Multi-agent Pathfinding with Radius Abstraction Deletion Solving

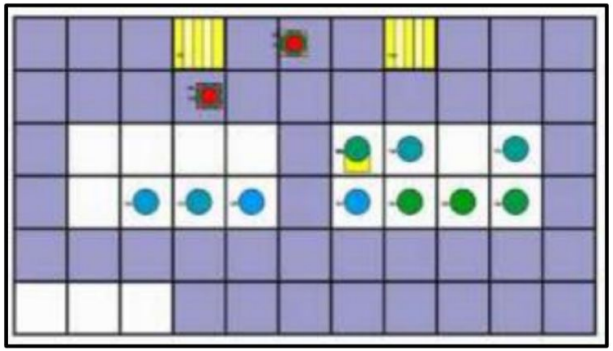
Sebastian Wilharm, Chung-Fan Tsai

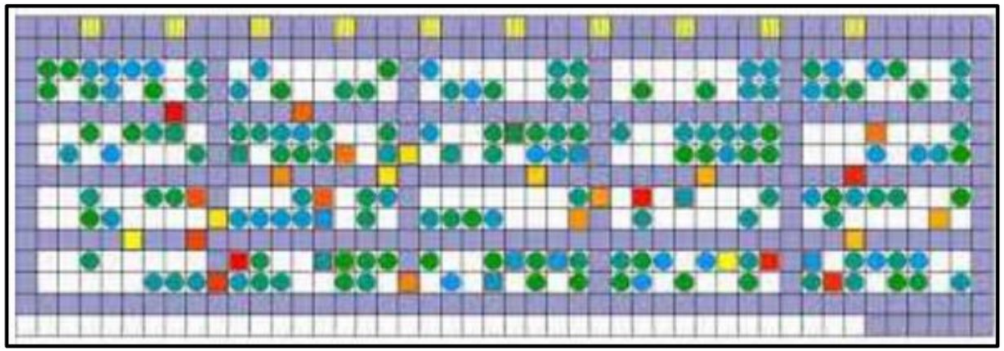
University of Potsdam
MAPF Final Project
28.03.2022

Table of Contents

MAPF

- Planning for multiple agents without collision
- **Plan** before **execute**

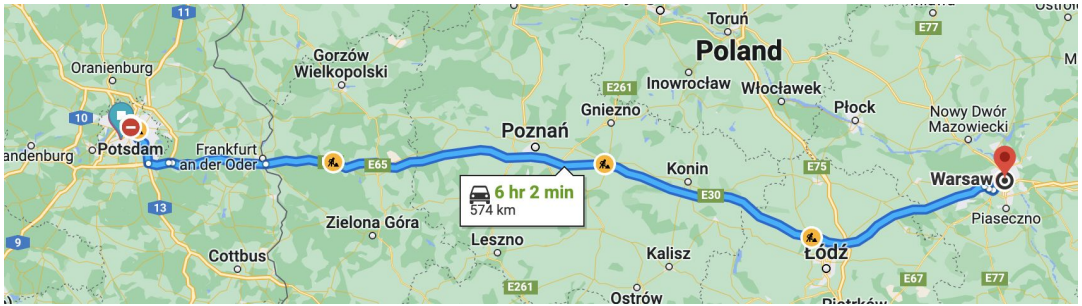




(Stern et al., 2019)

MAPF Problem - Dynamic Environments

- No time to or should not plan before act
- Interleaving **planning** and **execution**



Map Abstraction

- Extracting higher-level Information

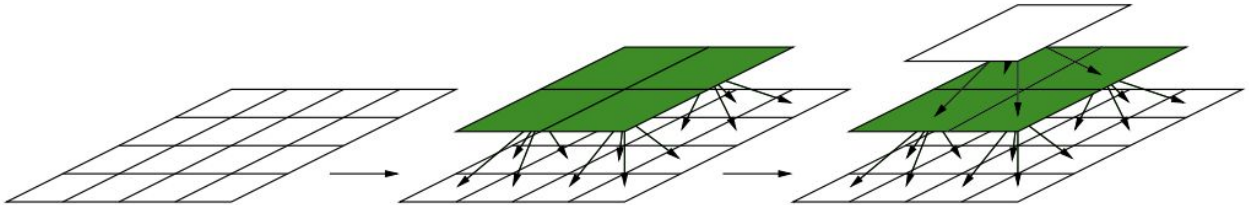


Figure 1: Abstracting tiles into a hierarchy

(Sturtevant & Buro, 2005)

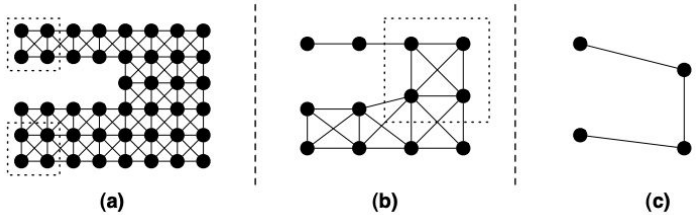
Abstraction Definition

- From a graph G1 to a graph G2
- Nodes in G1 can be mapped to abstracted nodes from G2
- An edge is added between two abstract nodes when there is at least 1 edge between their subnodes
- Abstract recursively from level 0 until the highest hierarchy is reached

(Sturtevant & Jansen, 2007)

Abstraction Methods

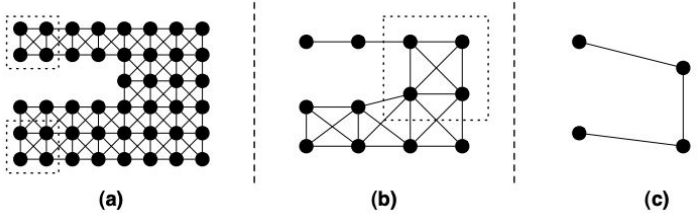
- Clique Abstraction (Sturtevant & Buro, 2005)



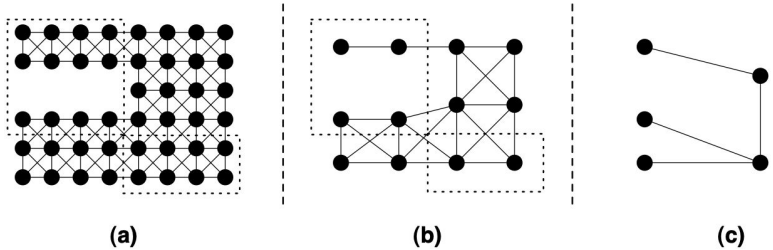
(Sturtevant & Jansen, 2007)

Abstraction Methods

- Clique Abstraction



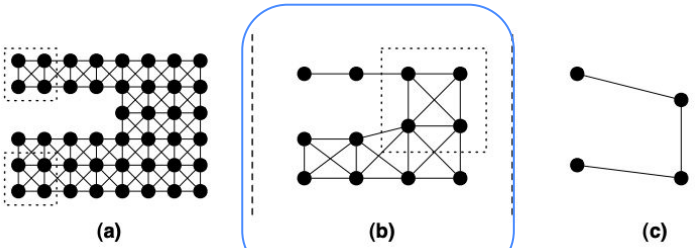
- Sector Abstraction



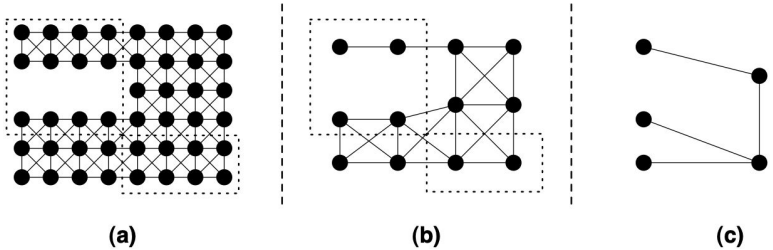
(Sturtevant & Jansen, 2007)

Abstraction Methods

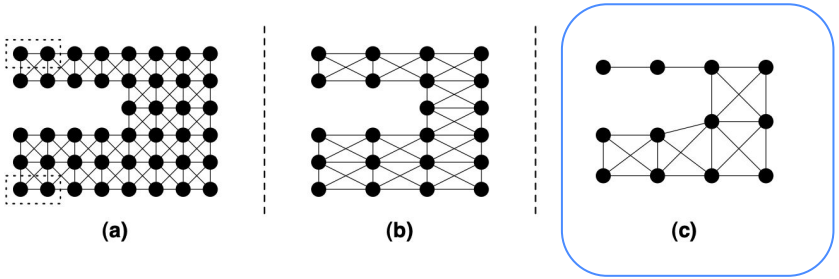
- Clique Abstraction



- Sector Abstraction



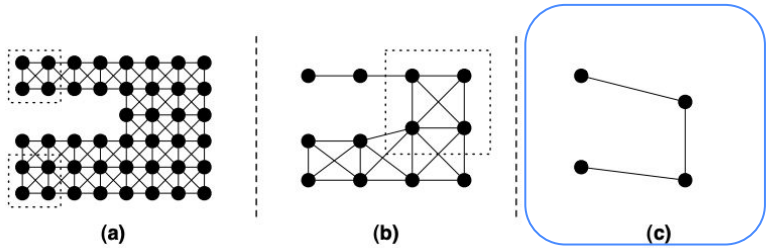
- Line Abstraction



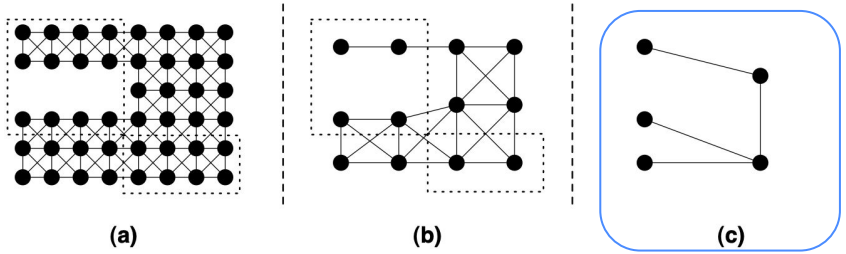
(Sturtevant & Jansen, 2007)

Abstraction Methods

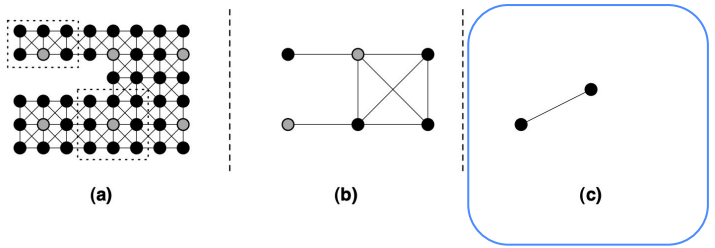
- Clique Abstraction



- Sector Abstraction



- Radius/Star Abstraction



(Sturtevant & Jansen, 2007)

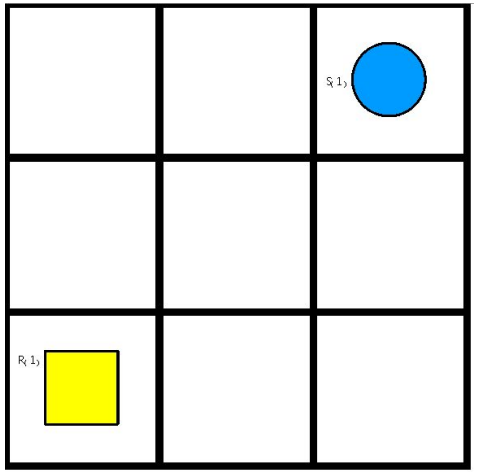
Radius Abstraction: Implementation

1. Initialize a list of abstract graphs with the original level 0 graph
2. First try abstracting with the number of center when each center is fully connected (each connects to 8 vertices)
3. Map vertices not yet merged to a center it's connected to
4. Add edge if some nodes under 2 centers are connected
5. If the amount of center cannot abstract the whole graph, increment the amount of center by one and abstract again for this level
6. Once the least number of center needed is found, and abstract graph is generated
7. Add the new graph to the end of the list and keep abstracting for higher level until no abstraction can be made

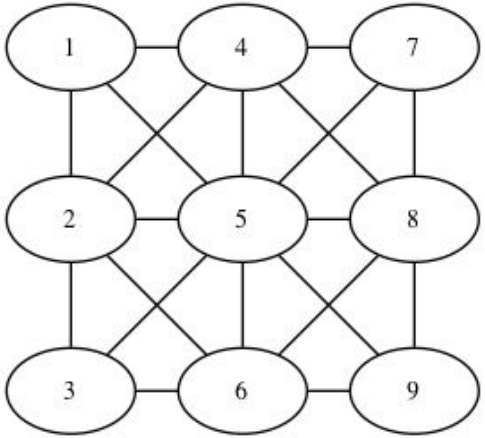
Radius Abstraction: N001.lp

Step 1. **Grid** to **graph**
Step 2. Recursively perform **radius abstraction**

grid

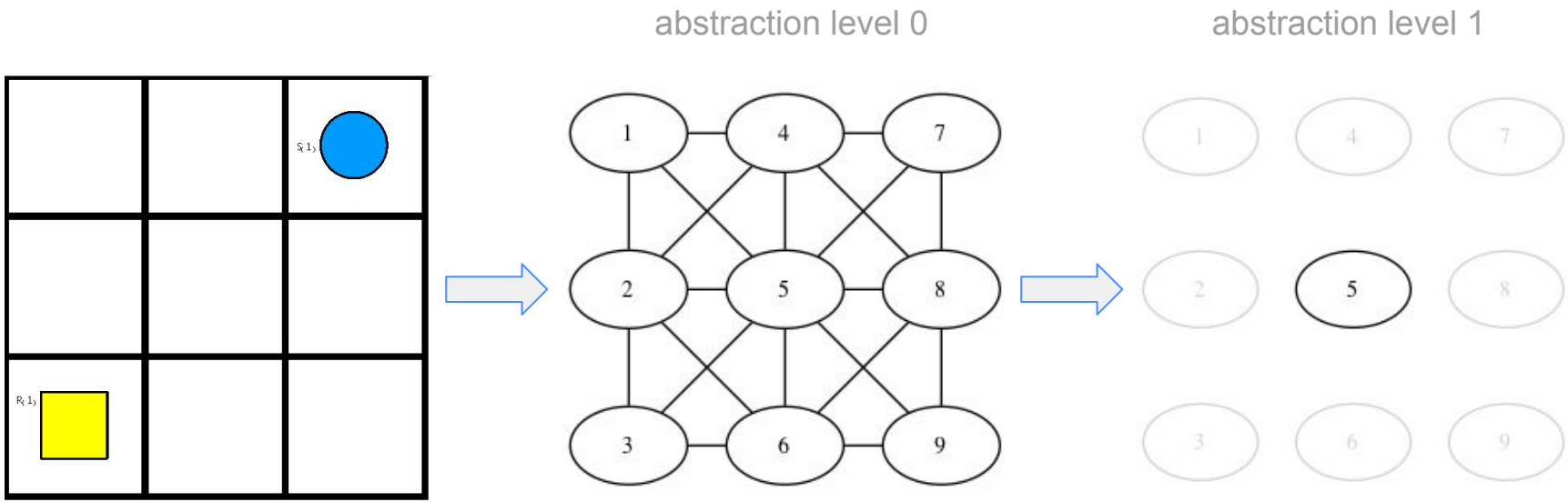


graph

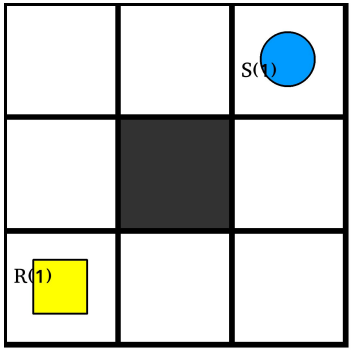


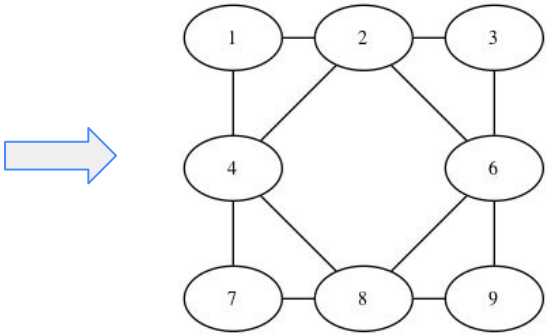
Radius Abstraction: N001.lp

Step 1. **Grid** to **graph**
Step 2. Recursively perform **radius abstraction**

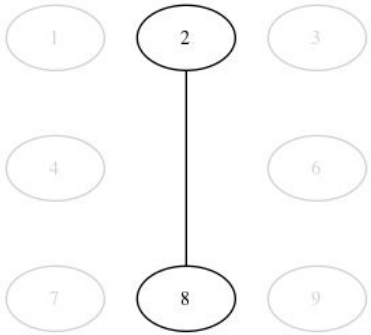


Radius Abstraction: 0.lp

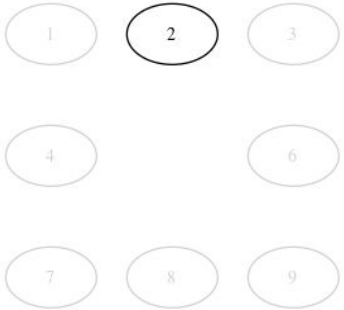




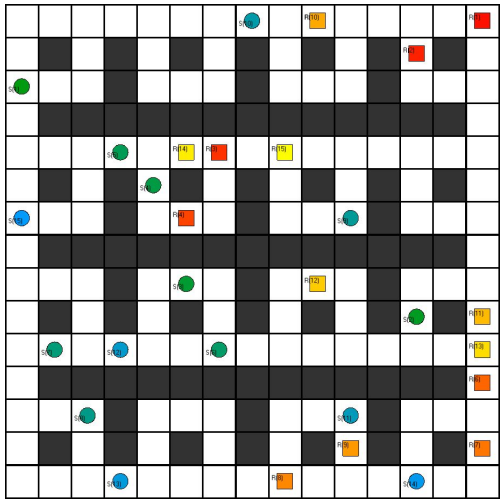
abstraction level 1

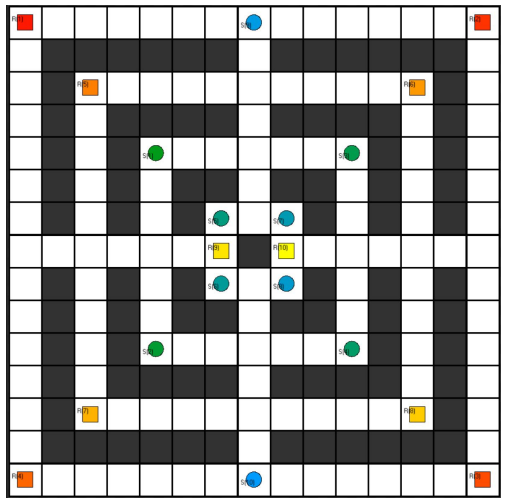


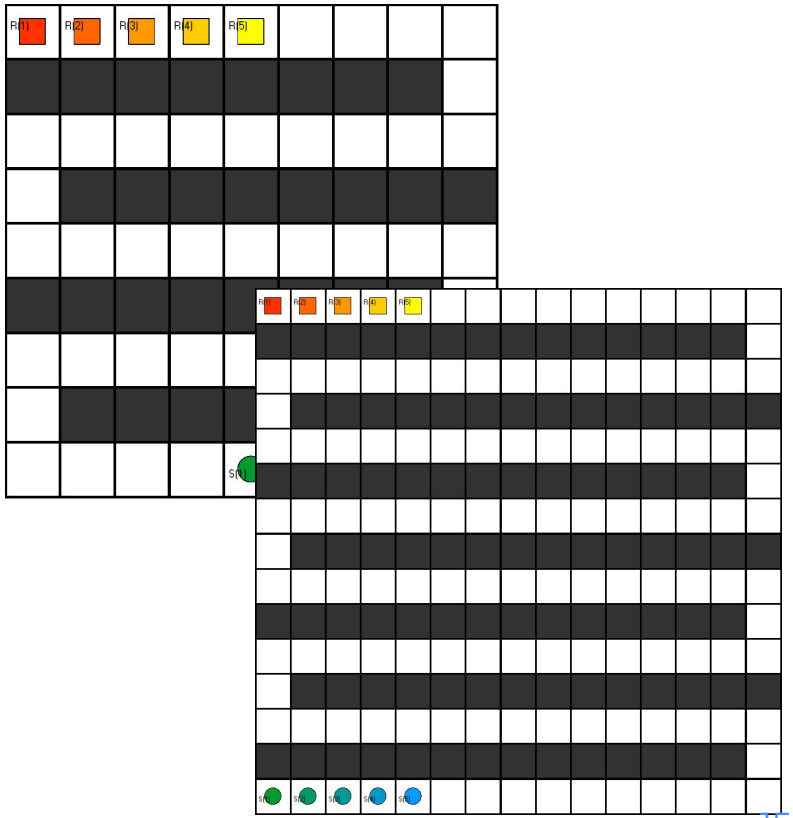
abstraction level 2



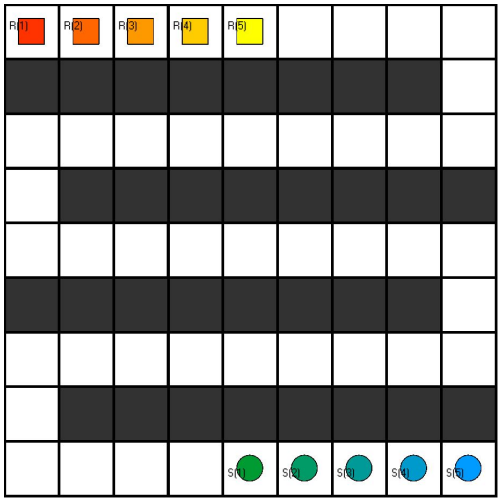
Radius Abstraction: Instances

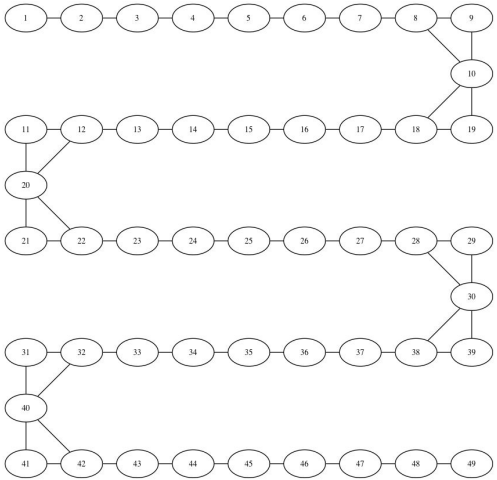


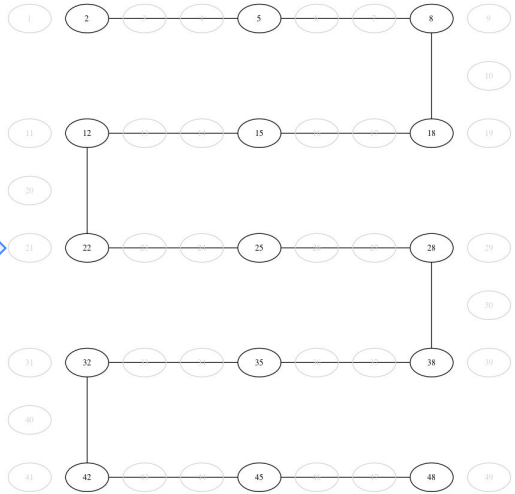




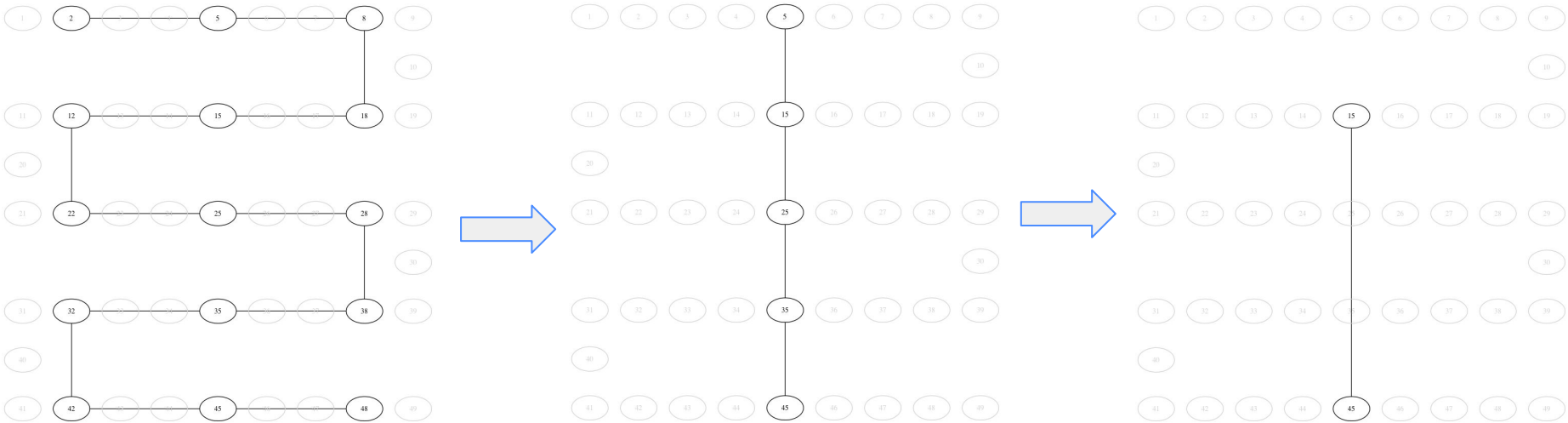
Radius Abstraction: Snake9x9





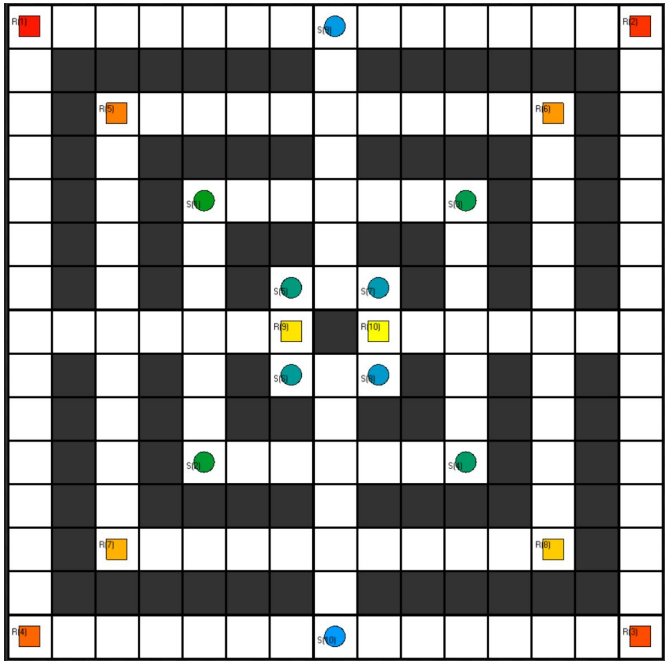


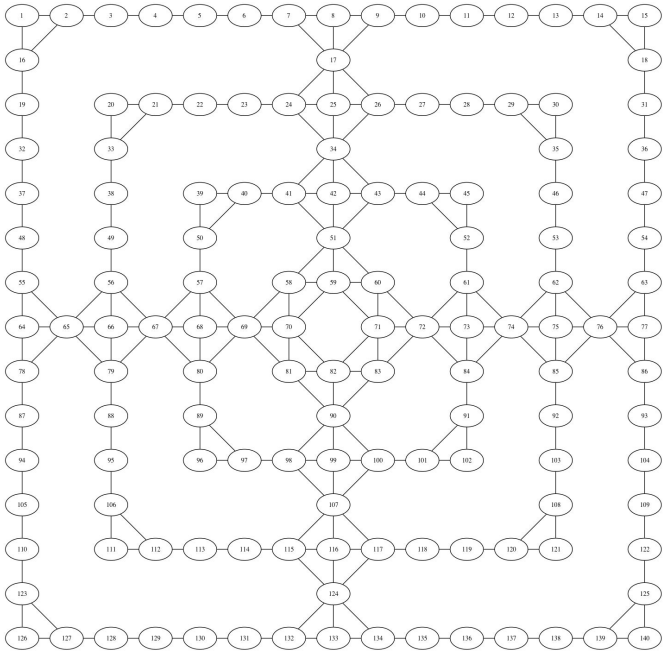
Radius Abstraction: Snake9x9



Radius Abstraction: Target

Step 1. **Grid** to **graph**

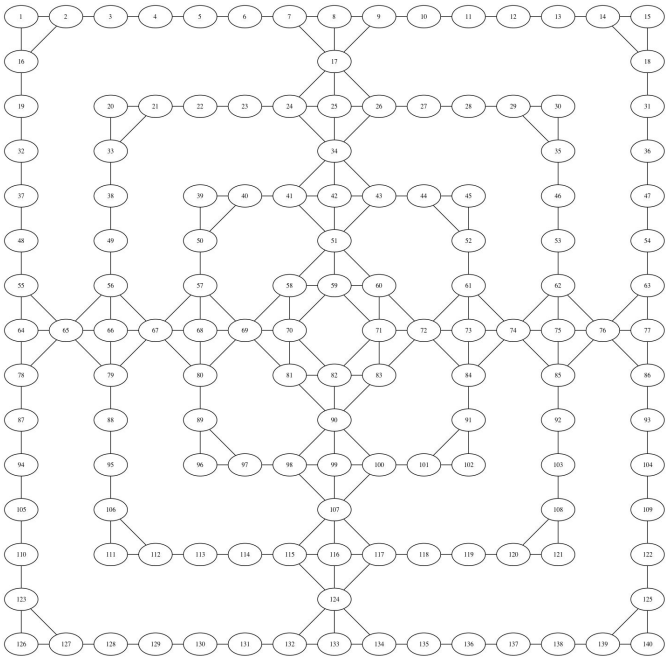




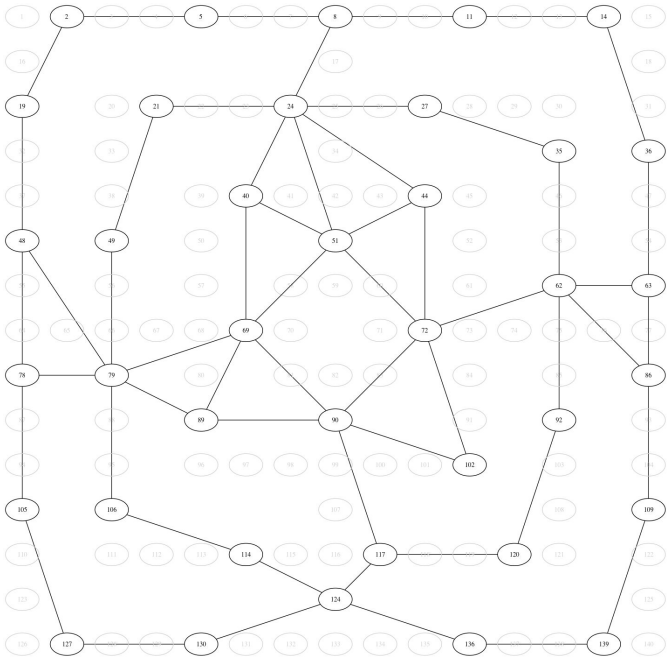
Radius Abstraction: Target

Step 2. Recursively perform **radius abstraction**

abstraction level 0



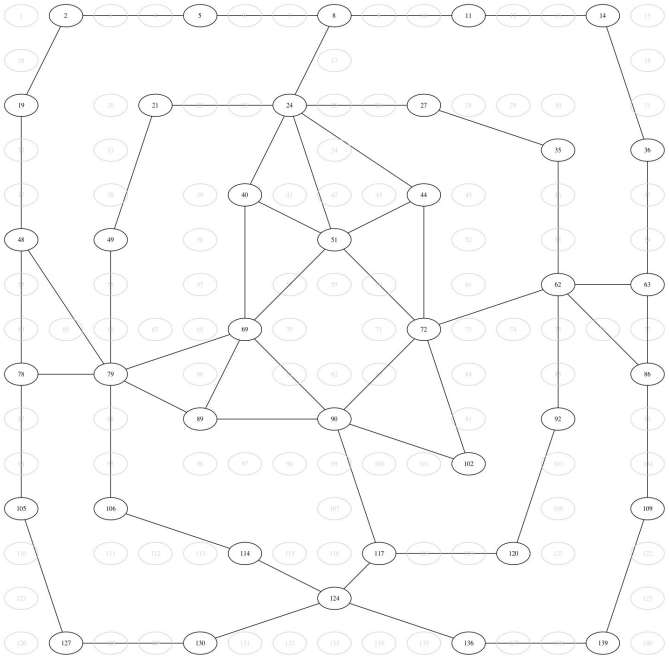
abstraction level 1



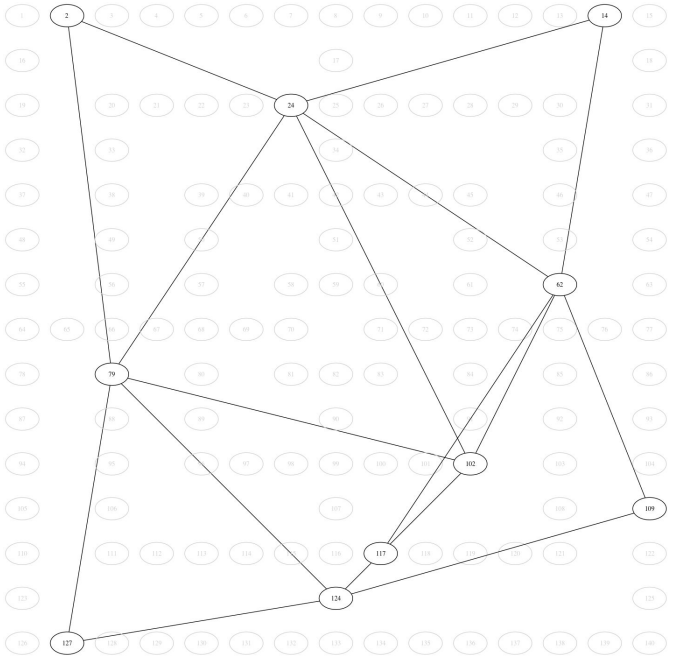
Radius Abstraction: Target

Step 2. Recursively perform **radius abstraction**

abstraction level 1



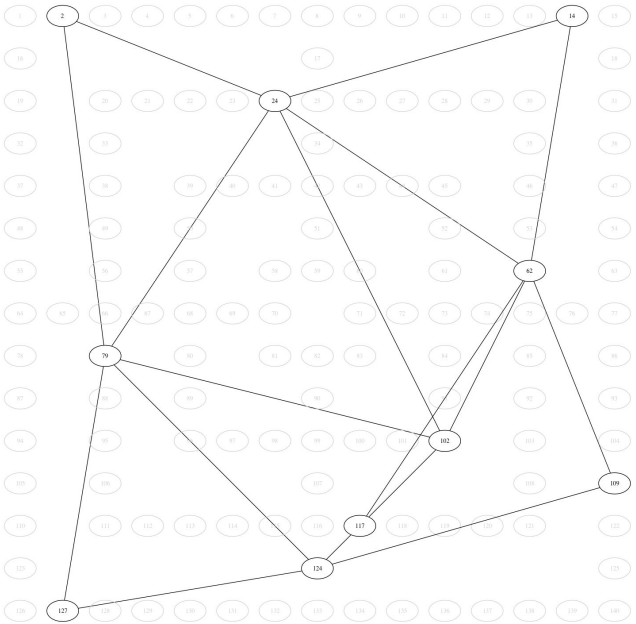
abstraction level 2



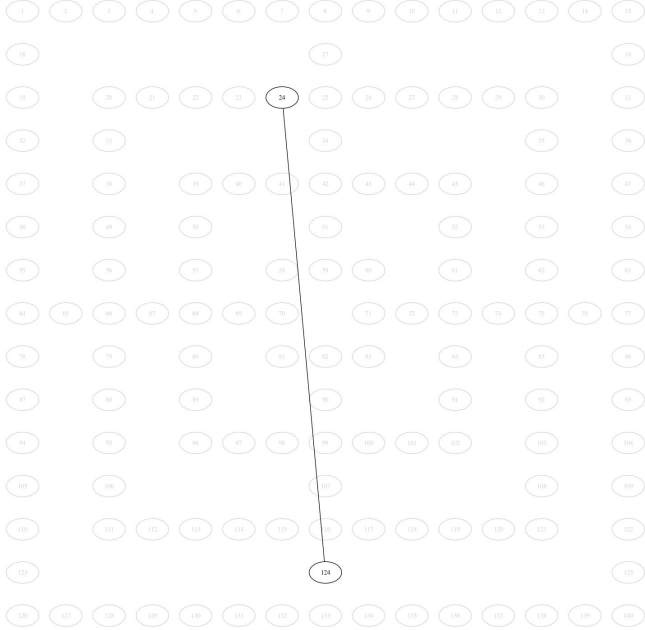
Radius Abstraction: Target

Step 2. Recursively perform **radius abstraction**

abstraction level 2



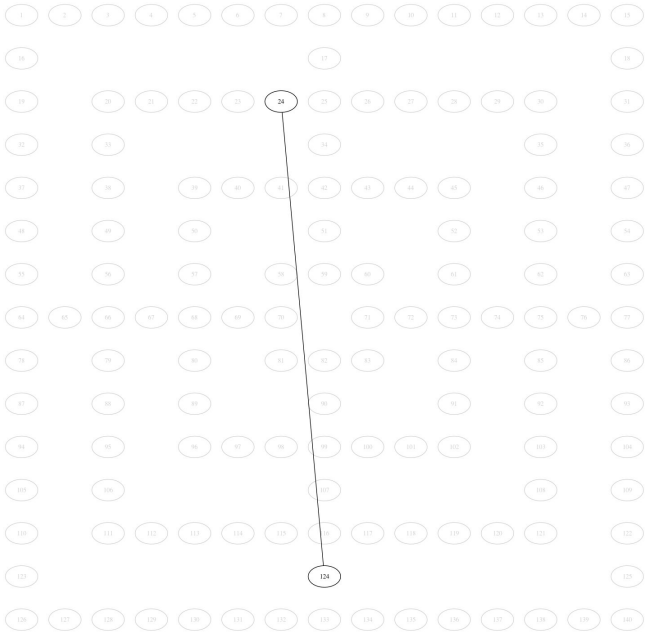
abstraction level 3



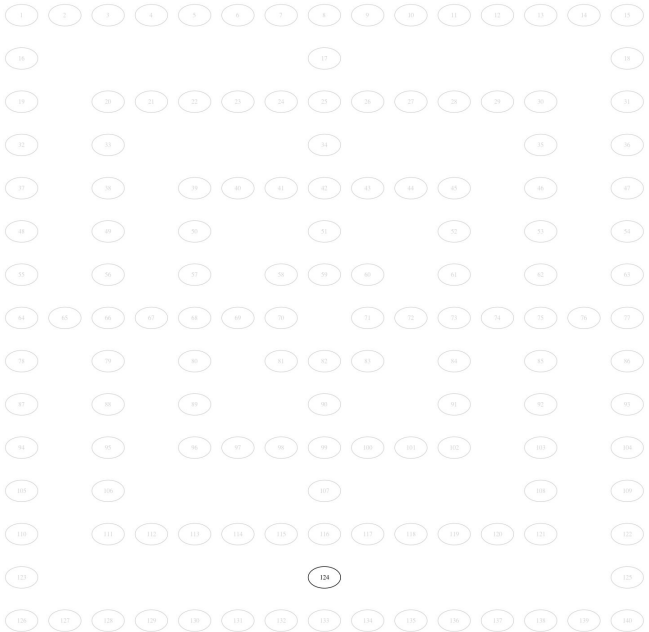
Radius Abstraction: Target

Step 2. Recursively perform **radius abstraction**

abstraction level 3

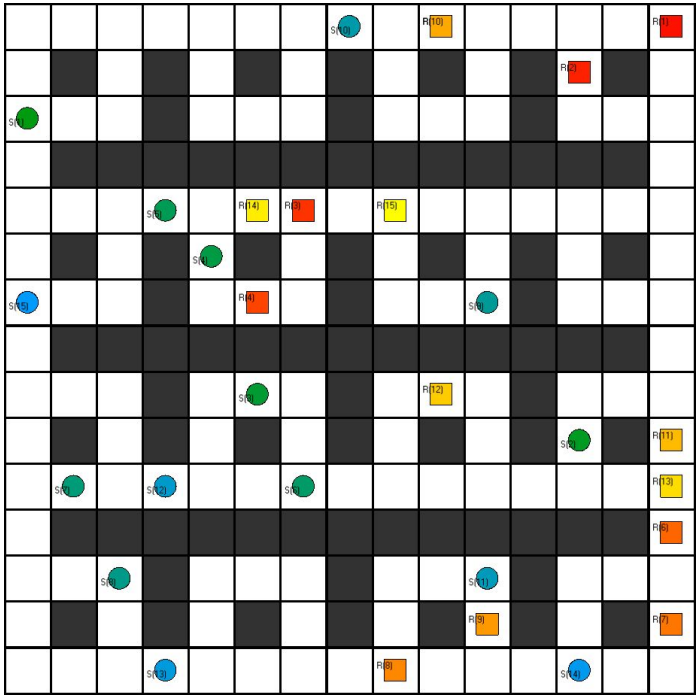


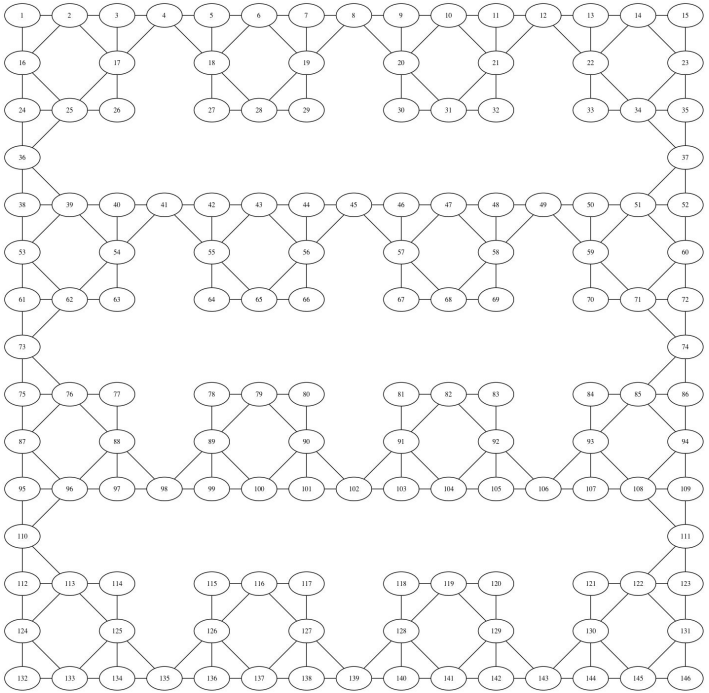
abstraction level 4



Radius Abstraction: Circles

Step 1. **Grid** to **graph**

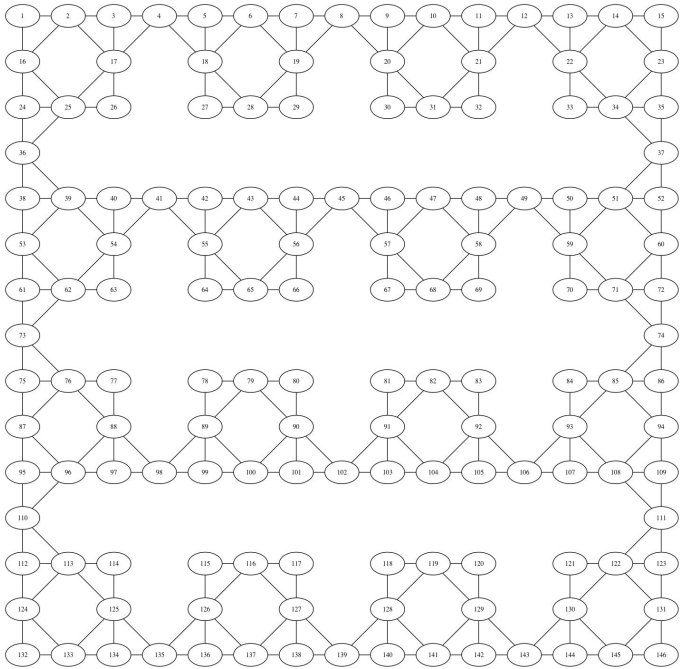




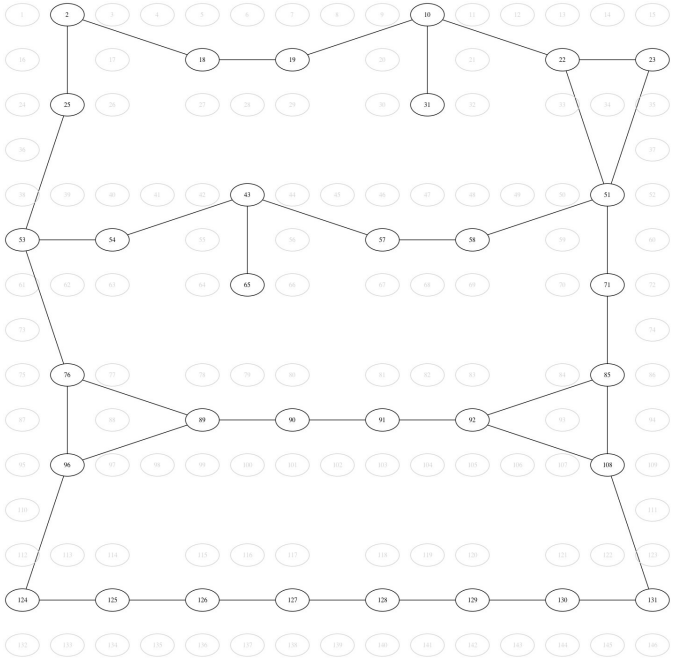
Radius Abstraction: Circles

Step 2. Recursively perform **radius abstraction**

abstraction level 0



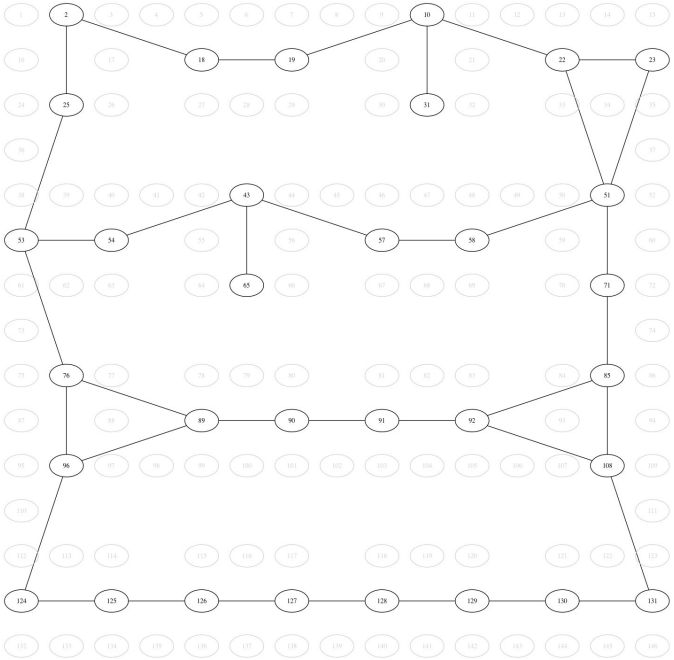
abstraction level 1



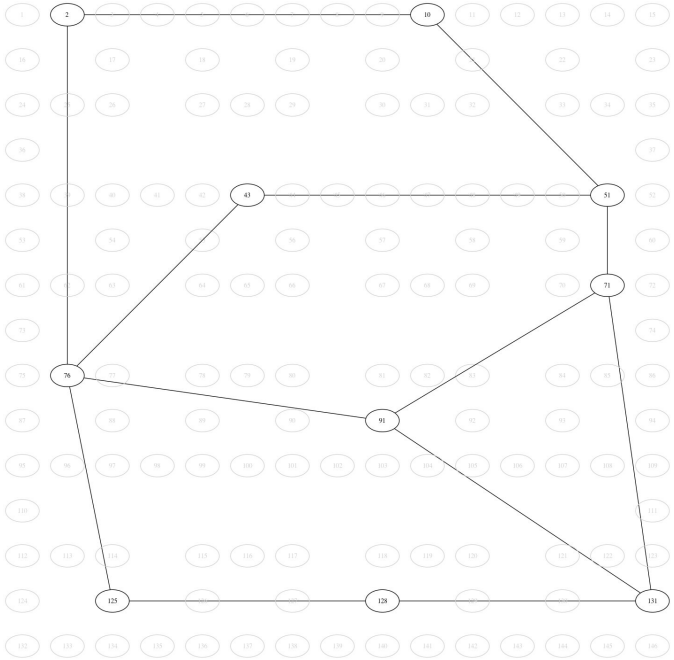
Radius Abstraction: Circles

Step 2. Recursively perform **radius abstraction**

abstraction level 1



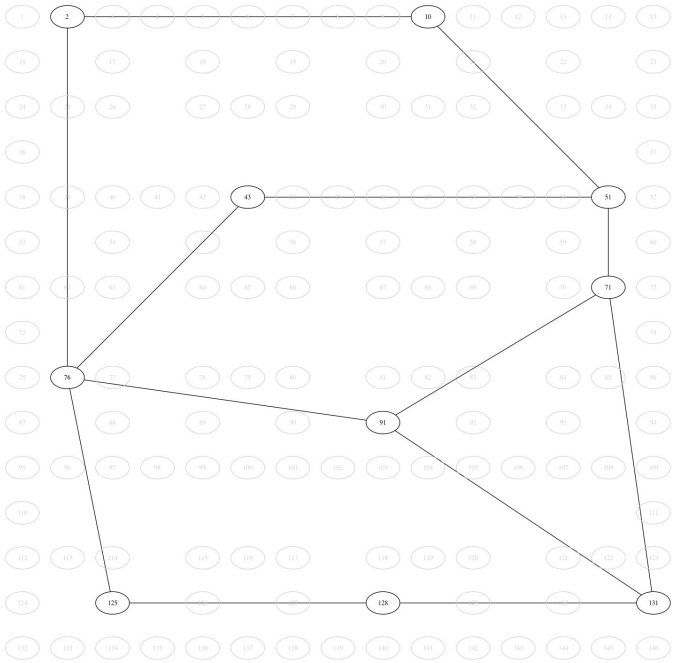
abstraction level 2



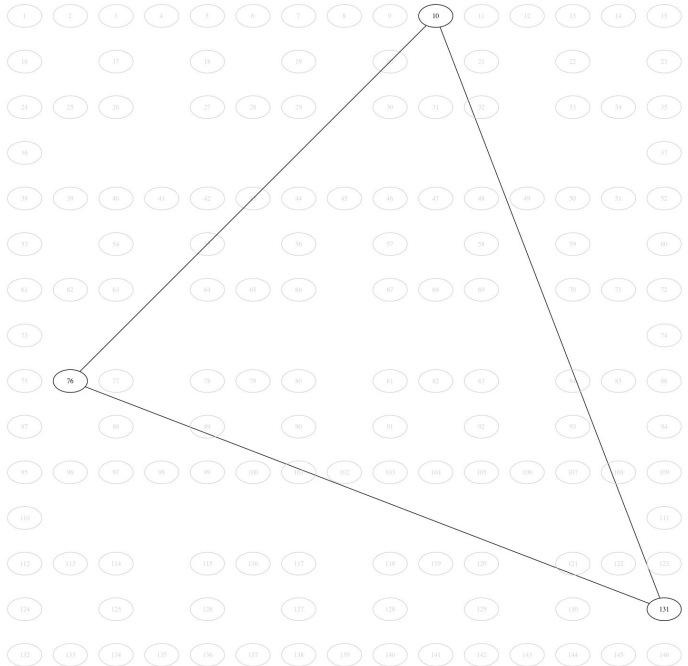
Radius Abstraction: Circles

Step 2. Recursively perform **radius abstraction**

abstraction level 2



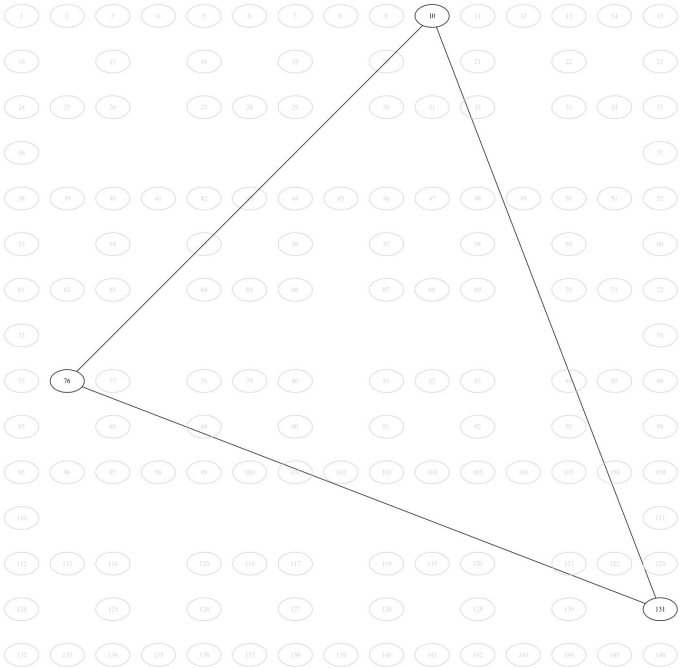
abstraction level 3



Radius Abstraction: Circles

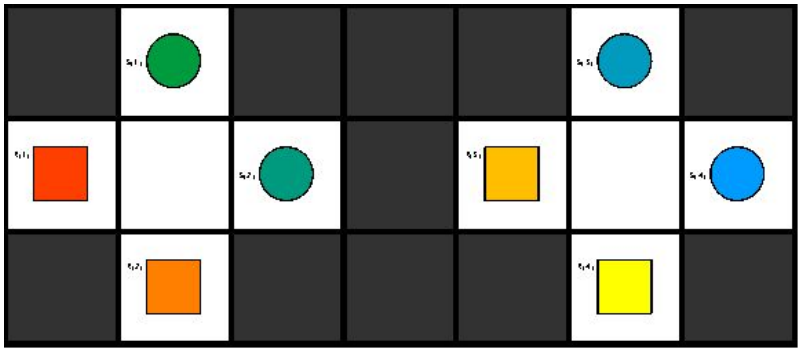
Step 2. Recursively perform **radius abstraction**

abstraction level 3

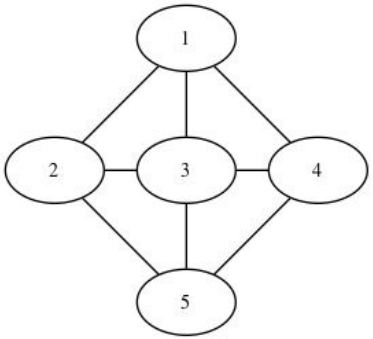


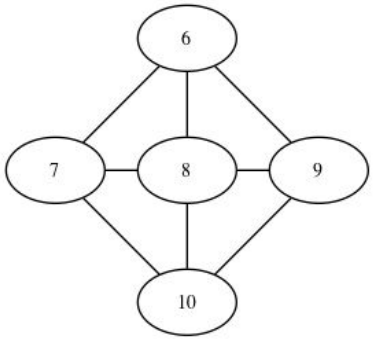
abstraction level 4

Radius Abstraction: Disjoint Subgraphs



abstraction level 0



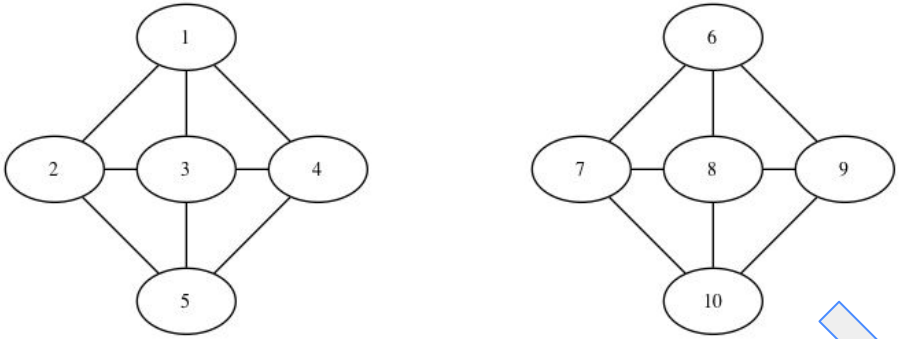


Radius Abstraction: Handling Disjoint Subgraphs

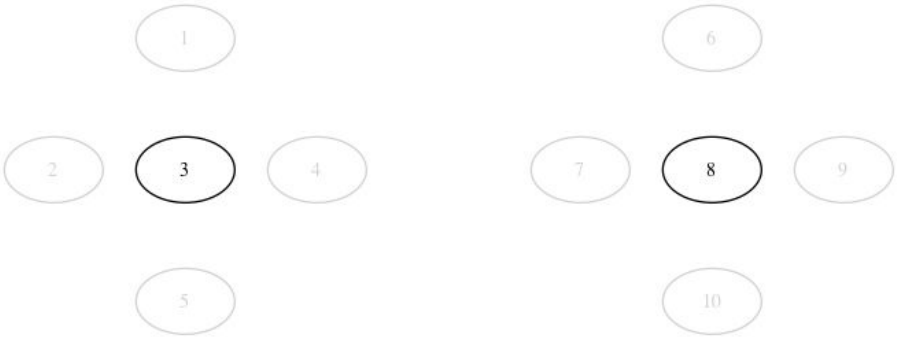
1. Initialize a list of abstract graphs with the original level 0 graph
2. First try abstracting with the number of center when each center is fully connected (each connects to 8 vertices)
3. Map vertices not yet merged to a center it's connected to
4. Add edge if some nodes under 2 centers are connected
5. If the amount of center cannot abstract the whole graph, increment the amount of center by one and abstract again for this level
6. Once the least number of center needed is found, abstract graph is generated, **and calculate a disjoint subgraph counter for each abstract graph to decide whether the highest abstract level has been reached**
7. Add the new graph to the end of the list and keep abstracting for higher level until no abstraction can be made, **when the number of vertex left is equal to the number of disjoint subgraphs**

Radius Abstraction: Disjoint Subgraphs

abstraction level 0



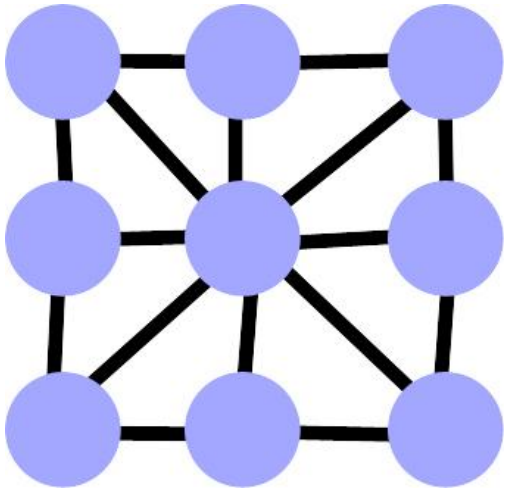
abstraction level 1



Solving

Why can the abstracted path be suboptimal?

- Maximum abstracted node is 9x9 grid, with center in the center
- Maximum steps within a node is 3 (including entry)
- From higher level, impossible to know if it takes 1, 2 or 3 steps to cross the node

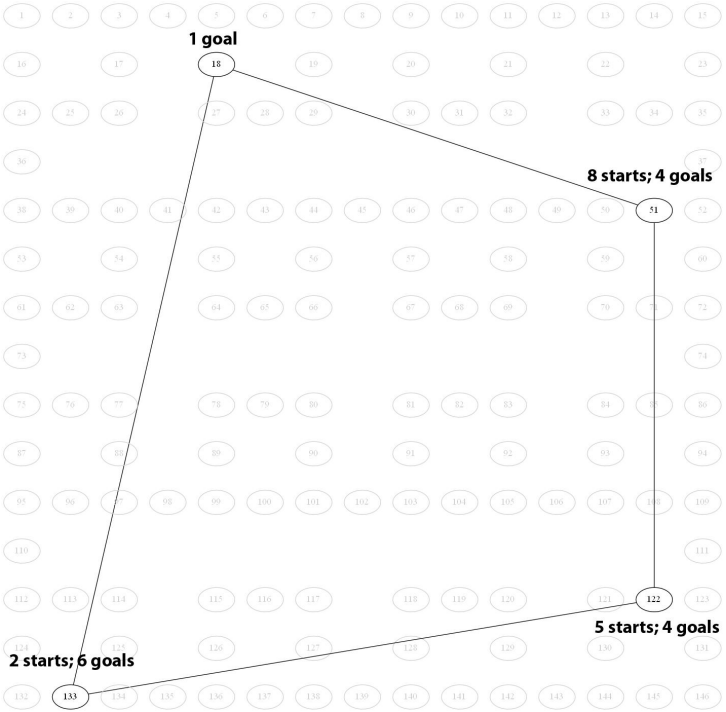


- Maximum makespan after expansion: 3n+2

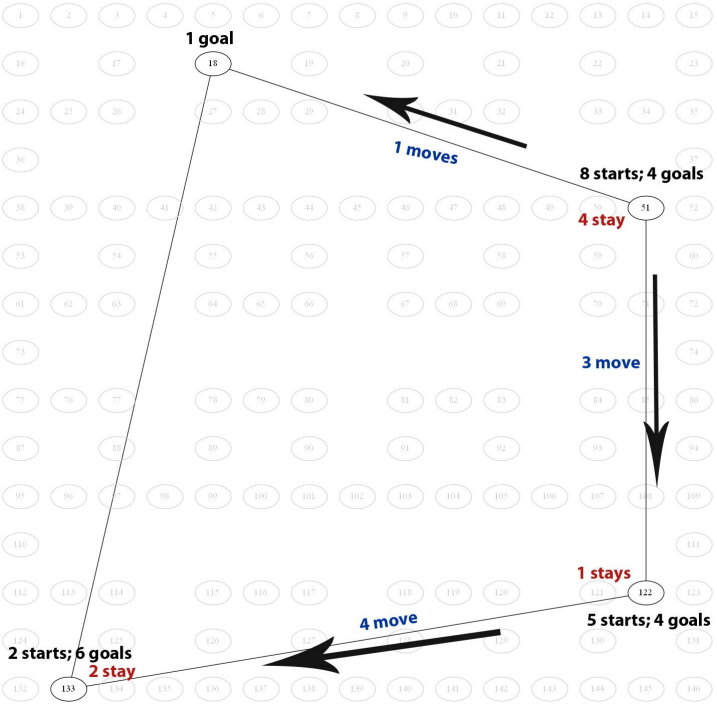
Deletion Solve

1. Start on highest abstraction level (1 vertex) and plan with upper bound 1
2. Group agents together that move together and plan them together on the next level
3. With previous plan length, calculate new upper bound
4. Reduce graph to those vertices that abstract into the visited ones
5. Load all starts and goals in graph and choose as many robots as moved on the higher layer
6. On lowest layer,
    a. load positions of already moved agents
    b. add conflicts constraints
7. Generate plan with minimum plan length and minimum moves

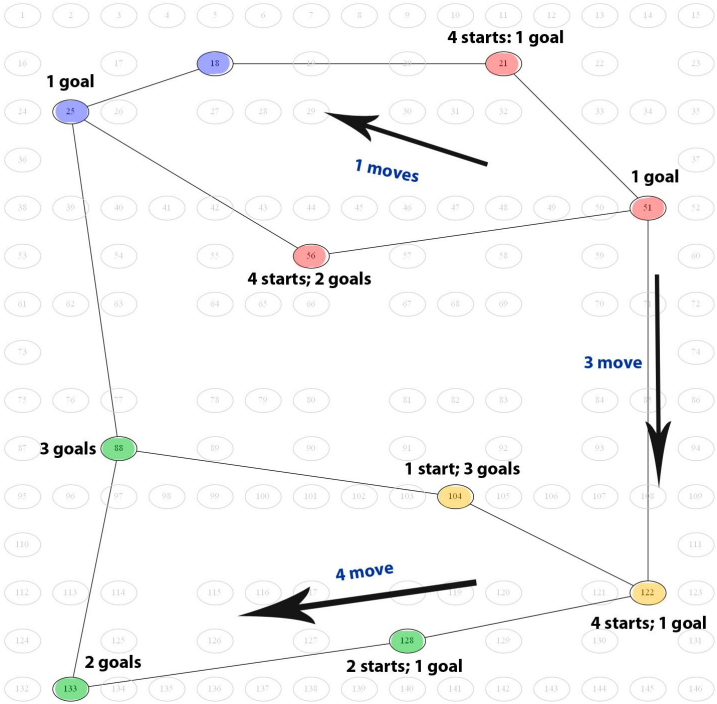
Deletion Solve



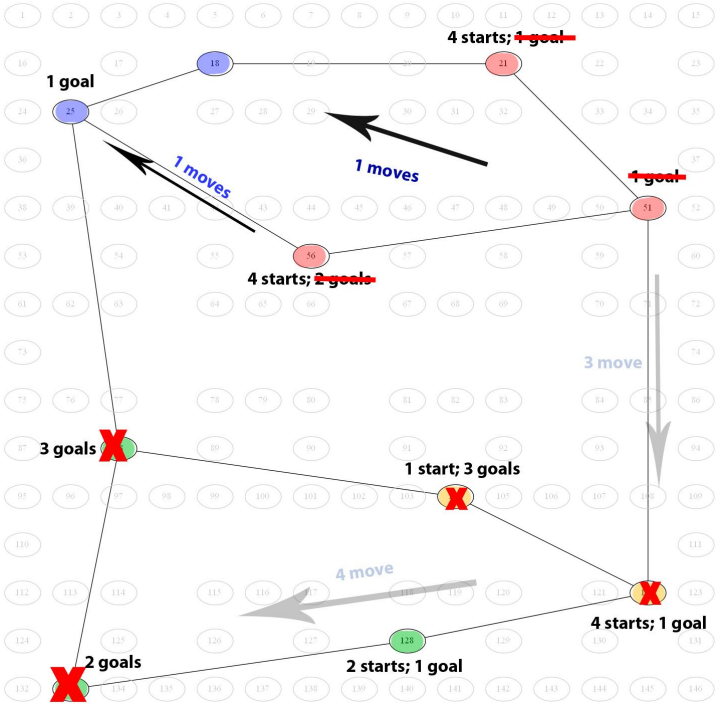
Deletion Solve



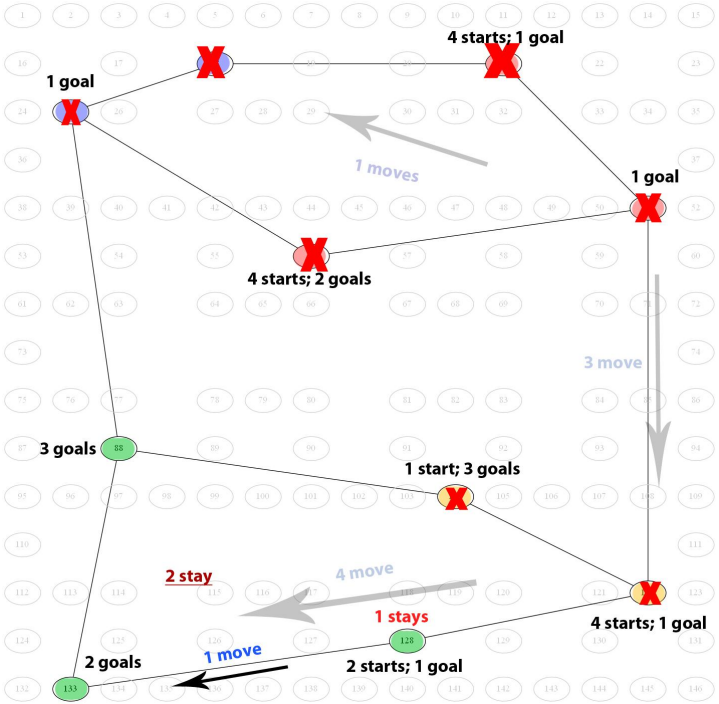
Deletion Solve



Deletion Solve



Deletion Solve



Benchmarks

Abstraction with optimization, solving incrementally

INSTANCE	OPTIMAL	ABSTRACT	SOLVE	TOTAL	PLAN	INSTANCE	OPTIMAL	ABSTRACT	SOLVE	TOTAL	PLAN
CIRCLES.LP	9	7032	1164	8196	12	N008.lp	3	536	72	608	3
TARGET.LP	12	11635	776	12411	13	N009.lp	4	504	56	560	4
SNAKE9X9.LP	36	6019	1779	7798	36	N010.lp	2	465	46	511	2
SNAKE15X15.LP	108	11825	30320	42145	108	N011.lp	3	448	43	491	3
0.LP	3	401	30	431	3	N012.lp	4	478	47	525	4
8.LP	5	612	55	667	6	N013.lp	33	6994	60383	67377	-1
20.LP	24	12625	6549	19174	32	N014.lp	24	7075	63735	70809	-1
XLARGE.LP	3	16247	61186	77433	-1	N015.lp	1	448	29	477	1
BENCHMARK1.LP	2	6085	150	6235	3	N016.lp	3	482	44	526	3
BENCHMARK2.LP	4	7324	279	7603	4	N017.lp	5	3343	214	3557	5
CORRIDOR_INSTANCE.LP	23	1534	1117	2651	23	N018.lp	3	498	60	558	3
PILLARS_INSTANCE.LP	8	6227	437	6664	12	N019.lp	6	880	60167	61047	-1
N001.LP	2	304	17	321	2	N020.lp	4	6833	933	7766	6
N002.LP	3	474	45	519	3	N021.lp	3	6173	340	6513	4
N003.LP	3	587	50	637	4	N022.lp	???	12576	61007	73584	-1
N005.LP	2	556	39	595	2	N023.lp	<=26	6618	61256	67874	-1
N006.LP	1	298	24	322	1	N024.lp	15	1120	637	1757	17
N007.LP	4	466	39	505	4						

circles.lp	775	snake9x9.lp	1547
target.lp	1072	snake15x15.lp	28308

instance	optimal_plan	abstraction	abstraction_time	solver	solving_time	total_time	plan_length
circles.lp	9	radius_incremental	6655	deletion_incremental	1287	7942	11
target.lp	12	radius_incremental	23669	deletion_incremental	792	24461	13
snake9x9.lp	36	radius_incremental	1316	deletion_incremental	2347	3663	36
snake15x15.lp	108	radius_incremental	4854	deletion_incremental	79496	84350	-1
0.lp	3	radius_incremental	569	deletion_incremental	34	603	3
8.lp	5	radius_incremental	484	deletion_incremental	41	525	6
20.lp	24	radius_incremental	40791	deletion_incremental	15176	55967	35
xlarge.lp	3	radius_incremental	-1	deletion_incremental	-1	-1	-1
N001.lp	2	radius_incremental	325	deletion_incremental	18	343	2
N002.lp	3	radius_incremental	631	deletion_incremental	44	675	3
N003.lp	3	radius_incremental	576	deletion_incremental	67	643	3
N005.lp	2	radius_incremental	523	deletion_incremental	29	552	2
N006.lp	1	radius_incremental	302	deletion_incremental	25	327	1
N007.lp	4	radius_incremental	470	deletion_incremental	43	513	4
N008.lp	3	radius_incremental	613	deletion_incremental	96	709	5
N009.lp	4	radius_incremental	482	deletion_incremental	60	542	4
N010.lp	2	radius_incremental	481	deletion_incremental	46	527	2
N011.lp	3	radius_incremental	445	deletion_incremental	34	479	3
N012.lp	4	radius_incremental	816	deletion_incremental	67	883	4
N013.lp	33	radius_incremental	-1	deletion_incremental	-1	-1	-1
N014.lp	24	radius_incremental	-1	deletion_incremental	-1	-1	-1
N015.lp	1	radius_incremental	434	deletion_incremental	32	466	1
N016.lp	3	radius_incremental	659	deletion_incremental	79	738	3
N017.lp	5	radius_incremental	1493	deletion_incremental	269	1762	5
N018.lp	3	radius_incremental	674	deletion_incremental	53	727	3
N019.lp	6	radius_incremental	1172	deletion_incremental	60368	61540	-1
N020.lp	4	radius_incremental	-1	deletion_incremental	-1	-1	-1
N021.lp	3	radius_incremental	-1	deletion_incremental	-1	-1	-1
N022.lp	???	radius_incremental	15599	deletion_incremental	69595	85194	-1
N023.lp	<=26	radius_incremental	3257	deletion_incremental	60221	63478	-1
N024.lp	15	radius_incremental	1115	deletion_incremental	563	1678	16
benchmark1.lp	2	radius_incremental	-1	deletion_incremental	-1	-1	-1
benchmark2.lp	4	radius_incremental	-1	deletion_incremental	-1	-1	-1
corridor_instance.lp	23	radius_incremental	1848	deletion_incremental	1489	3338	23
pillars_instance.lp	8	radius_incremental	-1	deletion_incremental	-1	-1	-1

Instance	Optimal	Abstraction	Abstract time	Solver	Solving time	Total time	Plan length
circles.lp	9	radius_optimize	7032	deletion_incremental	1164	8196	12
target.lp	12	radius_optimize	11635	deletion_incremental	776	12411	13
snake9x9.lp	36	radius_optimize	6019	deletion_incremental	1779	7798	36
snake15x15.lp	108	radius_optimize	11825	deletion_incremental	30320	42145	108
0.lp	3	radius_optimize	401	deletion_incremental	30	431	3
8.lp	5	radius_optimize	612	deletion_incremental	55	667	6
20.lp	24	radius_optimize	12625	deletion_incremental	6549	19174	32
xlarge.lp	3	radius_optimize	16247	deletion_incremental	61186	77433	-1
N001.lp	2	radius_optimize	304	deletion_incremental	17	321	2
N002.lp	3	radius_optimize	474	deletion_incremental	45	519	3
N003.lp	3	radius_optimize	587	deletion_incremental	50	637	4
N005.lp	2	radius_optimize	556	deletion_incremental	39	595	2
N006.lp	1	radius_optimize	298	deletion_incremental	24	322	1
N007.lp	4	radius_optimize	466	deletion_incremental	39	505	4
N008.lp	3	radius_optimize	536	deletion_incremental	72	608	3
N009.lp	4	radius_optimize	504	deletion_incremental	56	560	4
N010.lp	2	radius_optimize	465	deletion_incremental	46	511	2
N011.lp	3	radius_optimize	448	deletion_incremental	43	491	3
N012.lp	4	radius_optimize	478	deletion_incremental	47	525	4
N013.lp	33	radius_optimize	6994	deletion_incremental	60383	67377	-1
N014.lp	24	radius_optimize	7075	deletion_incremental	63735	70809	-1
N015.lp	1	radius_optimize	448	deletion_incremental	29	477	1
N016.lp	3	radius_optimize	482	deletion_incremental	44	526	3
N017.lp	5	radius_optimize	3343	deletion_incremental	214	3557	5
N018.lp	3	radius_optimize	498	deletion_incremental	60	558	3
N019.lp	6	radius_optimize	880	deletion_incremental	60167	61047	-1
N020.lp	4	radius_optimize	6833	deletion_incremental	933	7766	6
N021.lp	3	radius_optimize	6173	deletion_incremental	340	6513	4
N022.lp	???	radius_optimize	12576	deletion_incremental	61007	73584	-1
N023.lp	<=26	radius_optimize	6618	deletion_incremental	61256	67874	-1
N024.lp	15	radius_optimize	1120	deletion_incremental	637	1757	17
benchmark1.lp	2	radius_optimize	6085	deletion_incremental	150	6235	3
benchmark2.lp	4	radius_optimize	7324	deletion_incremental	279	7603	4
corridor_instance.lp	23	radius_optimize	1534	deletion_incremental	1117	2651	23
pillars_instance.lp	8	radius_optimize	6227	deletion_incremental	437	6664	12

instance	optimal_plan	abstraction	abstraction_time	solver	solving_time	total_time	plan_length
circles.lp	9	radius_optimize	7554	deletion_optimize	15326	22880	12
target.lp	12	radius_optimize	11740	deletion_optimize	13230	24969	13
snake9x9.lp	36	radius_optimize	6247	deletion_optimize	11348	17595	39
snake15x15.lp	108	radius_optimize	12113	deletion_optimize	50070	62184	406
0.lp	3	radius_optimize	477	deletion_optimize	60	537	3
8.lp	5	radius_optimize	975	deletion_optimize	153	1128	5
20.lp	24	radius_optimize	12929	deletion_optimize	39015	51944	63
xlarge.lp	3	radius_optimize	16700	deletion_optimize	172721	189421	32
N001.lp	2	radius_optimize	357	deletion_optimize	22	379	2
N002.lp	3	radius_optimize	571	deletion_optimize	133	704	3
N003.lp	3	radius_optimize	509	deletion_optimize	79	588	3
N005.lp	2	radius_optimize	455	deletion_optimize	46	501	2
N006.lp	1	radius_optimize	442	deletion_optimize	45	487	1
N007.lp	4	radius_optimize	661	deletion_optimize	75	736	4
N008.lp	3	radius_optimize	573	deletion_optimize	211	784	3
N009.lp	4	radius_optimize	476	deletion_optimize	250	726	4
N010.lp	2	radius_optimize	556	deletion_optimize	124	680	2
N011.lp	3	radius_optimize	453	deletion_optimize	144	597	3
N012.lp	4	radius_optimize	466	deletion_optimize	168	634	4
N013.lp	33	radius_optimize	6950	deletion_optimize	51879	58828	88
N014.lp	24	radius_optimize	7069	deletion_optimize	76542	83611	58
N015.lp	1	radius_optimize	450	deletion_optimize	42	492	1
N016.lp	3	radius_optimize	594	deletion_optimize	107	701	3
N017.lp	5	radius_optimize	3863	deletion_optimize	1854	5717	10
N018.lp	3	radius_optimize	727	deletion_optimize	191	918	3
N019.lp	6	radius_optimize	866	deletion_optimize	1724	2590	5
N020.lp	4	radius_optimize	6830	deletion_optimize	14483	21313	10
N021.lp	3	radius_optimize	6364	deletion_optimize	3585	9949	8
N022.lp	???	radius_optimize	12695	deletion_optimize	176244	188939	76
N023.lp	<=26	radius_optimize	6605	deletion_optimize	20054	26659	15
N024.lp	15	radius_optimize	1326	deletion_optimize	2978	4304	10
benchmark1.lp	2	radius_optimize	5882	deletion_optimize	3935	9817	8
benchmark2.lp	4	radius_optimize	7371	deletion_optimize	4592	11963	9
corridor_instance.lp	23	radius_optimize	1558	deletion_optimize	2731	4289	20
pillars_instance.lp	8	radius_optimize	6373	deletion_optimize	3342	9715	13

Conclusions

- Abstractions create a significant overhead
- Trade off between abstraction speed and abstraction optimality (levels and centers)
- Least abstraction level and least number of centers at each level does not guarantee optimal solving
- Very open maps take a long time to abstract
- More restricted maps, the benefit of abstraction is lower
    - If agents traverse the whole graph, might as well solve original graph

References

- Holte, R.C., Mkadmi, T., Zimmer, R.M., MacDonald, A.J.: Speeding up problem solving by abstraction: A graph oriented approach. Artificial Intelligence 85 (1996) 321–361
- Botea, A., Muller, M., Schaeffer, J.: Near optimal hierarchical path–finding. Journal of Game ¨ Development 1(1) (2004) 7–28
- Sturtevant, N.R., Buro, M.: Partial pathfinding using map abstraction and refinement. In: AAAI. (2005) 1392–1397
- Fernandez., A., Gonzalez, J.: Multi-Hierarchical Representation of Large-Scale Space. Kluwer (2001)
- Yang, Q., Tenenberg, J., Woods, S.: On the implementation and evaluation of ABTweak. Computational Intelligence Journal 12 (1996) 295–318
- Sturtevant N., Jansen R. (2007) An Analysis of Map-Based Abstraction and Refinement. In: Miguel I., Ruml W. (eds) Abstraction, Reformulation, and Approximation. SARA 2007. Lecture Notes in Computer Science, vol 4612. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-73580-9_27 Stern, Roni & Sturtevant,
- Nathan & Felner, Ariel & Koenig, Sven & Ma, Hang & Walker, Thayne & Li, Jiaoyang & Atzmon, Dor & Cohen, Liron & Kumar, T. & Boyarski, Eli & Barták, Roman. (2019). Multi-Agent Pathfinding: Definitions, Variants, and Benchmarks.
- Nathan Sturtevant and Michael Buro. (2005). Partial pathfinding using map abstraction and refinement. In Proceedings of the 20th national conference on Artificial intelligence - Volume 3 (AAAI'05). AAAI Press, 1392–1397. Write to Dave Hersh