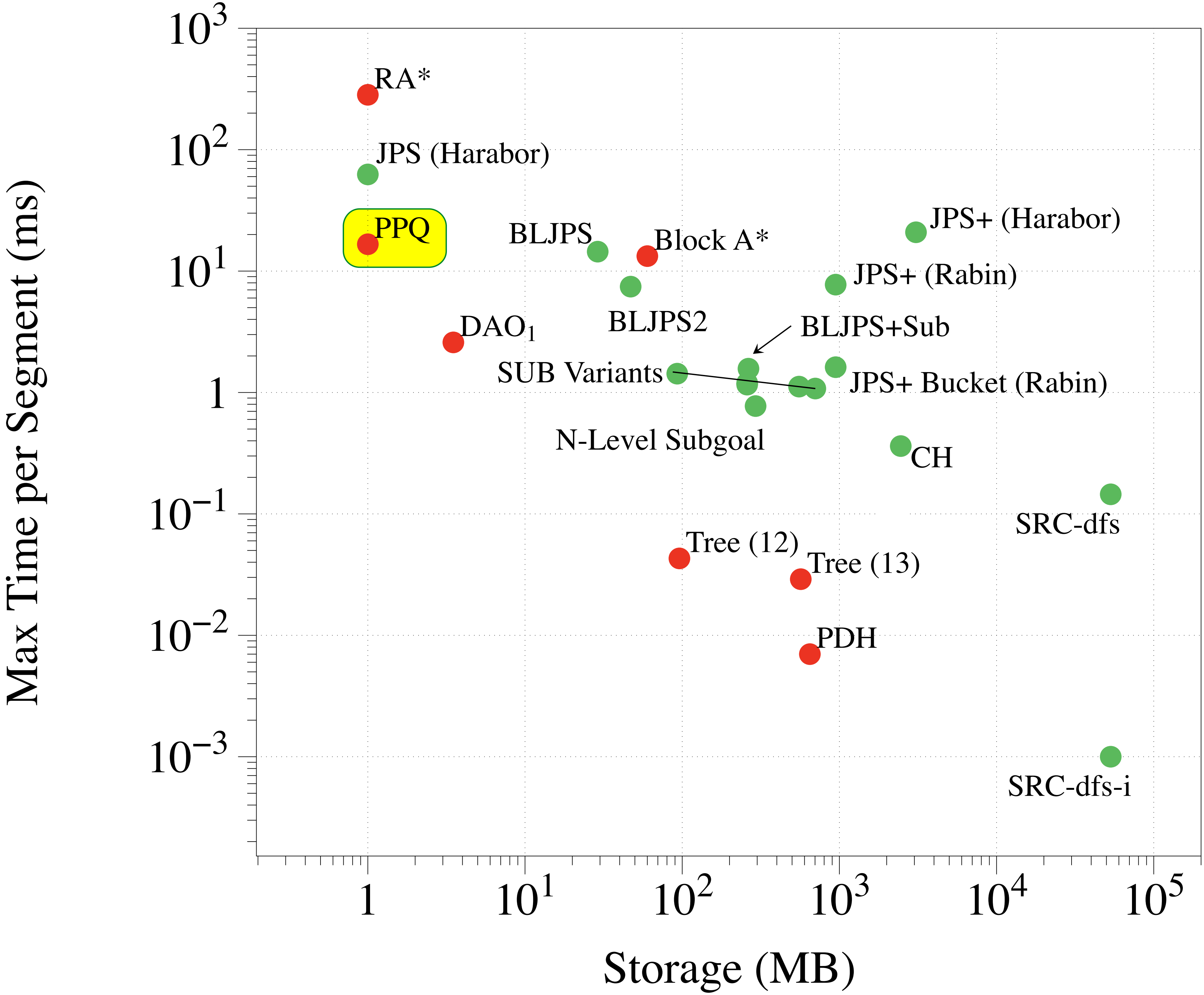
Entry	Average (ms)	Storage
RA*	282.995	0
BLJPS	14.453	20 MB
BLJPS2	7.444	47 MB
RA* Subgoal	1.688	264 MB
NSubgoal	0.773	293 MB
CH	0.362	2.4 GB
SRC-dfs	0.145	28 GB

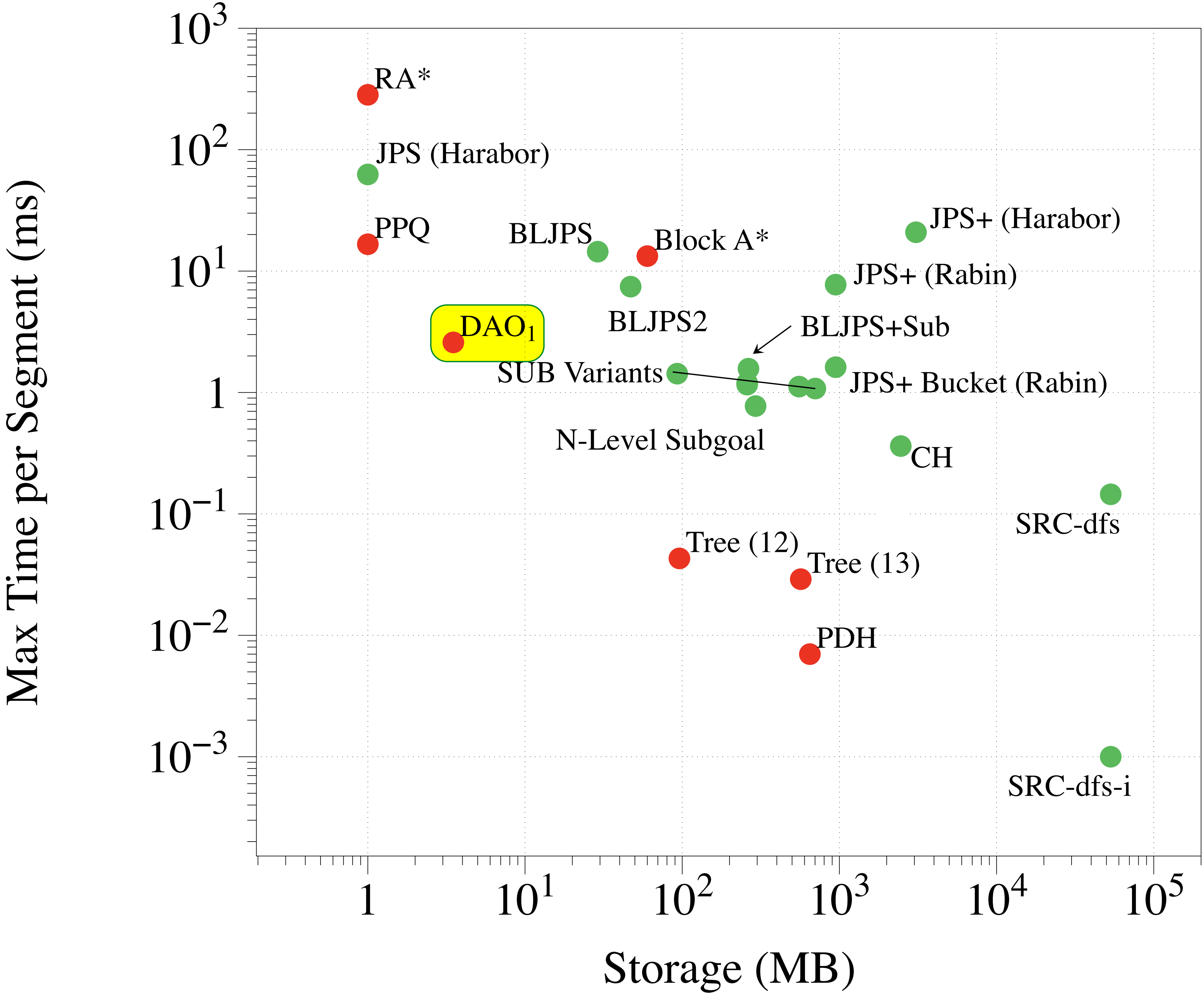


Max Time per Segment (ms)
Storage (MB)
RA*
JPS (Harabor)
PPQ
BLJPS
Block A*
JPS+ (Harabor)
JPS+ (Rabin)
BLJPS2
DAO1
BLJPS+Sub
SUB Variants
JPS+ Bucket (Rabin)
N-Level Subgoal
CH
SRC-dfs
Tree (12)
Tree (13)
PDH
SRC-dfs-i

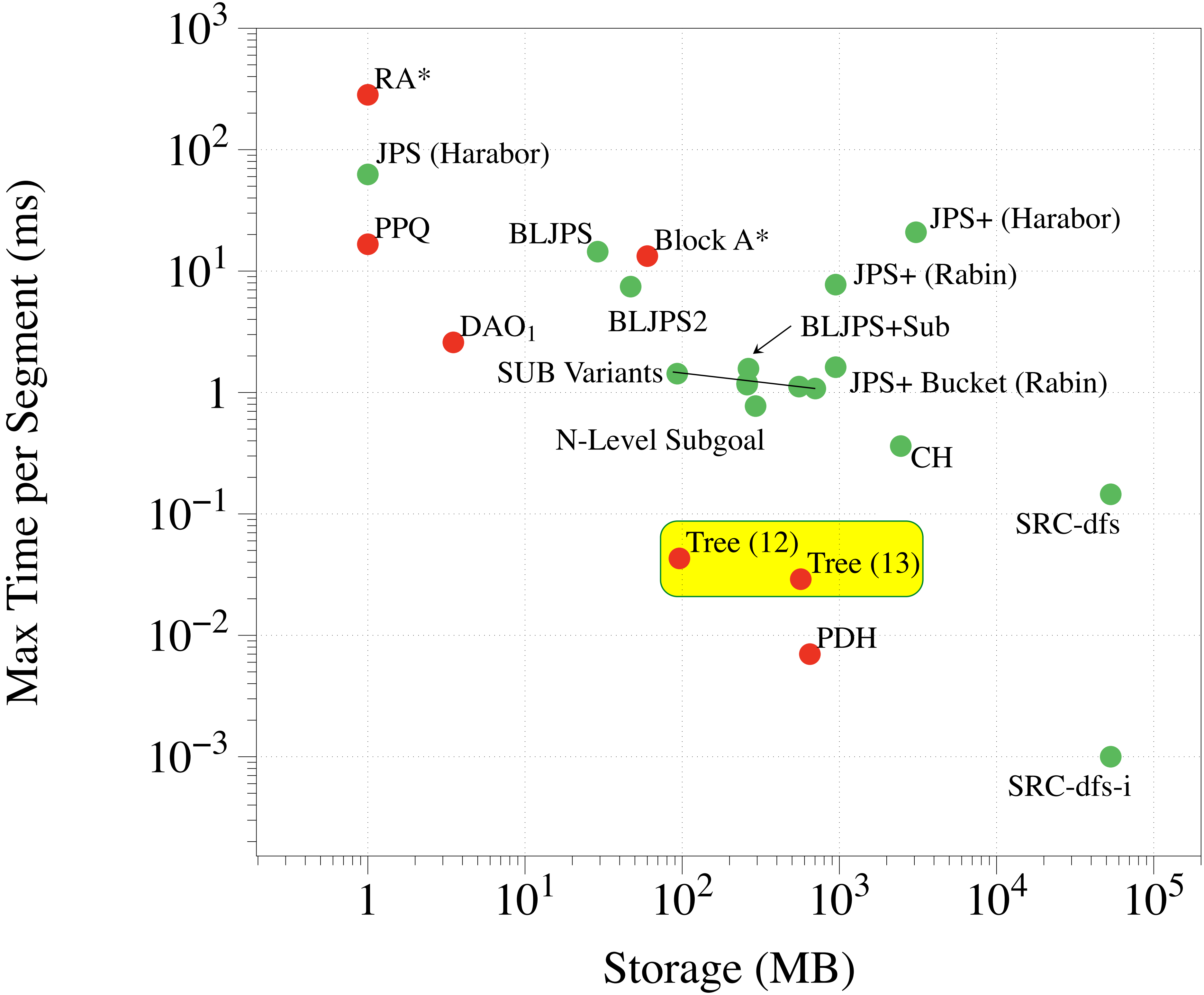


RA*
JPS (Harabor)
PPQ
BLJPS
Block A*
JPS+ (Harabor)
JPS+ (Rabin)
BLJPS2
DAO1
BLJPS+Sub
SUB Variants
JPS+ Bucket (Rabin)
N-Level Subgoal
CH
SRC-dfs
Tree (12)
Tree (13)
PDH
SRC-dfs-i
Max Time per Segment (ms)
Storage (MB)

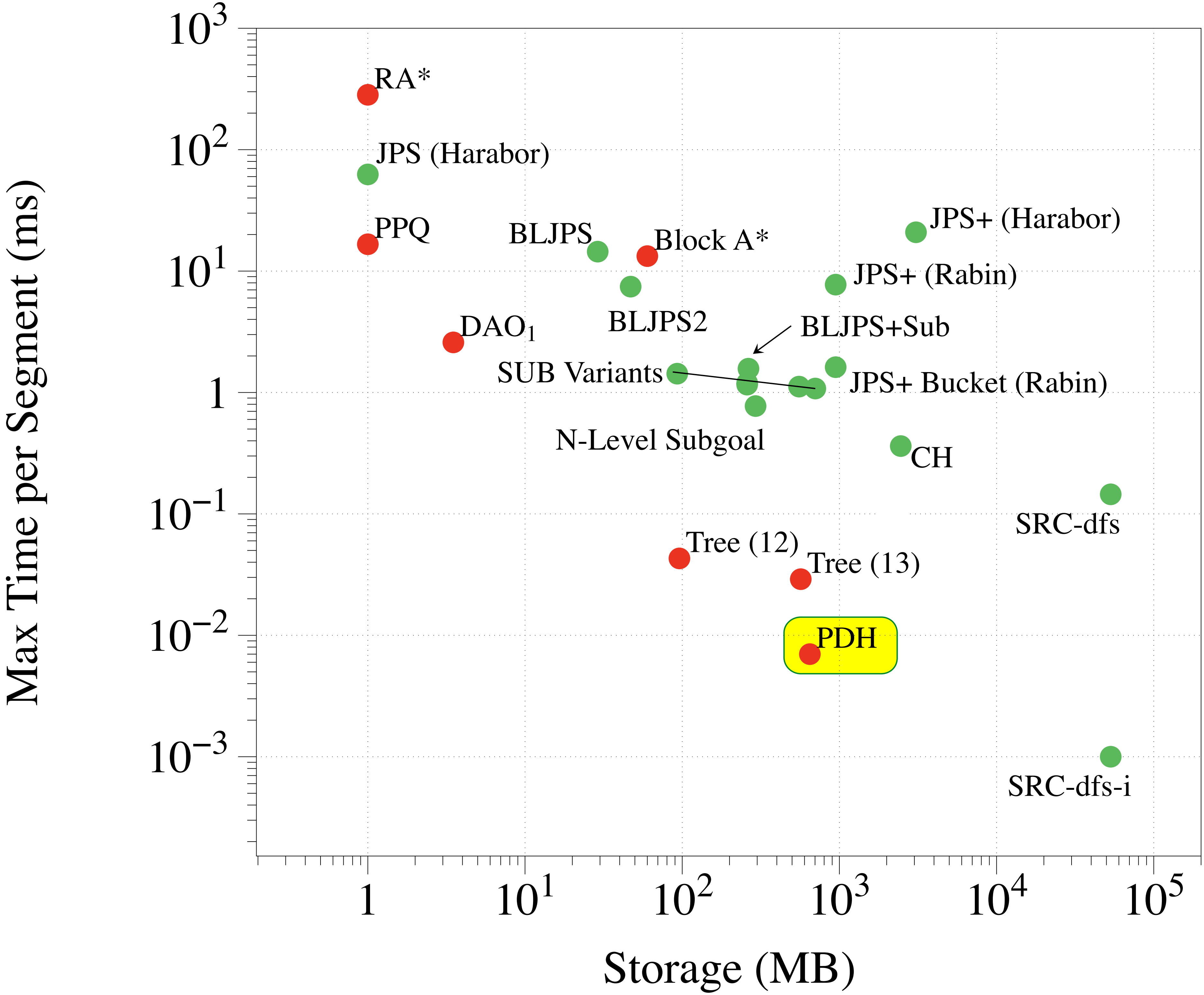

RA*
JPS (Harabor)
PPQ
BLJPS
Block A*
JPS+ (Harabor)
JPS+ (Rabin)
BLJPS2
DAO$_1$
BLJPS+Sub
SUB Variants
JPS+ Bucket (Rabin)
N-Level Subgoal
CH
SRC-dfs
Tree (12)
Tree (13)
PDH
SRC-dfs-i
Max Time per Segment (ms)
Storage (MB)
10^3
10^2
10^1
1
10^{-1}
10^{-2}
10^{-3}
1
10^1
10^2
10^3
10^4
10^5



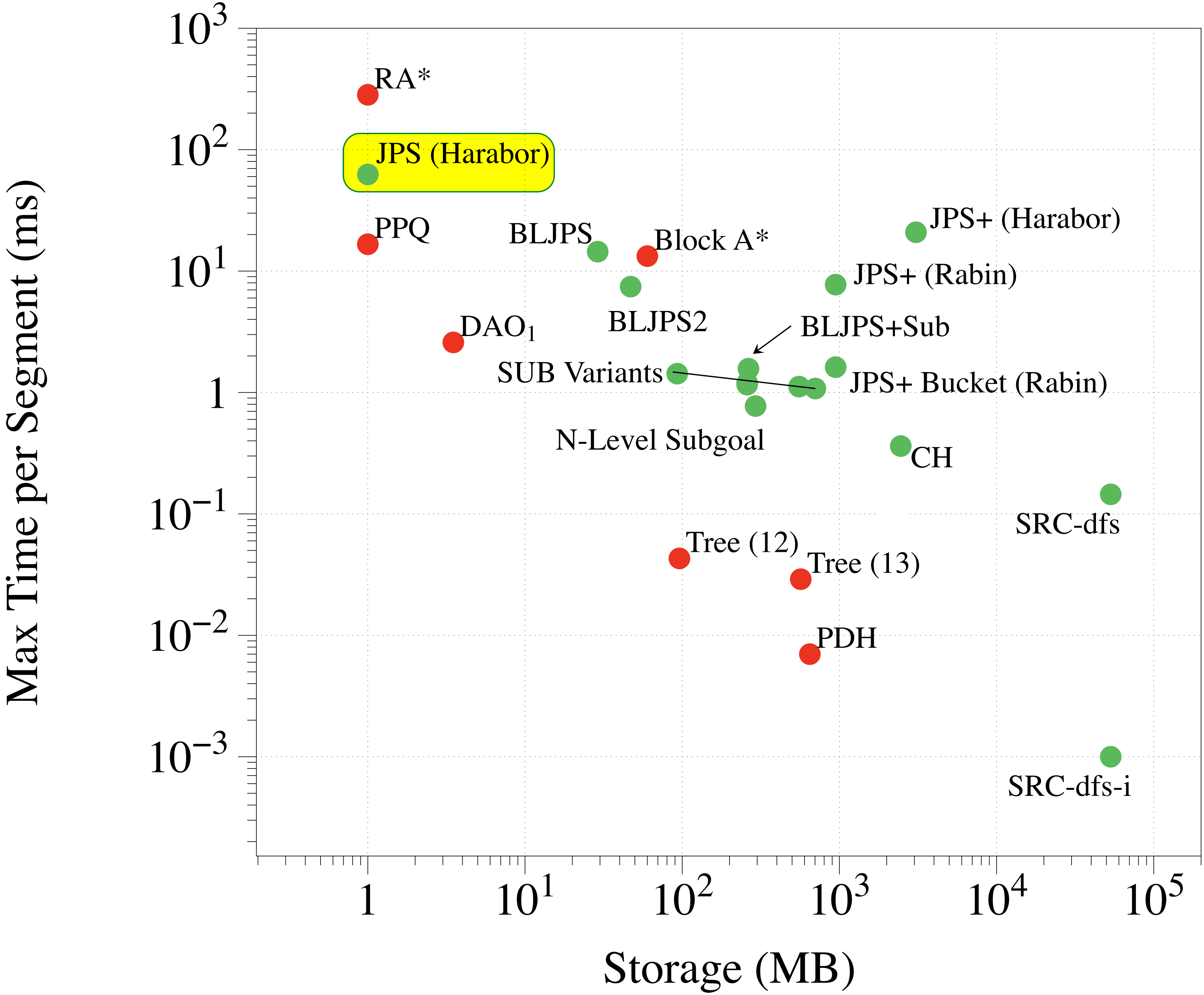
RA*
JPS (Harabor)
PPQ
BLJPS
Block A*
JPS+ (Harabor)
JPS+ (Rabin)
BLJPS2
DAO1
BLJPS+Sub
SUB Variants
JPS+ Bucket (Rabin)
N-Level Subgoal
CH
SRC-dfs
Tree (12)
Tree (13)
PDH
SRC-dfs-i
Max Time per Segment (ms)
Storage (MB)



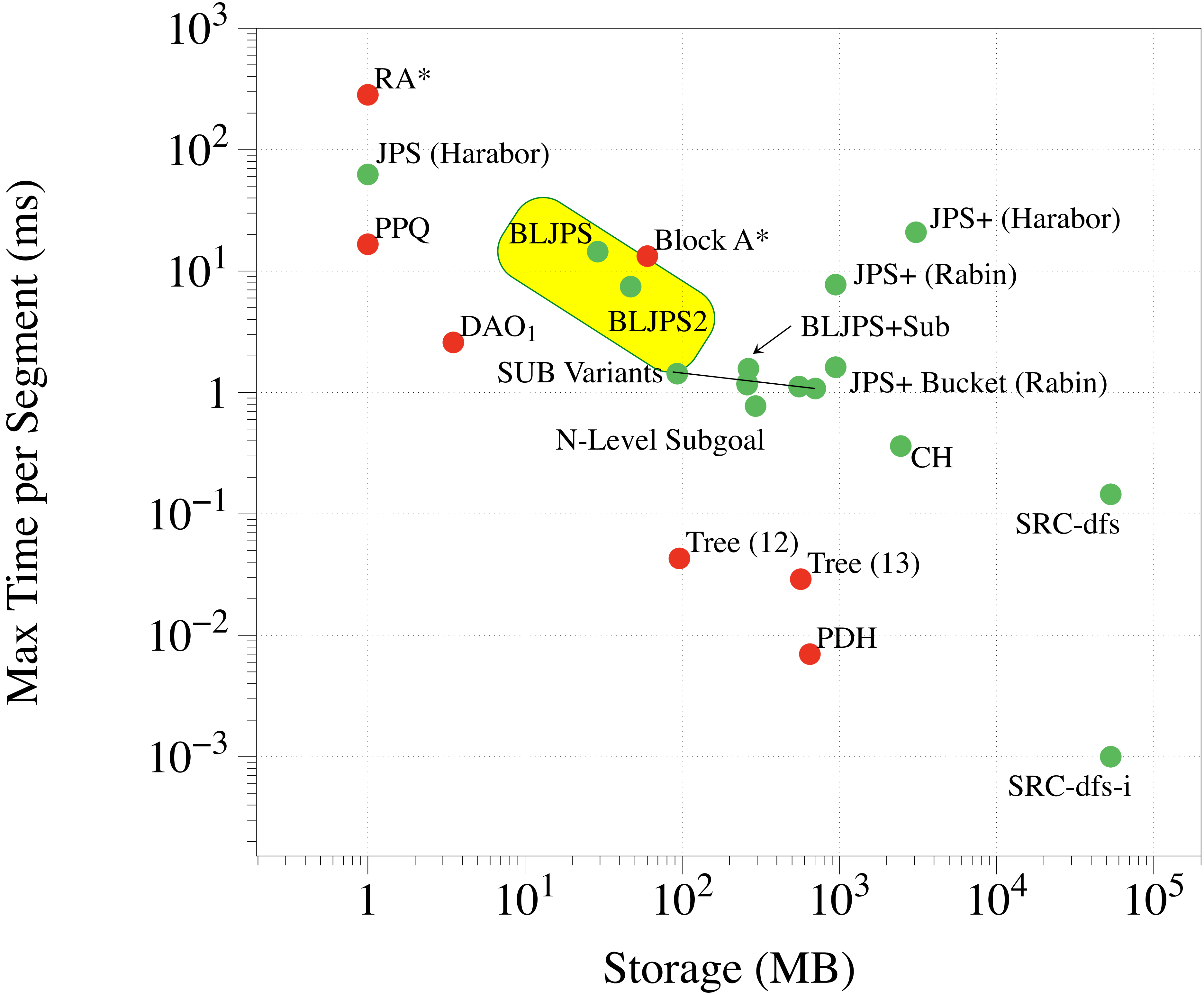
Max Time per Segment (ms)
Storage (MB)
RA*
JPS (Harabor)
PPQ
BLJPS
Block A*
JPS+ (Harabor)
JPS+ (Rabin)
BLJPS2
DAO$_1$
BLJPS+Sub
SUB Variants
JPS+ Bucket (Rabin)
N-Level Subgoal
CH
SRC-dfs
Tree (12)
Tree (13)
PDH
SRC-dfs-i



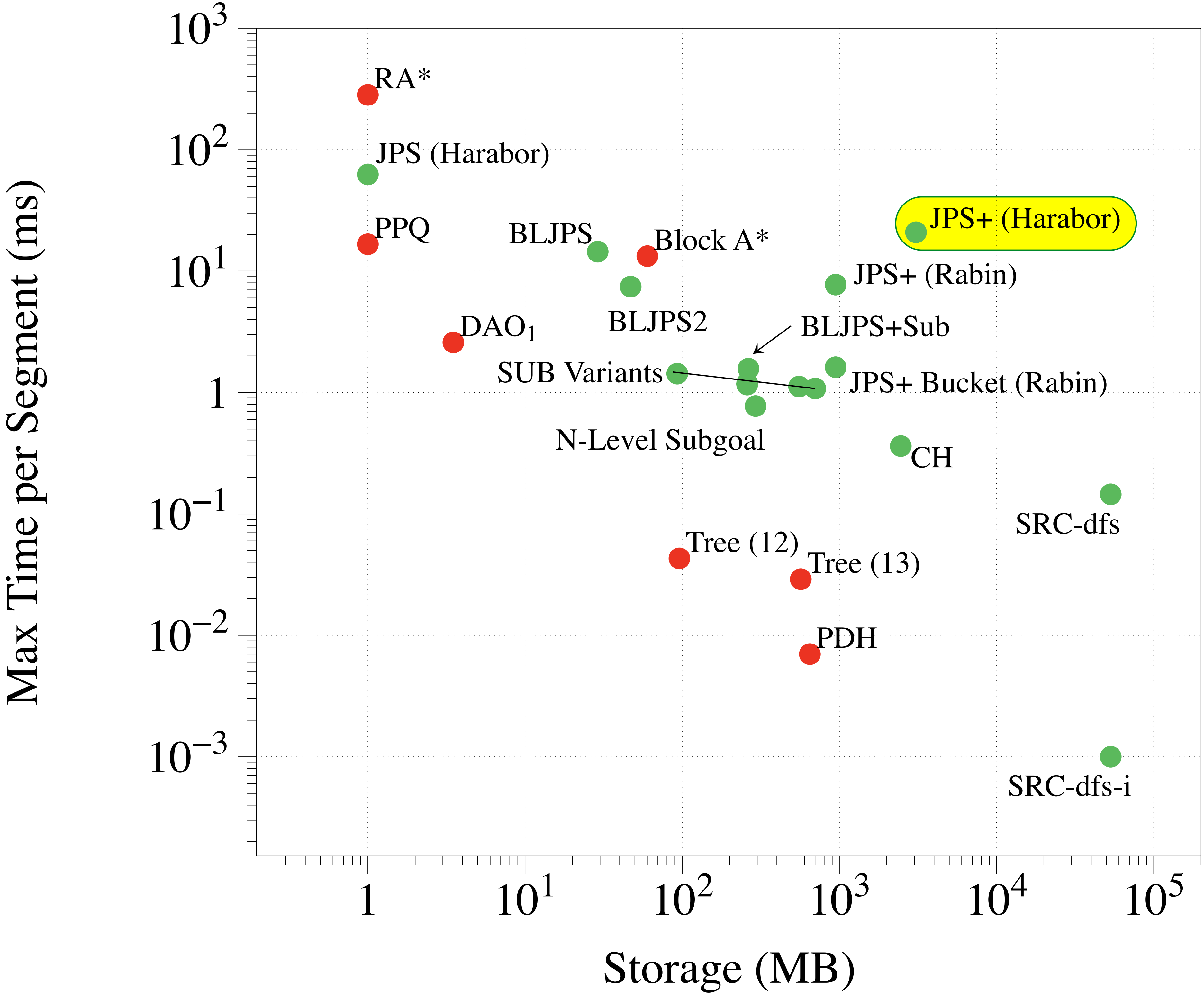
RA*
JPS (Harabor)
PPQ
BLJPS
Block A*
JPS+ (Harabor)
JPS+ (Rabin)
BLJPS2
DAO_1
BLJPS+Sub
SUB Variants
JPS+ Bucket (Rabin)
N-Level Subgoal
CH
SRC-dfs
Tree (12)
Tree (13)
PDH
SRC-dfs-i
10^3
10^2
10^1
1
10^{-1}
10^{-2}
10^{-3}
1
10^1
10^2
10^3
10^4
10^5
Max Time per Segment (ms)
Storage (MB)



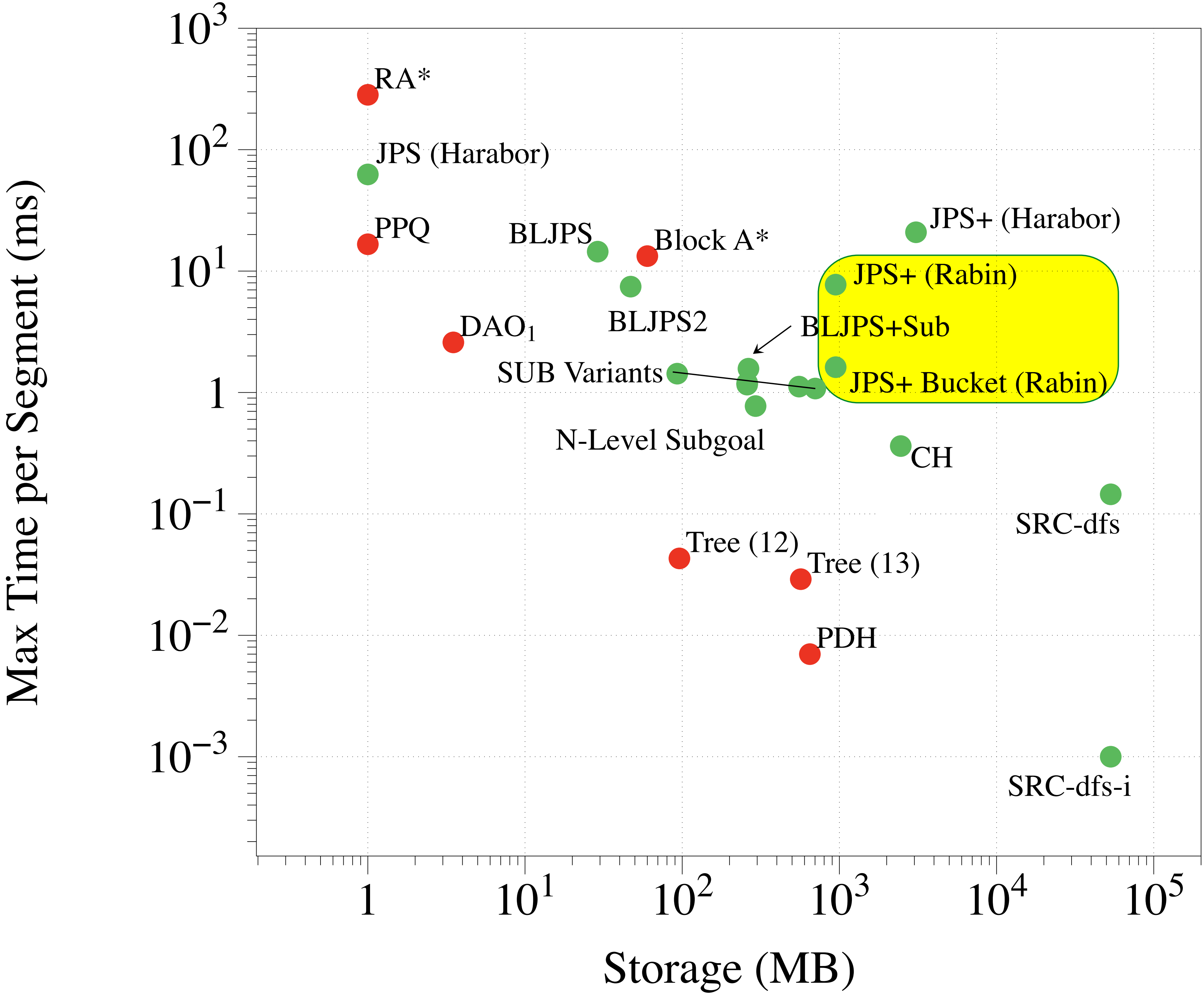
RA*
JPS (Harabor)
PPQ
BLJPS
Block A*
JPS+ (Harabor)
JPS+ (Rabin)
BLJPS2
DAO1
BLJPS+Sub
SUB Variants
JPS+ Bucket (Rabin)
N-Level Subgoal
CH
SRC-dfs
Tree (12)
Tree (13)
PDH
SRC-dfs-i
Max Time per Segment (ms)
Storage (MB)



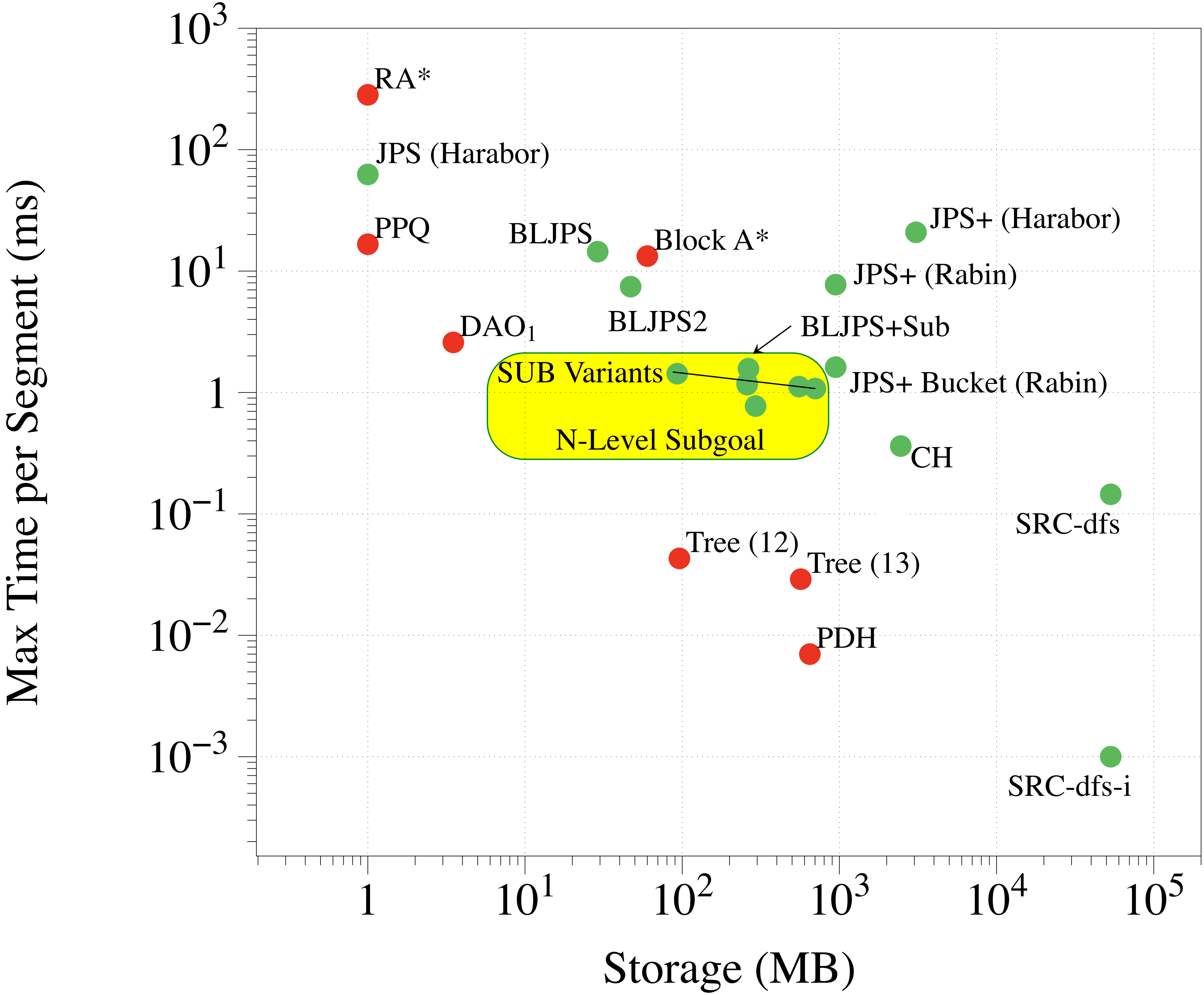
Max Time per Segment (ms)
Storage (MB)
10^3
10^2
10^1
1
10^-1
10^-2
10^-3
10^1
10^2
10^3
10^4
10^5
RA*
JPS (Harabor)
PPQ
BLJPS
Block A*
BLJPS2
DAO1
SUB Variants
BLJPS+Sub
JPS+ (Harabor)
JPS+ (Rabin)
JPS+ Bucket (Rabin)
N-Level Subgoal
CH
SRC-dfs
Tree (12)
Tree (13)
PDH
SRC-dfs-i



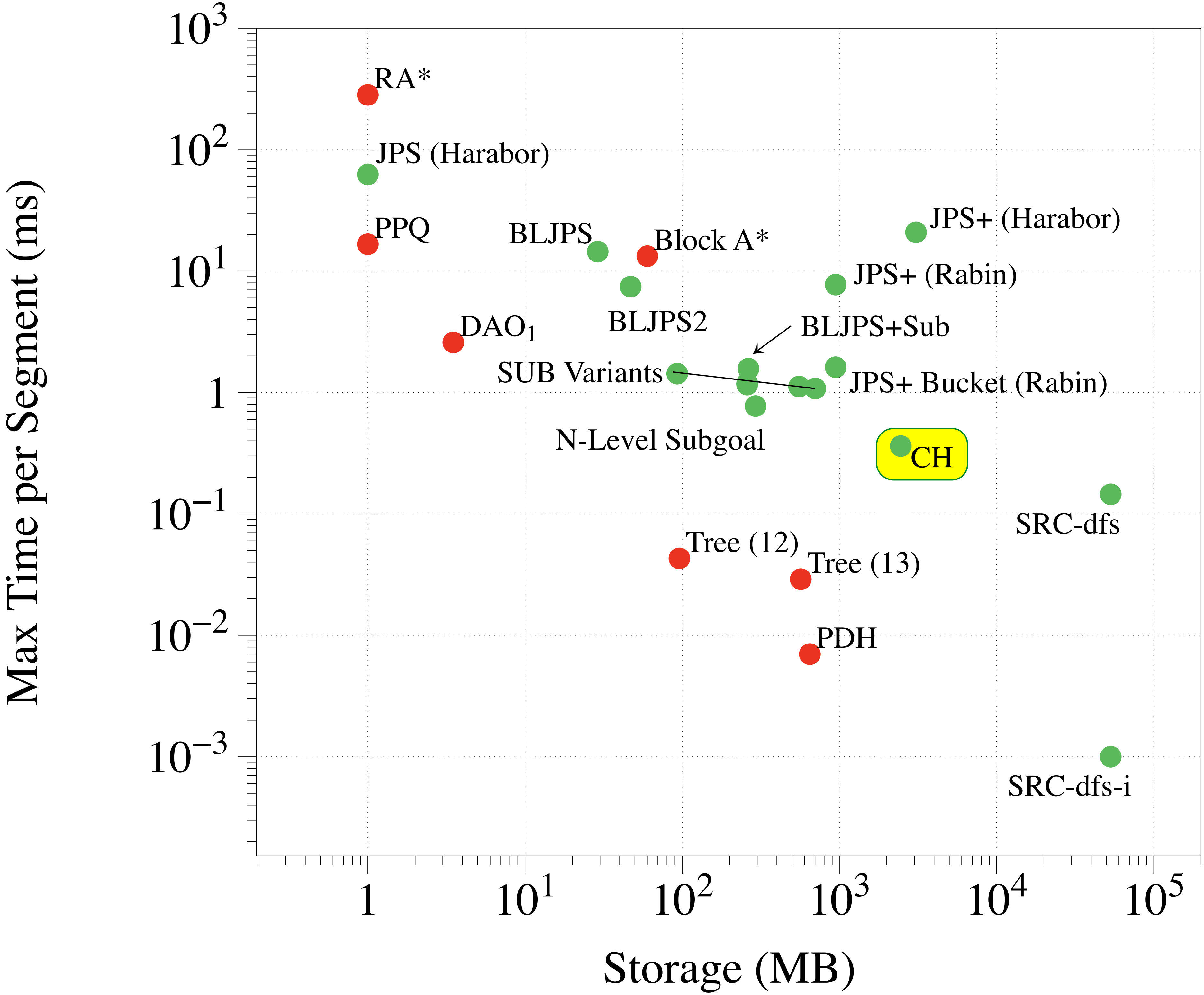
Max Time per Segment (ms)
Storage (MB)
RA*
JPS (Harabor)
PPQ
BLJPS
Block A*
JPS+ (Harabor)
JPS+ (Rabin)
BLJPS2
DAO$_1$
BLJPS+Sub
SUB Variants
JPS+ Bucket (Rabin)
N-Level Subgoal
CH
SRC-dfs
Tree (12)
Tree (13)
PDH
SRC-dfs-i



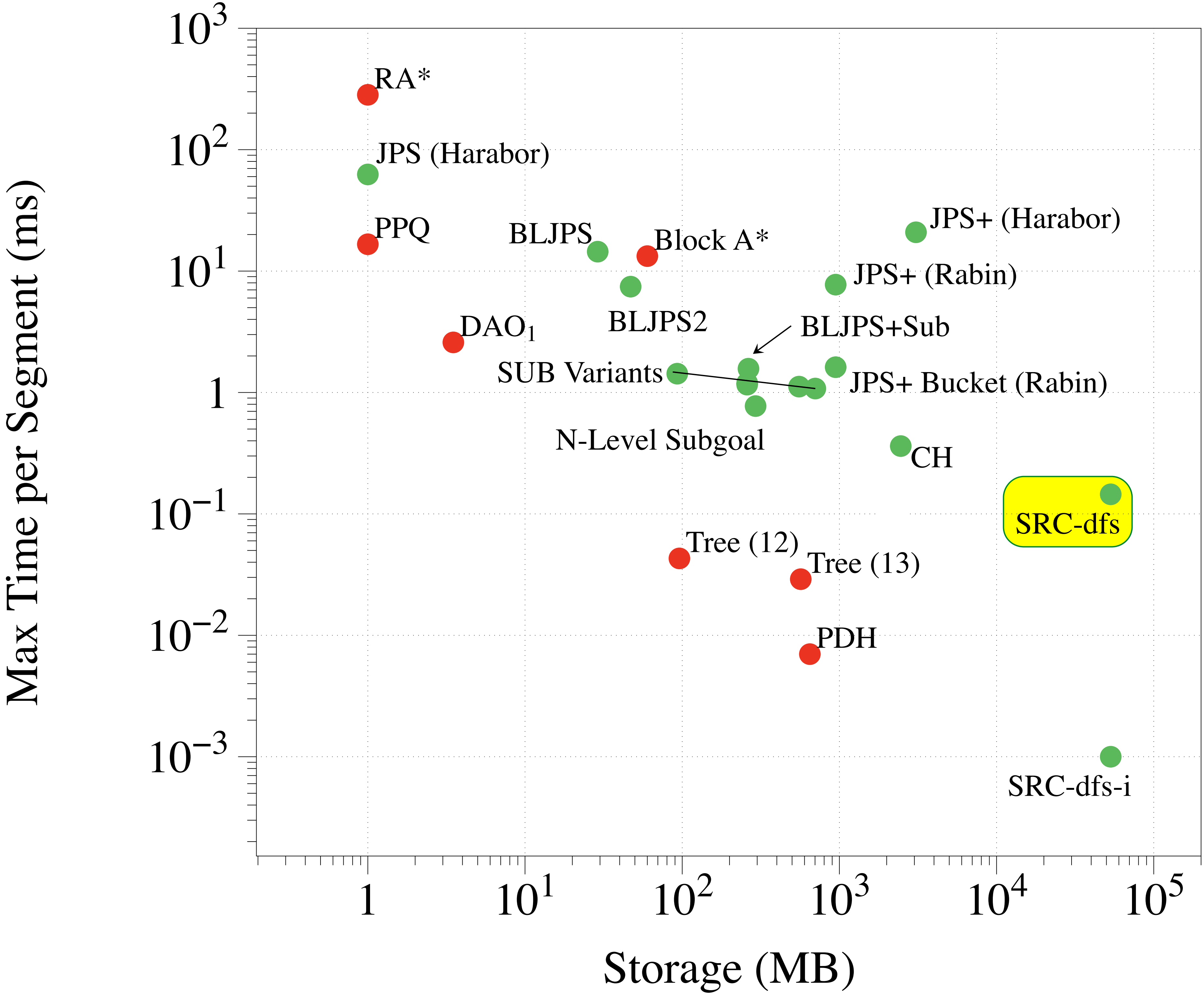
Max Time per Segment (ms)
Storage (MB)
RA*
JPS (Harabor)
PPQ
BLJPS
Block A*
JPS+ (Harabor)
JPS+ (Rabin)
BLJPS2
DAO1
BLJPS+Sub
SUB Variants
JPS+ Bucket (Rabin)
N-Level Subgoal
CH
SRC-dfs
Tree (12)
Tree (13)
PDH
SRC-dfs-i



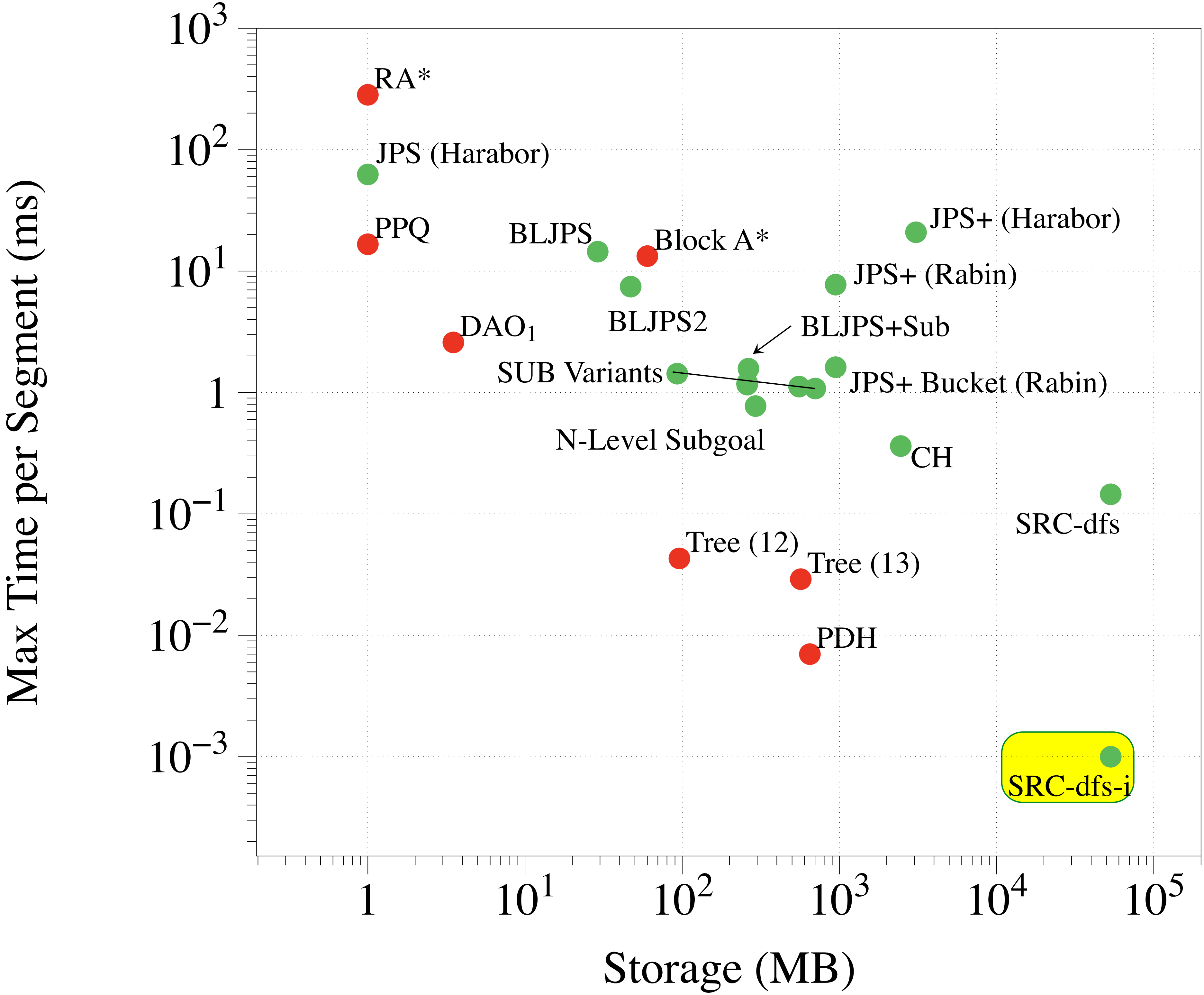
RA*
JPS (Harabor)
PPQ
BLJPS
Block A*
JPS+ (Harabor)
JPS+ (Rabin)
BLJPS2
DAO$_1$
BLJPS+Sub
SUB Variants
JPS+ Bucket (Rabin)
N-Level Subgoal
CH
SRC-dfs
Tree (12)
Tree (13)
PDH
SRC-dfs-i
Max Time per Segment (ms)
Storage (MB)
10^3
10^2
10^1
1
10^{-1}
10^{-2}
10^{-3}
1
10^1
10^2
10^3
10^4
10^5



RA*
JPS (Harabor)
PPQ
BLJPS
Block A*
JPS+ (Harabor)
JPS+ (Rabin)
BLJPS2
DAO$_1$
BLJPS+Sub
SUB Variants
JPS+ Bucket (Rabin)
N-Level Subgoal
CH
SRC-dfs
Tree (12)
Tree (13)
PDH
SRC-dfs-i
Max Time per Segment (ms)
Storage (MB)
10^3
10^2
10^1
1
10^{-1}
10^{-2}
10^{-3}
1
10^1
10^2
10^3
10^4
10^5



Max Time per Segment (ms)
Storage (MB)
RA*
JPS (Harabor)
PPQ
BLJPS
Block A*
JPS+ (Harabor)
JPS+ (Rabin)
BLJPS2
DAO1
BLJPS+Sub
SUB Variants
JPS+ Bucket (Rabin)
N-Level Subgoal
CH
SRC-dfs
Tree (12)
Tree (13)
PDH
SRC-dfs-i



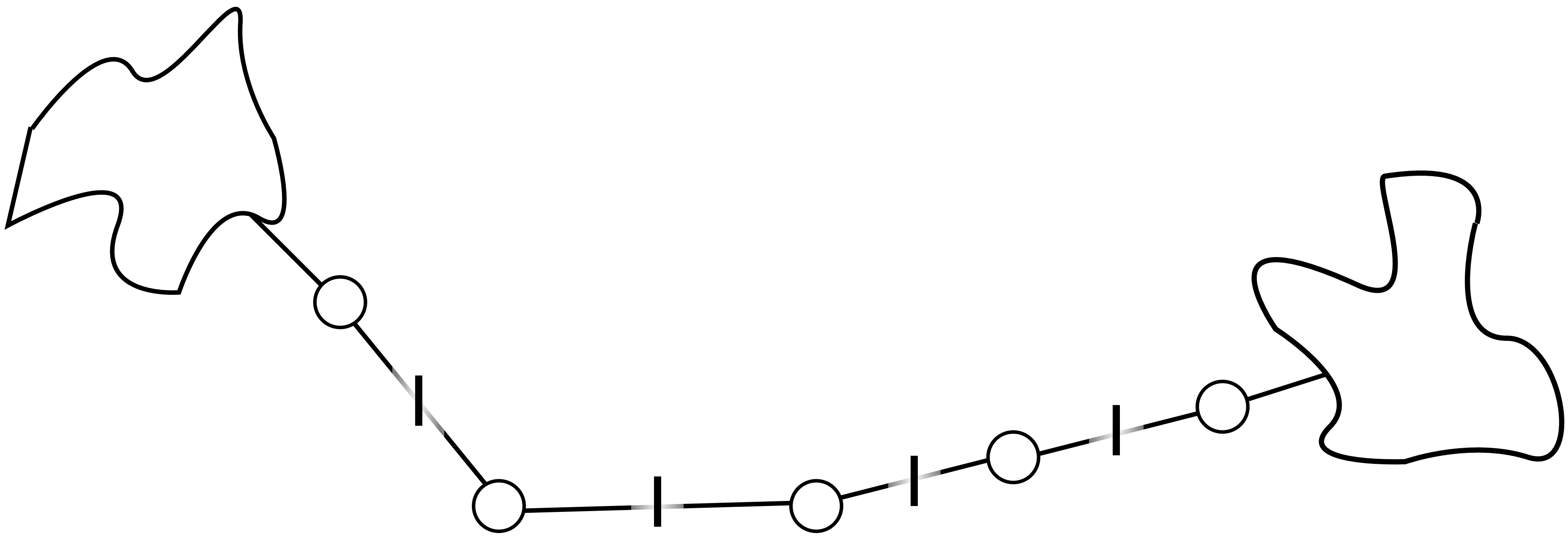
RA*
JPS (Harabor)
PPQ
BLJPS
Block A*
JPS+ (Harabor)
JPS+ (Rabin)
BLJPS2
DAO_1
BLJPS+Sub
SUB Variants
JPS+ Bucket (Rabin)
N-Level Subgoal
CH
SRC-dfs
Tree (12)
Tree (13)
PDH
SRC-dfs-i
Max Time per Segment (ms)
Storage (MB)
10^3
10^2
10^1
1
10^{-1}
10^{-2}
10^{-3}
1
10^1
10^2
10^3
10^4
10^5

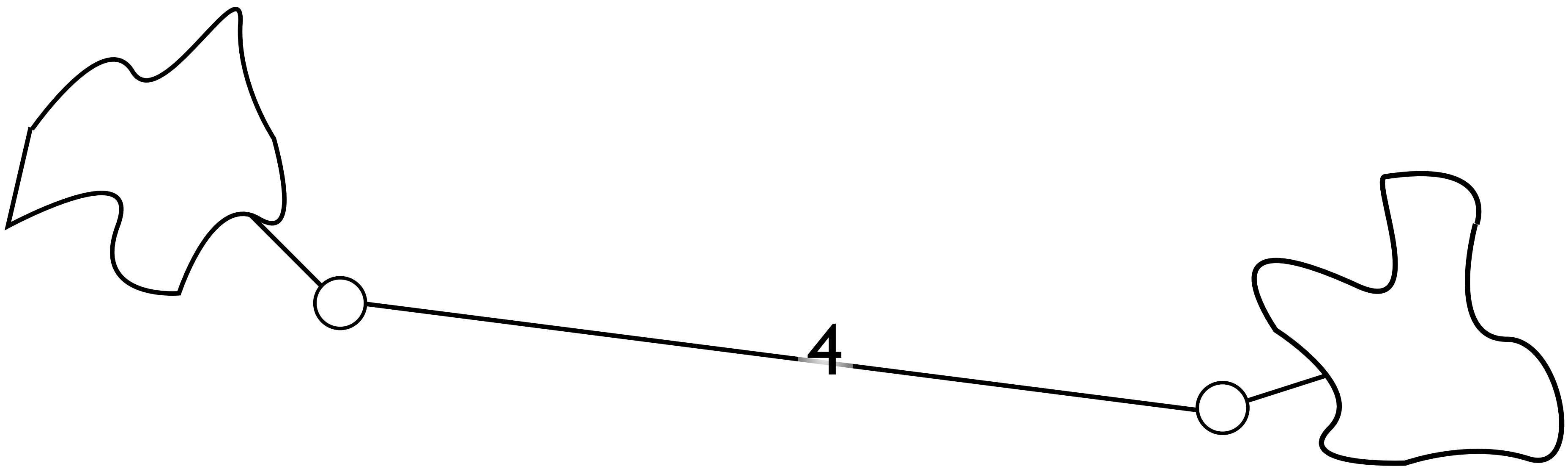
Contractions everywhere!

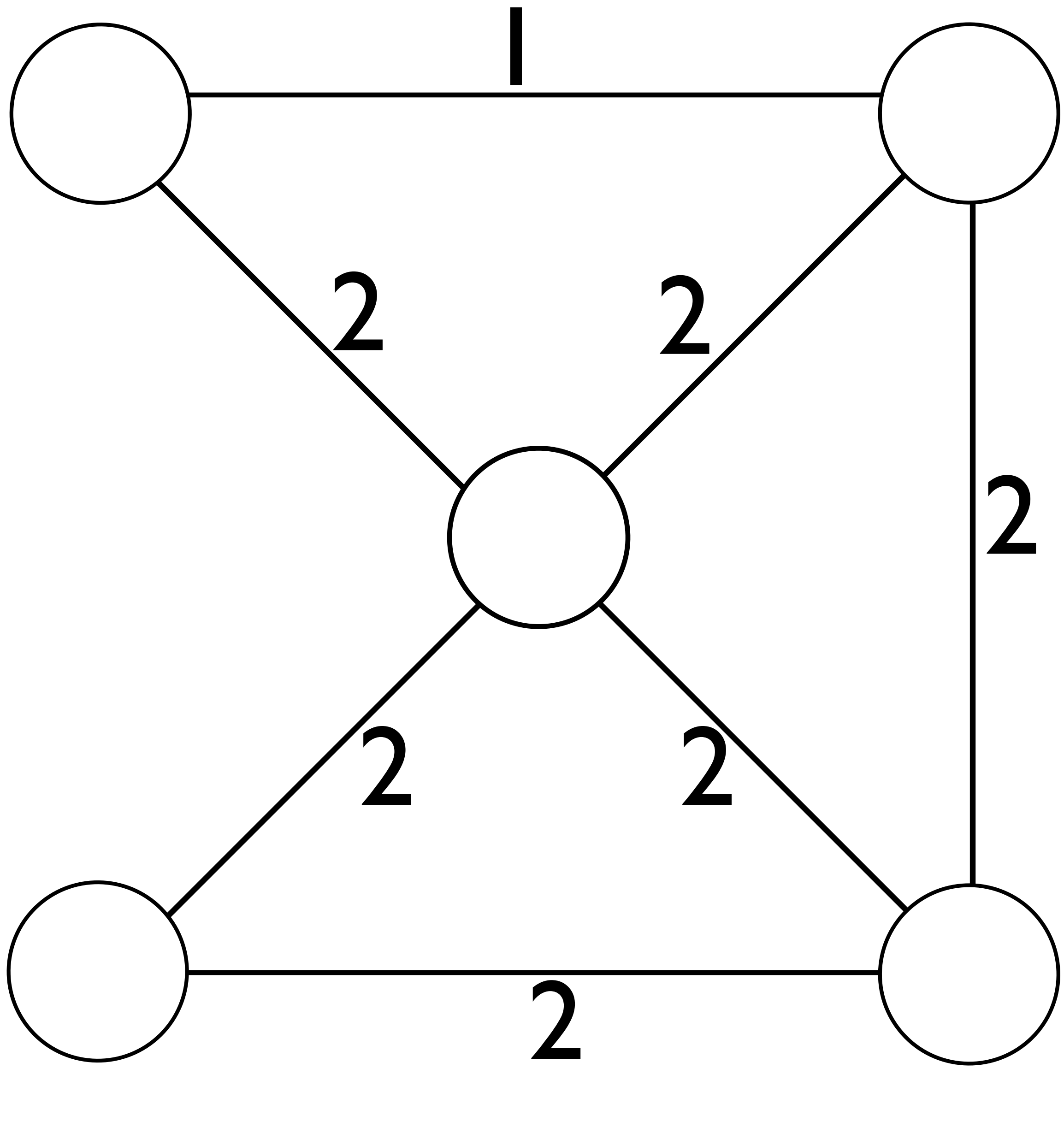
- Describe majority of approaches as contraction:
  - Contraction Hierarchies
  - DAO abstraction
  - Subgoal Graphs
  - Jump Point Search

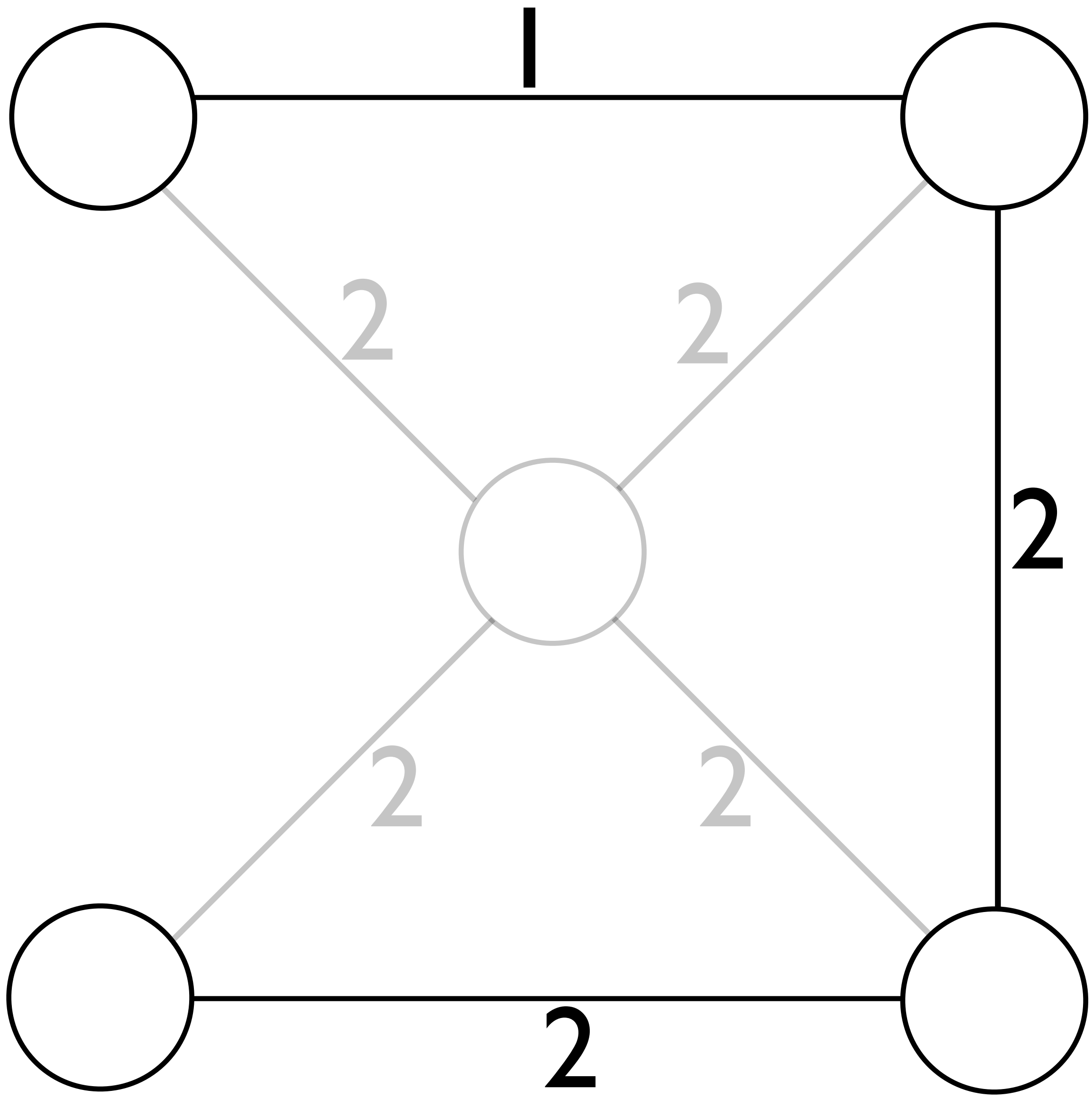
Graph Contraction

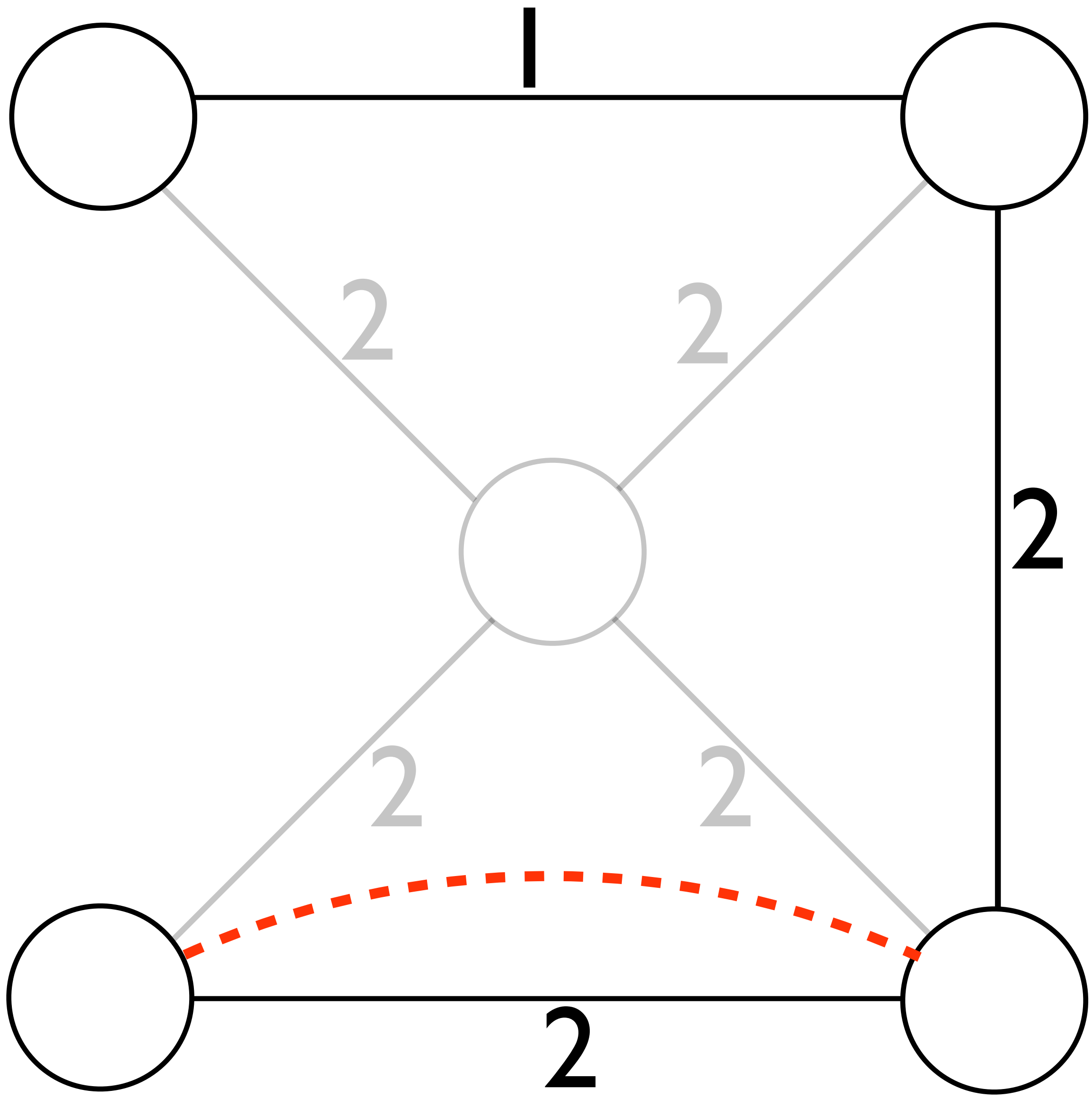
- Remove a set S of states from a graph G
- Optimal:
  - Ensure that the shortest path between all states G \ S are not changed
  - Add edges (shortcuts) to preserve shortest paths
- Suboptimal:
  - Ensure that completeness is not lost

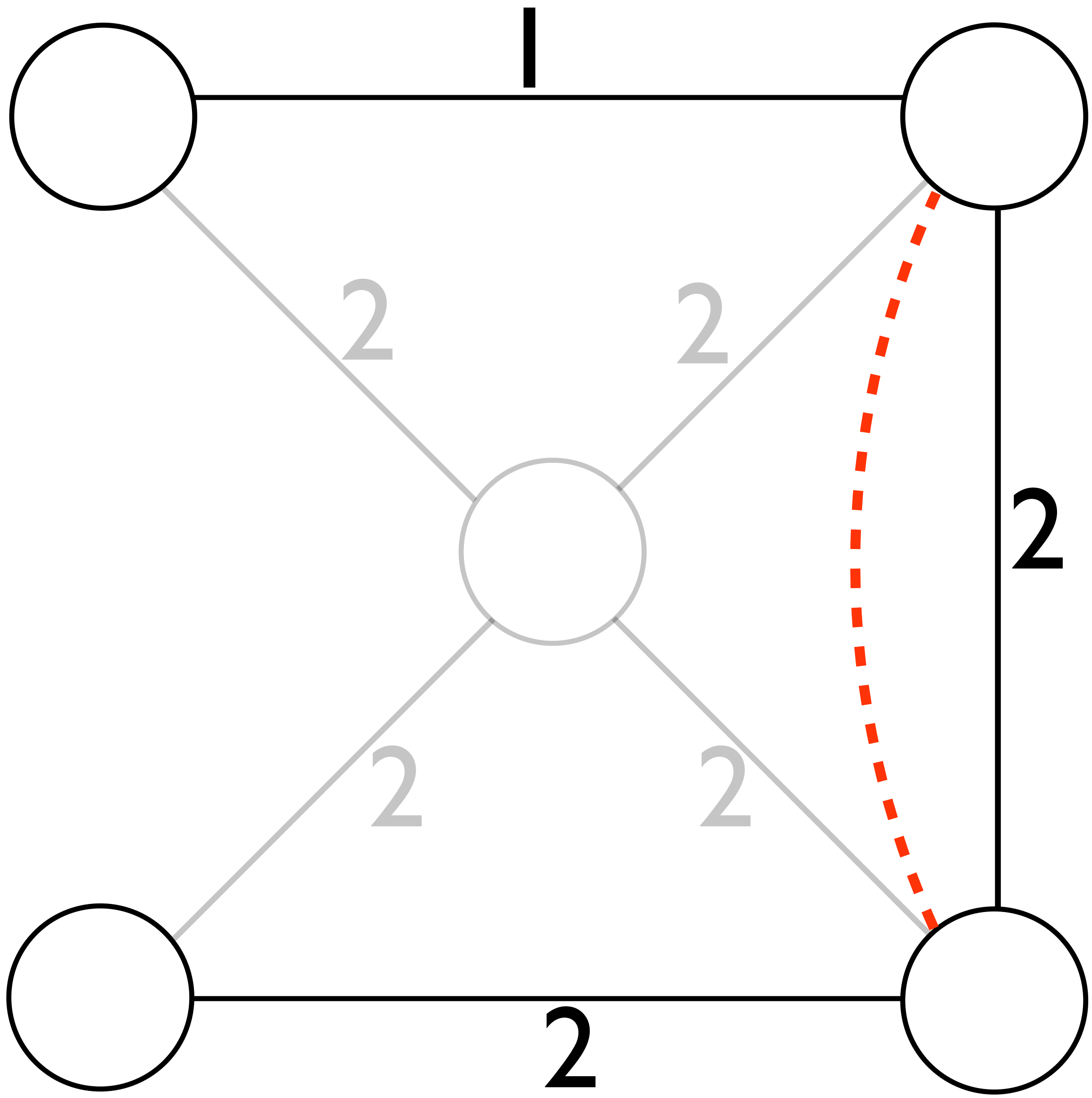



4

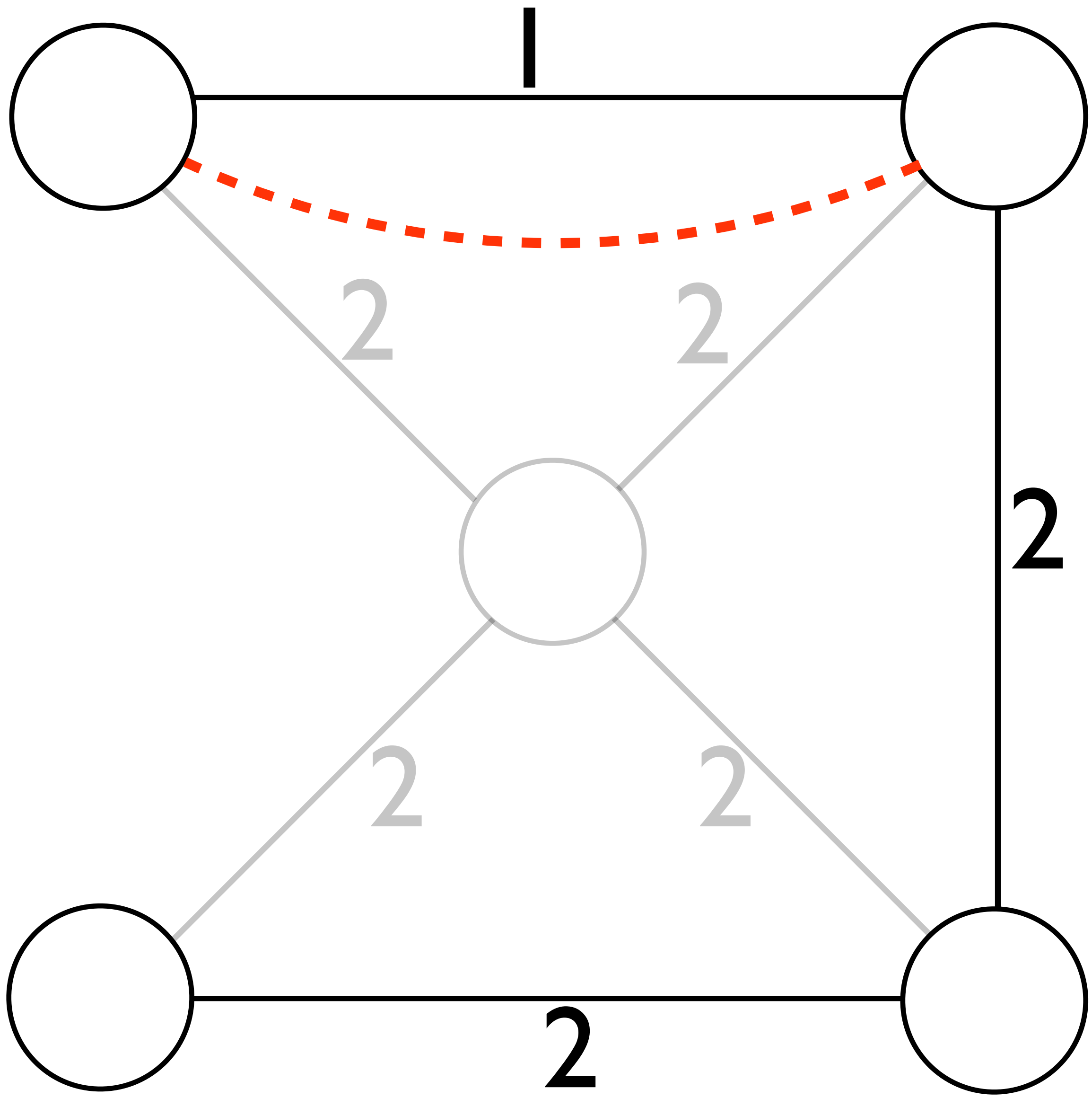

1
2
2
2
2
2
2

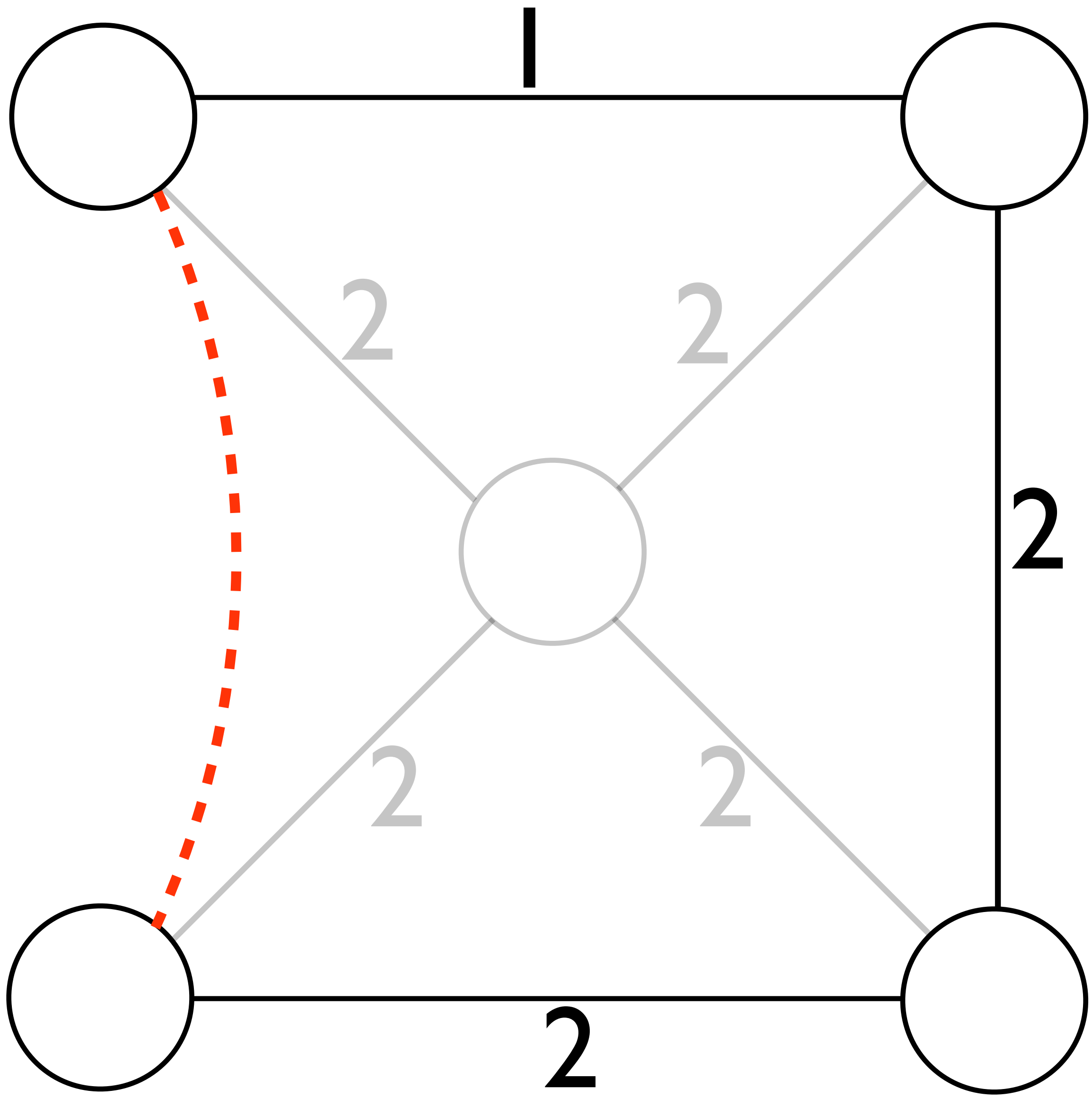

1
2
2
2
2
2
2
2


1
2
2
2
2
2
2

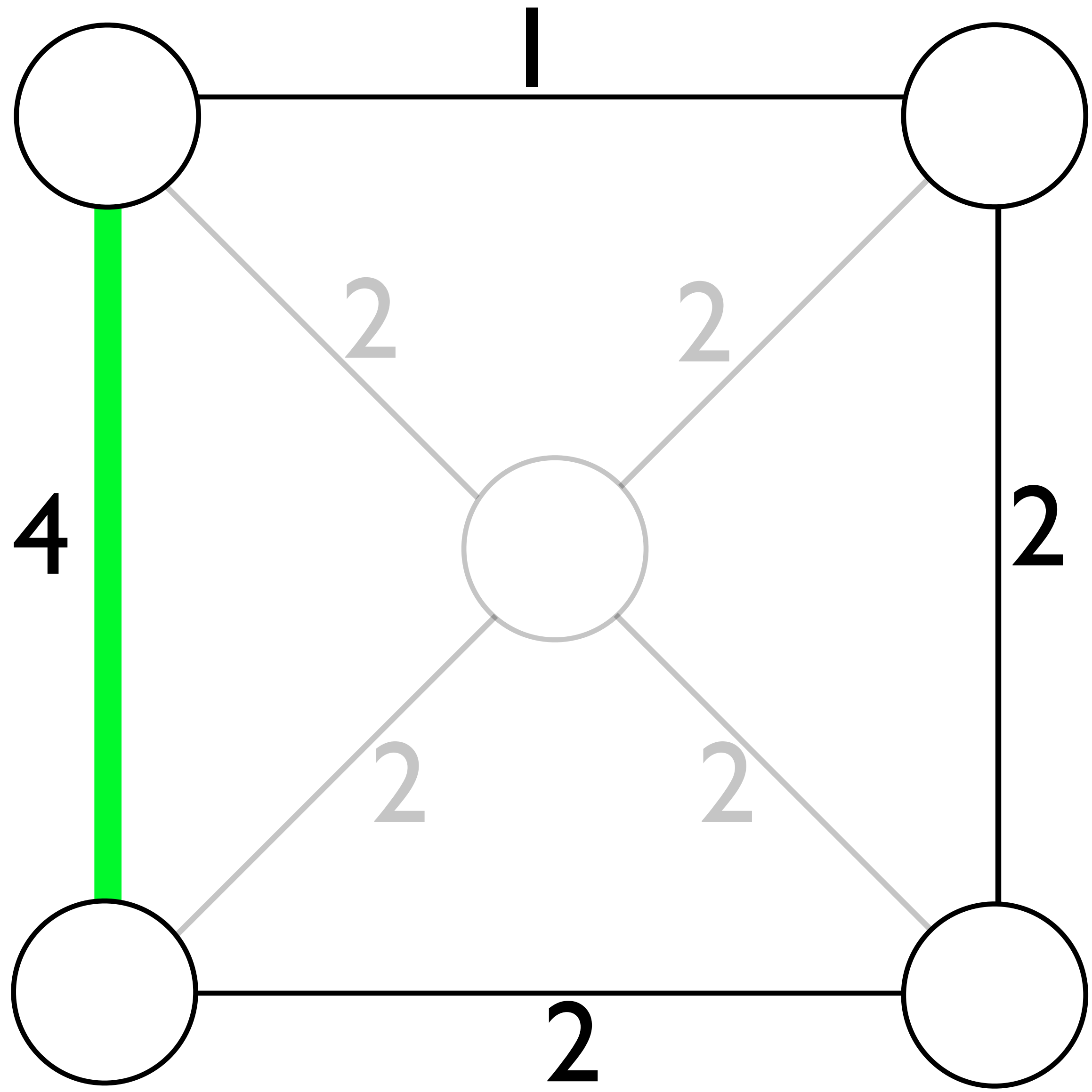


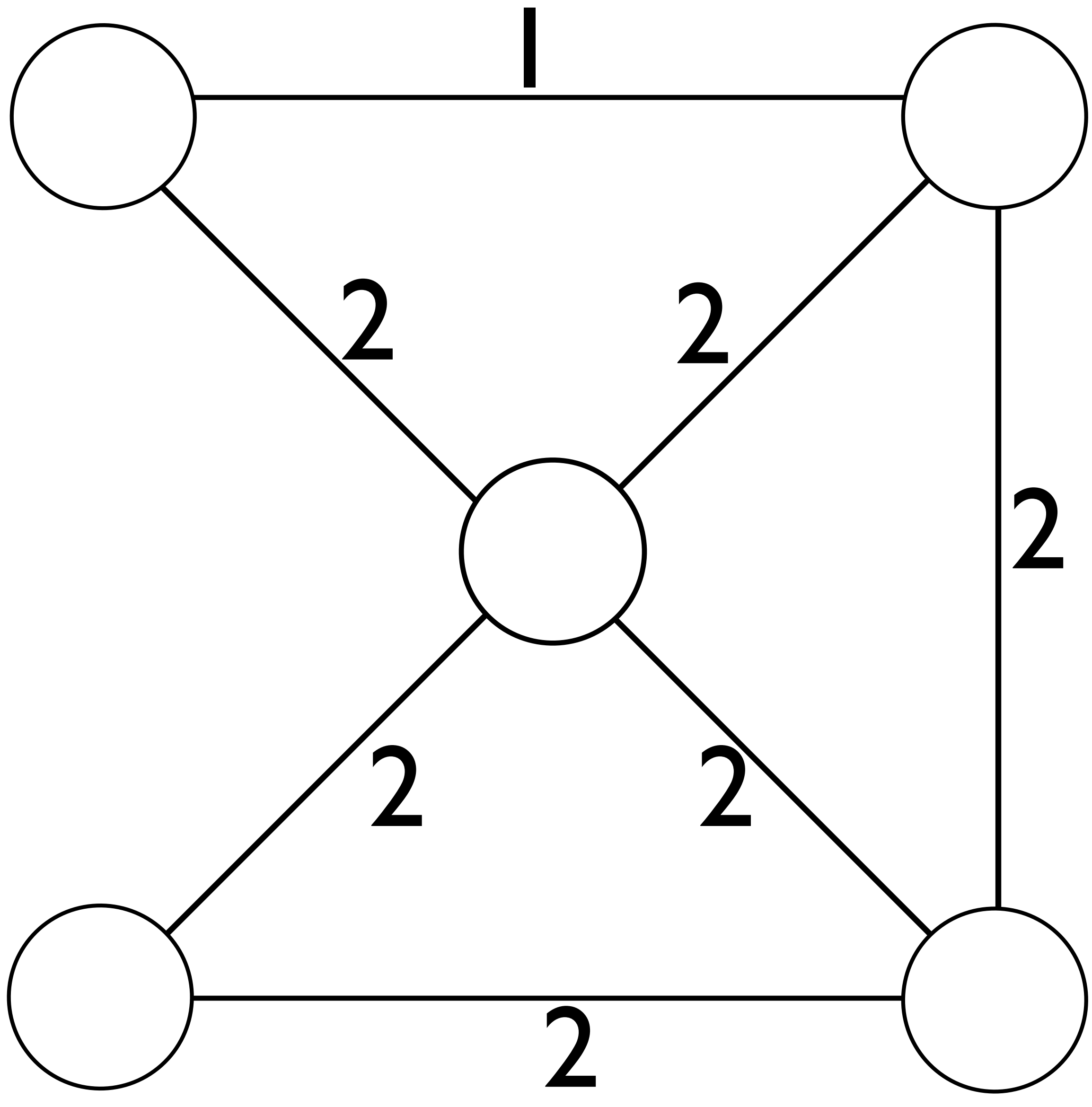
1
2
2
2
2
2
2

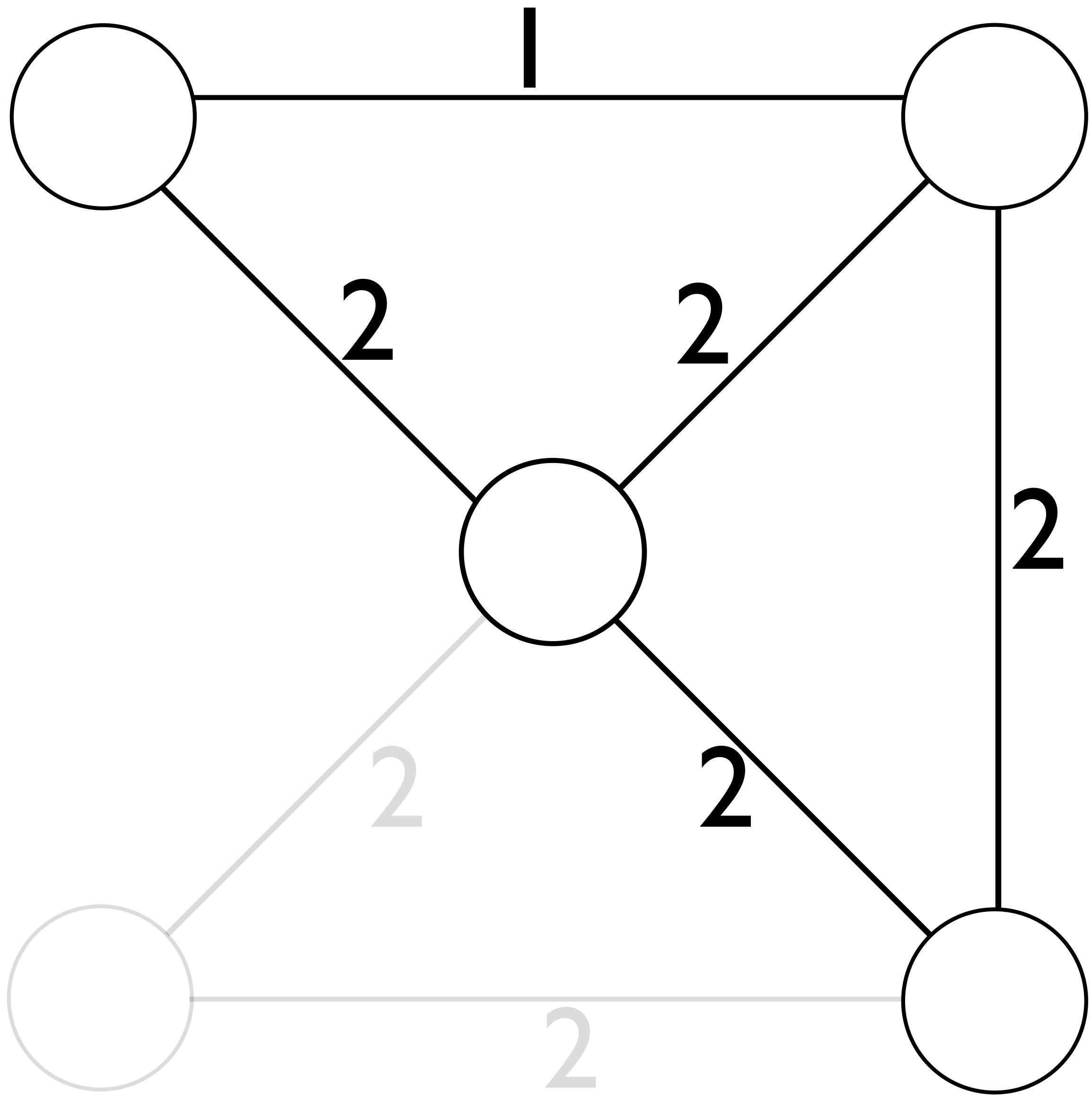

1
2
2
2
2
2
2



1
2
2
2
2
2
2


1
2
2
4
2
2
2
2
2


1
2
2
2
2
2
2



1
2
2
2
2
2
2

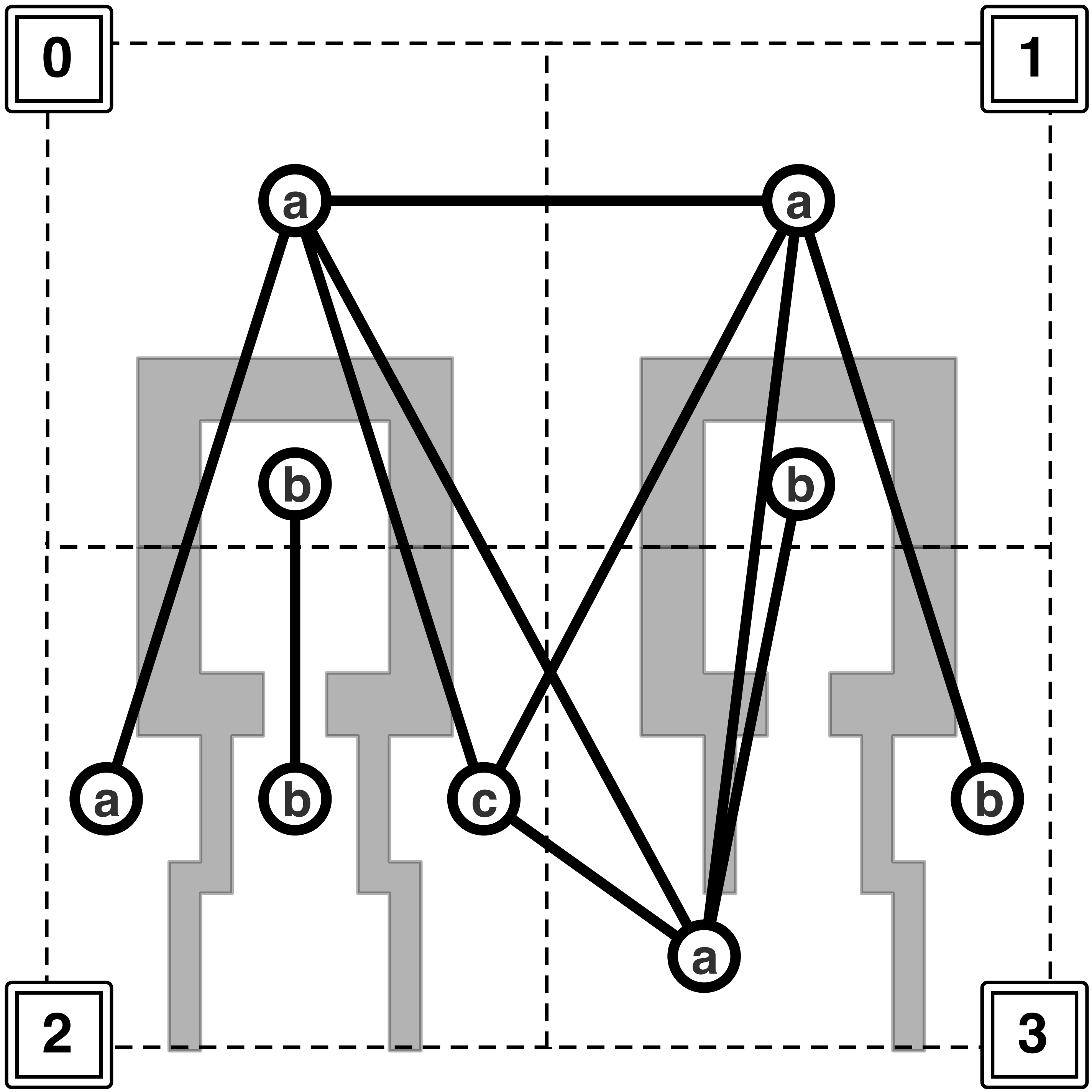
Contraction Hierarchies

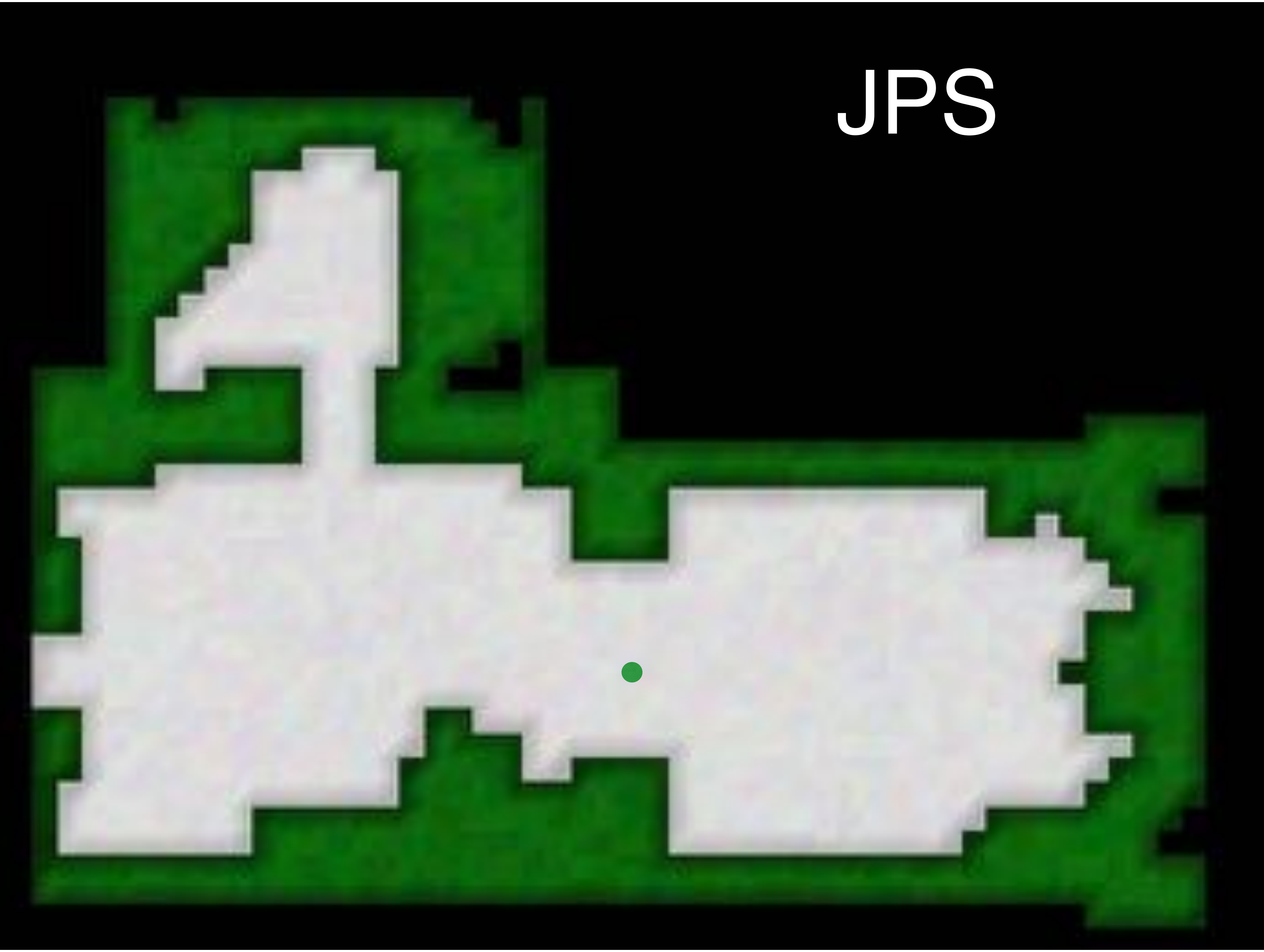
- Optimal approach
- Key idea is finding good orderings for contracting nodes from the graph

- Contract one node at a time
  - Add shortcut edges

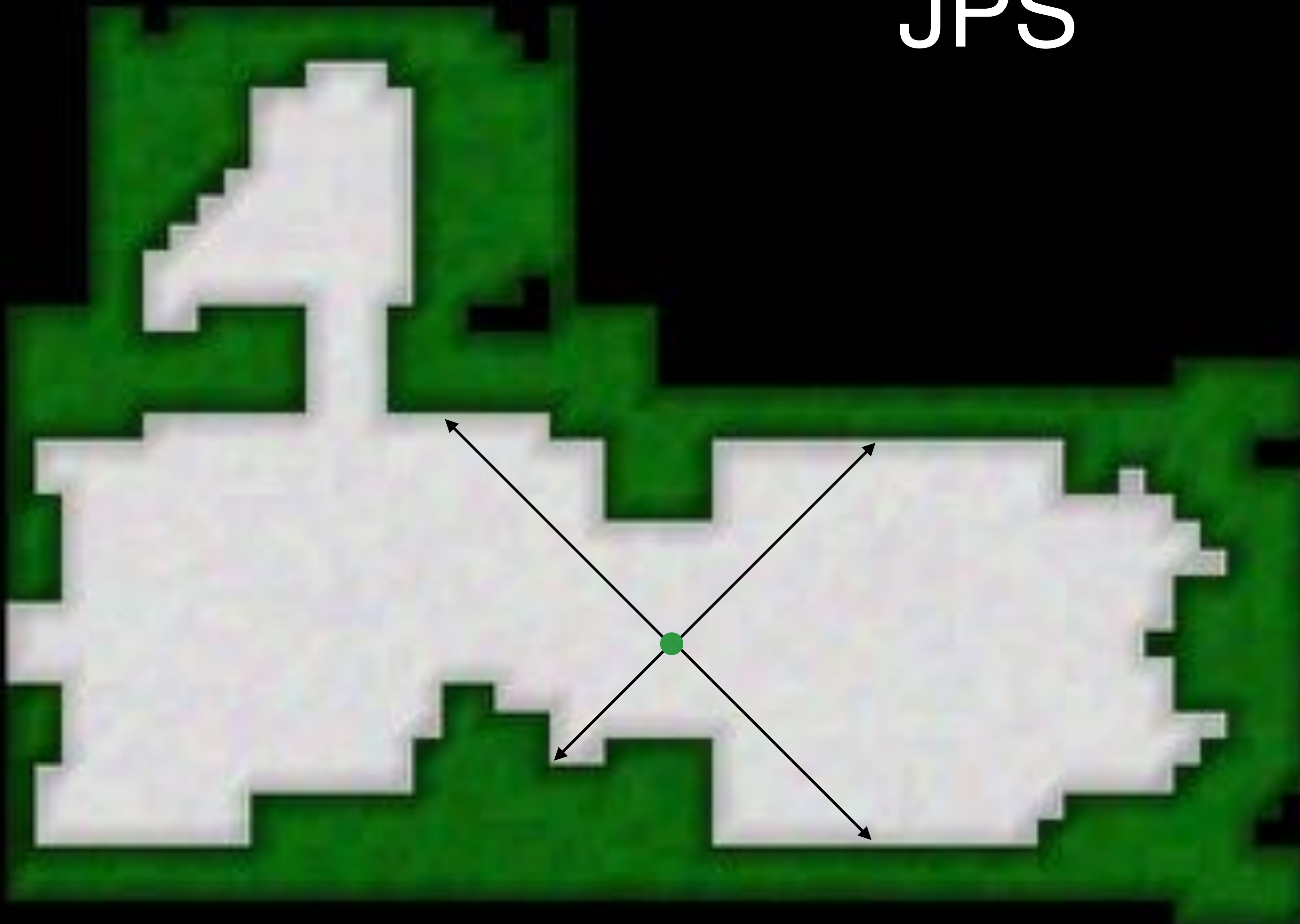
Traditional Abstraction/ Refinement

- Find strongly connected components
  - Abstract them together into a single abstract state
  - (Holte et al, 1996), (Fernández, & González, 2002), (Botea et. at., 2004), (Sturtevant & Buro, 2005), (Sturtevant 2007)
  - *Downward refinement property*
- Approach discarded in road networks after optimal approaches discovered

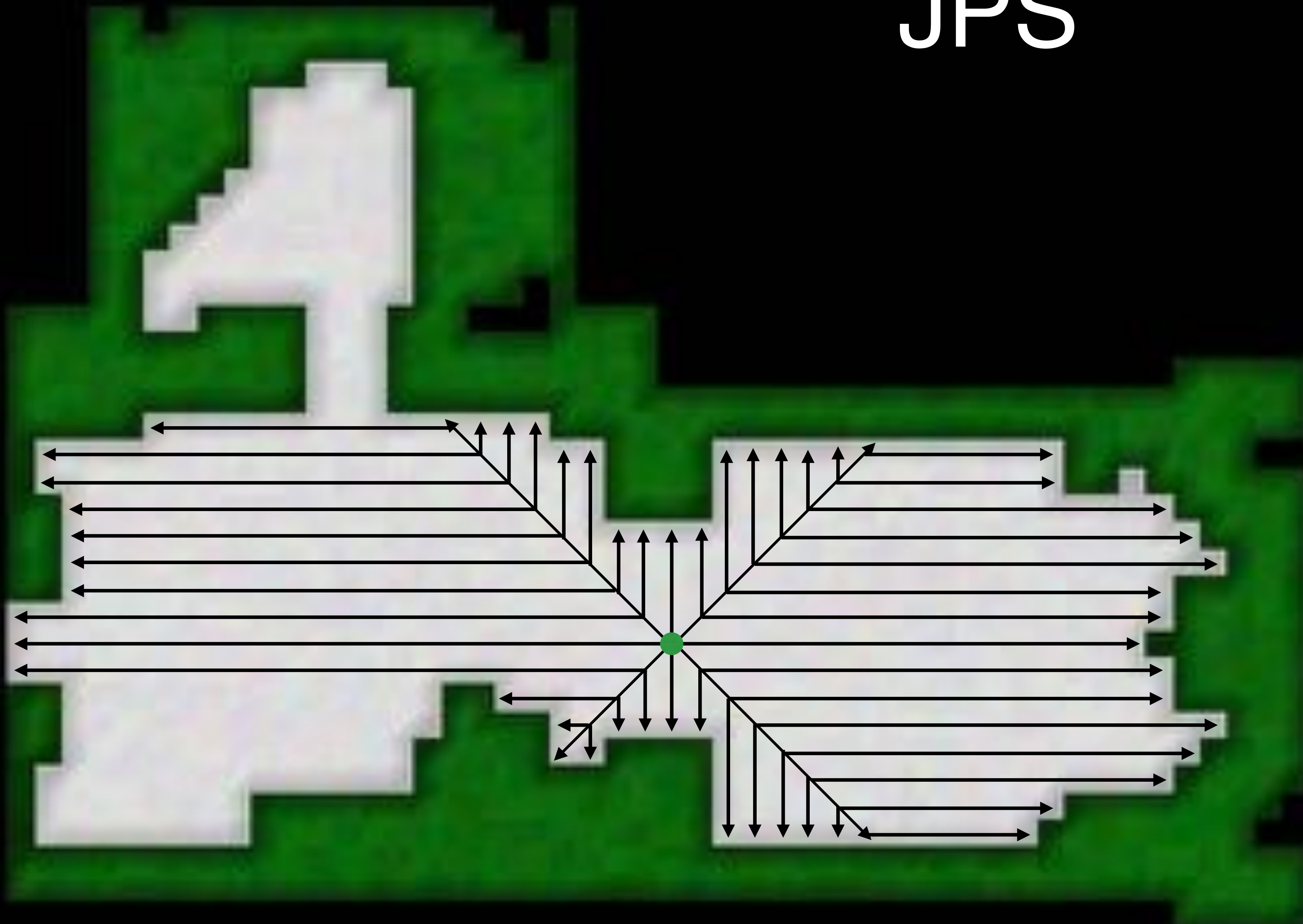

0
1
2
3
a
a
b
b
a
b
c
b
a

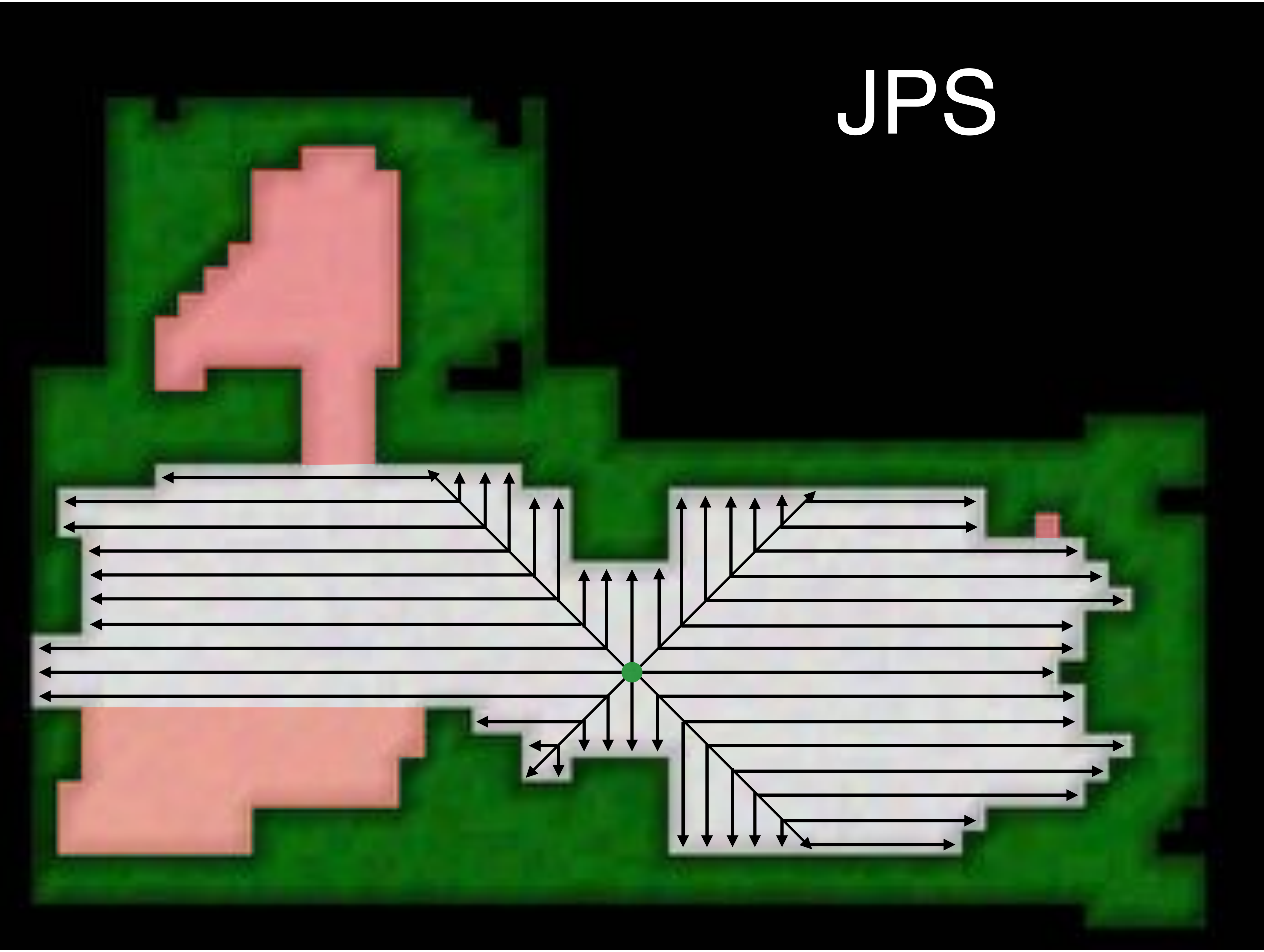

JPS

JPS

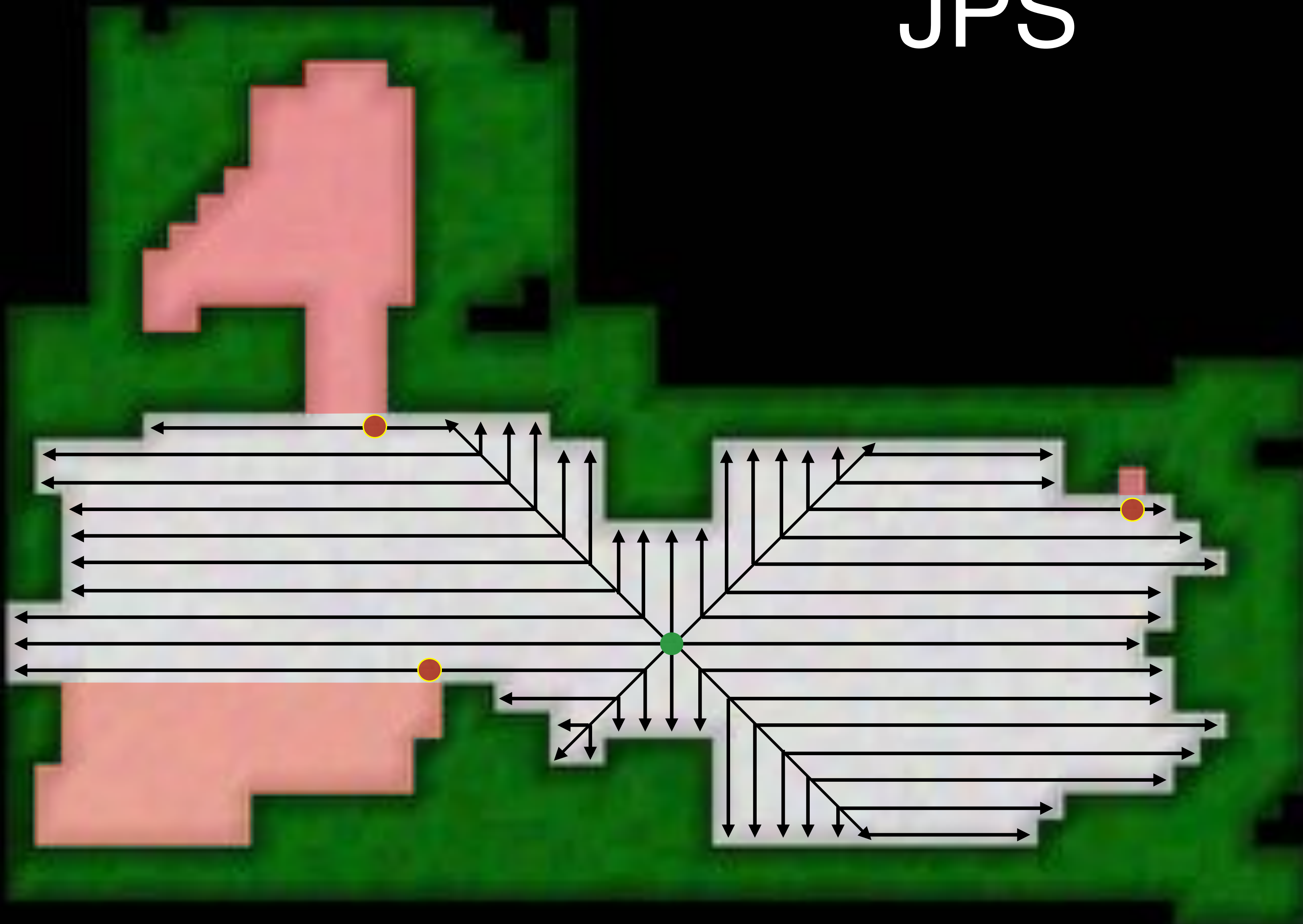


JPS

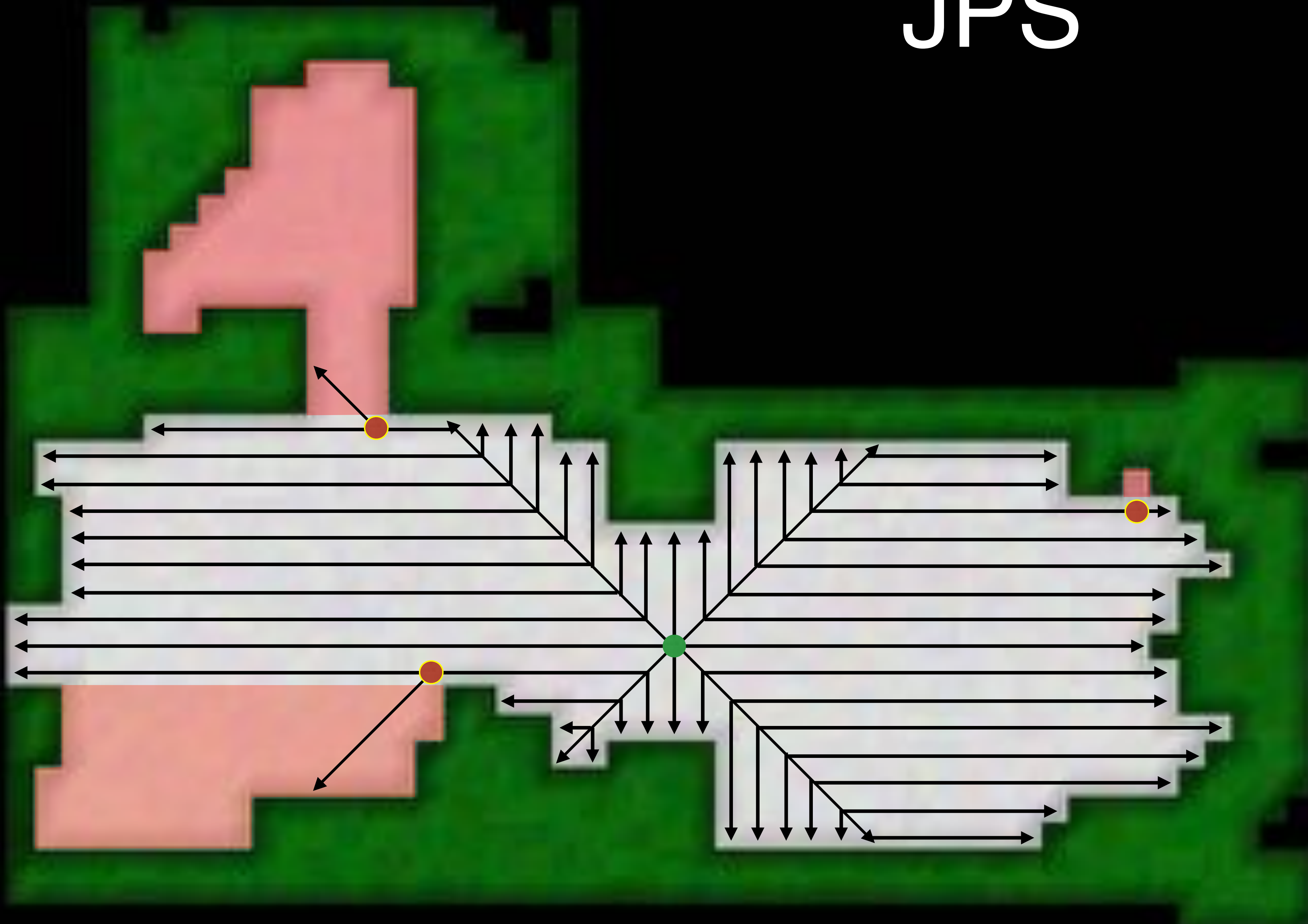




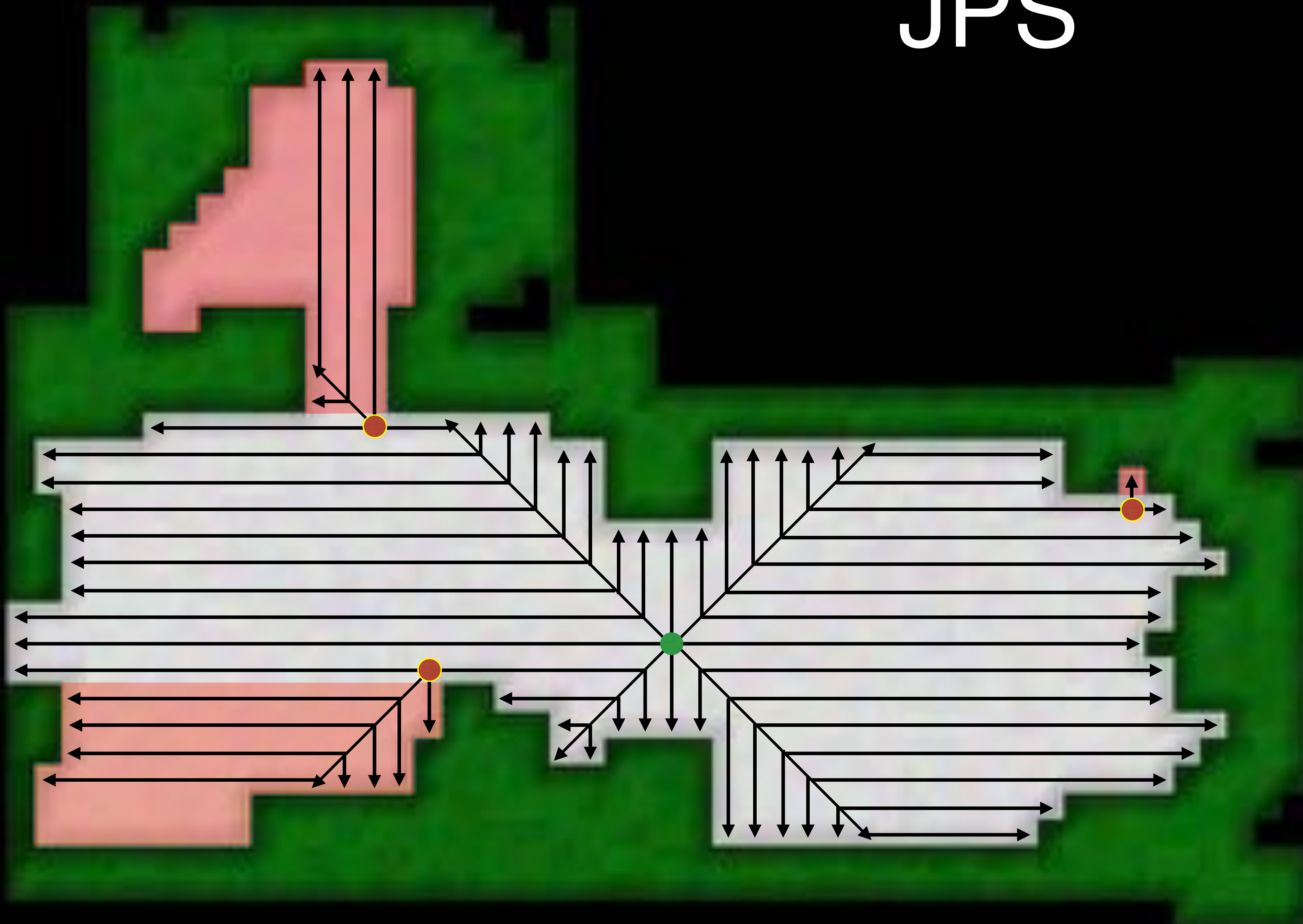
JPS



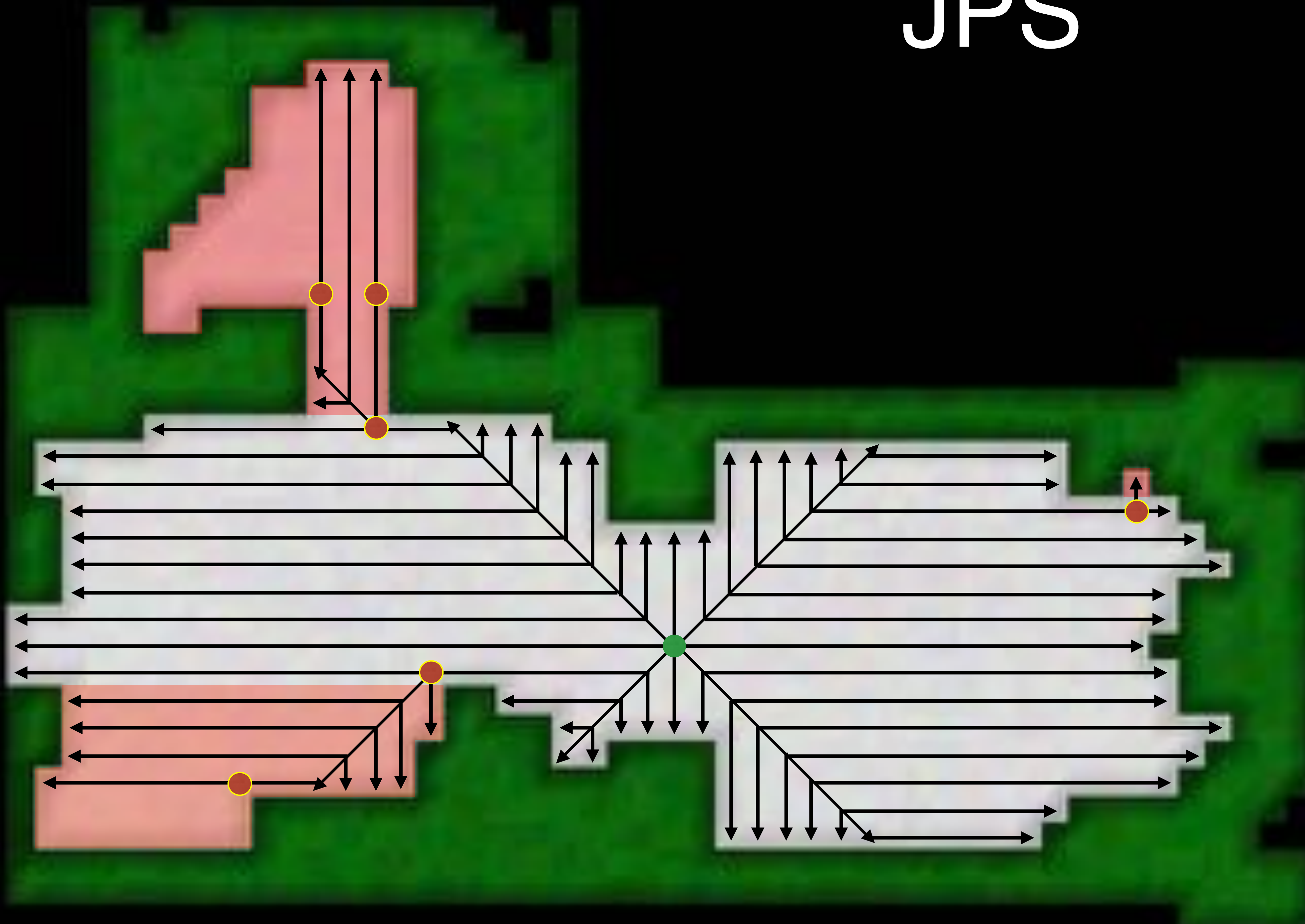
JPS



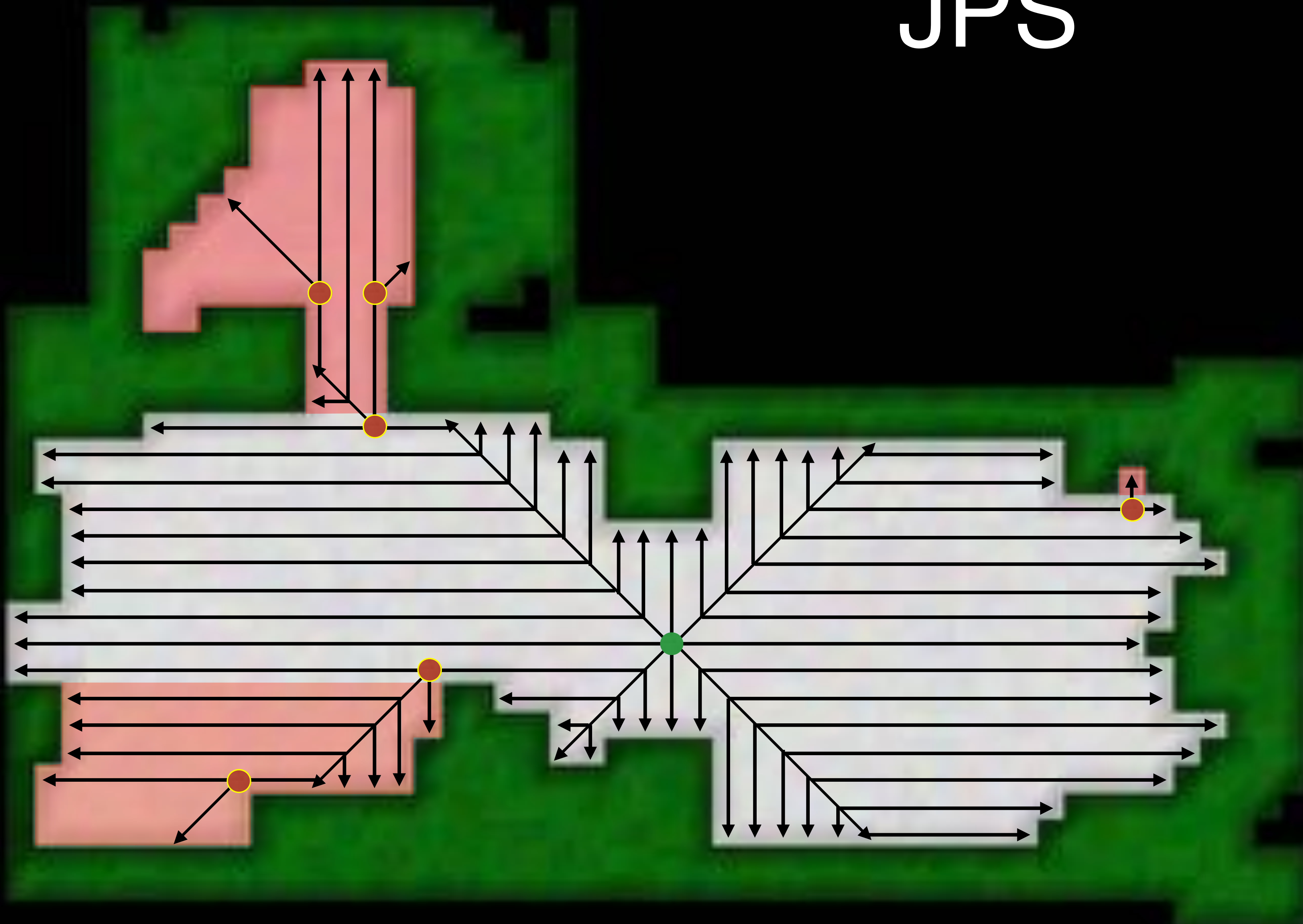
JPS



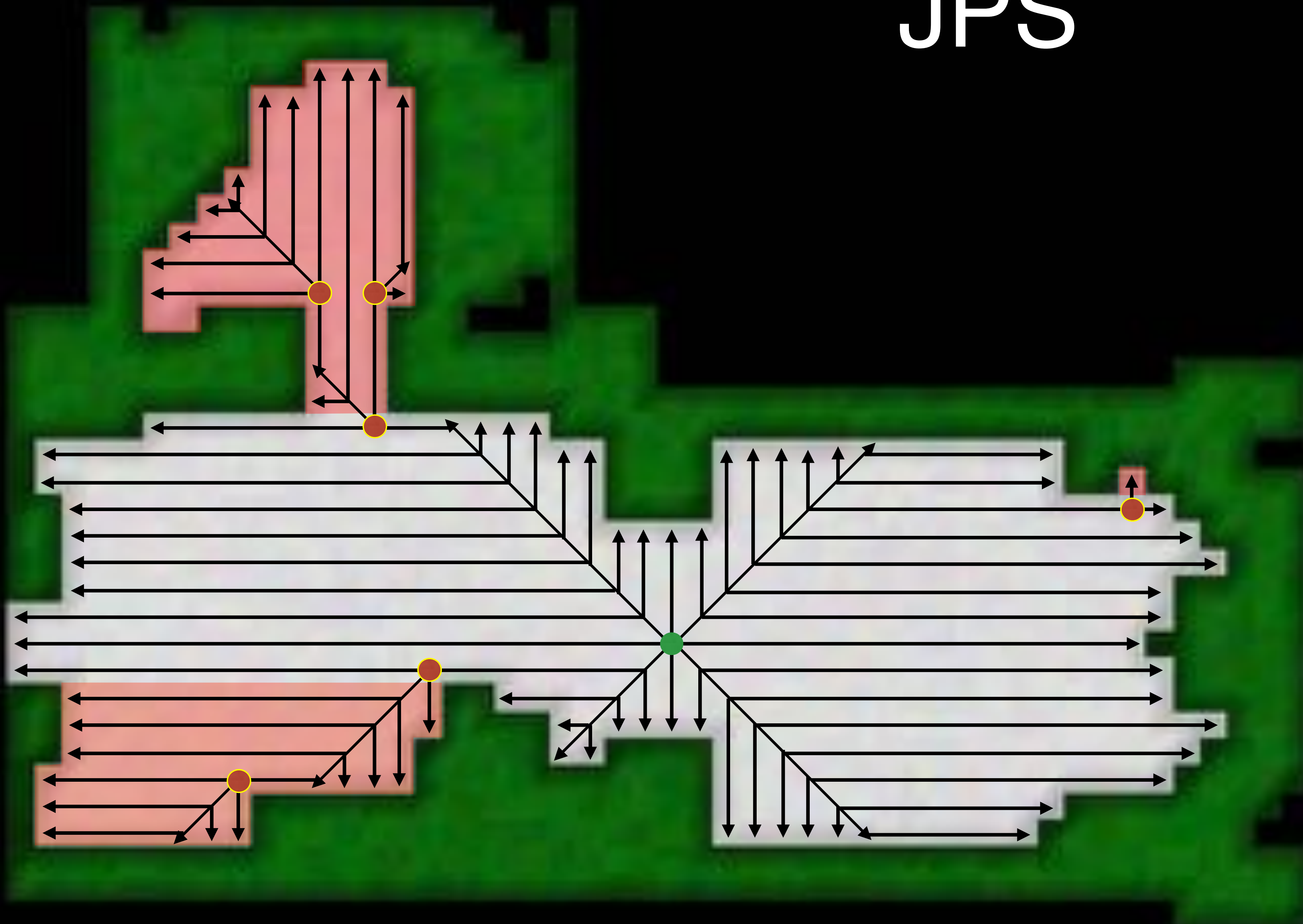
JPS



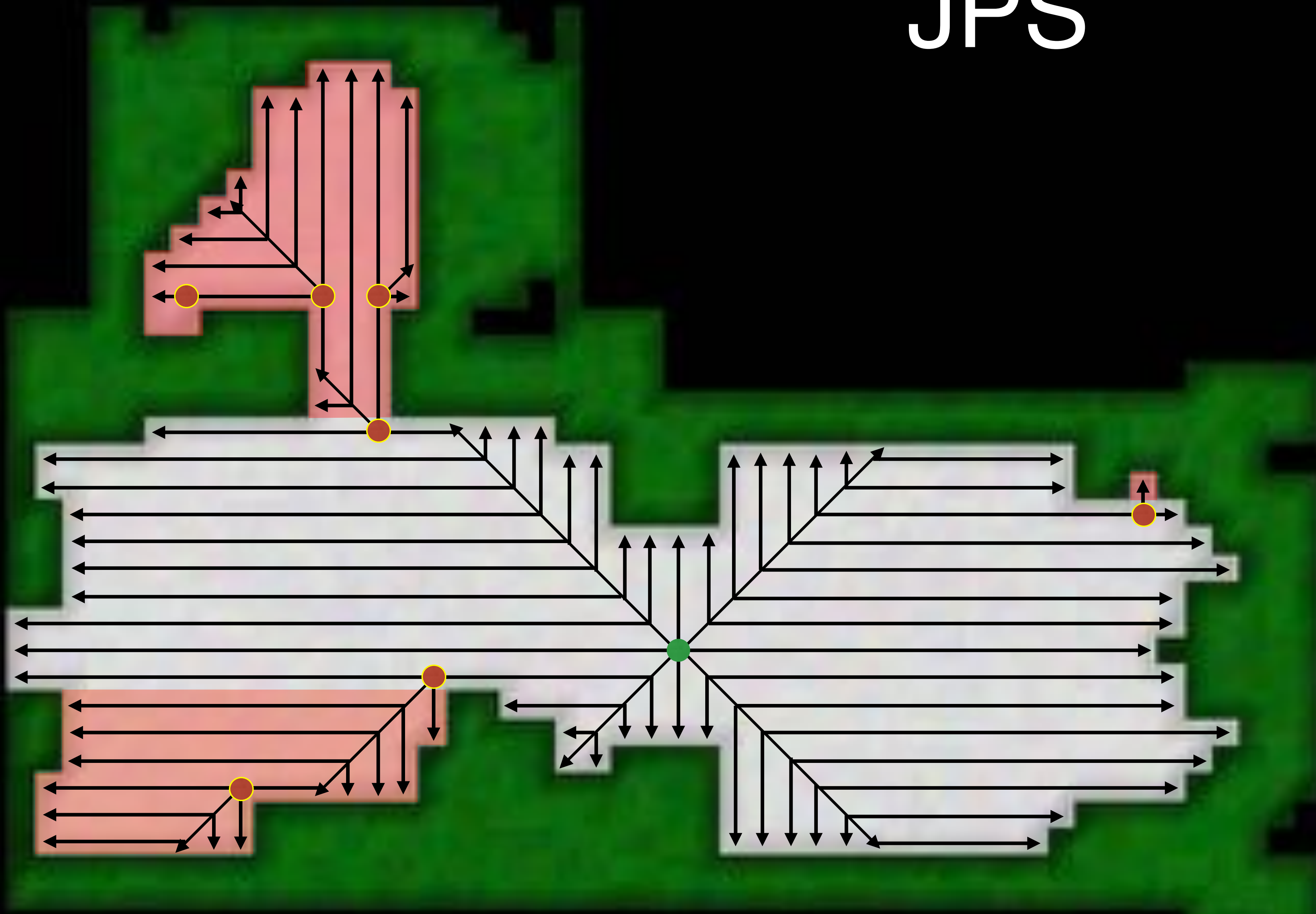
JPS



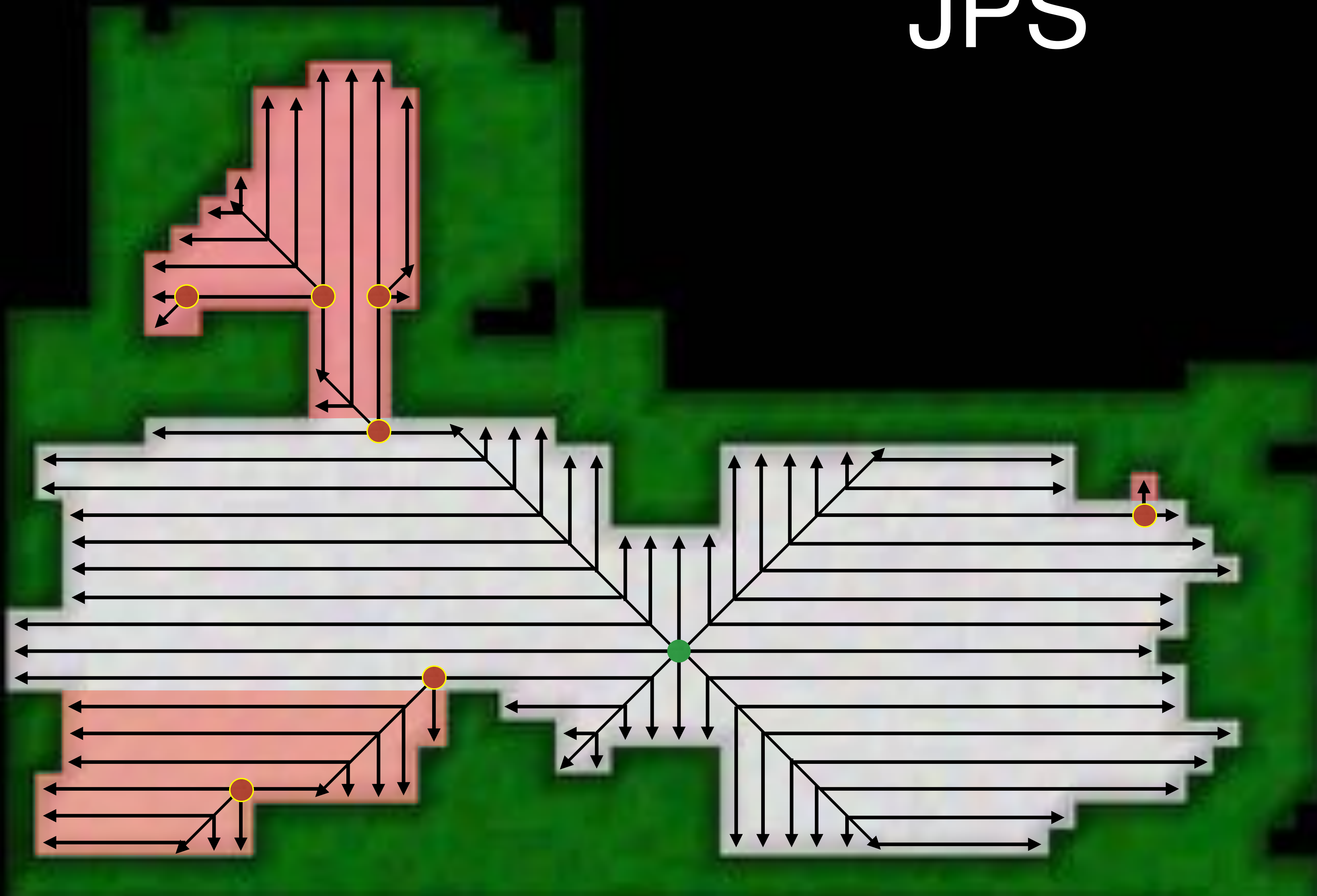
JPS

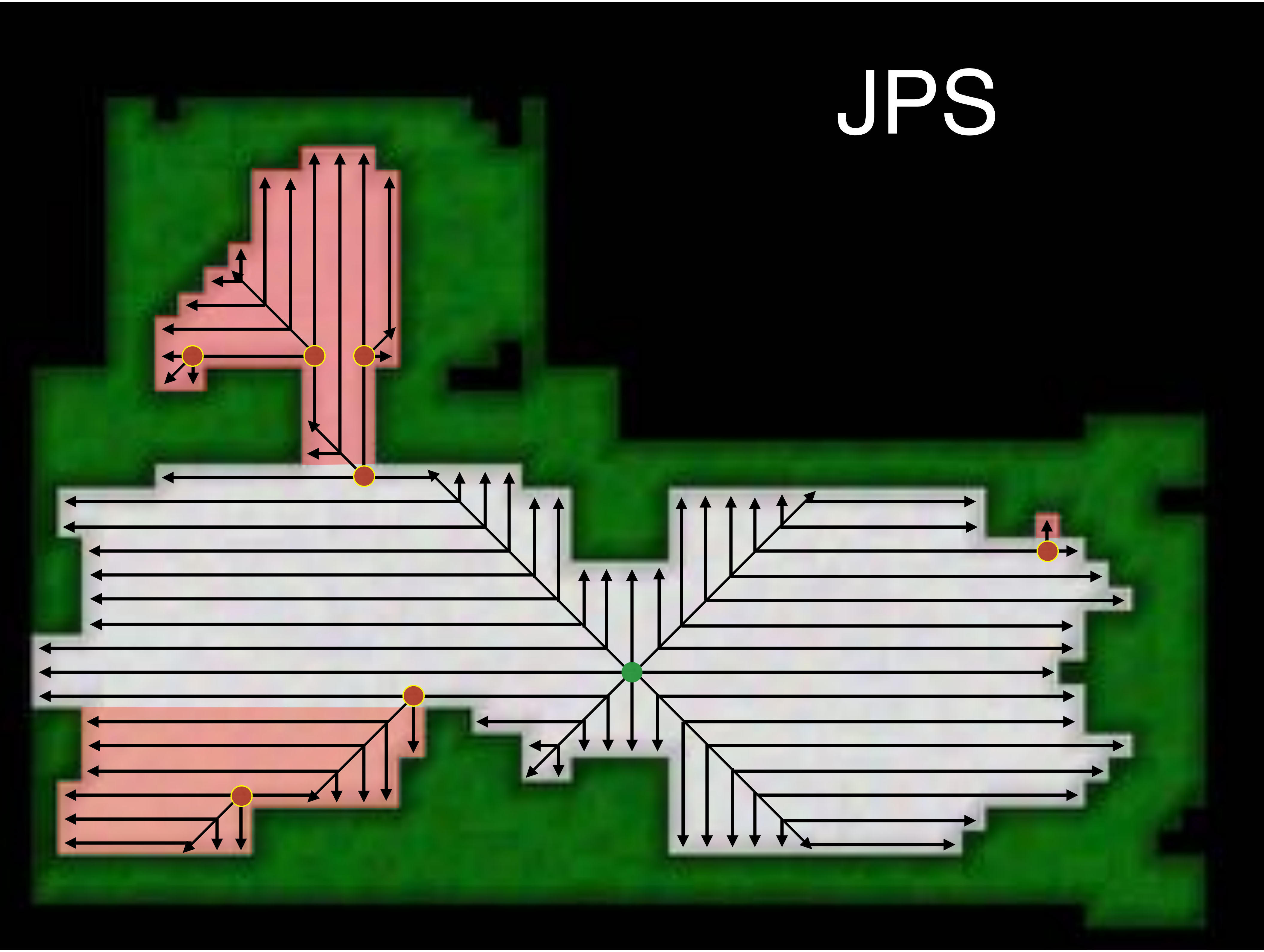


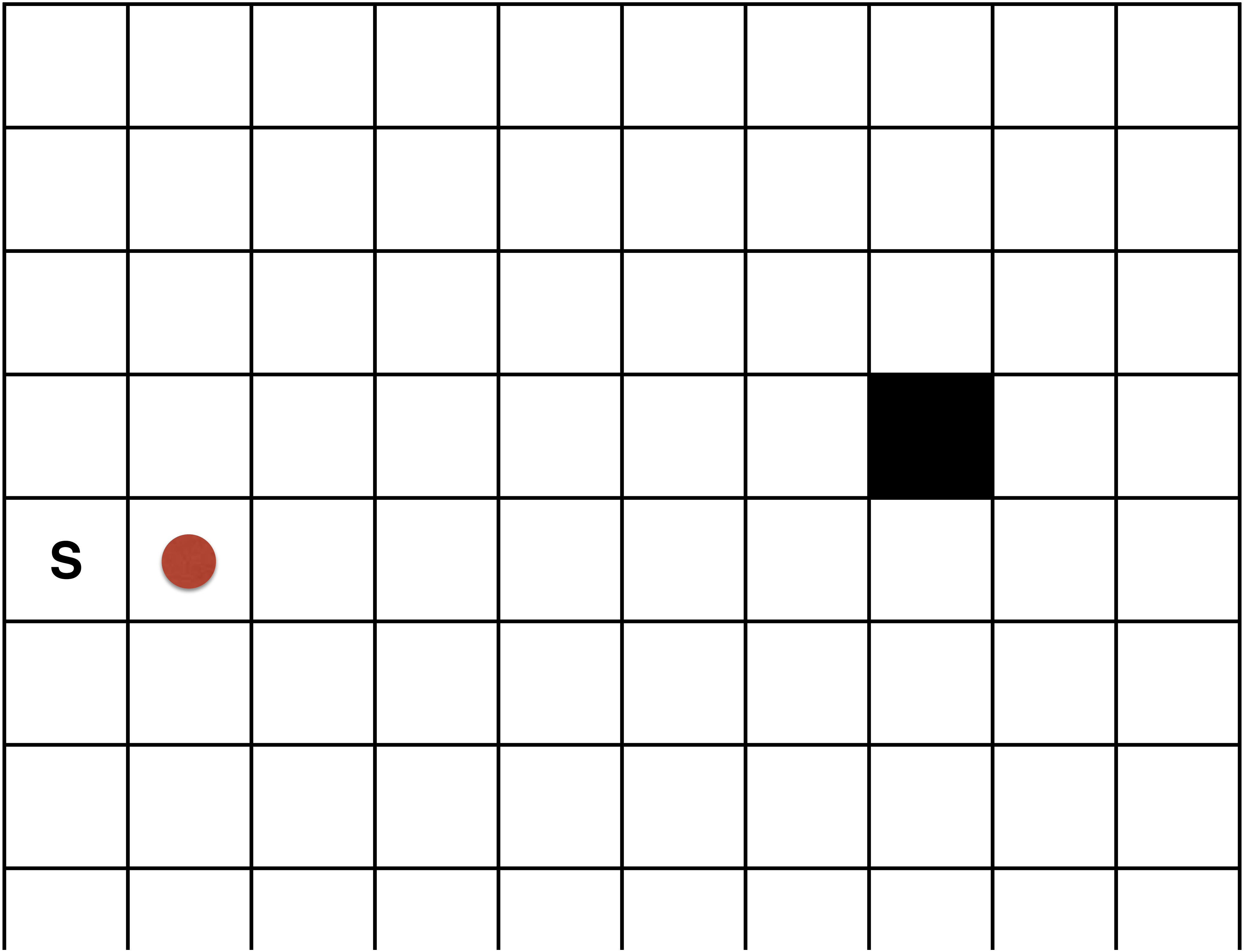
JPS

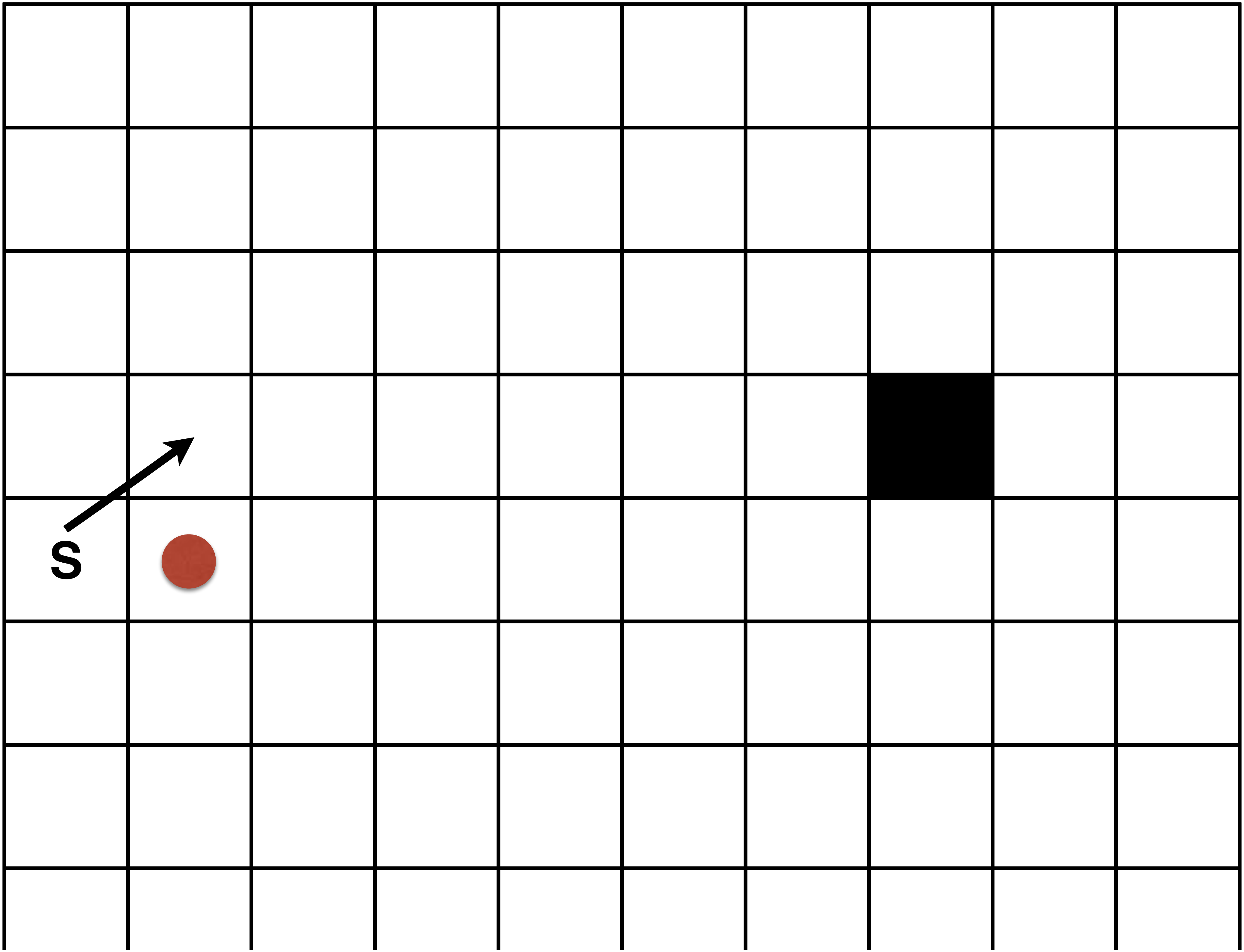


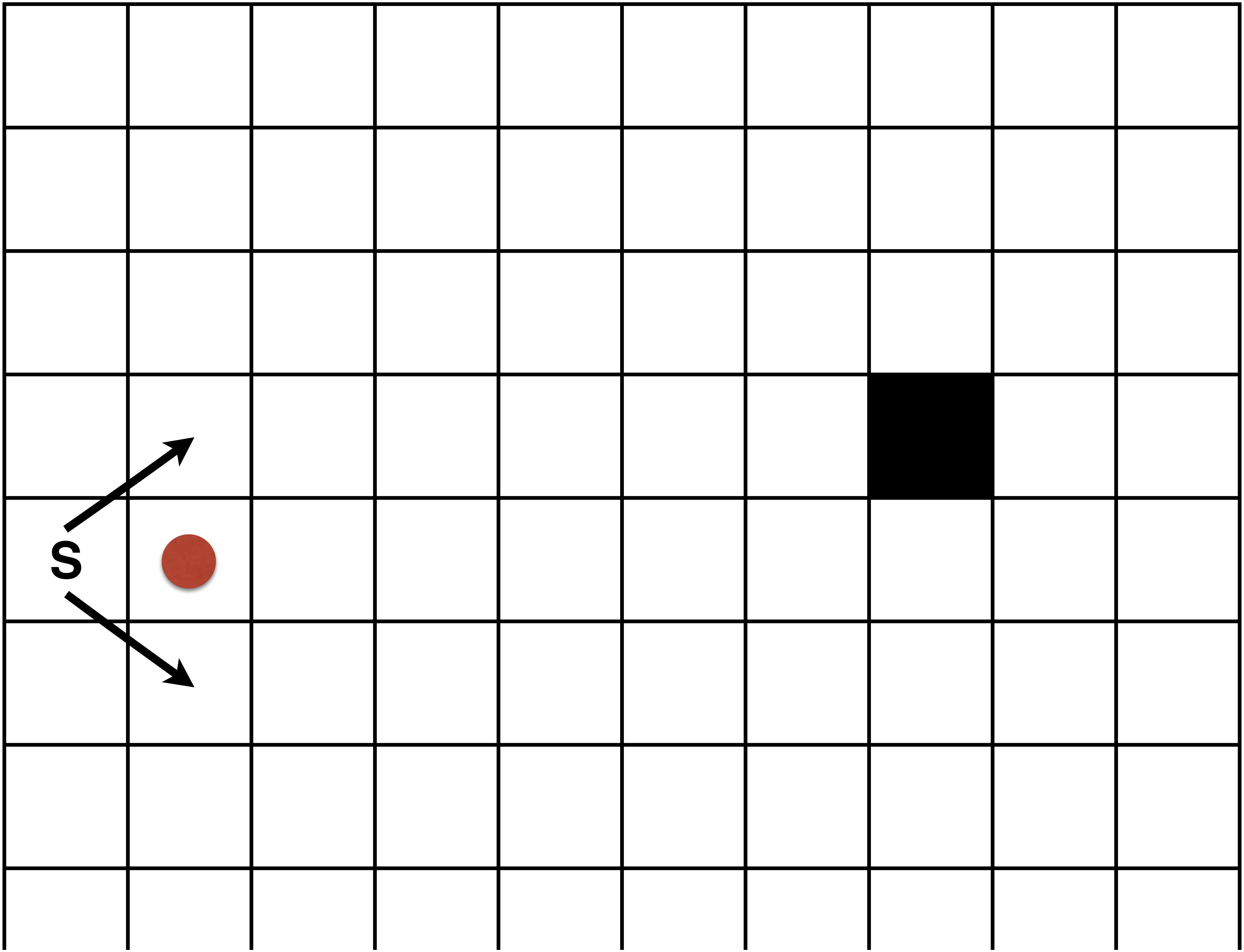
JPS

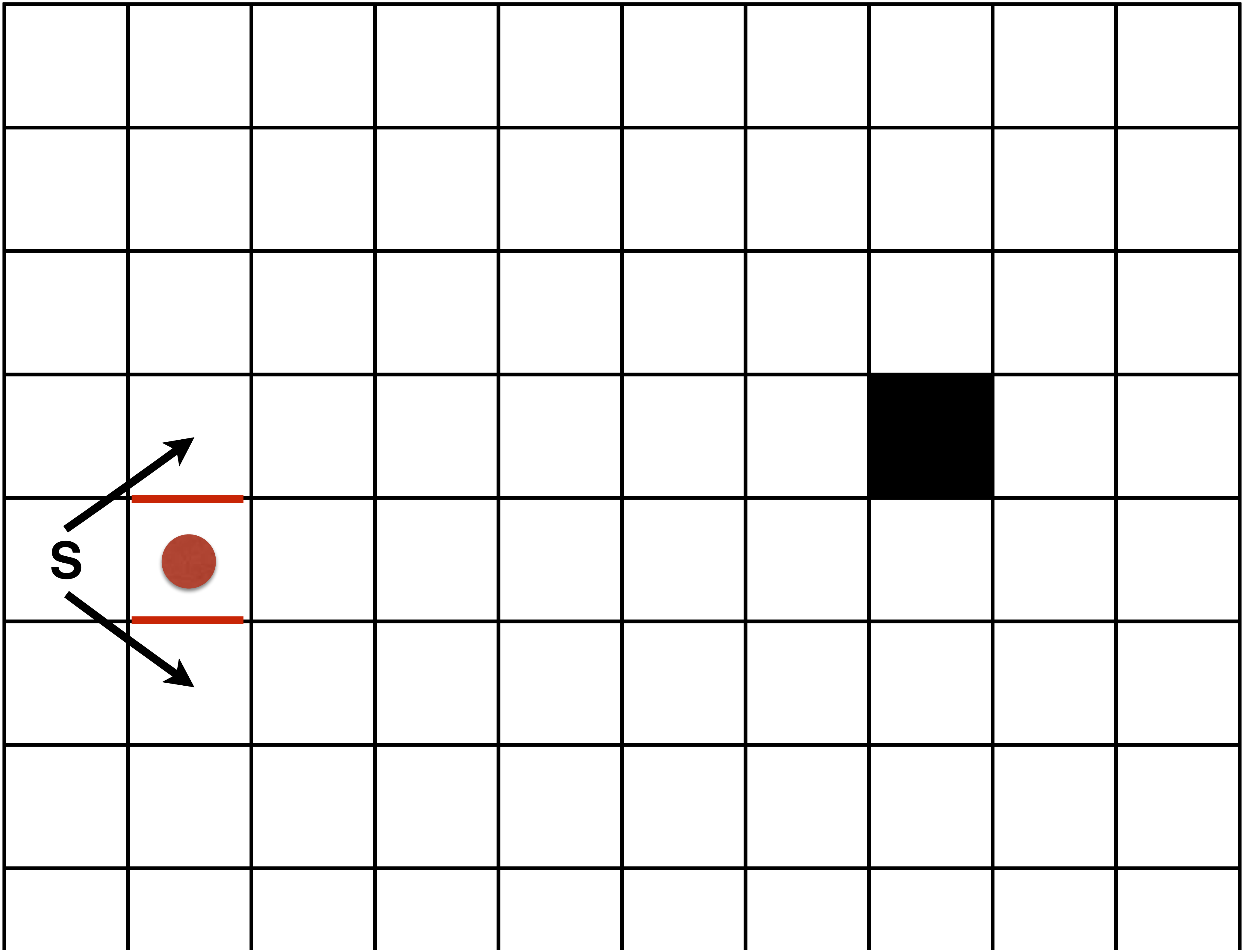


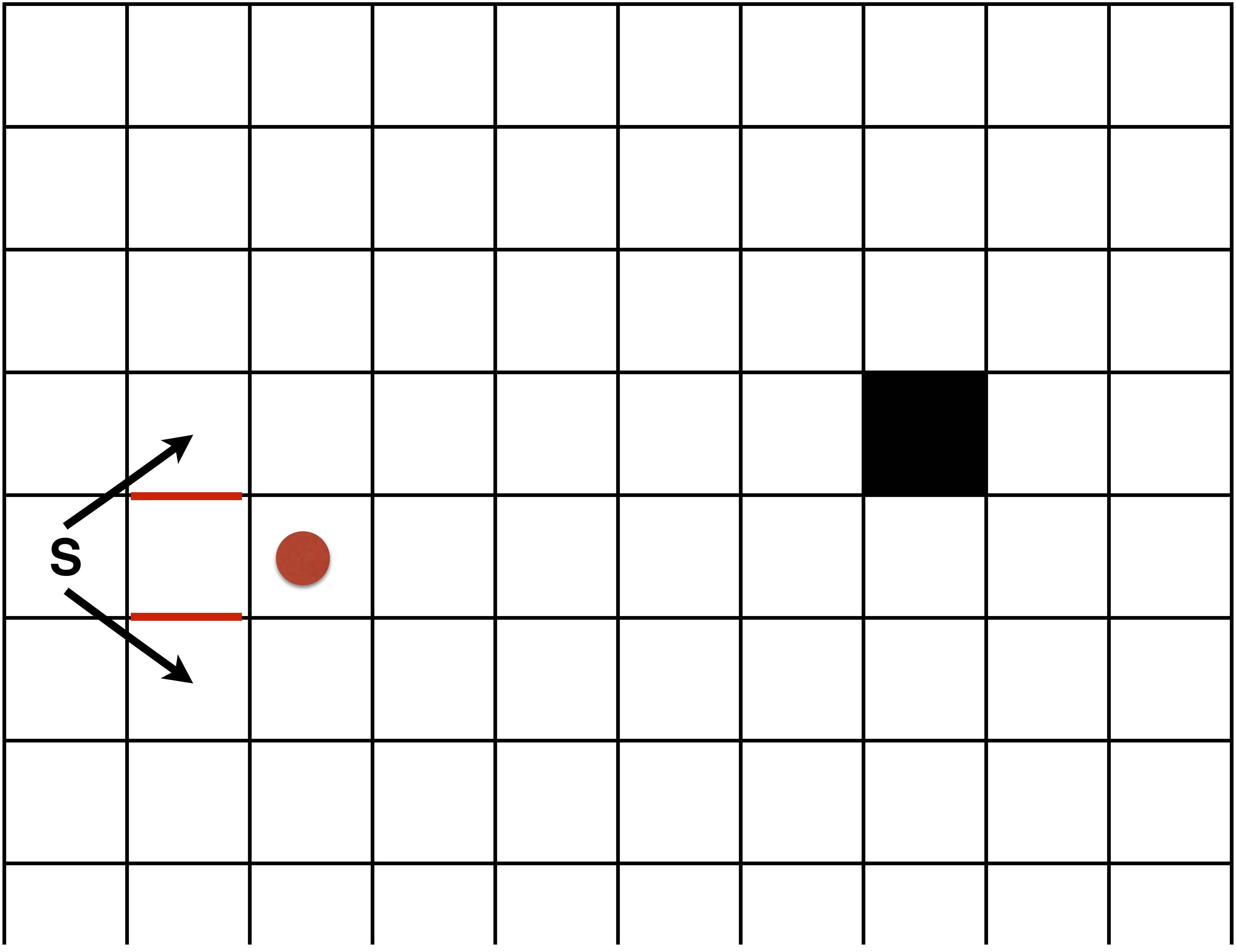

JPS

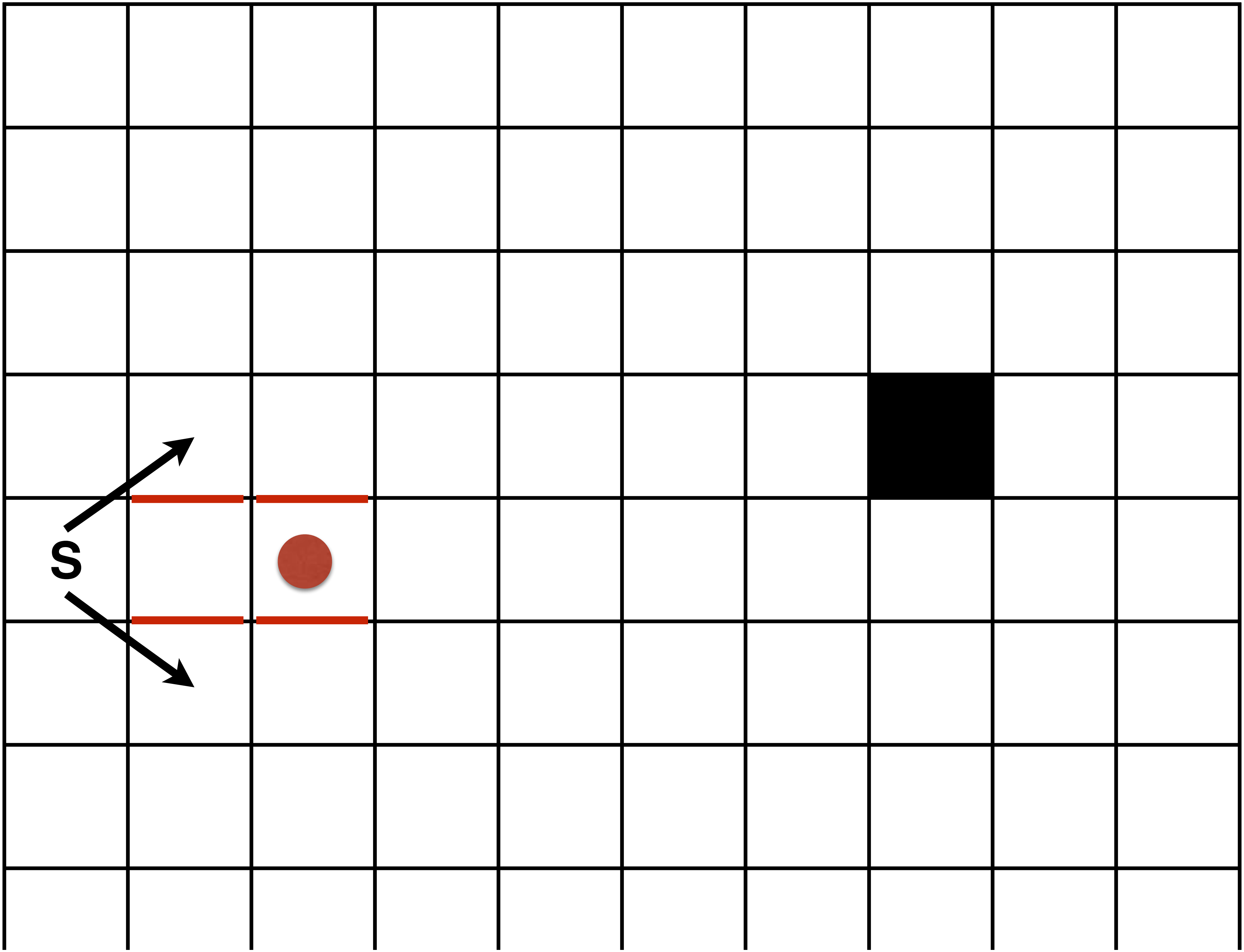

S

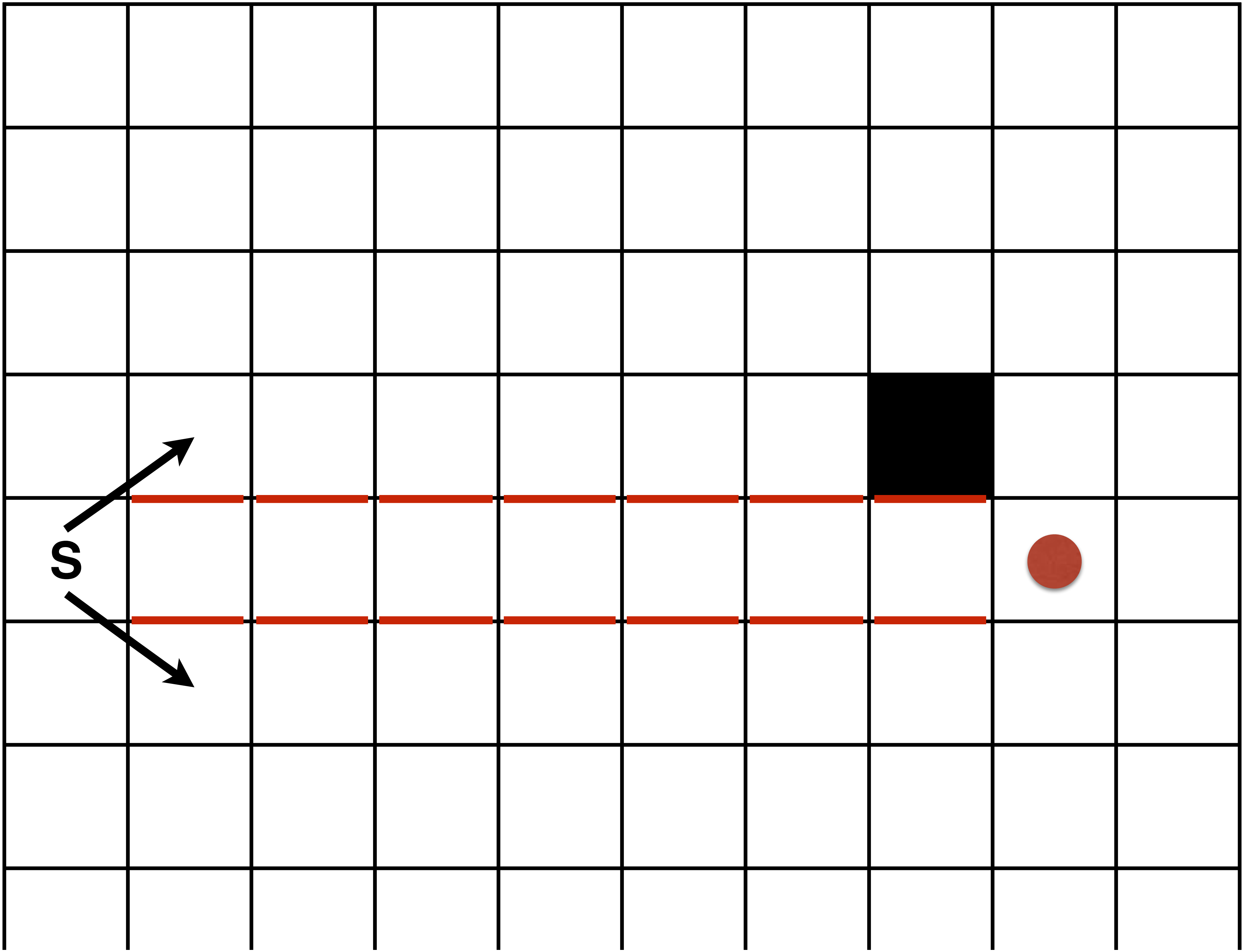

S

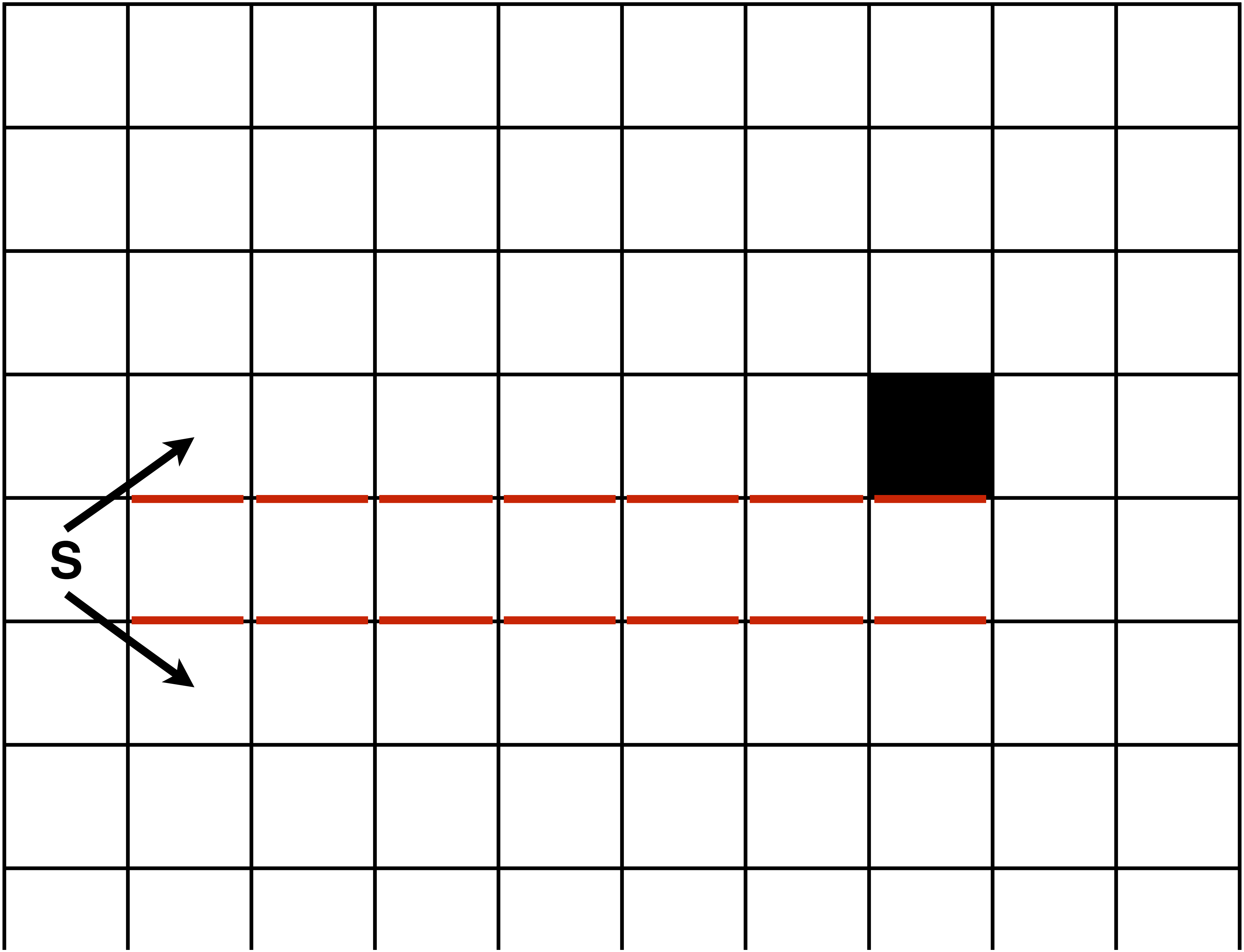

S

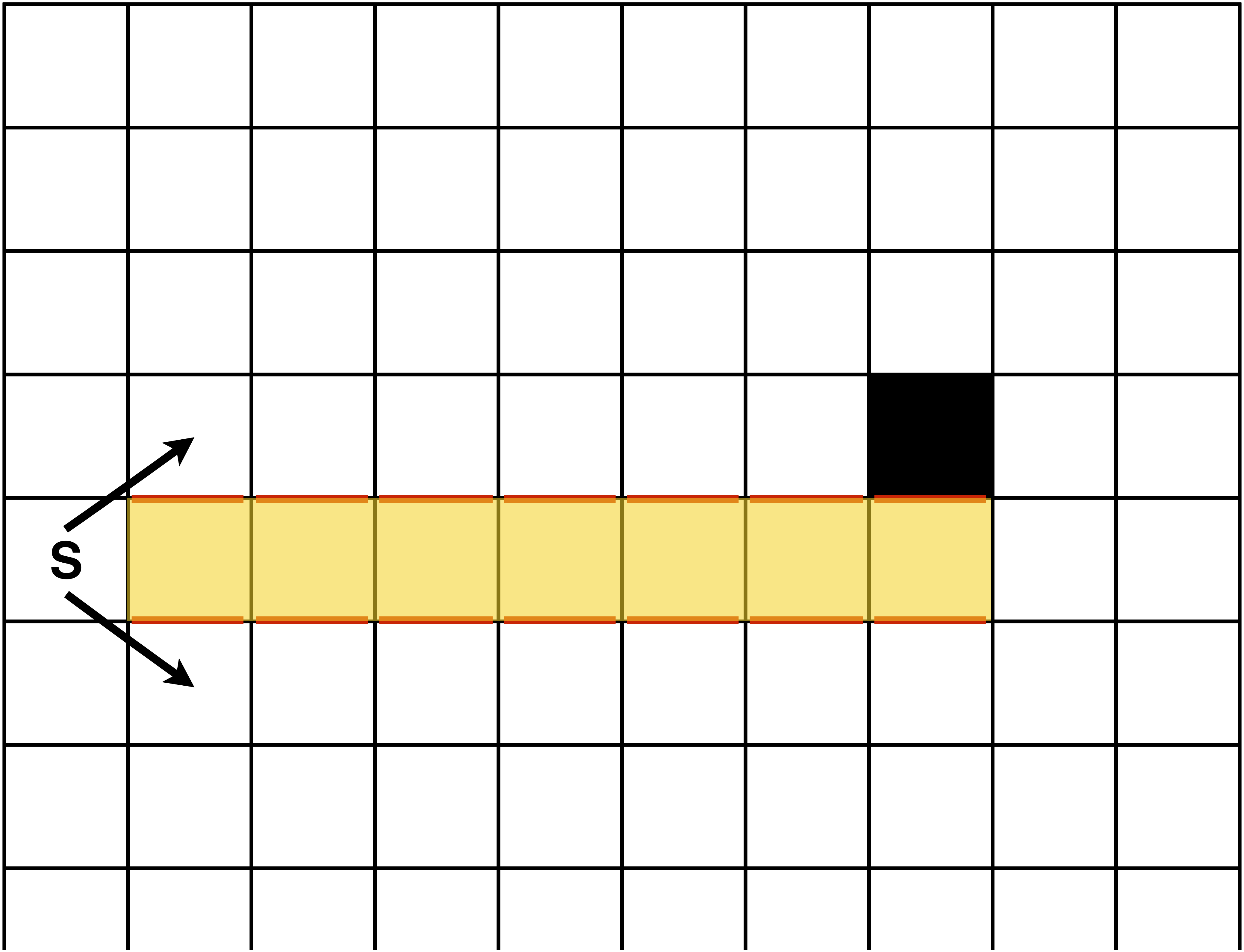

S

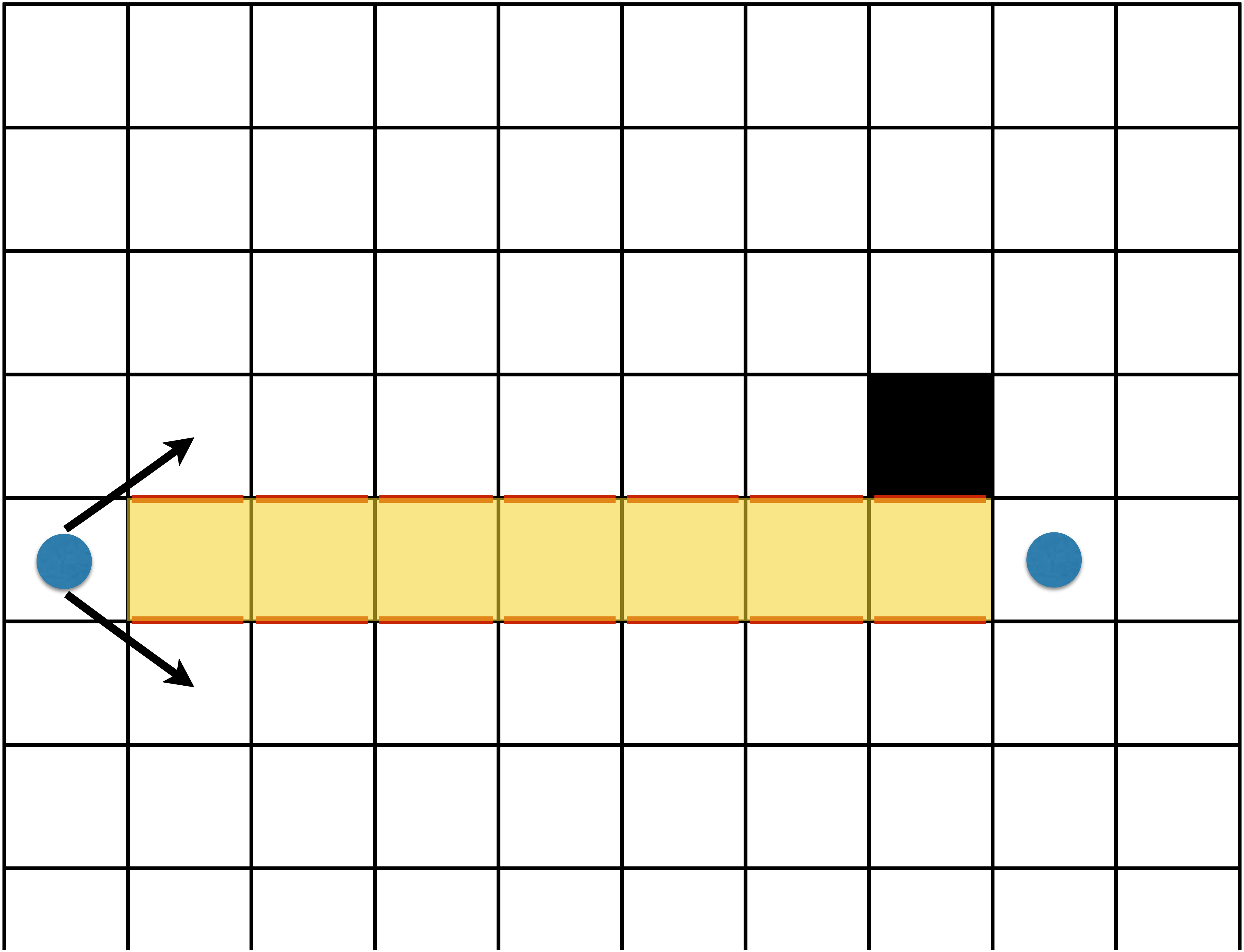

S

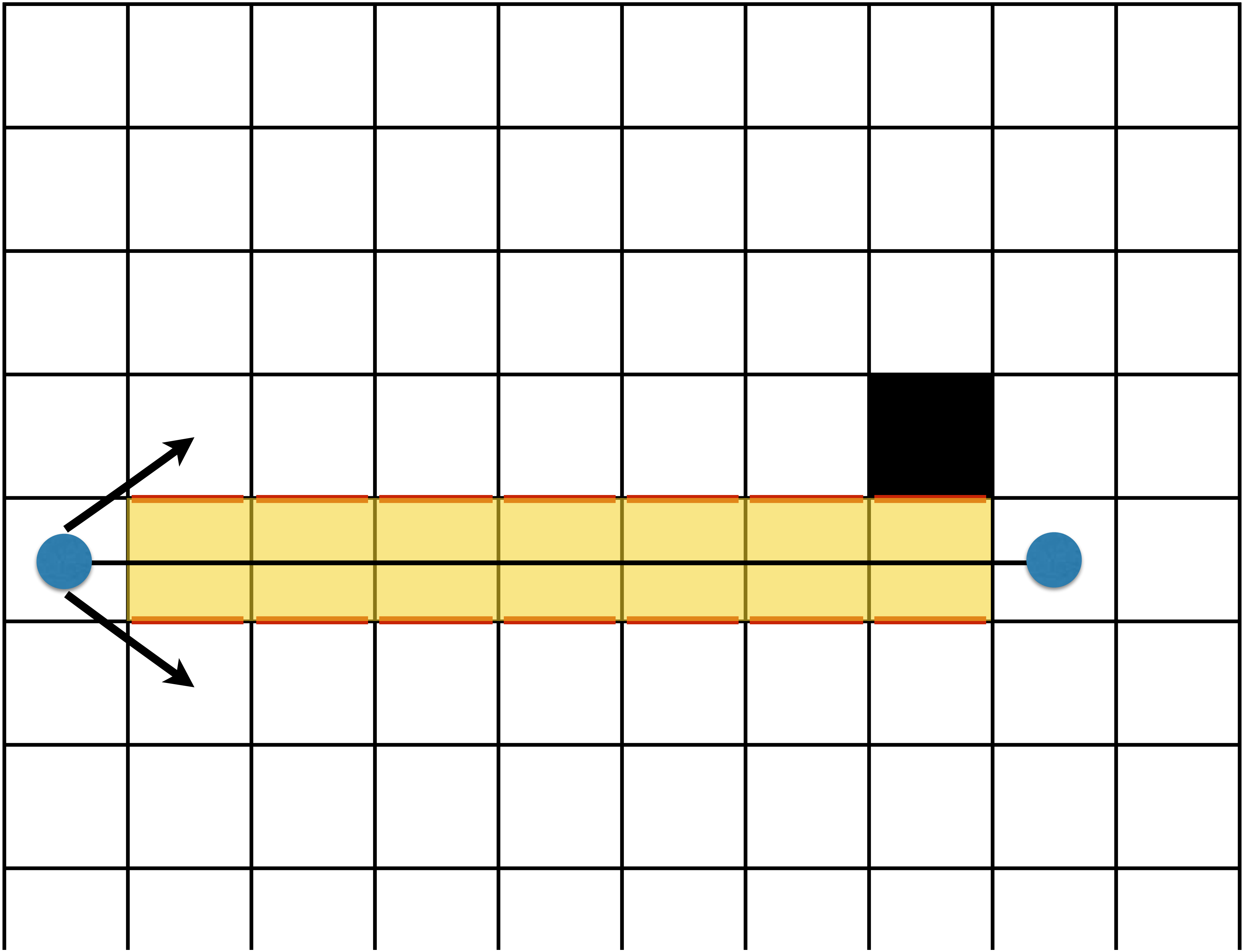

S


S


S


S

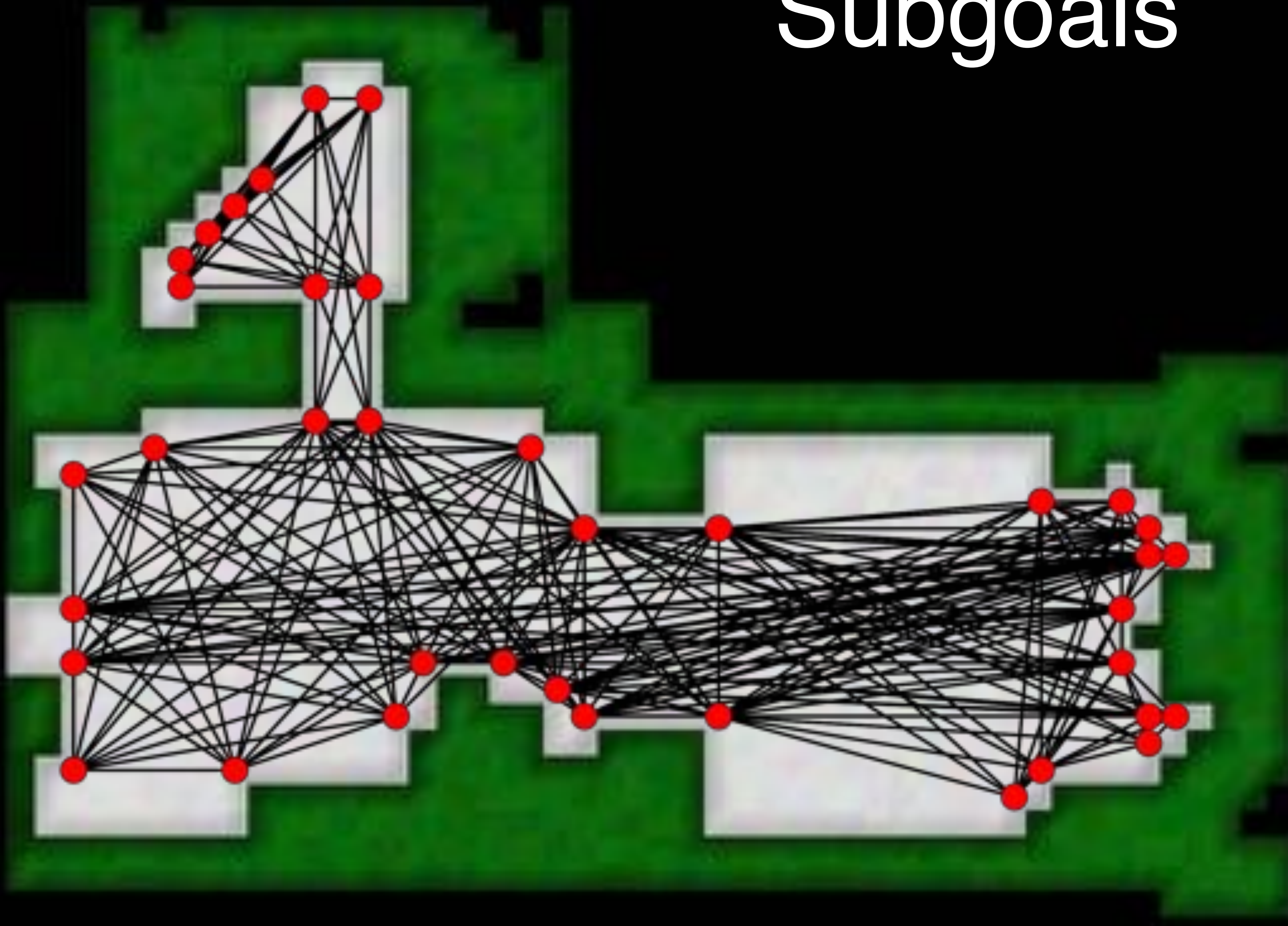




Subgoals

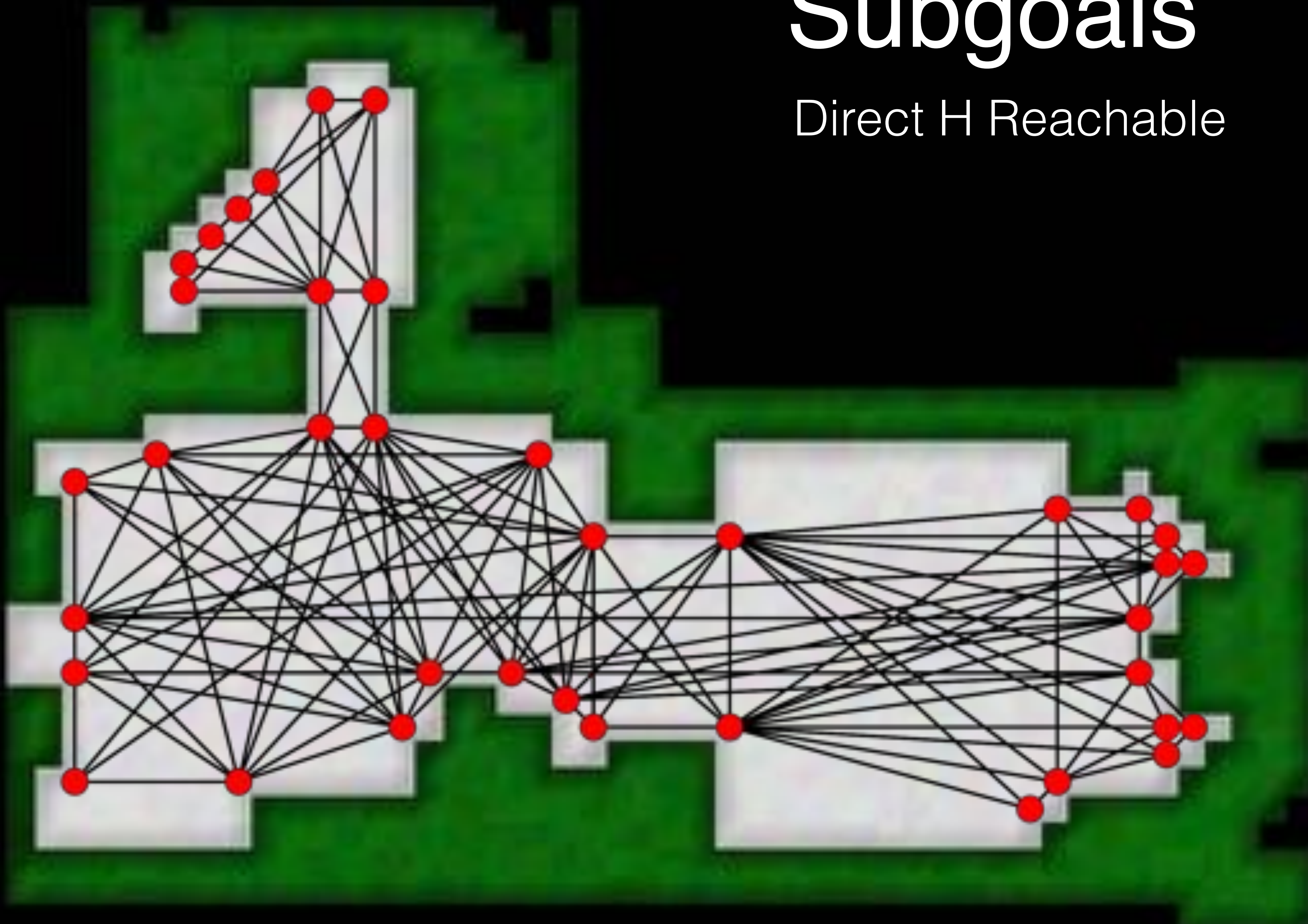
Subgoals

Subgoals



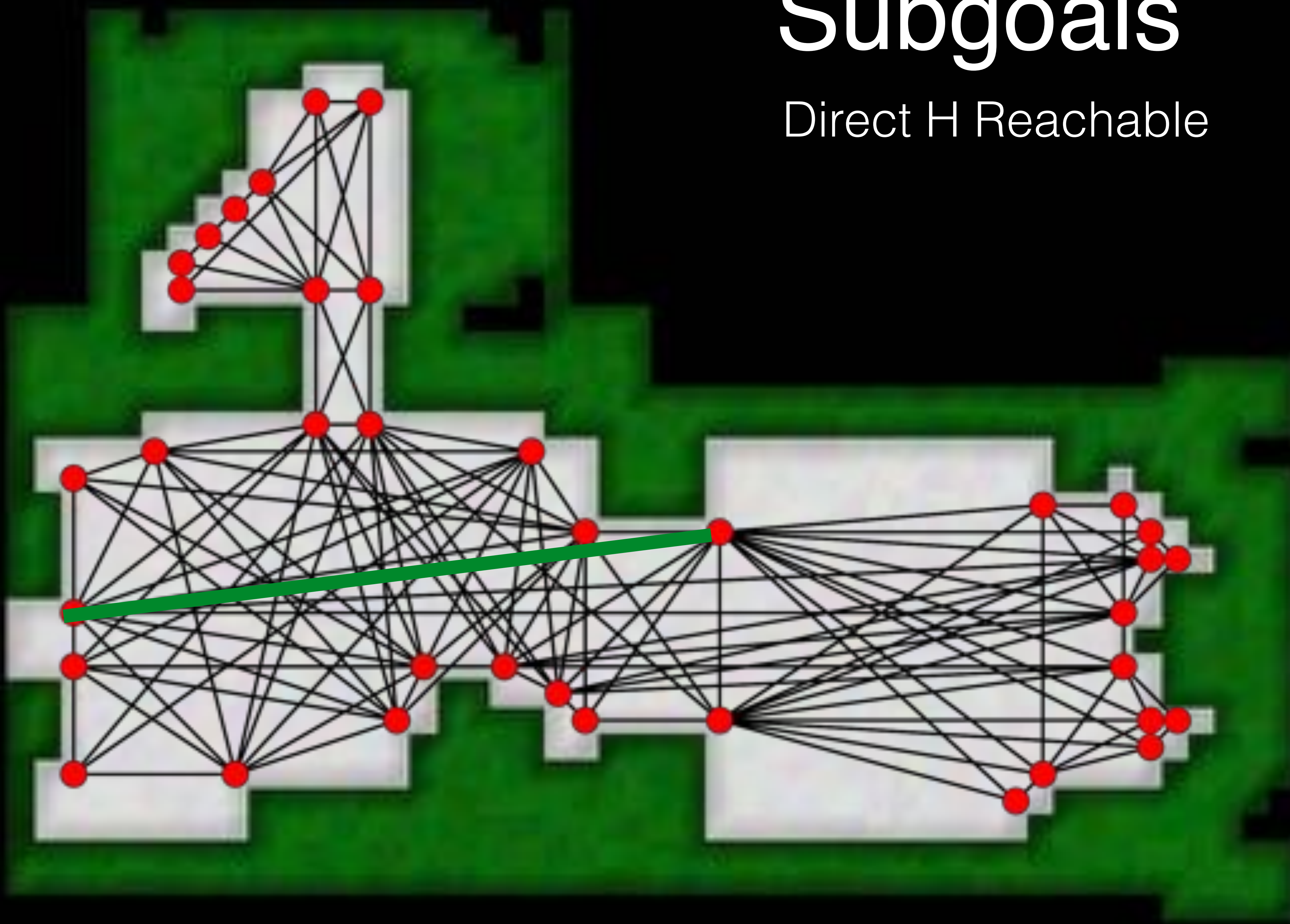
Subgoals

Direct H Reachable



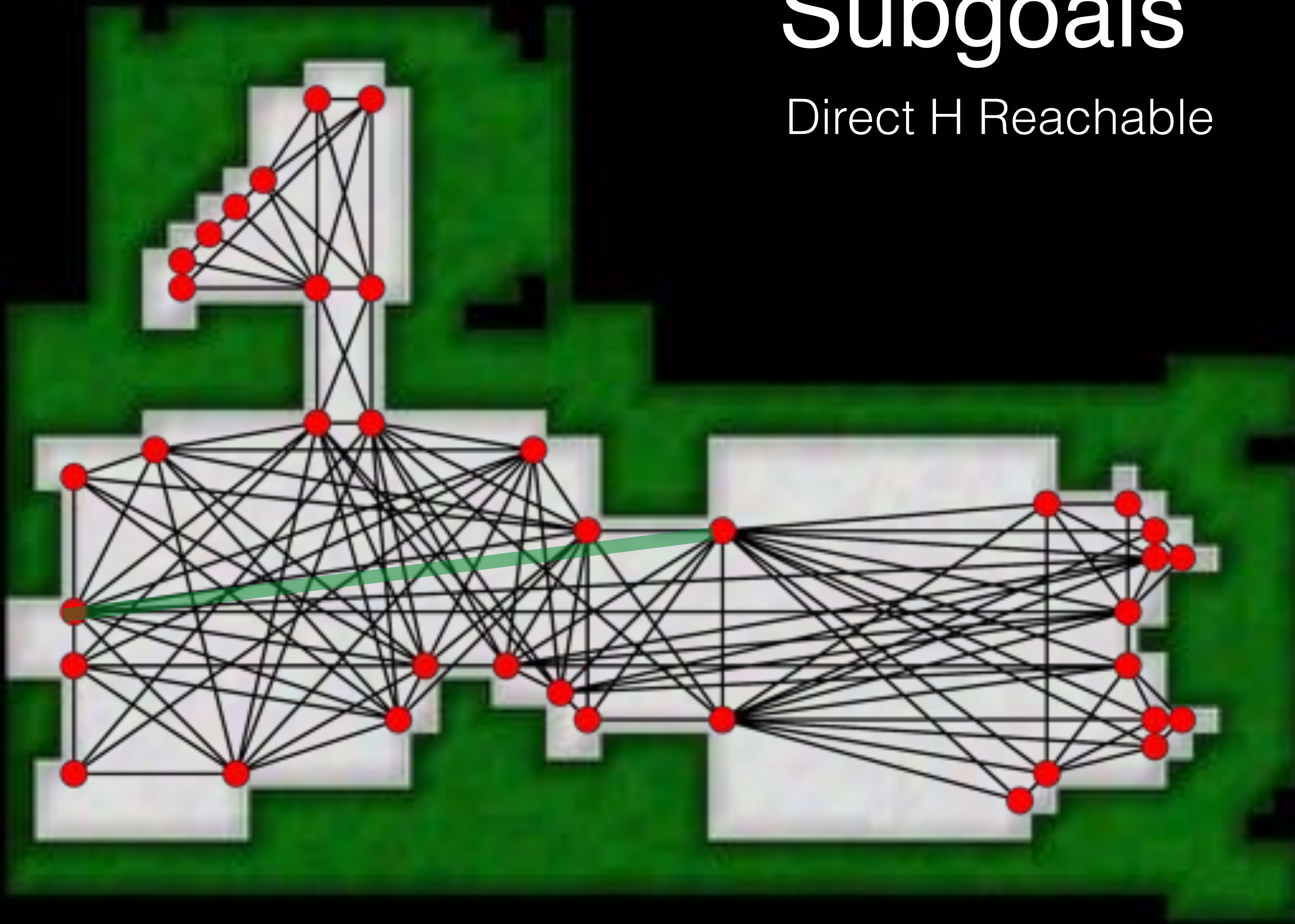
Subgoals

Direct H Reachable



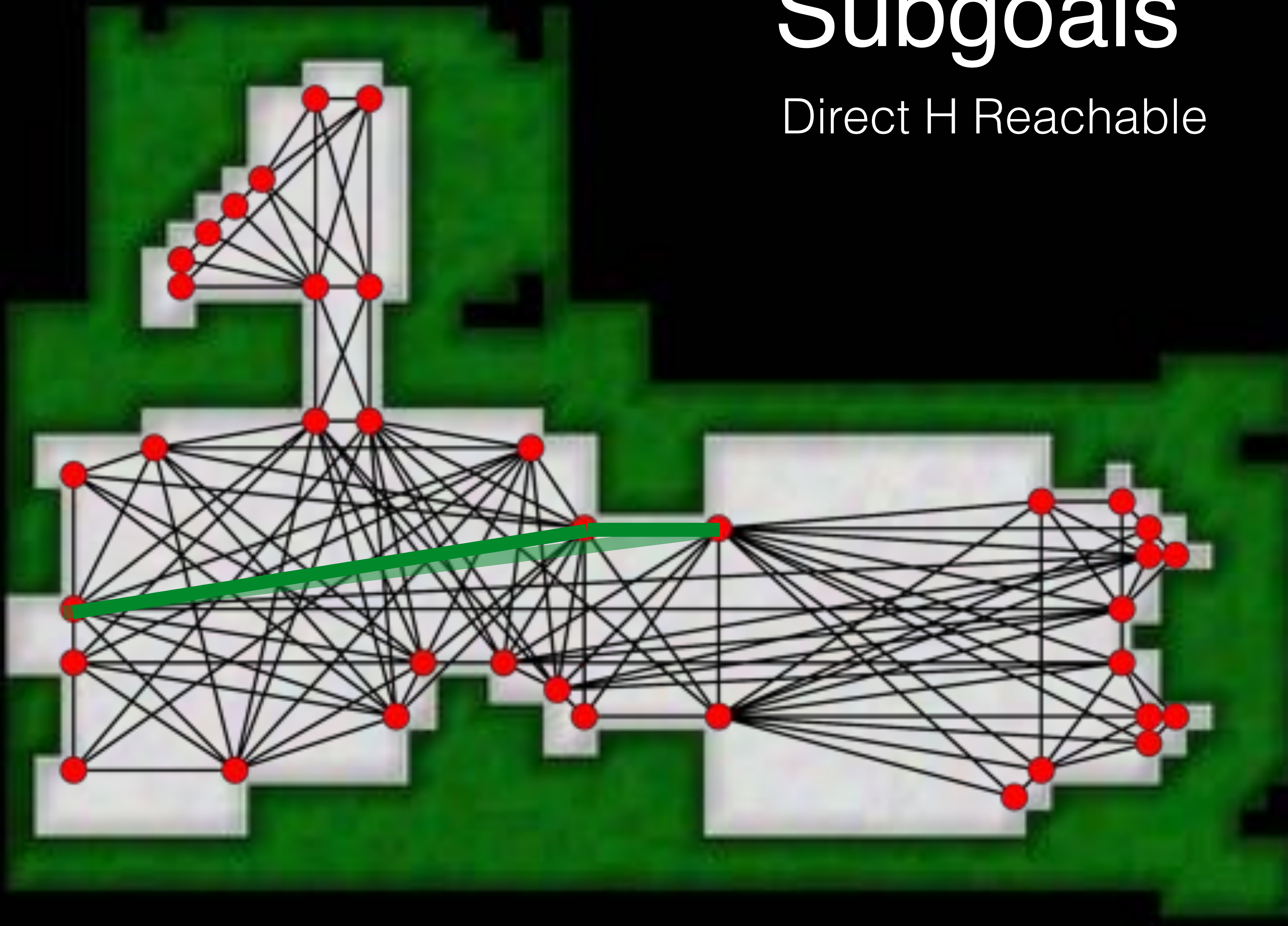
Subgoals

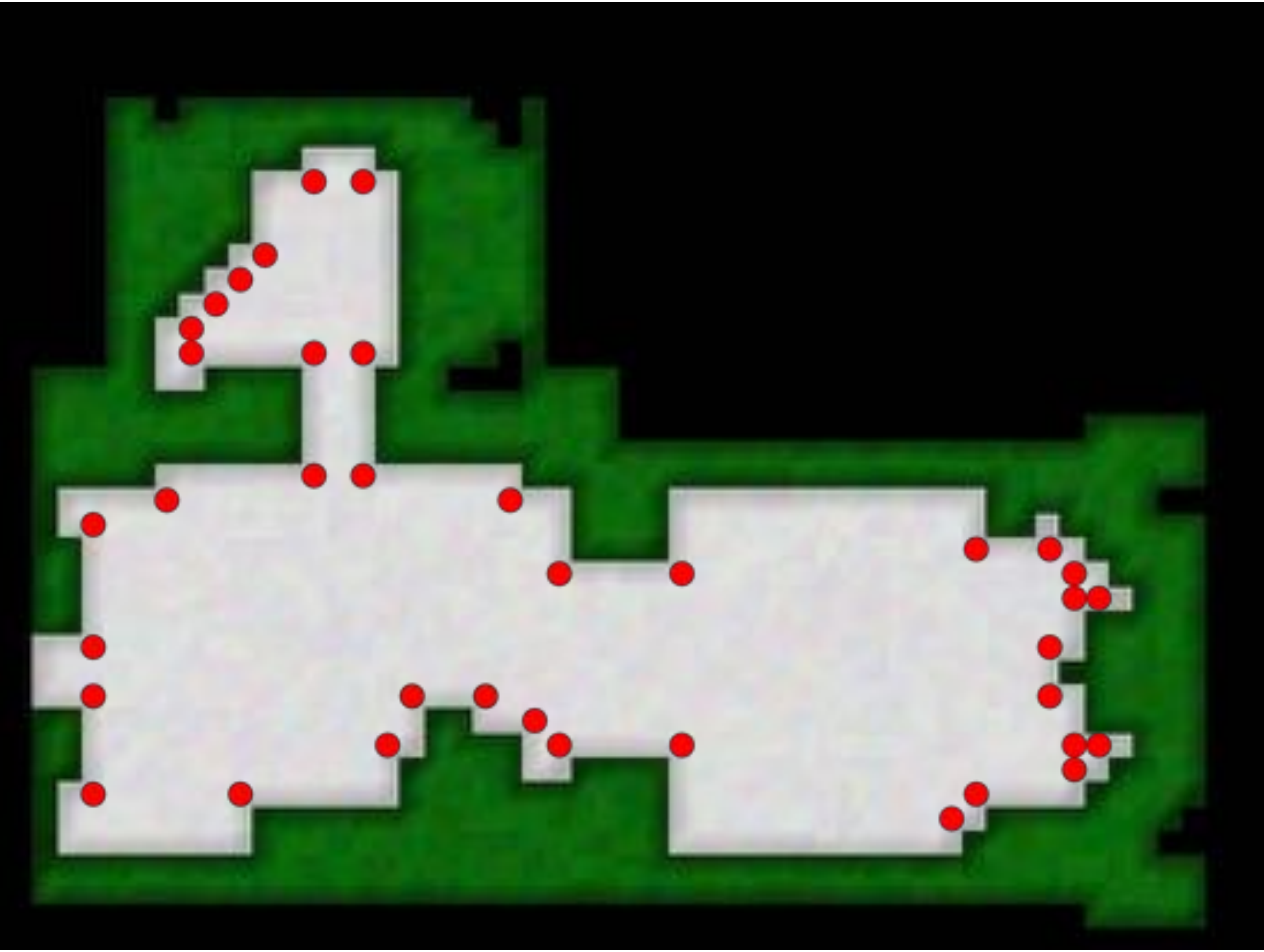
Direct H Reachable

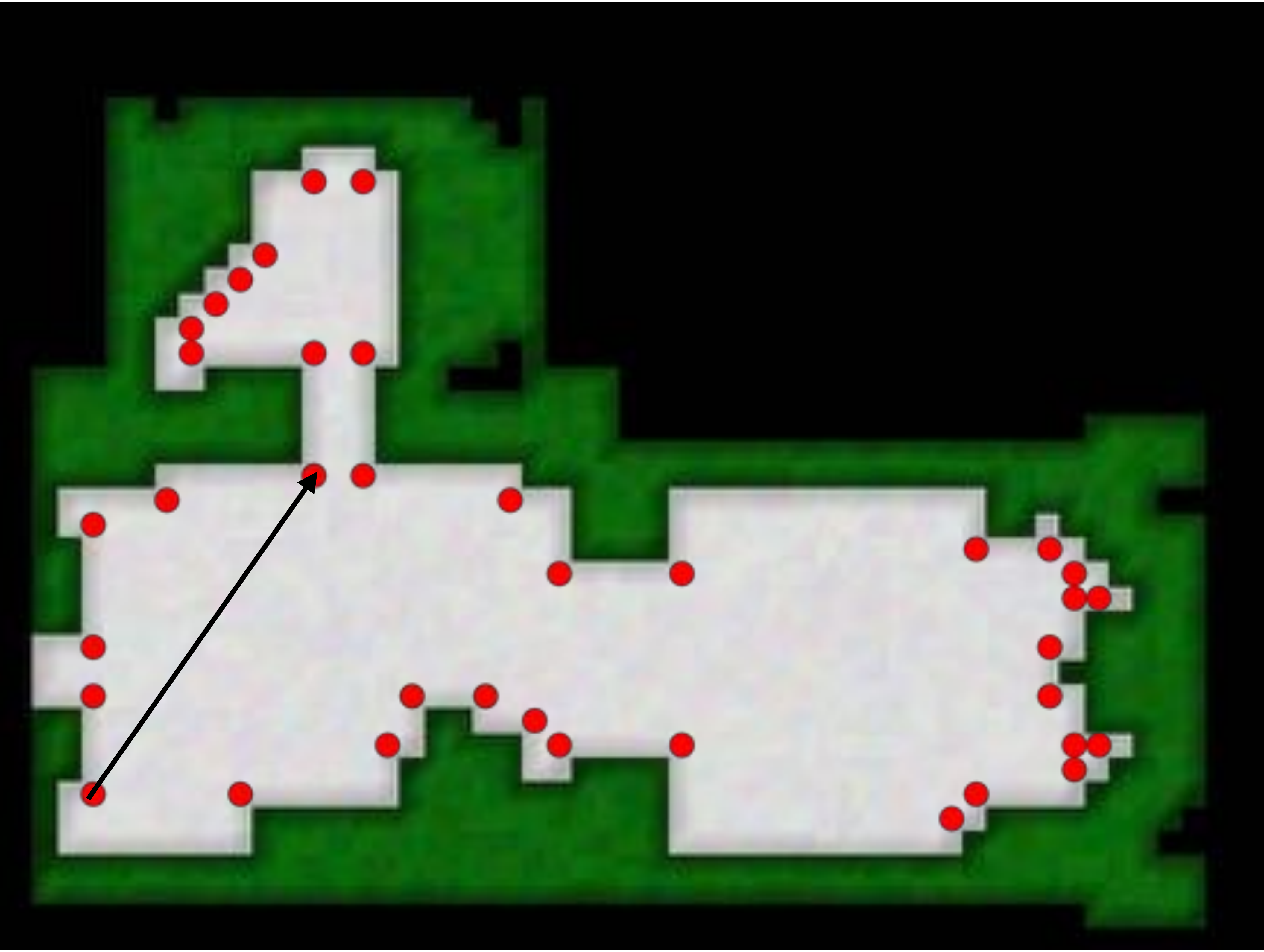


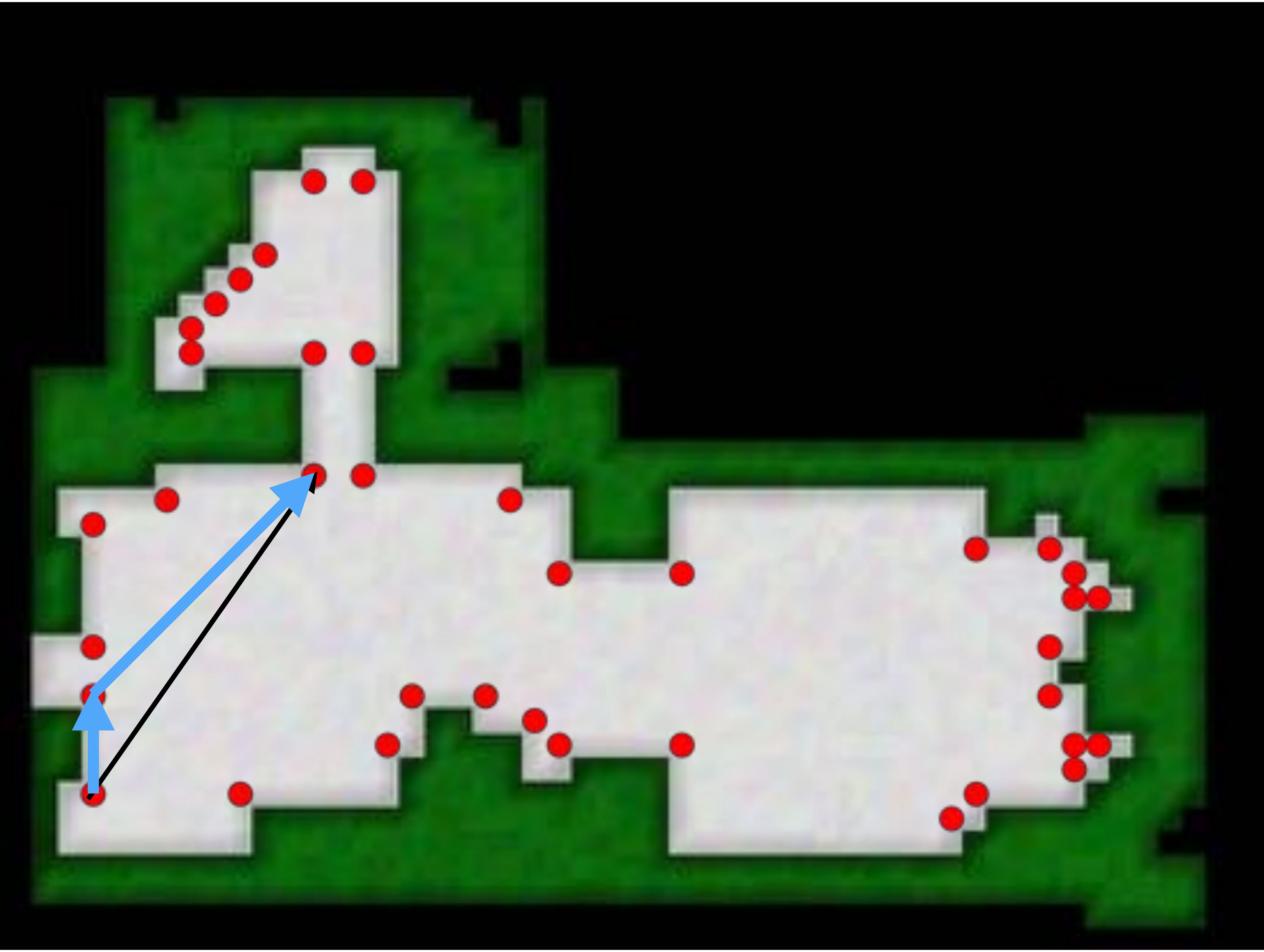
Subgoals

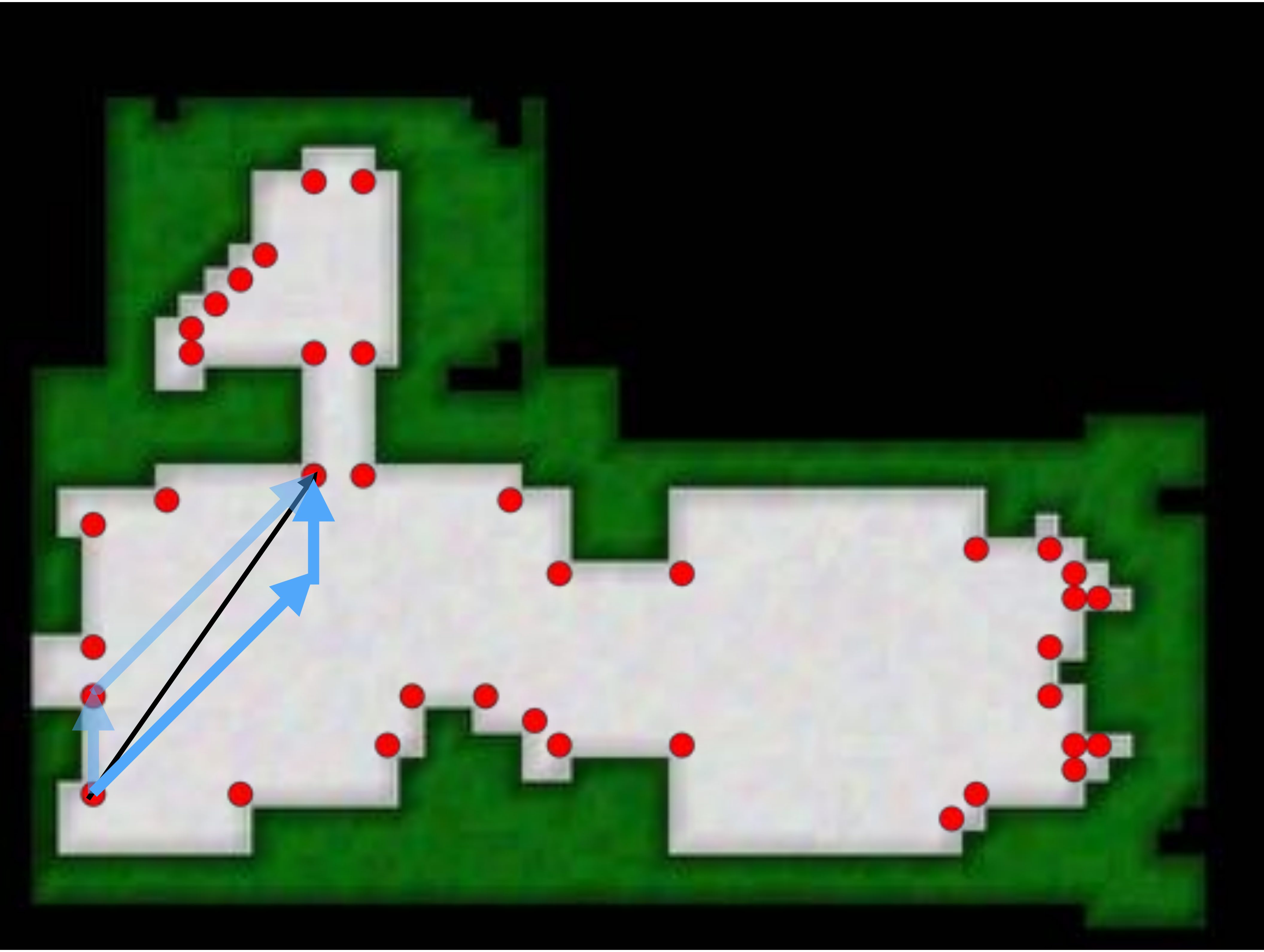
Direct H Reachable

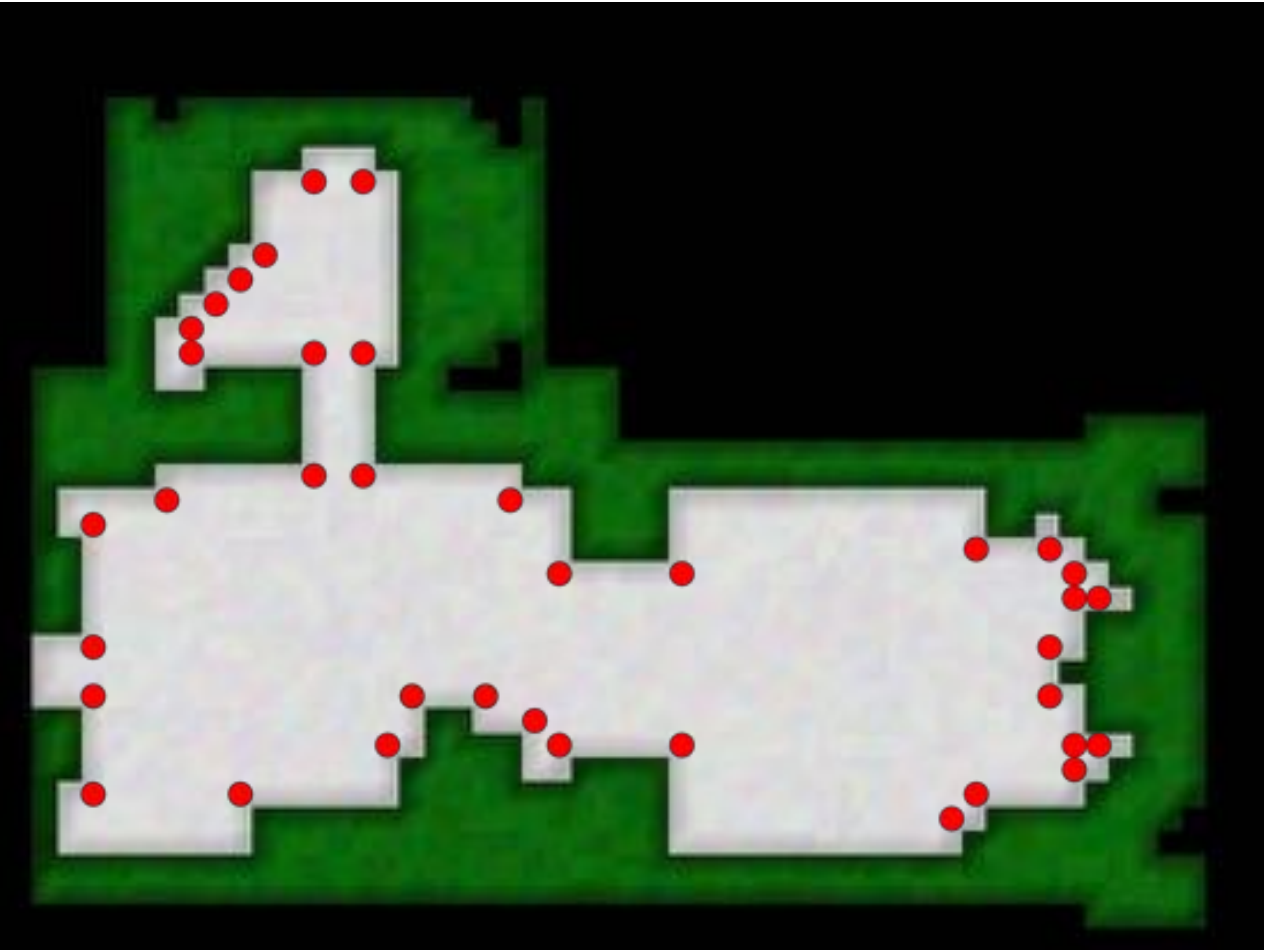












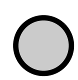
Other contractions

- Dead-end heuristic
  - Yngvi Björnsson and Kári Halldórsson, 2006
- Swamps
  - Nir Pochter, Aviv Zohar, Jeffrey S. Rosenschein, Ariel Felner [2008 - 2010]
- Dead/redundant states
  - N.R. Sturtevant, V. Bulitko and Y. Bjornsson, 2010

- What is the correct way to contract a map?

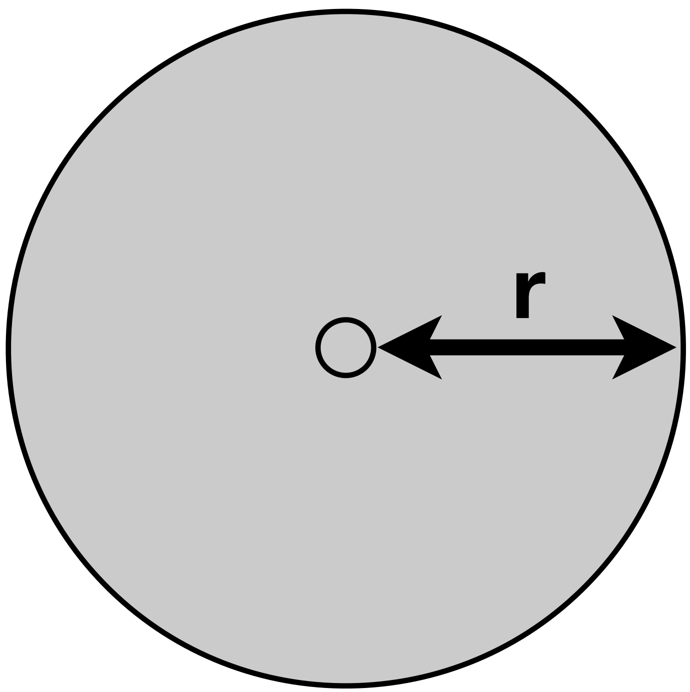
Highway Dimension

Highway Dimension, Shortest Paths, and Provably Efficient Algorithms [Abraham et. al., 2010]



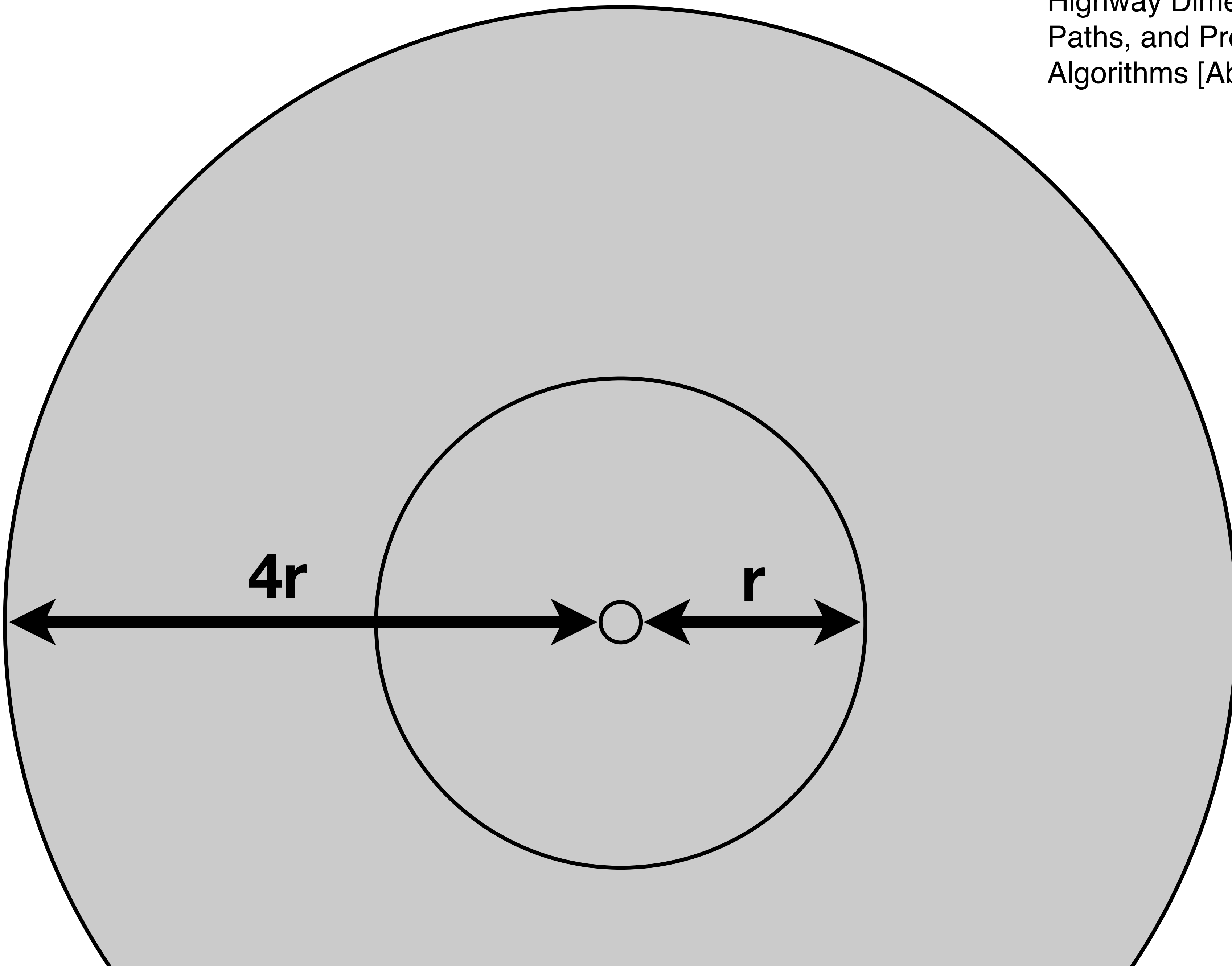
Highway Dimension

Highway Dimension, Shortest Paths, and Provably Efficient Algorithms [Abraham et. al., 2010]



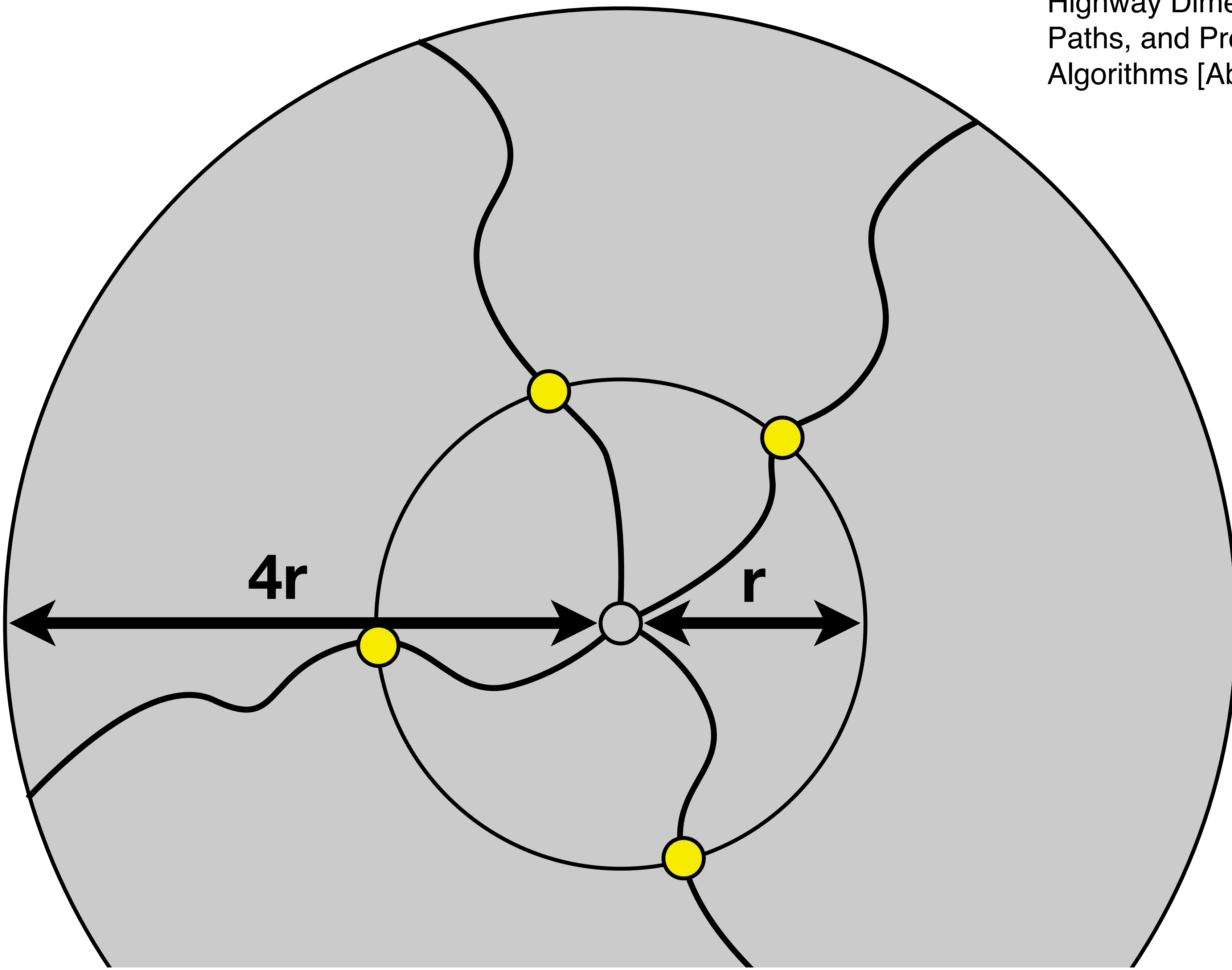
Highway Dimension

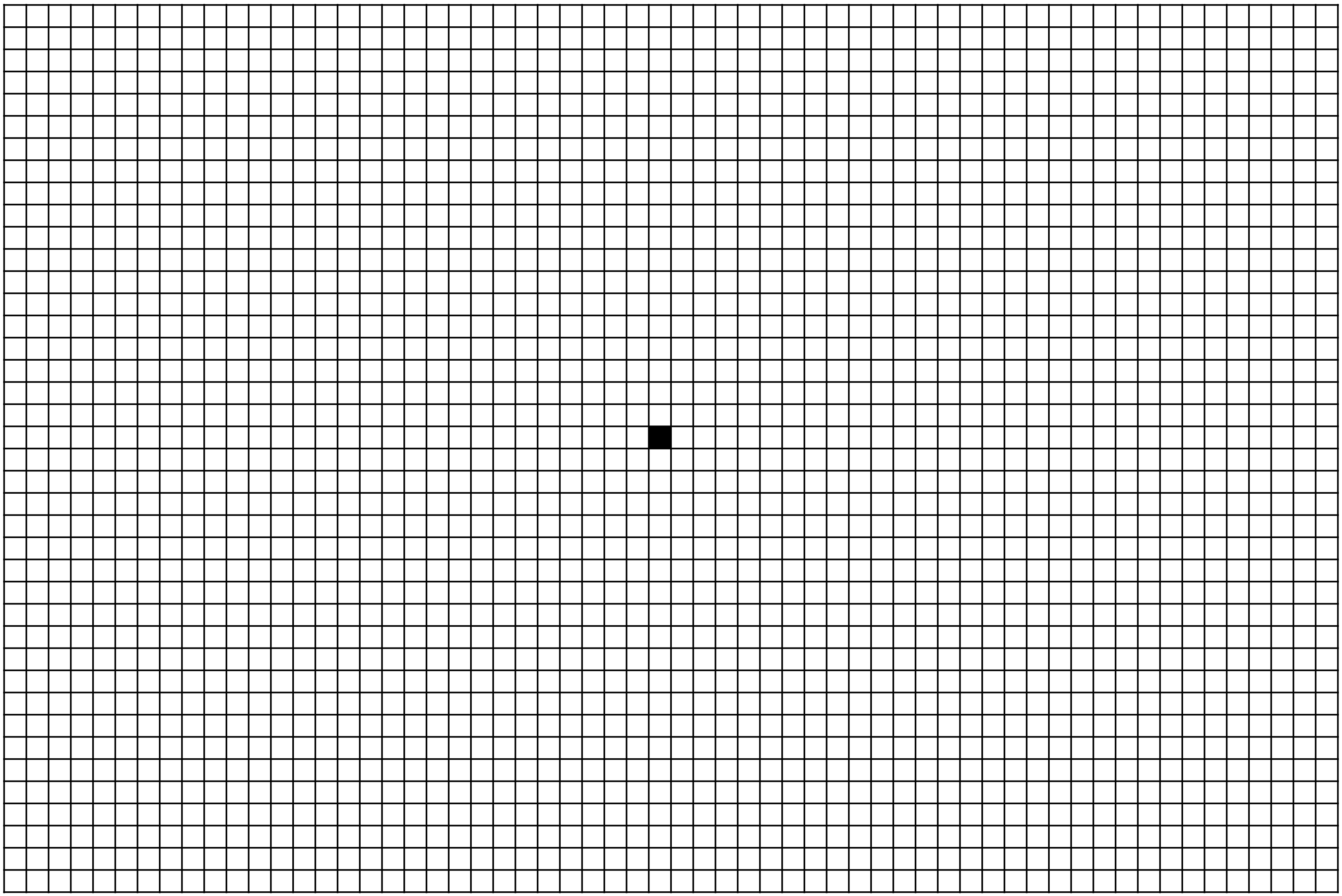
Highway Dimension, Shortest Paths, and Provably Efficient Algorithms [Abraham et. al., 2010]

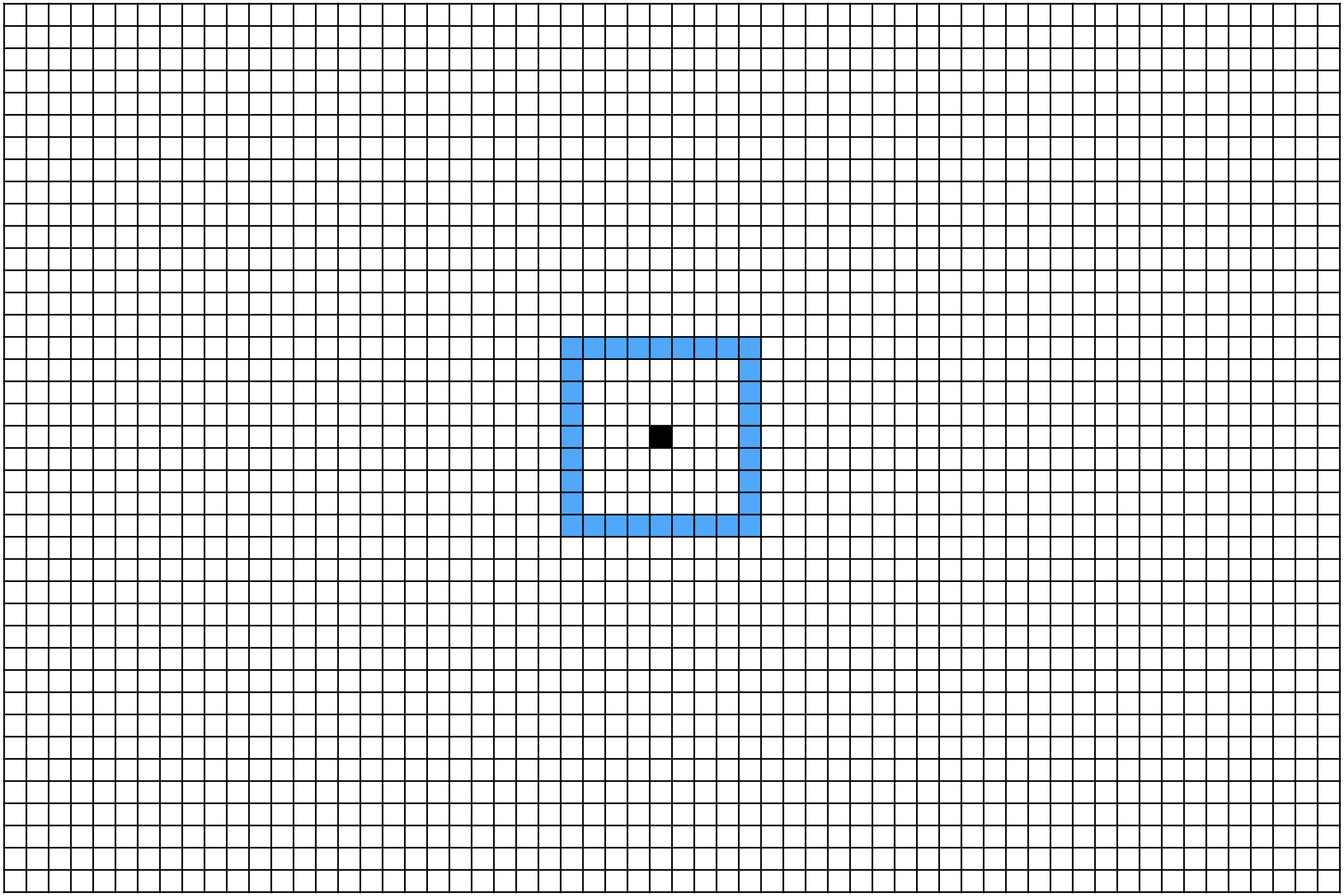


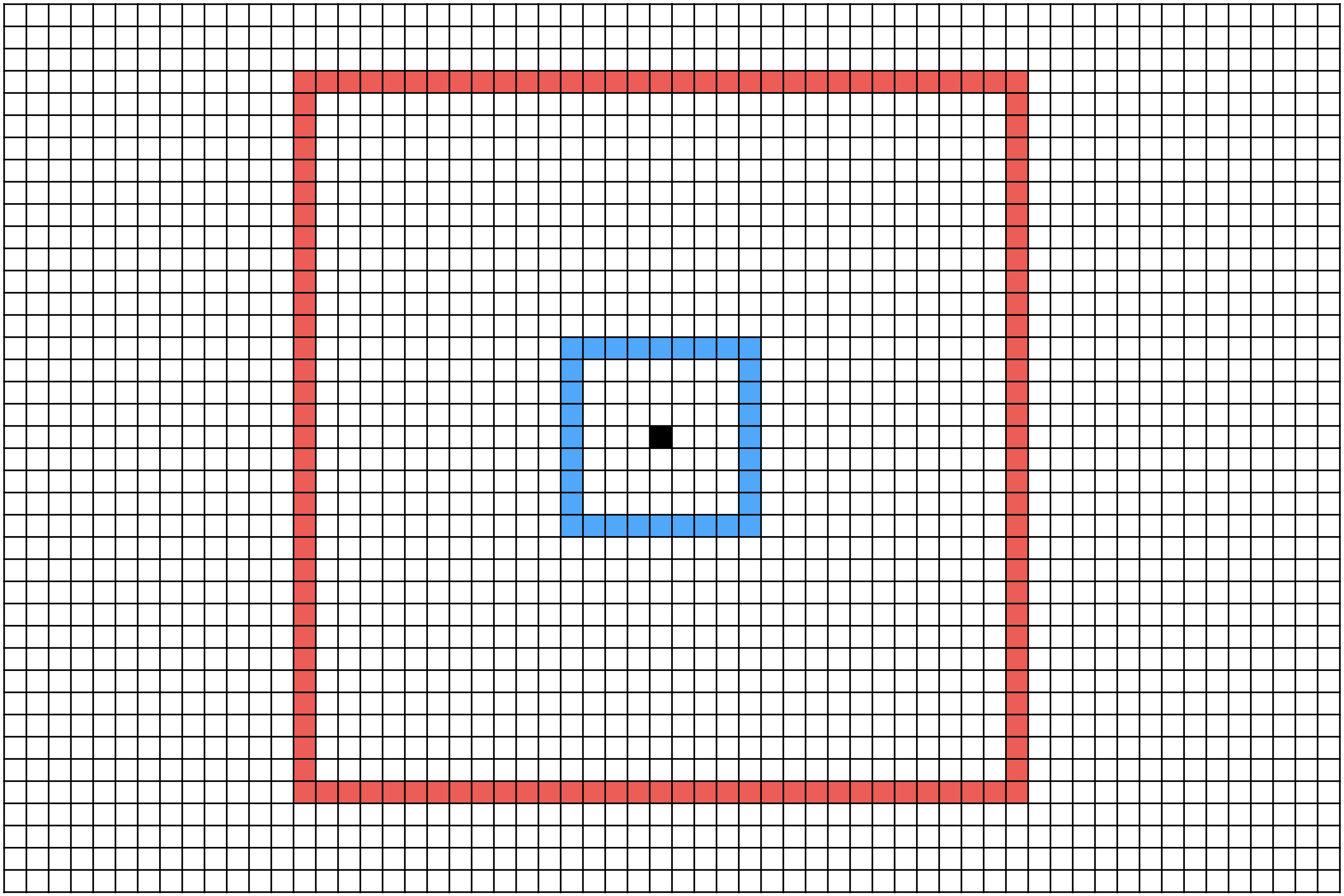
Highway Dimension

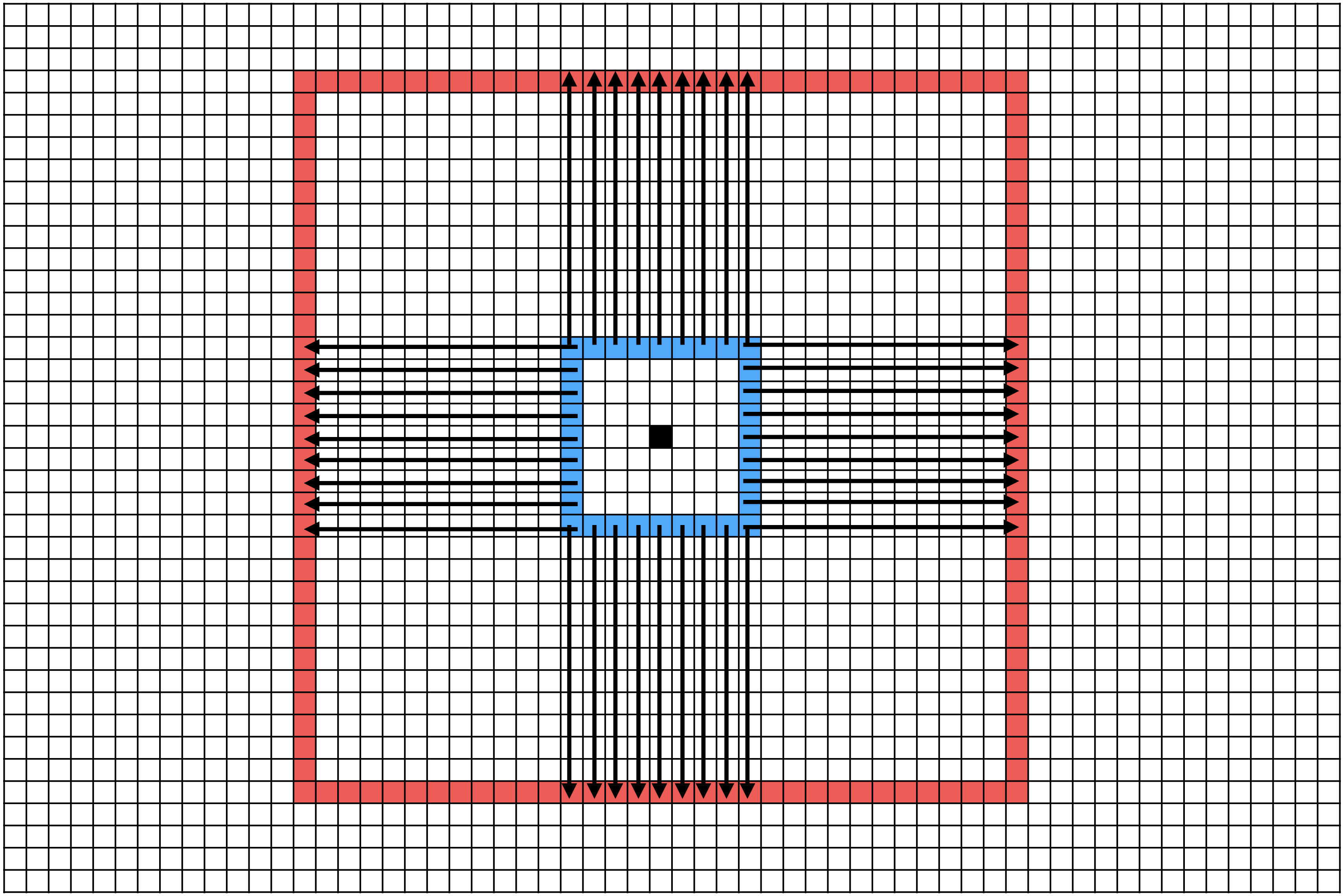
Highway Dimension, Shortest Paths, and Provably Efficient Algorithms [Abraham et. al., 2010]

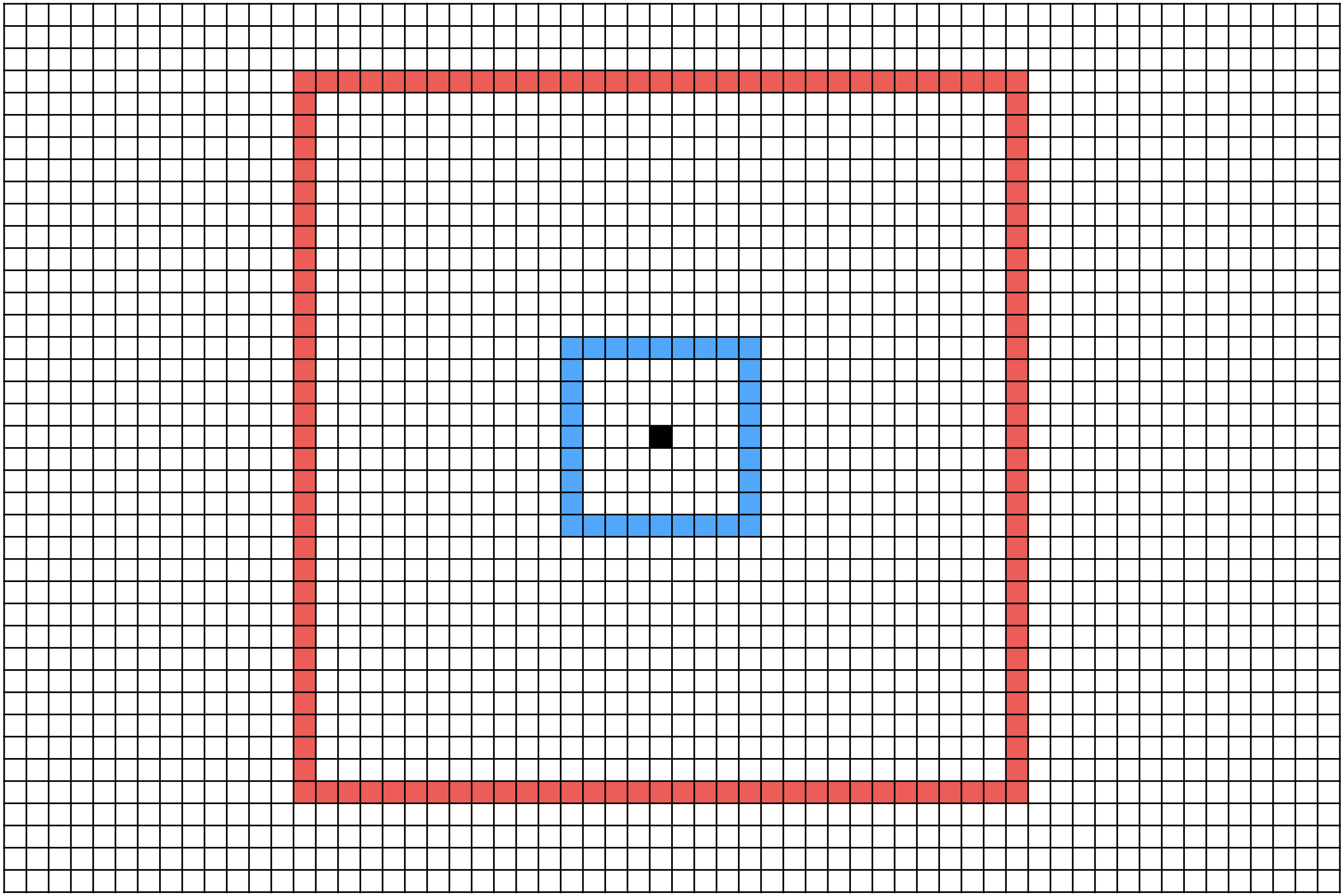


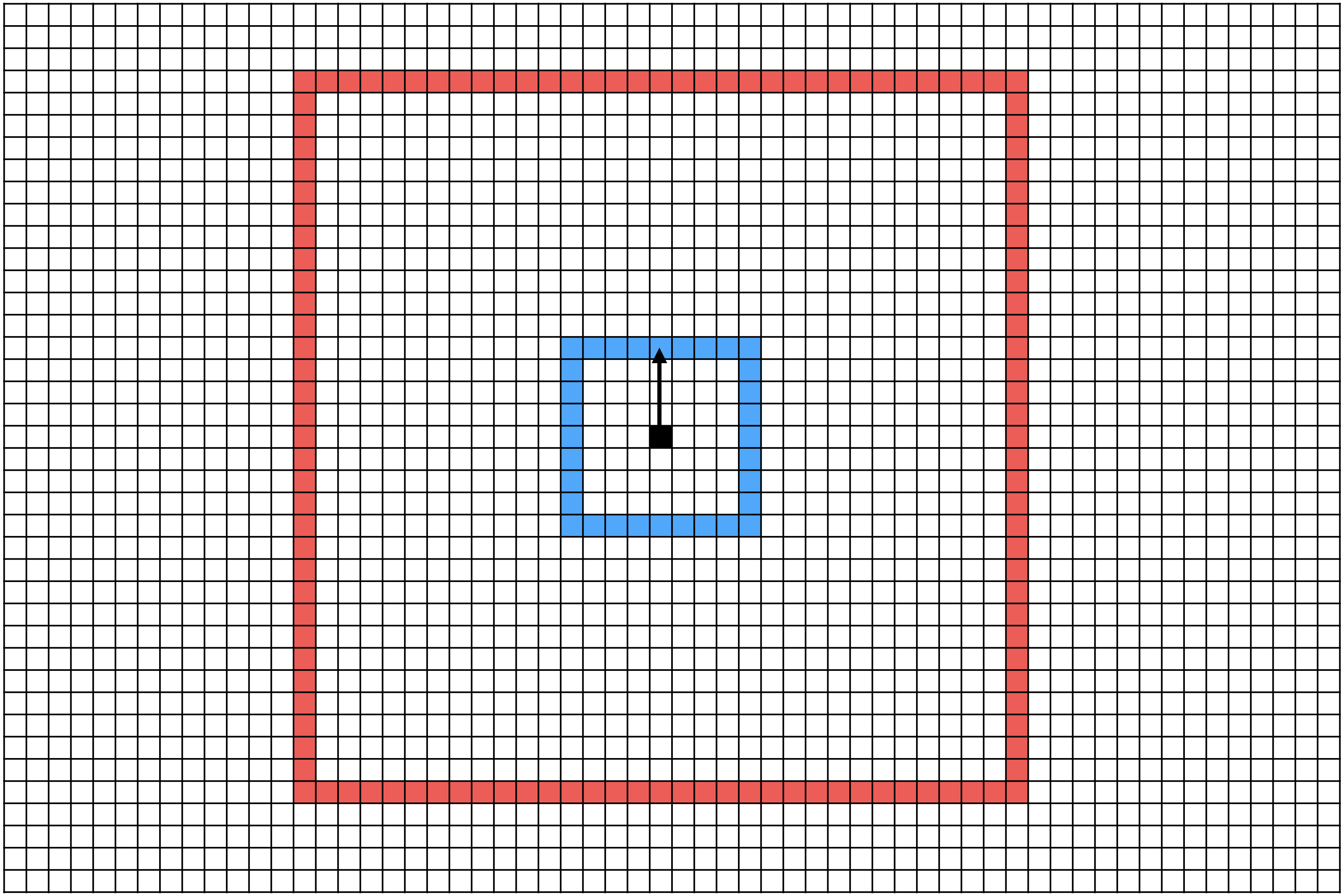


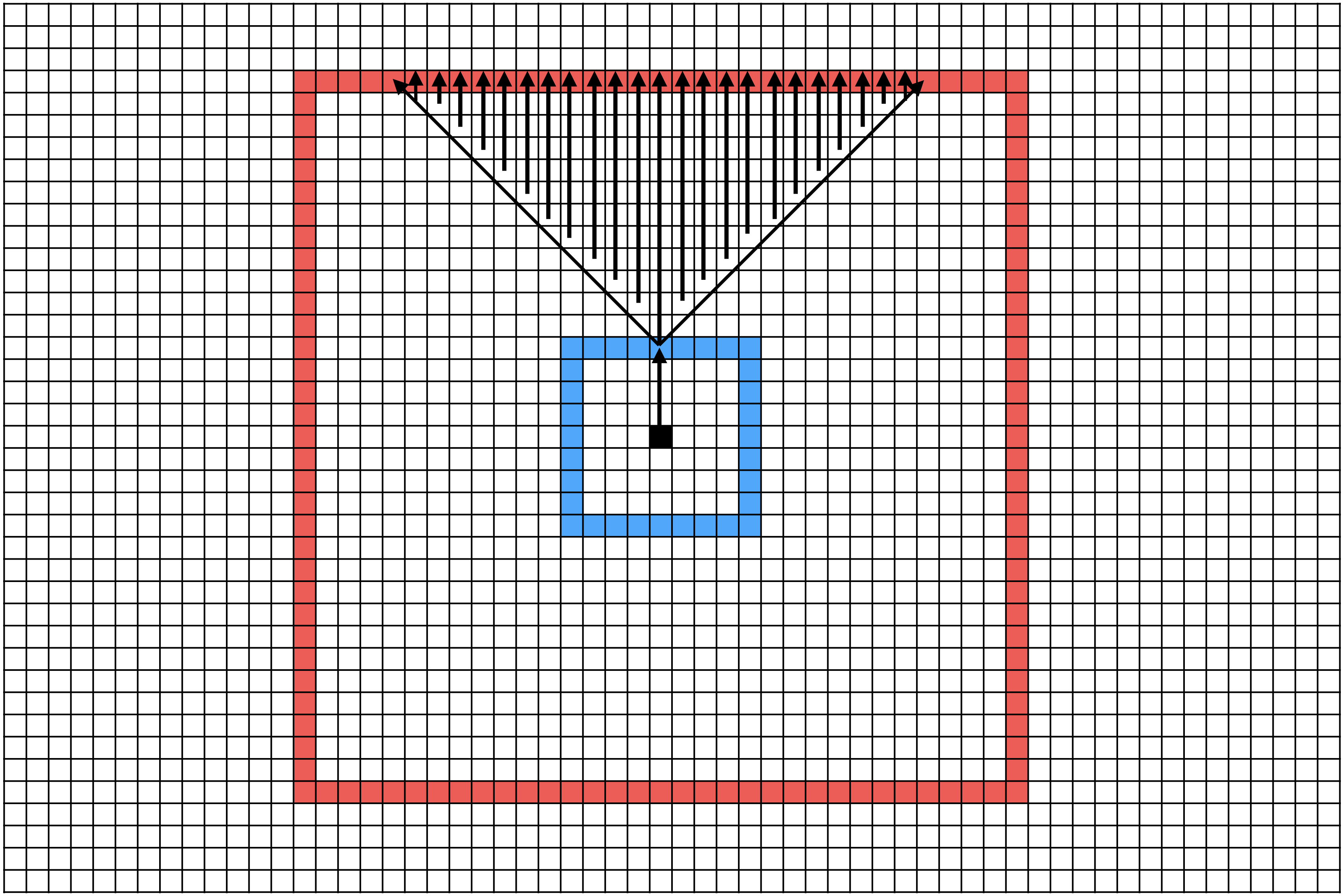


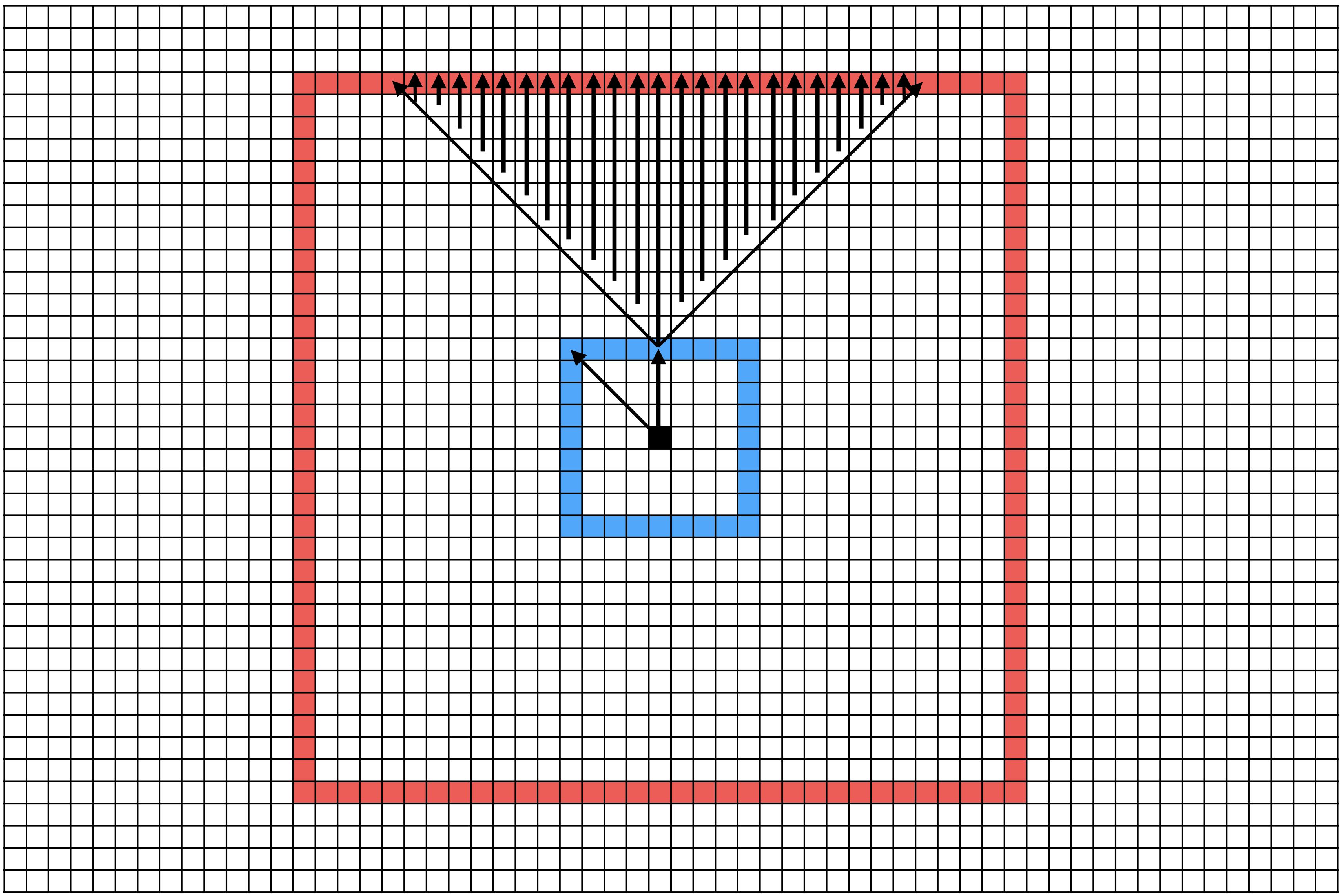


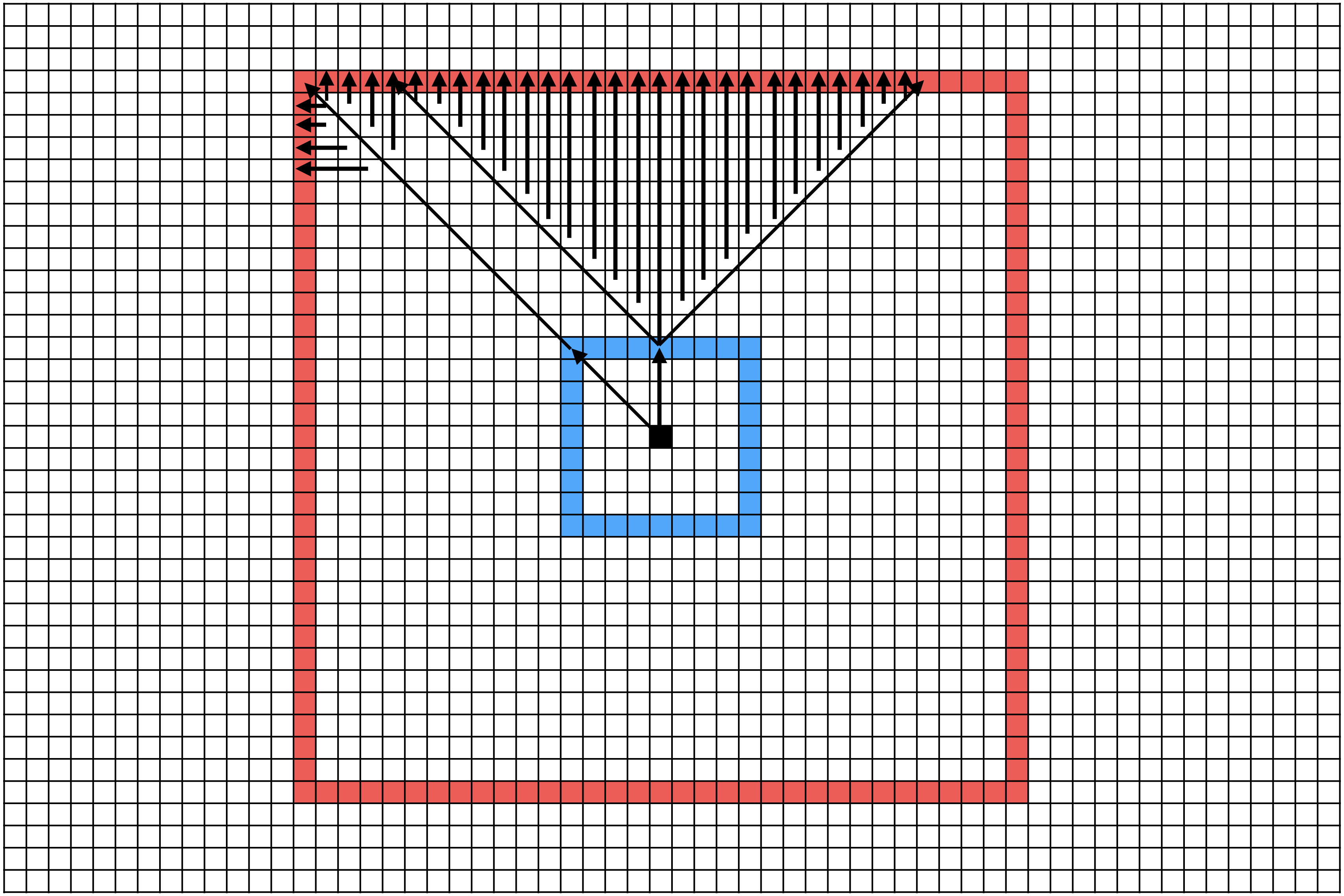


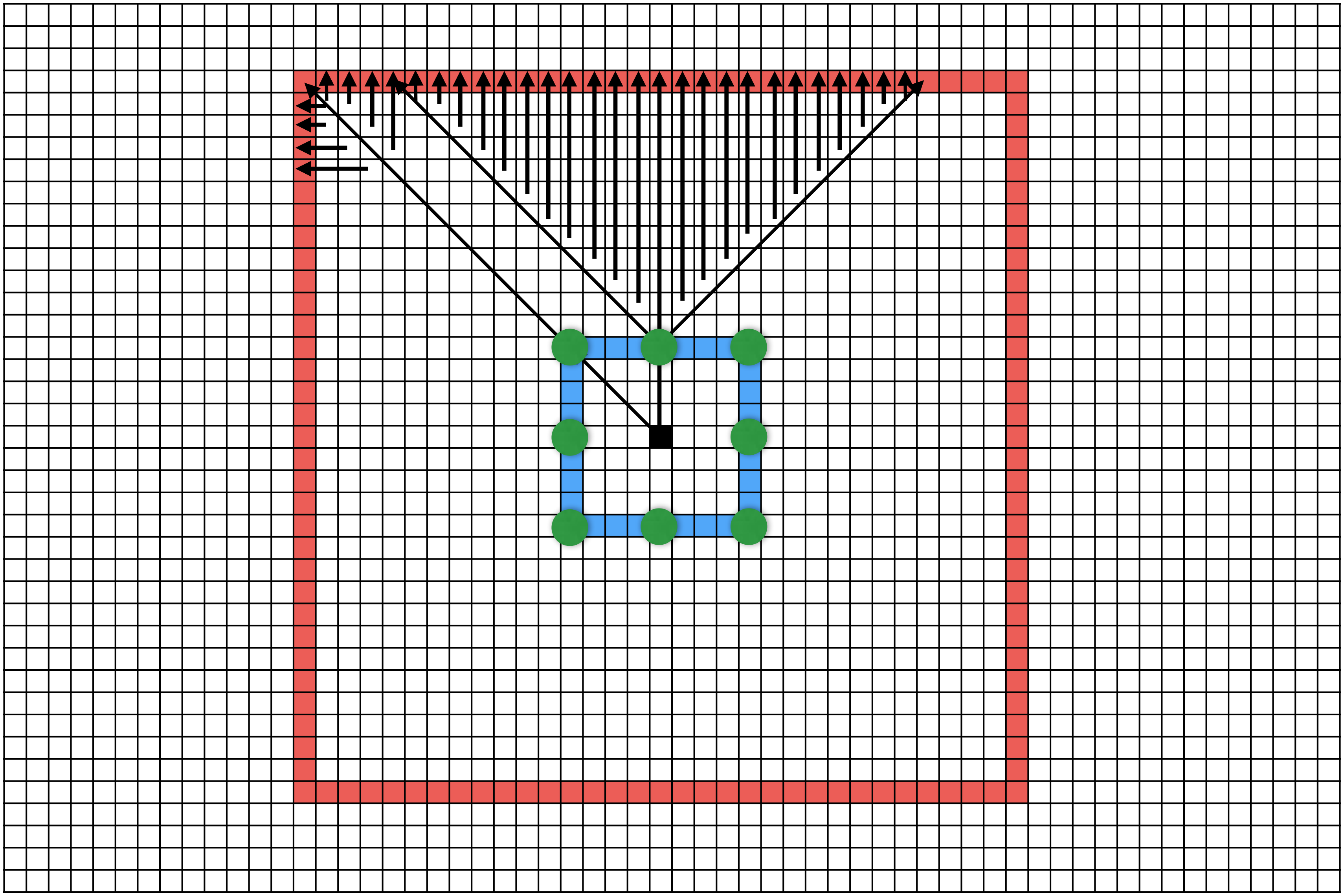












Cross-Fertilization

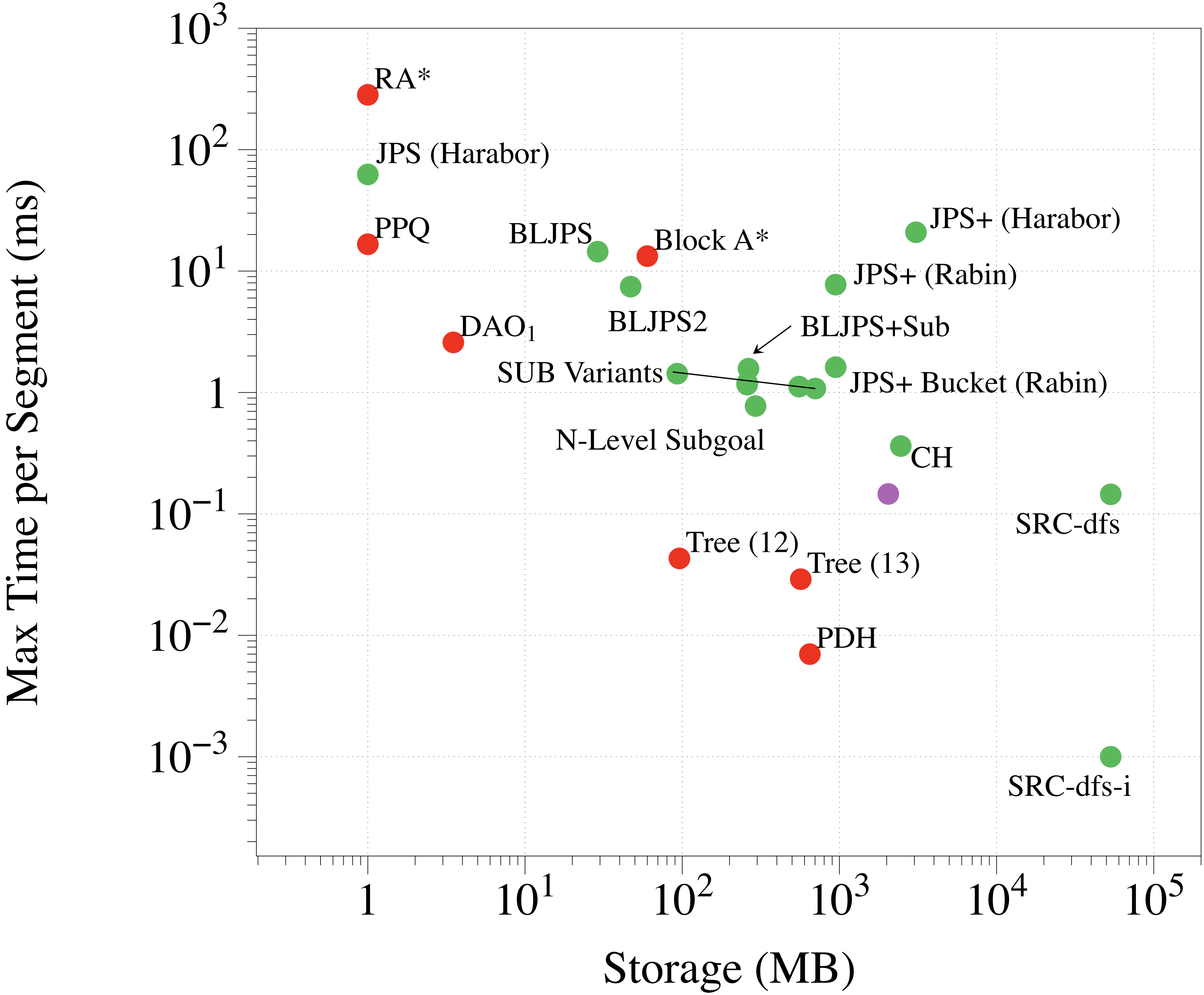
- Many road network ideas being used in grids
- Didn't originally know how to apply them properly
  - Contraction Hierarchies on Grid Graphs
    - Storandt, 2013
  - TRANSIT Routing on Video Game Maps
    - Antsfeld, Harabor, Kilby & Walsh, 2012

Further use of contractions?

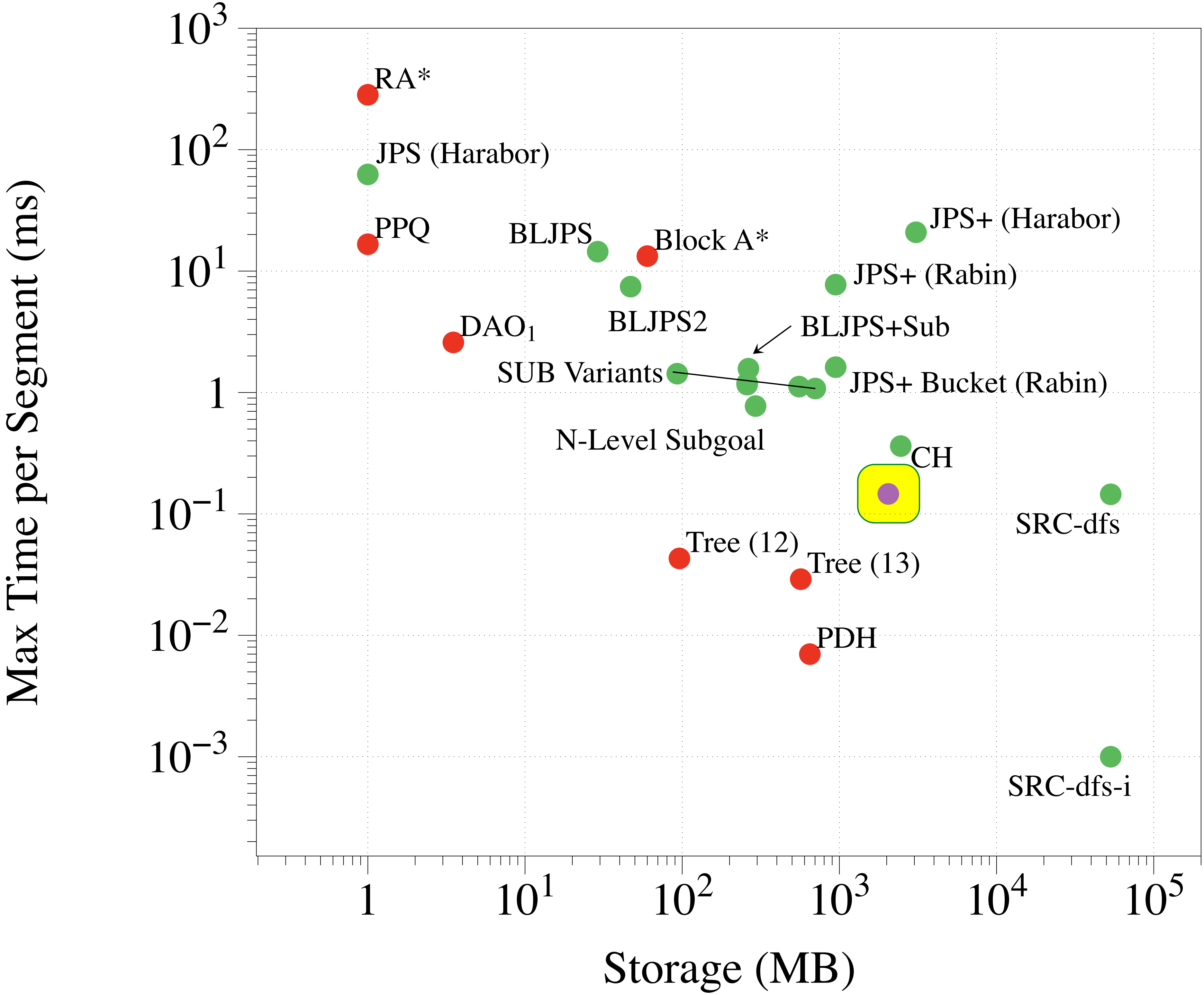
- Where else could we apply contractions?
    - Robotics?
    - 3D worlds?
    - Alternate representations? (polygons)

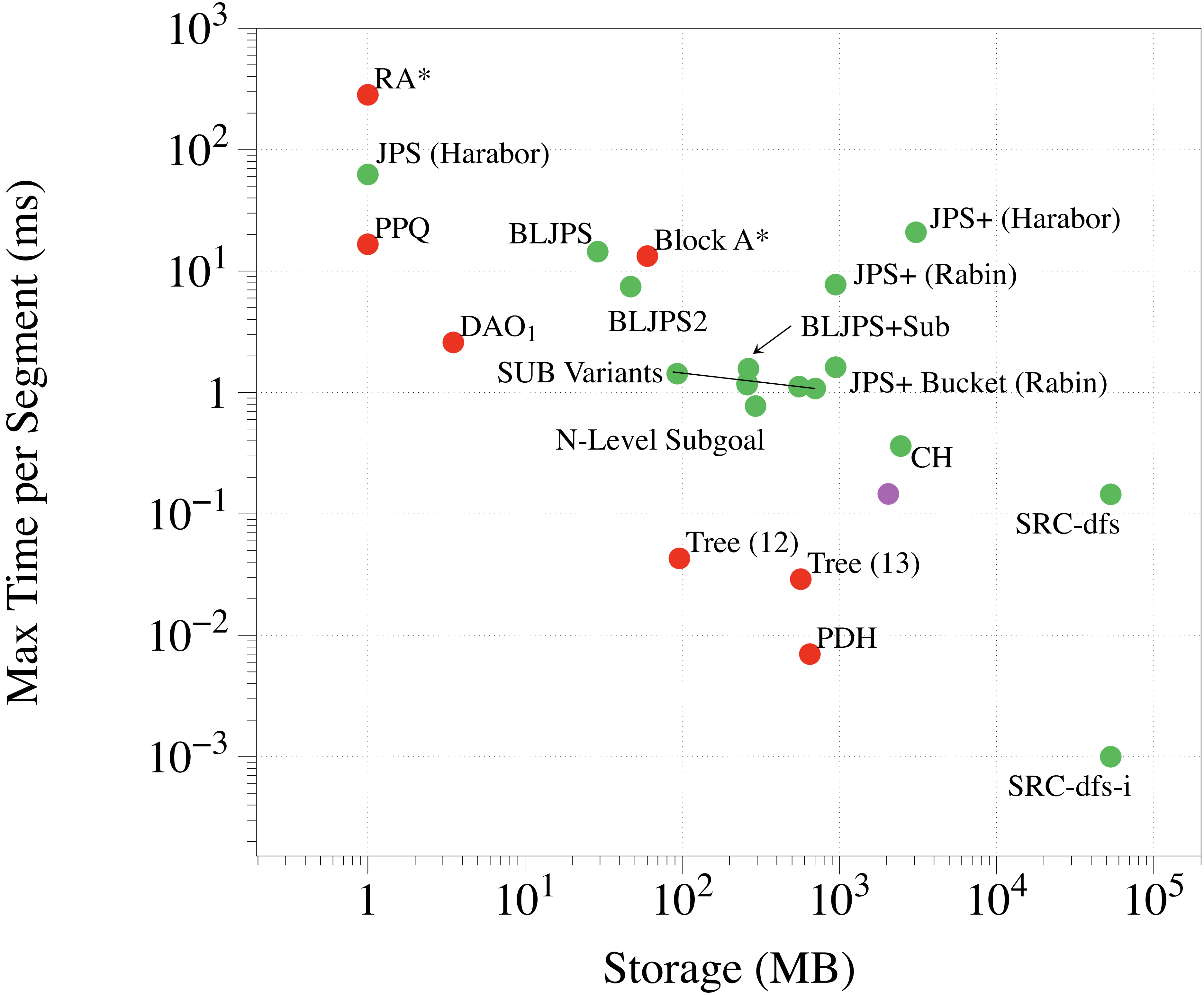
The Future

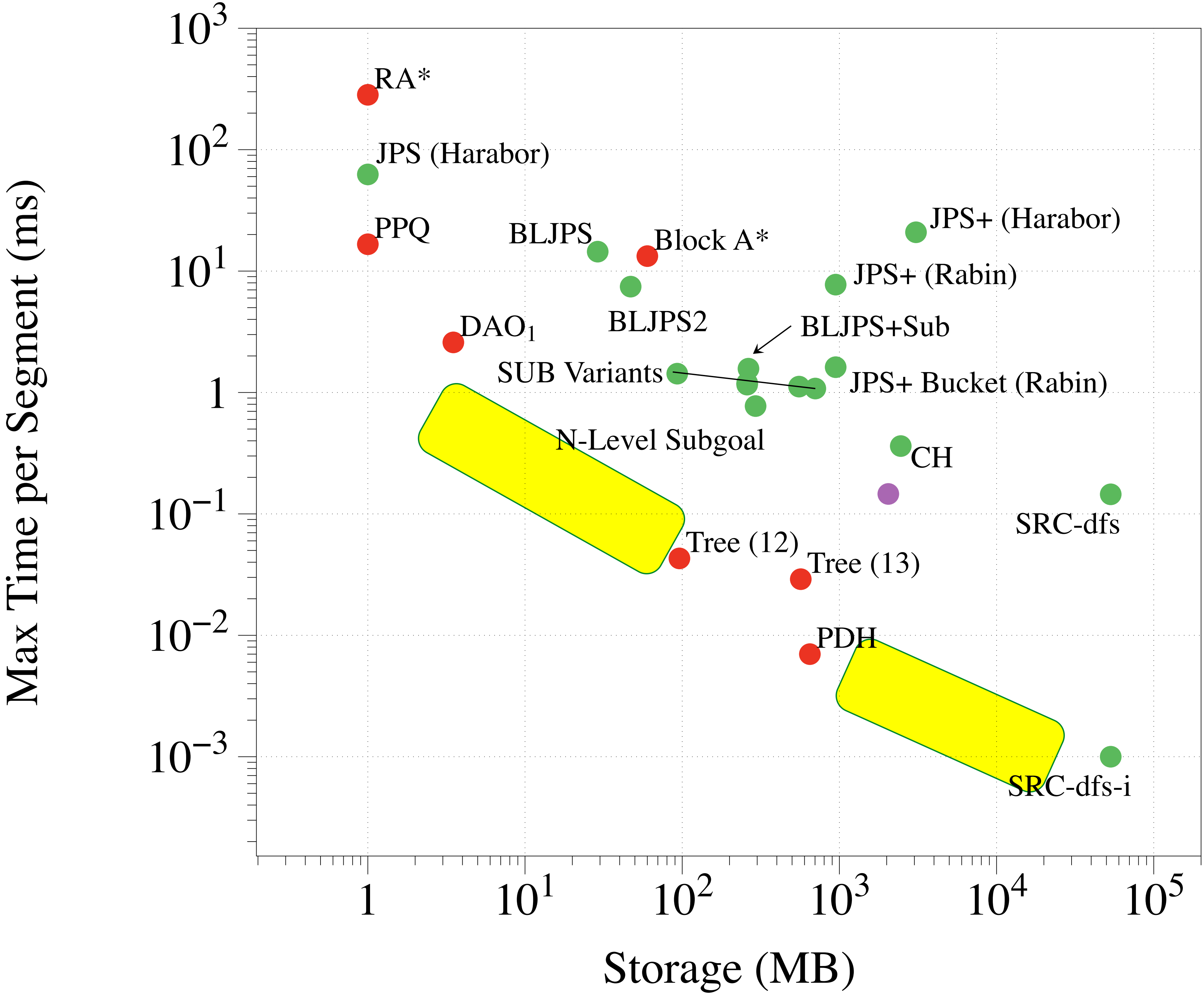
- Have we reached the limits of our performance?
- Where is there significant room for improvement?
- Other future directions?



Max Time per Segment (ms)
Storage (MB)
10^3
10^2
10^1
1
10^{-1}
10^{-2}
10^{-3}
1
10^1
10^2
10^3
10^4
10^5
RA*
JPS (Harabor)
PPQ
BLJPS
Block A*
JPS+ (Harabor)
JPS+ (Rabin)
BLJPS2
DAO_1
BLJPS+Sub
SUB Variants
JPS+ Bucket (Rabin)
N-Level Subgoal
CH
SRC-dfs
Tree (12)
Tree (13)
PDH
SRC-dfs-i


Max Time per Segment (ms)
Storage (MB)
10^3
10^2
10^1
1
10^{-1}
10^{-2}
10^{-3}
1
10^1
10^2
10^3
10^4
10^5
RA*
JPS (Harabor)
PPQ
BLJPS
Block A*
JPS+ (Harabor)
JPS+ (Rabin)
BLJPS2
DAO_1
BLJPS+Sub
SUB Variants
JPS+ Bucket (Rabin)
N-Level Subgoal
CH
SRC-dfs
Tree (12)
Tree (13)
PDH
SRC-dfs-i


Max Time per Segment (ms)
Storage (MB)
RA*
JPS (Harabor)
PPQ
BLJPS
Block A*
JPS+ (Harabor)
JPS+ (Rabin)
BLJPS2
DAO$_1$
BLJPS+Sub
SUB Variants
JPS+ Bucket (Rabin)
N-Level Subgoal
CH
SRC-dfs
Tree (12)
Tree (13)
PDH
SRC-dfs-i



Max Time per Segment (ms)
Storage (MB)
RA*
JPS (Harabor)
PPQ
BLJPS
Block A*
JPS+ (Harabor)
JPS+ (Rabin)
BLJPS2
DAO$_1$
BLJPS+Sub
SUB Variants
JPS+ Bucket (Rabin)
N-Level Subgoal
CH
SRC-dfs
Tree (12)
Tree (13)
PDH
SRC-dfs-i

Is this the real problem?

- Dynamic world
  - Costs change, connectivity changes
- Cell costs

Open Question

- Are we building insight?

Open Question

- Are we building insight?
  - The benchmarks are helping drive research

Open Question

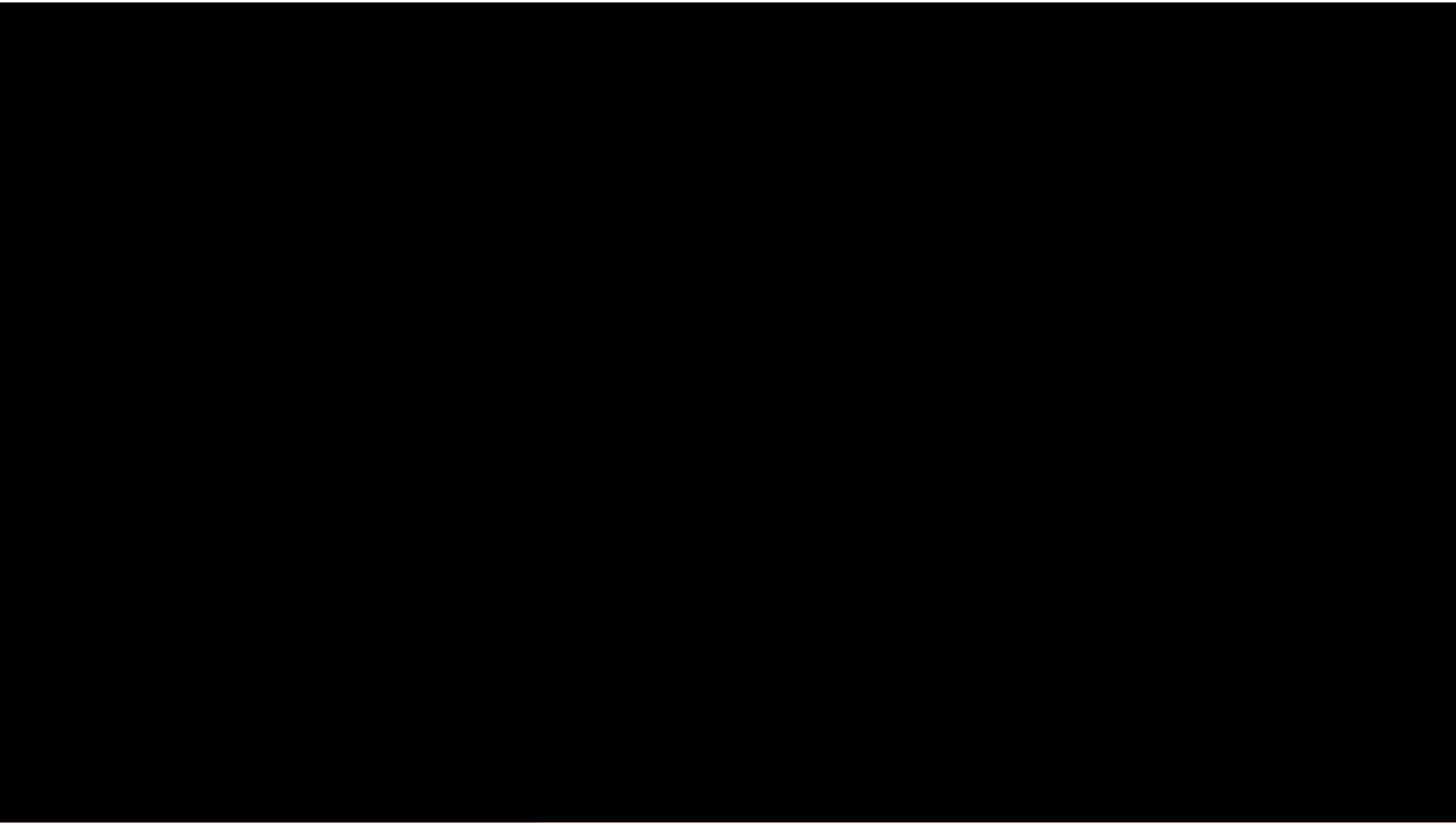
- Are we building insight?
  - The benchmarks are helping drive research
  - Work is leading to new theoretical understanding

Open Question

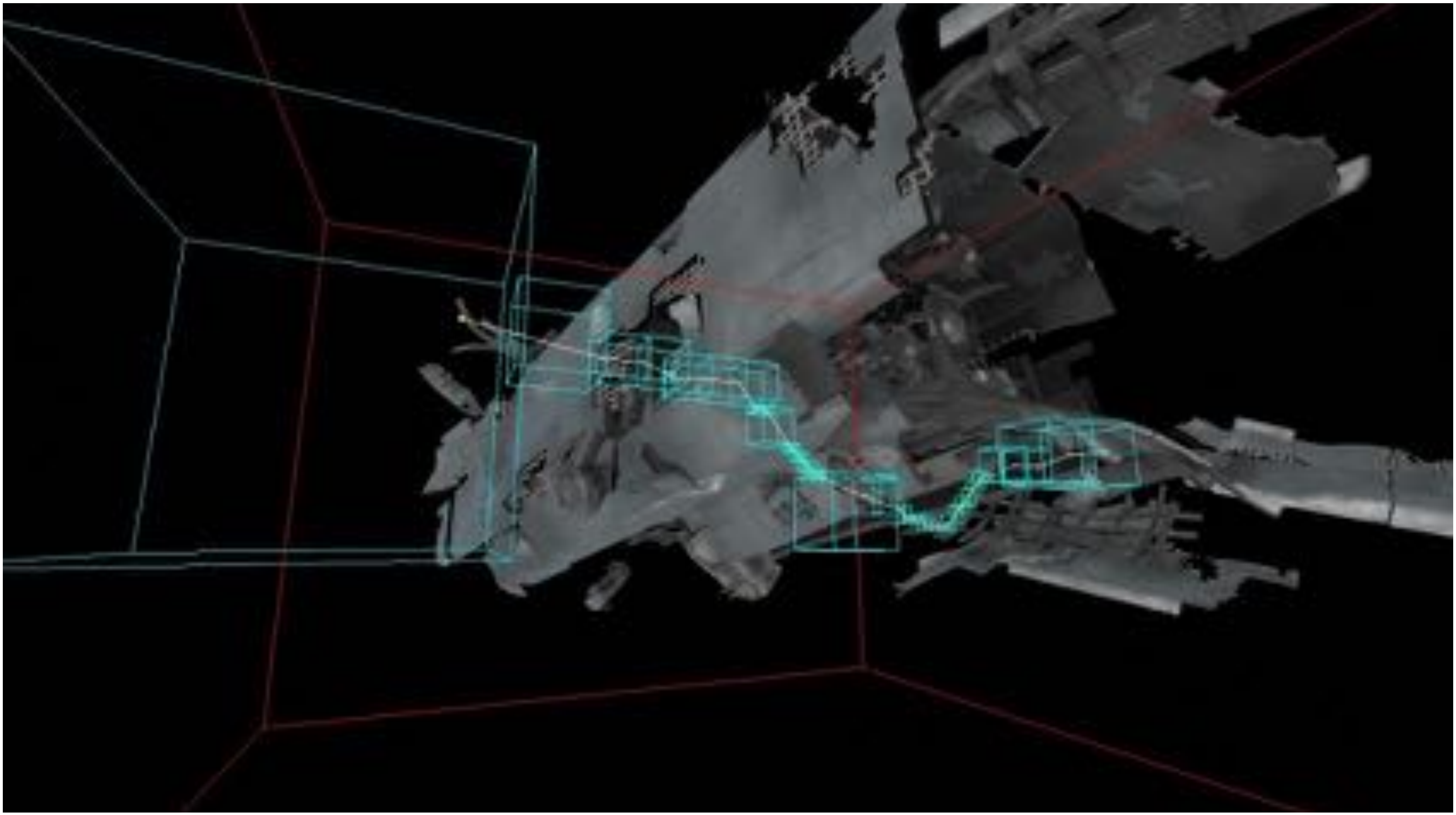
- Are we building insight?
  - The benchmarks are helping drive research
  - Work is leading to new theoretical understanding

- Thanks to SoCS chairs for the chance to contribute to the understanding

One more thing!



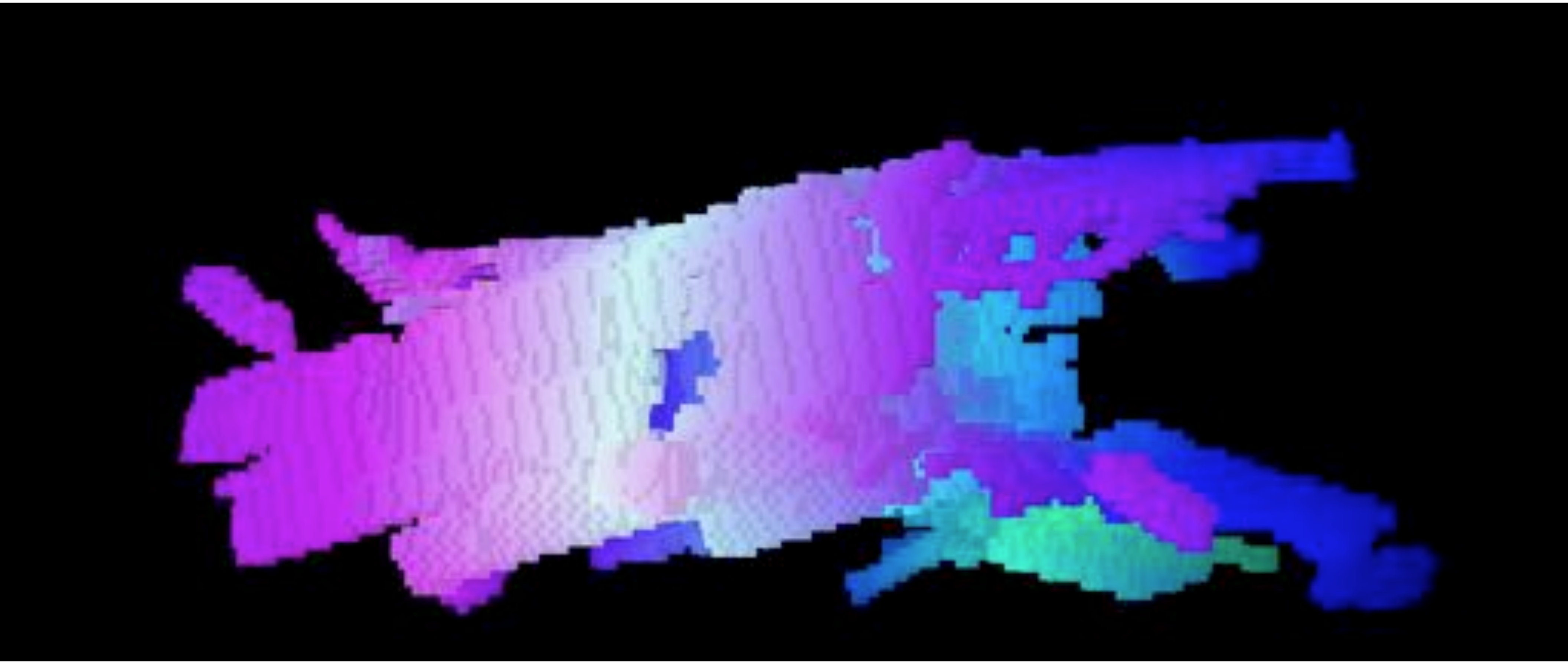
Warframe - Digital Extremes

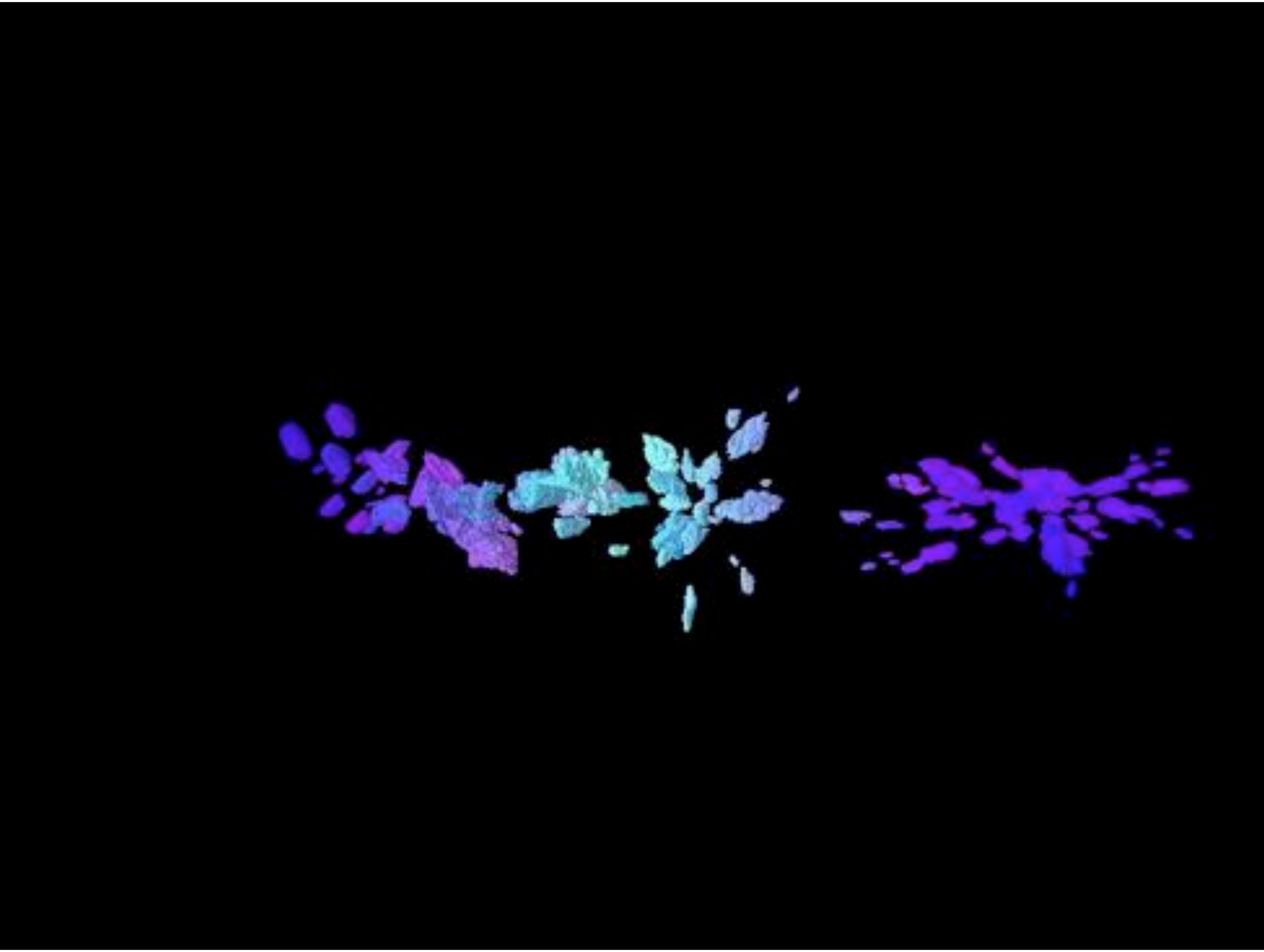


UNIVERSITY of DENVER
DANIEL FELIX RITCHIE SCHOOL OF ENGINEERING & COMPUTER SCIENCE

New repository coming

- 3D voxel world
- Challenging new path planning domain
- Thanks to Digital Extremes and Daniel Brewer!





Thanks

- Students/Collaborators
  - Renee Jansen
  - Sally Li
  - Michael Buro
  - Vadim Bulitko
  - Robert Geisberger
- http://movingai.com/GPPC/
- Games Industry
  - BioWare
  - Digital Extremes
    - Daniel Brewer
  - Troy Humphreys
- GPPC Participants
- SoCS Organizers