Hierarchical path planning for multi-size agents in heterogeneous environments

Daniel Harabor, Adi Botea
NICTA & The Australian National University

IEEE Symposium on Computational Intelligence and Games
December 18, 2008

Outline

NICTA

- Motivation
- Prior Work
- Planning with Clearance Values
- Abstraction and Hierarchical Planning
- Results
- Conclusion

Motivation

NICTA

Path planning literature is full of assumptions...
That don't always hold in reality!

Knowledge engineering challenges:

- Identifying relevant domain-specific information.
- Extracting it automatically.
- Exploiting it to guide search.

Application areas:

- Video games.
- GPS systems.
- Any path planning system with heterogeneous agents.

GPS Fail

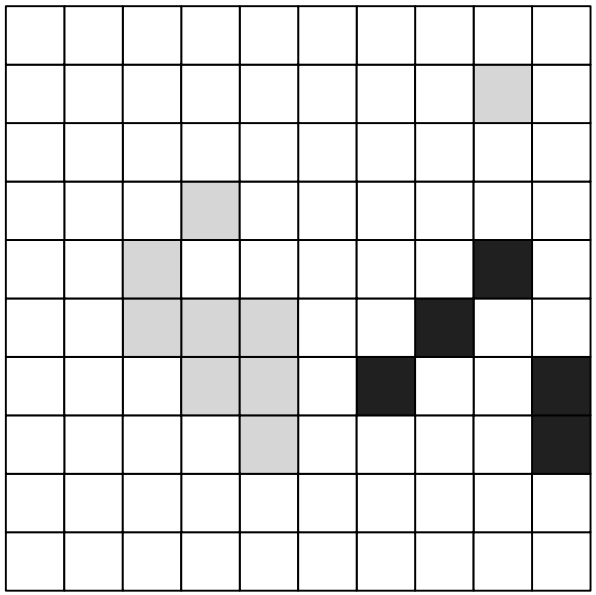
NICTA

Pictures: Danfung Dennis, NYTimes (04.12.2007)

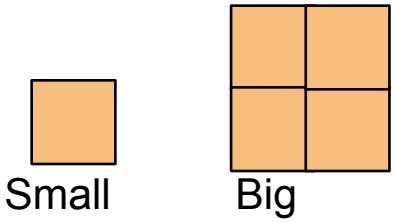
Problem definition

NICTA

Example map



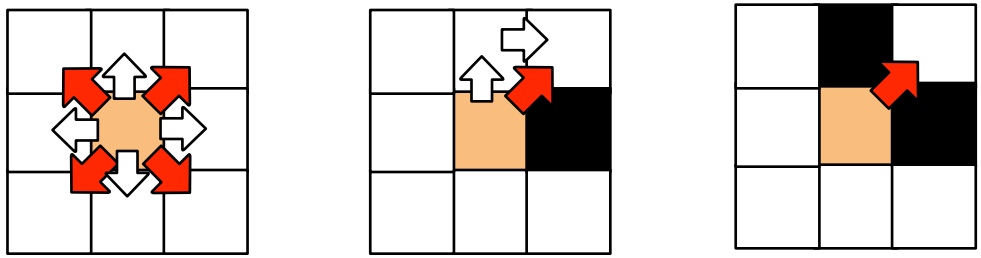
Example agent types



Terrain traversal capabilities:

- Ground
- Trees
- Ground or Trees

Movement rules



Previous Work

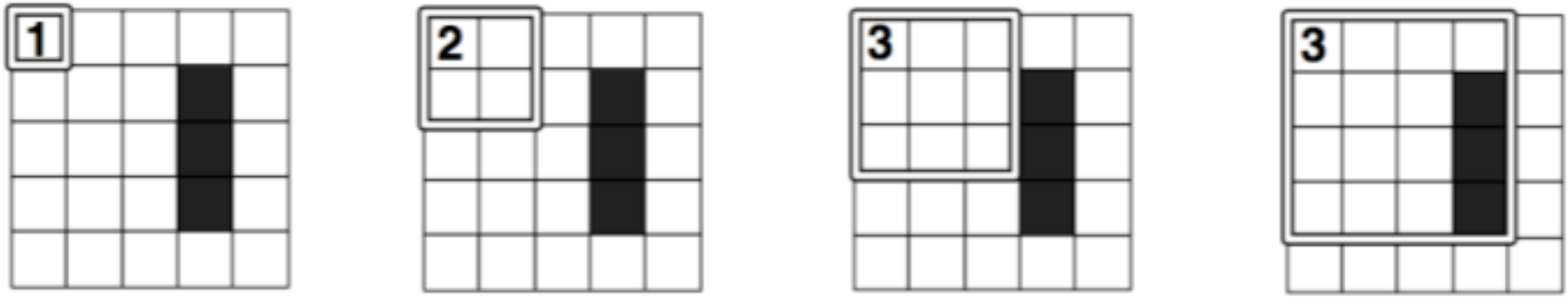
NICTA

- **A* [Hart et al, 1968]**
- Brushfire [Latombe, 1991]
- **HPA* [Botea et al, 2004]**
- PRA* [Sturtevant & Buro, 2005]
- TA*/TRA* [Demyen & Buro, 2006]
- **CMM [Geraerts & Overmars, 2007]**

Clearance Annotations

NICTA

Intuition: Calculate size of maximum traversable area at each octile (clearance value).

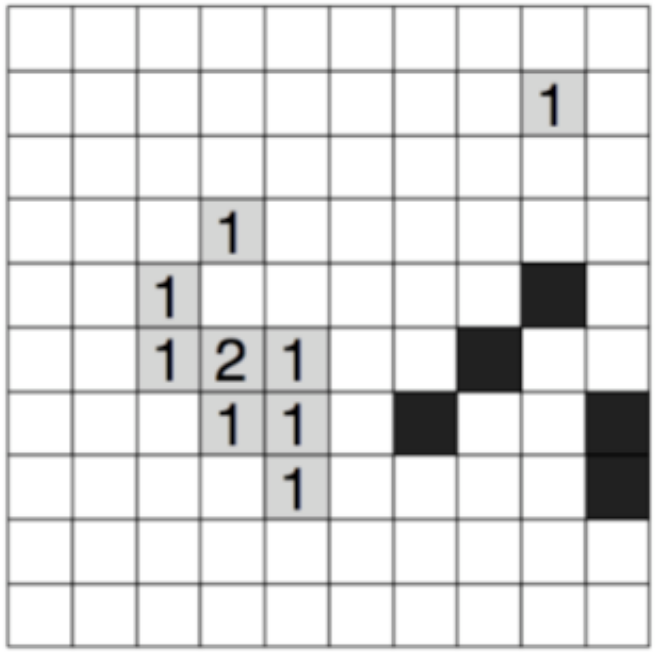


Results (toymap)

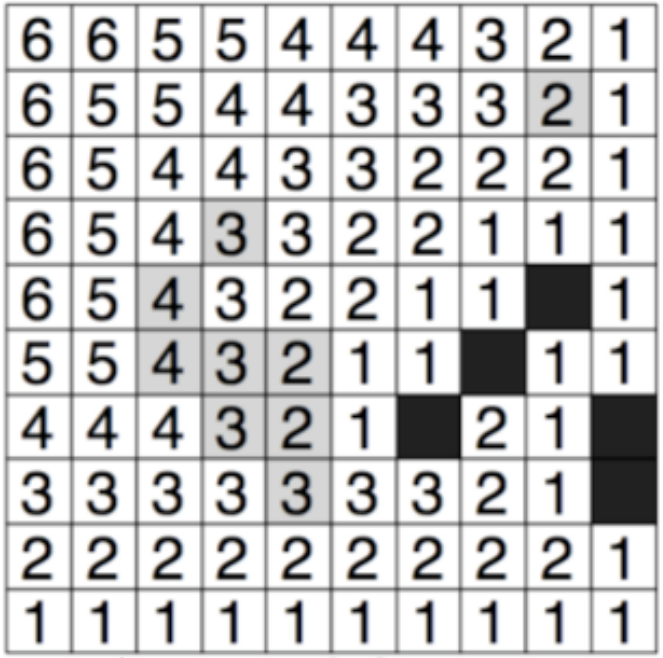
NICTA

Clearance values for different capabilities:

Ground



Trees



Ground or Trees

Results (toymap)

NICTA

Clearance values for different capabilities:

3	3	3	3	4	3	2	1	1	1
3	2	2	2	3	3	2	1		1
2	2	1	1	3	3	2	2	2	1
2	1	1		2	2	2	1	1	1
2	1		1	1	2	1	1	■	1
2	1				1	1	■	1	1
3	2	1			1	■	2	1	■
3	3	2	1		3	3	2	1	■
2	2	2	2	2	2	2	2	2	1
1	1	1	1	1	1	1	1	1	1

Ground

								1	
			1						
		1						■	
		1	2	1			■		
			1	1		■			■
				1					■

Trees

6	6	5	5	4	4	4	3	2	1
6	5	5	4	4	3	3	3	2	1
6	5	4	4	3	3	2	2	2	1
6	5	4	3	3	2	2	1	1	1
6	5	4	3	2	2	1	1	■	1
5	5	4	3	2	1	1	■	1	1
4	4	4	3	2	1	■	2	1	■
3	3	3	3	3	3	3	2	1	■
2	2	2	2	2	2	2	2	2	1
1	1	1	1	1	1	1	1	1	1

Ground or Trees

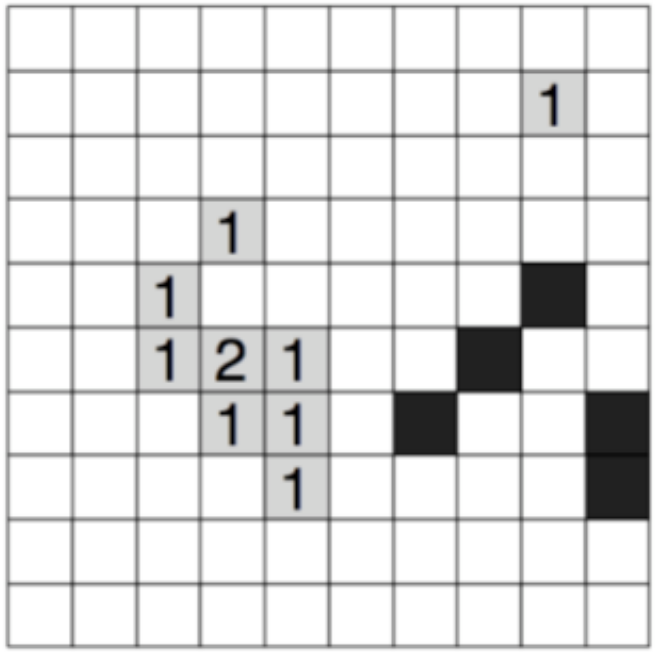
Space complexity: $|CV| = (|V| - |V_{HO}|) \times 2^{r-1}$

Results (toymap)

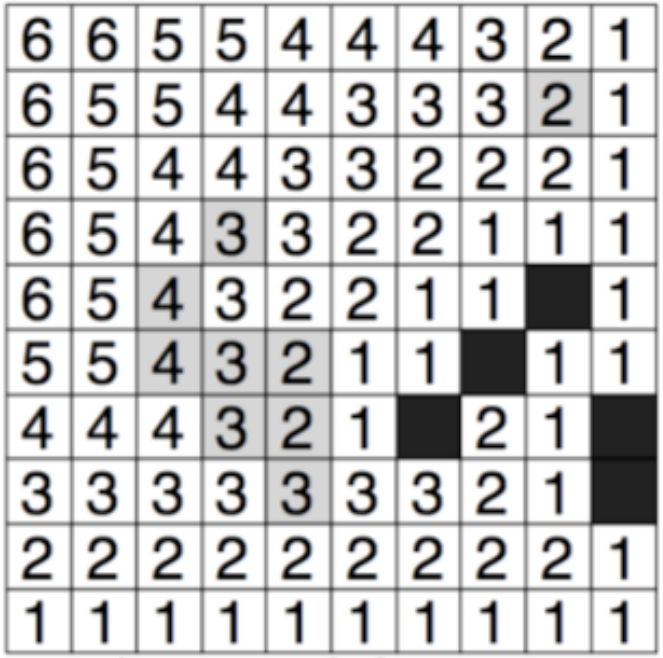
NICTA

Clearance values for different capabilities:

Ground



Trees



Ground or Trees

Space complexity: $|CV| = (|V| - |V_{HO}|) \times 2^{r-1}$

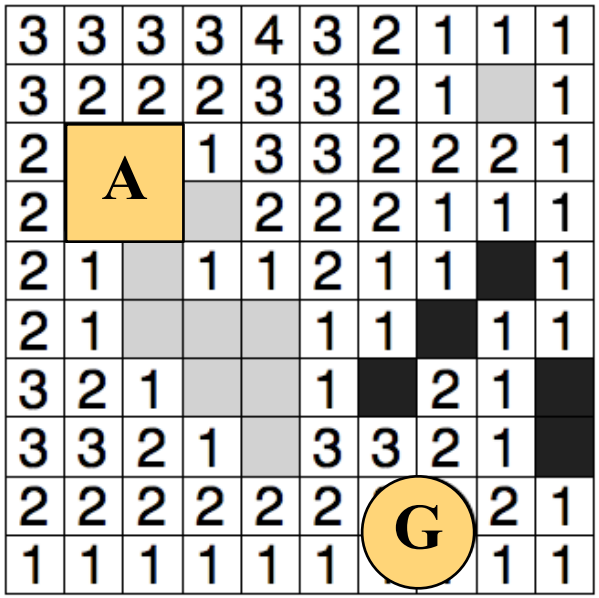
Compute on demand!

Emoticons: Wikimedia Foundation

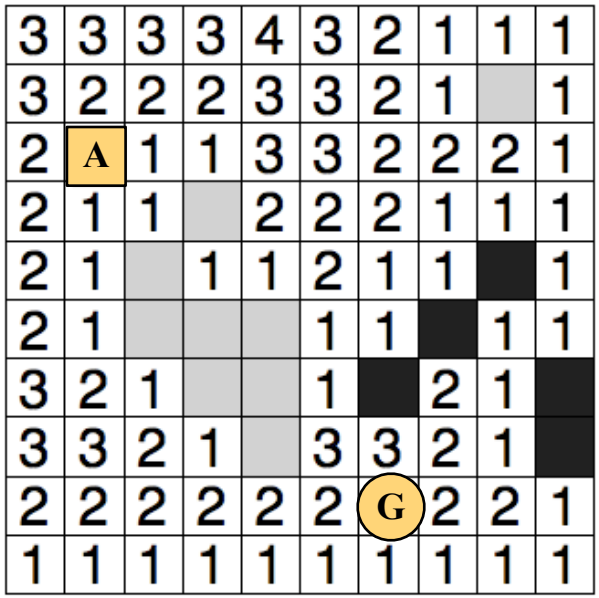
Reducing the problem

NICTA

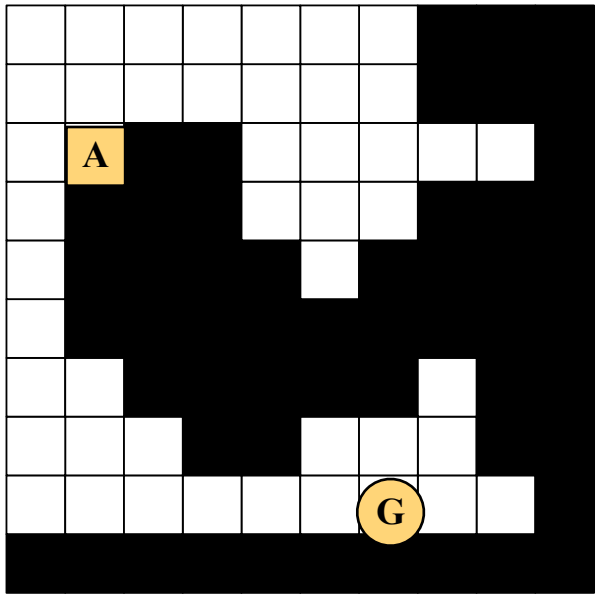
- Theorem 1: Any problem involving an agent of arbitrary size and capability can be reduced into a canonical problem (agent size = 1, capability = 1 terrain).



Initial problem



Reduced size



Reduced size and capability

Annotated A*

NICTA

Search process:

- Similar to A*.
- Extra parameters: Agent's size and capability.
- Only expand nodes with clearance > agent size.

Pros:

- Works great!

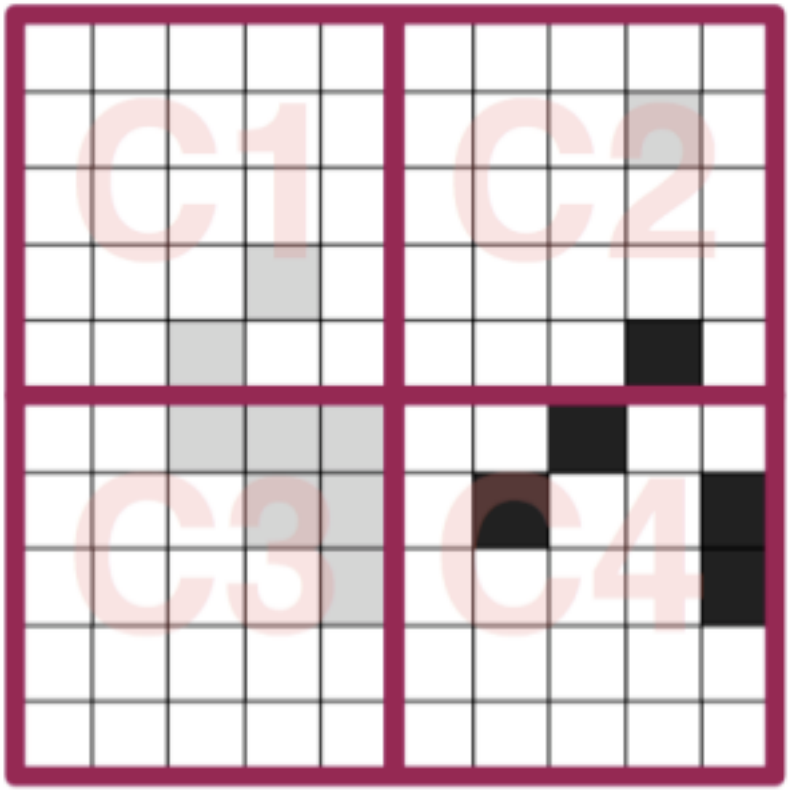
Cons:

- For small problem sizes...

Abstraction

NICTA

Intuition: Use hierarchical search. Apply cluster-based abstraction as per [Botea et al, 2004]

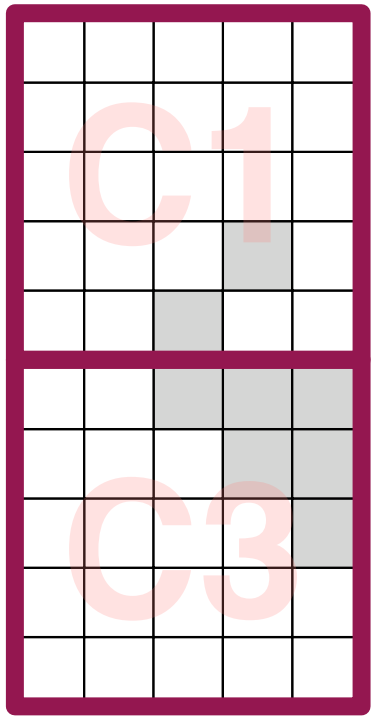


5x5 Clusters

Inter-cluster Transitions

NICTA

Approach: Build abstract graph by finding all entrances between clusters.

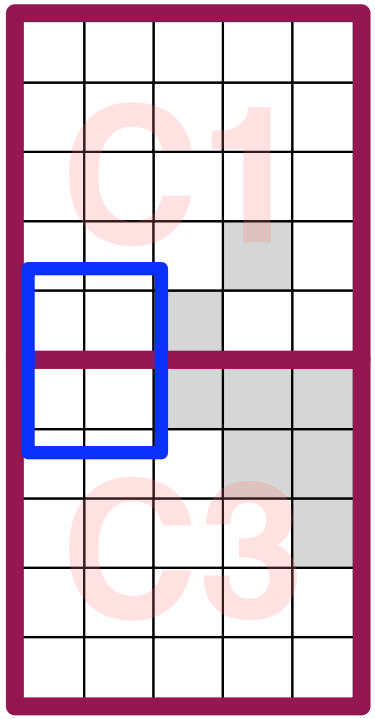


Identify entrances

Inter-cluster Transitions

NICTA

Approach: Build abstract graph by finding all entrances between clusters.

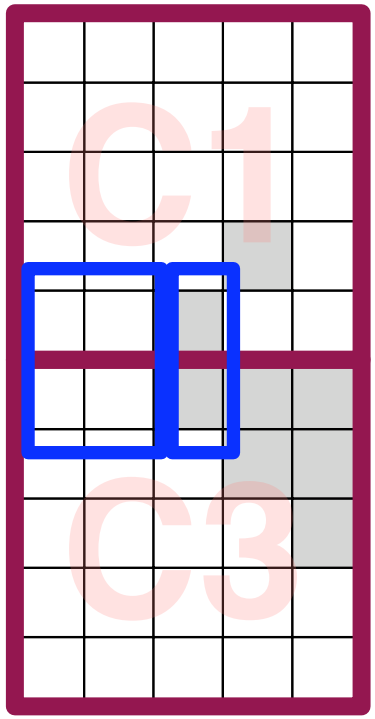


Identify entrances

Inter-cluster Transitions

NICTA

Approach: Build abstract graph by finding all entrances between clusters.

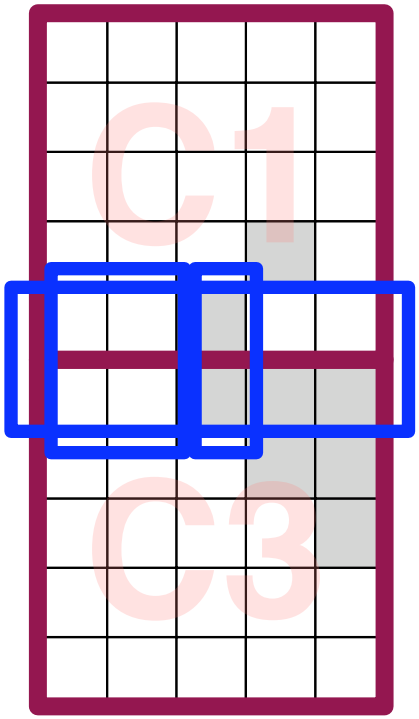


Identify entrances

Inter-cluster Transitions

NICTA

Approach: Build abstract graph by finding all entrances between clusters.

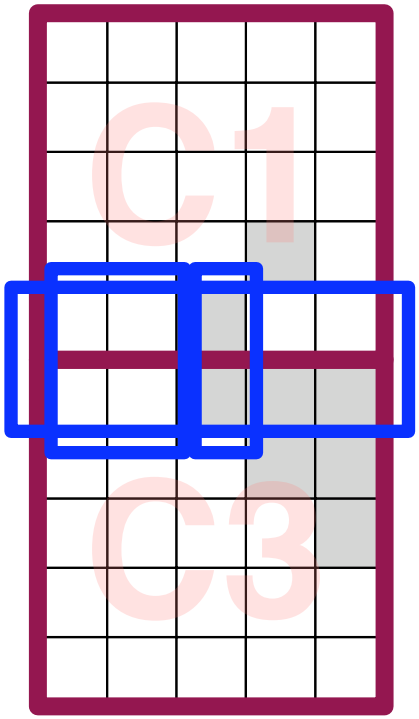


Identify entrances

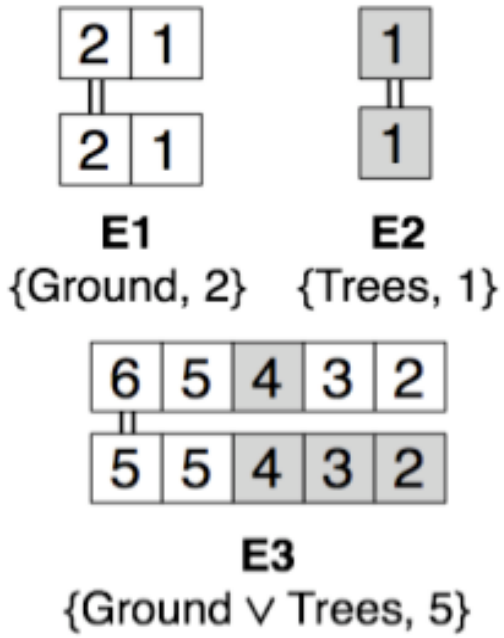
Inter-cluster Transitions

NICTA

Approach: Build abstract graph by finding all entrances between clusters.



Identify entrances

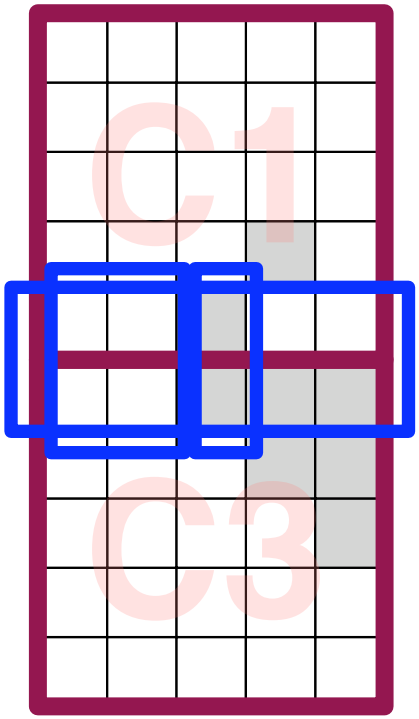


Identify transition points

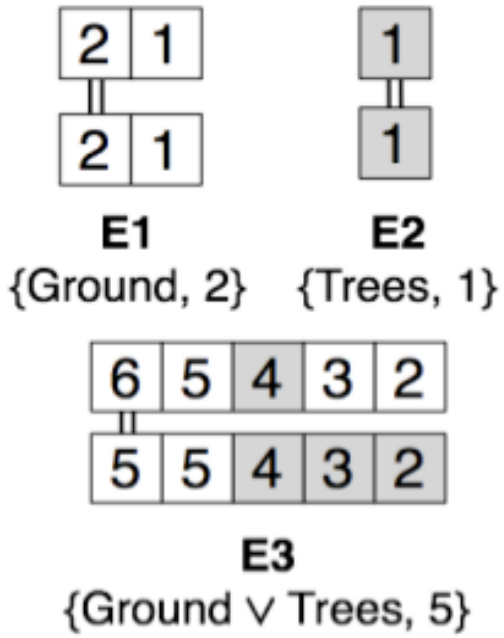
Inter-cluster Transitions

NICTA

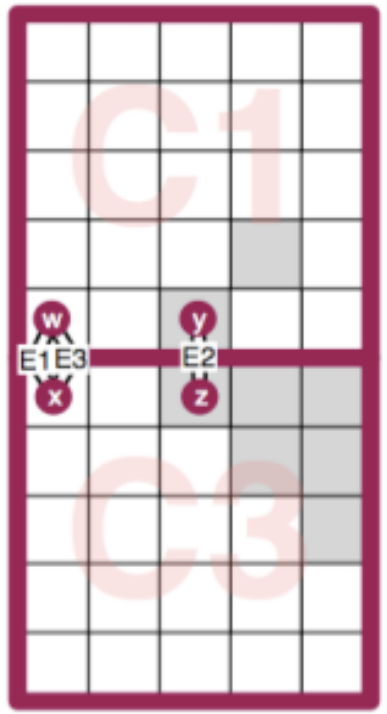
Approach: Build abstract graph by finding all entrances between clusters.



Identify entrances



Identify transition points

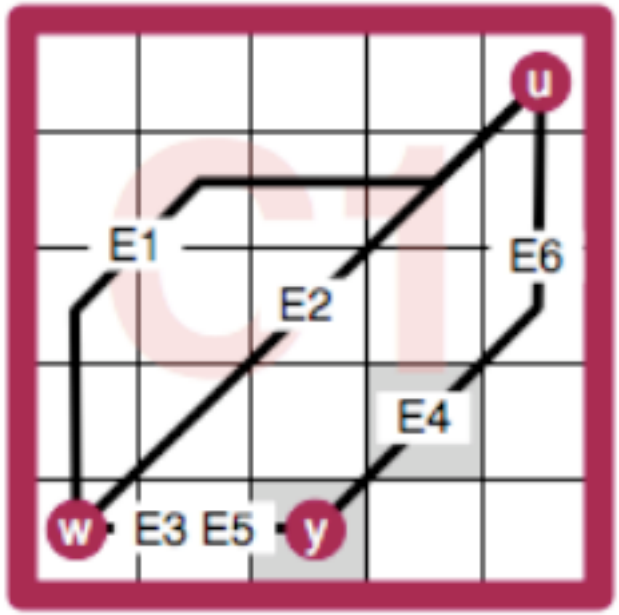


Final result

Intra-cluster Transitions

NICTA

Approach: Use AA* to find all paths between each pair of nodes inside a cluster.



Edge Annotations {Terrain, Clearance}:

E1 = {Ground, 2}
E2 = {Ground, 1}
E3 = {Ground V Trees,2}
E4 = {Ground V Trees, 2}
E5 = {Ground V Trees,1}
E6 = {Ground V Trees, 1}

If a path exists, add a new intra-edge edge to abstract graph; annotated with the capability and size parameters used by AA*

Compacting the abstract graph

Method produces a representationally complete graph but can get rather large.

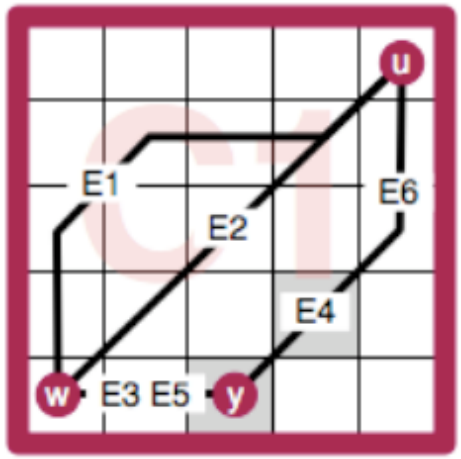
Solutions:

- Strong dominance
- Weak dominance.

Strong dominance example

NICTA

Intuition: Retain paths with larger clearance, all else being equal.



Edge Annotations {Terrain, Clearance}:

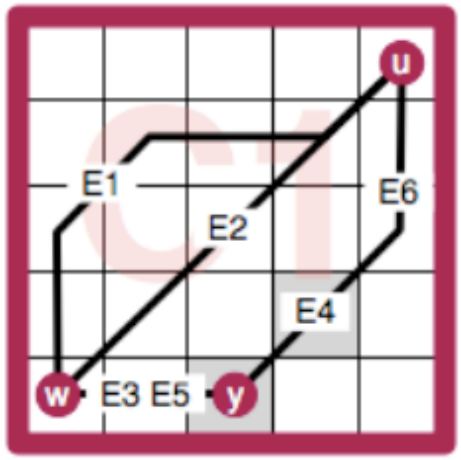
E1 = {Ground, 2}
E2 = {Ground, 1}
E3 = {Ground V Trees,2}
E4 = {Ground V Trees, 2}
E5 = {Ground V Trees,1}
E6 = {Ground V Trees, 1}

Result: High quality abstraction.

Strong dominance example

NICTA

Intuition: Retain paths with larger clearance, all else being equal.



Edge Annotations {Terrain, Clearance}:

E1 = {Ground, 2}
E2 = {Ground, 1}
E3 = {Ground V Trees,2}
E4 = {Ground V Trees, 2}
E5 = {Ground V Trees,1}
E6 = {Ground V Trees, 1}

$E3 \succ E5$

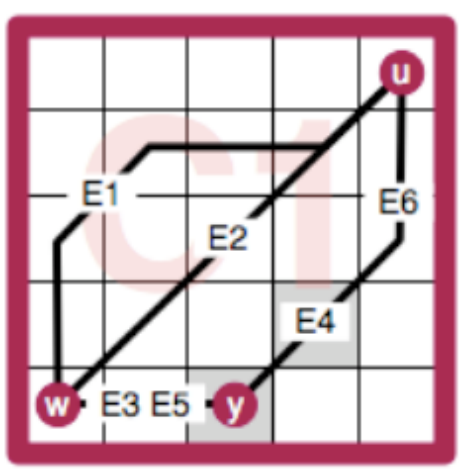
$E4 \succ E6$

Result: High quality abstraction.

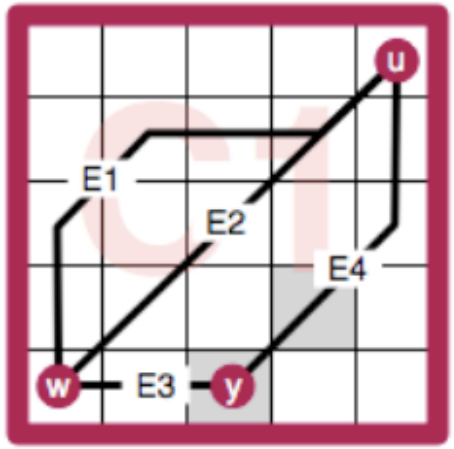
Strong dominance example

NICTA

Intuition: Retain paths with larger clearance, all else being equal.



Edge Annotations {Terrain, Clearance}:
E1 = {Ground, 2}
E2 = {Ground, 1}
E3 = {Ground V Trees,2}
E4 = {Ground V Trees, 2}
E5 = {Ground V Trees,1}
E6 = {Ground V Trees, 1}



Edge Annotations {Terrain, Clearance}:
E1 = {Ground, 2}
E2 = {Ground, 1}
E3 = {Ground V Trees,2}
E4 = {Ground V Trees, 2}

$E3 \succ E5$

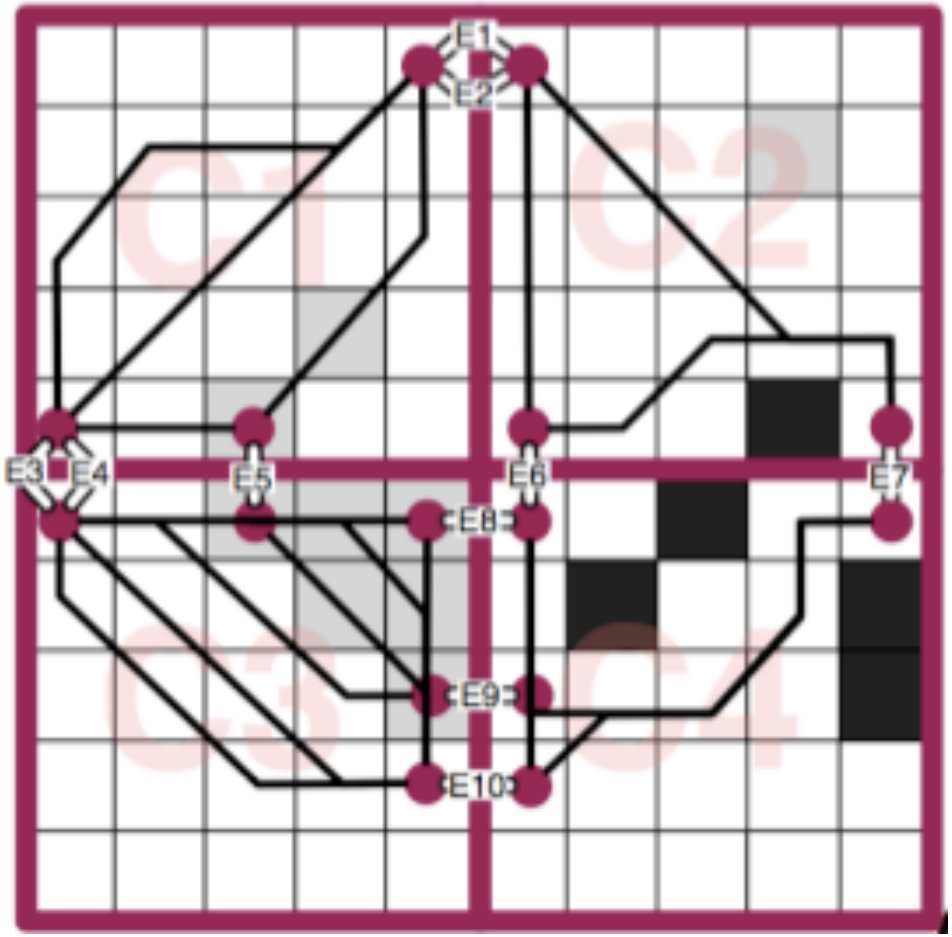
$E4 \succ E6$

Result: High quality abstraction.

Weak dominance example

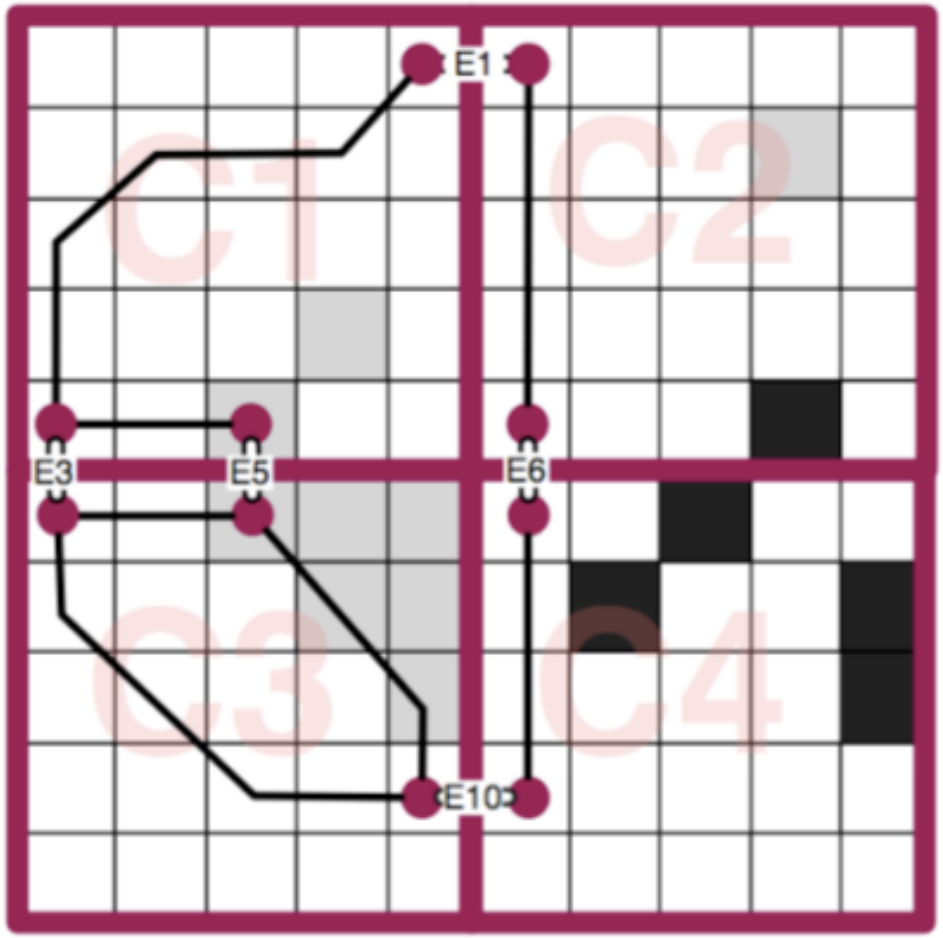
NICTA

Intuition: Retain edges with large clearance traversable by many agents (freeways vs. trails)



$E1 \succsim E2$ $E10 \succsim E9$ $E6 \succsim E7$

$E3 \succsim E4$ $E10 \succsim E8$

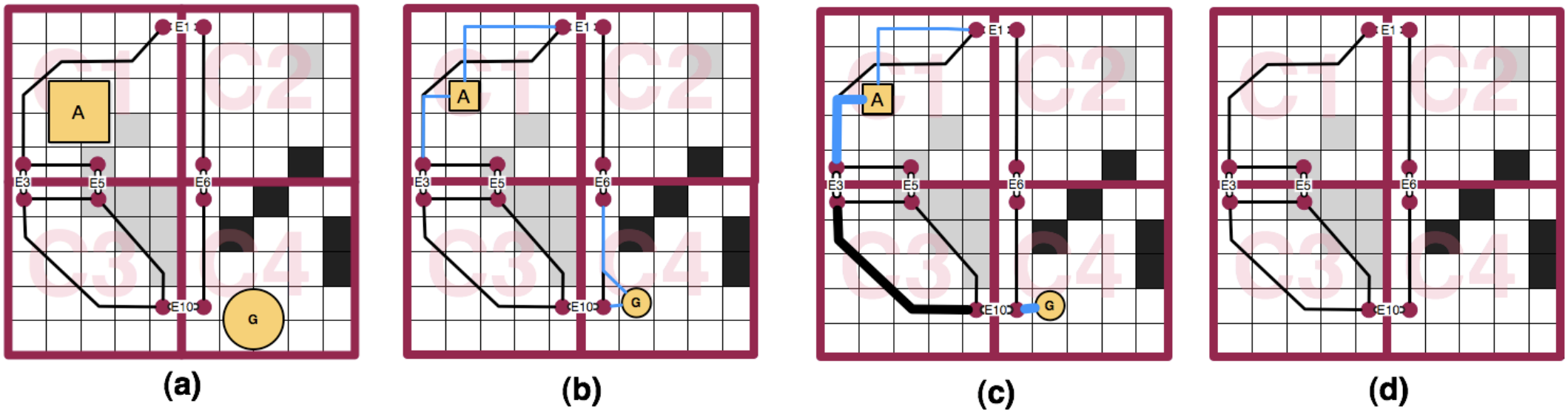


Result: low quality abstraction

Hierarchical Annotated A*

NICTA

- Extends HPA* [Botea et al 2004]
  - Insert start and goal into abstract graph
  - Find a hierarchical solution
  - Refine
- AA* for insertion step.
- Hierarchical search is a variation on A*
  - Requires agent size and capability as parameters
  - Only add successors to open list if edge is traversable



Experiments: Setup

NICTA

- 120 maps from Baldur's Gate.
- 3 cluster sizes (10, 15, 20)
- 5 derivative sets
  - Randomly interspersed each map with second terrain type (10%, 20%, 30%, 40% and 50%).
- 2 agent sizes (1 and 2).
- Randomly assigned capability
- 100 valid problems per map.
- Each agent size solves each problem.
- Intel Core2 Duo @ 2.4GHz w/ 1GB RAM (OSX 10.5.2)
- Implemented using Hierarchical Open Graph
- Source code at: http://ahastar.googlecode.com

Experiments: Graph size

NICTA

Original gridmaps averaged 4469 nodes & 16420 edges.

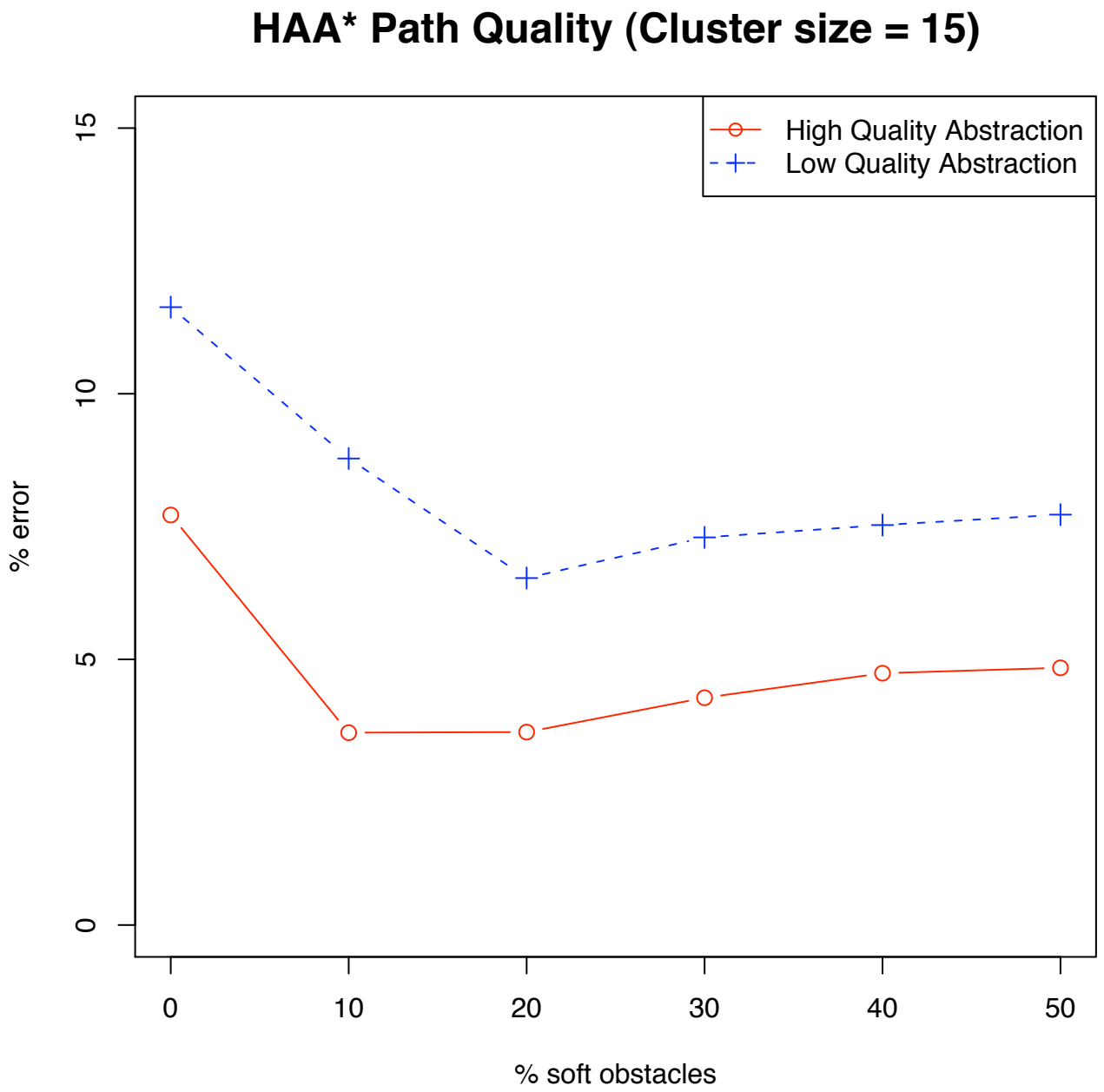
Best case (Cluster size = 20)

	HQ	LQ
Nodes	4.0%	2.0%
Edges	5.0%	0.9%

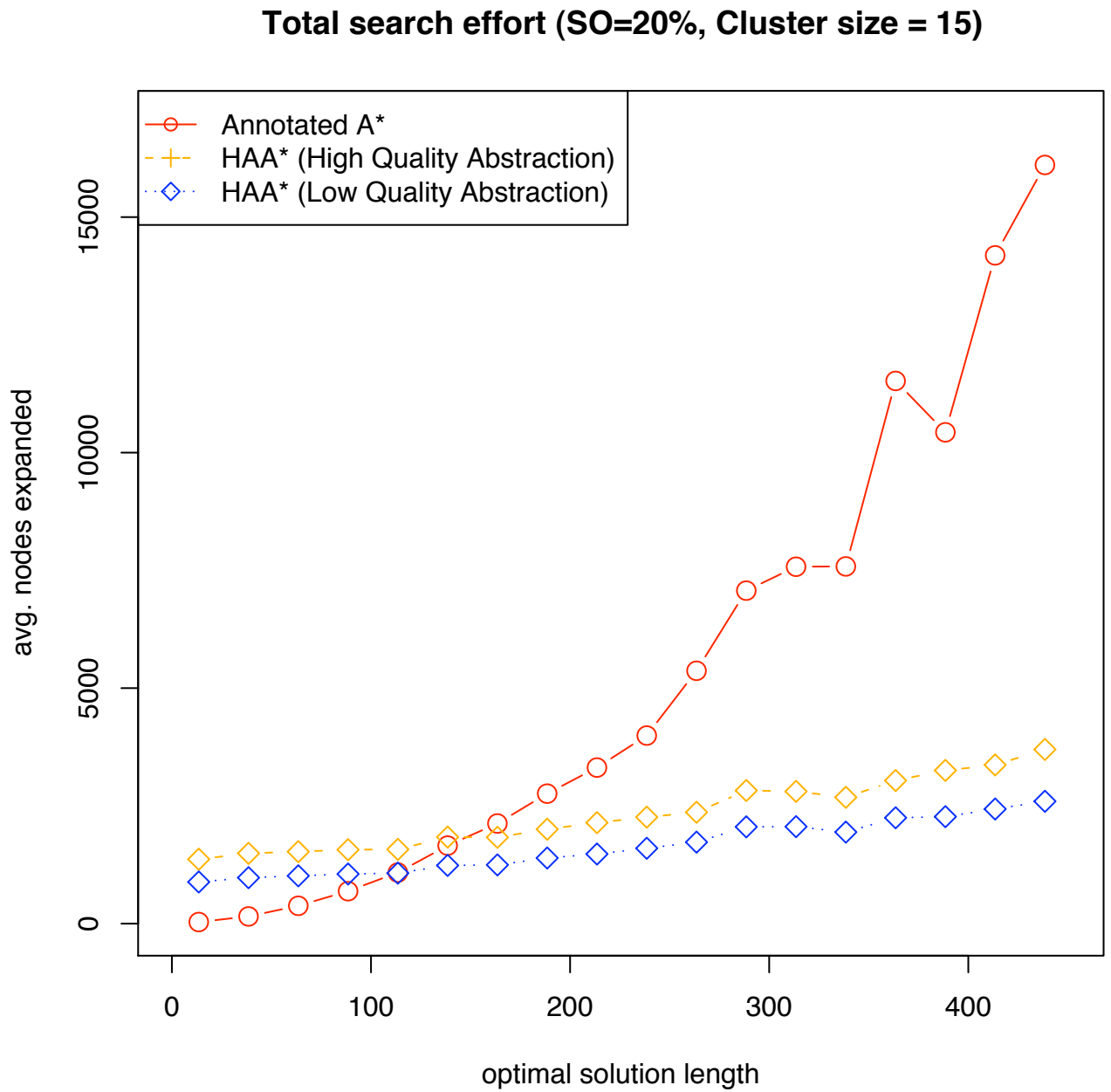
Worst case (Cluster size = 10)

	HQ	LQ
Nodes	16.6%	15.7%
Edges	38.4%	23.6%

Experiments: Path quality



Experiments: Search effort



Conclusion

NICTA

- Presented solutions for an overlooked but important problem in single-agent pathfinding.
  - Clearance value based pathfinding is simple and powerful.
  - Possible to build efficient hierarchical representations of complex environments.
  - Detailed empirical analysis shows method is very effective.
    - Near optimal solutions to complex problems.
    - Small memory overhead in practice.

- Future:
    - Reducing insertion effort.
    - Extend ideas to multi-agent case.
    - Apply to non-tile map encodings (like navigation meshes).

Questions

NICTA