# *Flexible Dynamic Space Partitioning for Robotic Rescue*

Jacky Baltes and John Anderson
Autonomous Agent Laboratory
Department of Computer Science
University of Manitoba
Canada, R3T 2N2

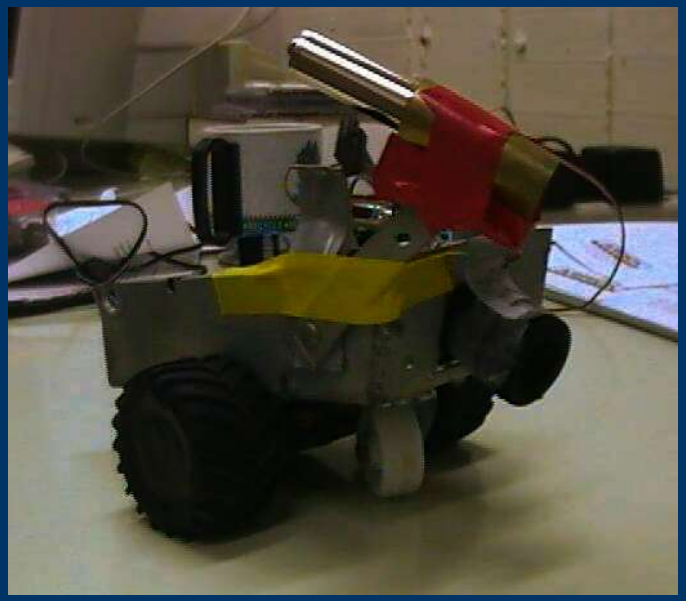
Email: {jacky,andersj}@cs.umanitoba.ca
WWW: http://www.cs.umanitoba.ca/~jacky

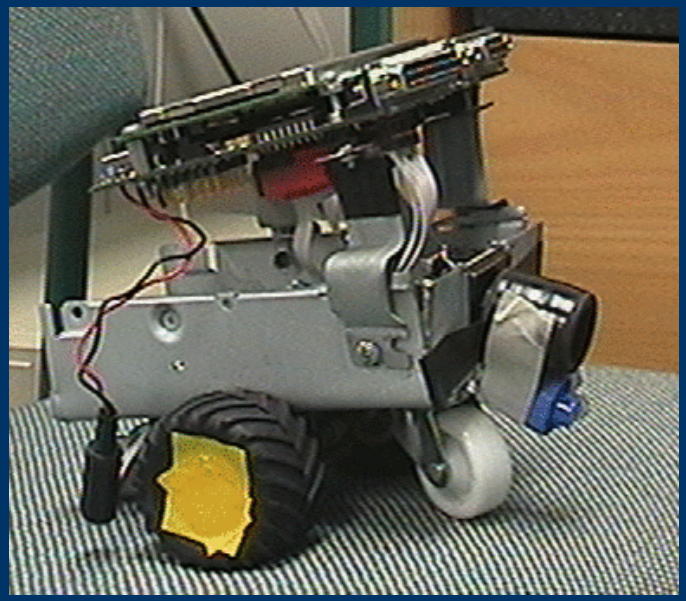
# Outline

- Introduction
- Path Planning
- Binary Space Partitioning
- Entropy and Information Gain Heuristic
- Empirical Evaluation
- Conclusion

# *Introduction*

- Keystone Rescue Team
  - Cheap robotic platforms
  - Computer vision as only/main sensor



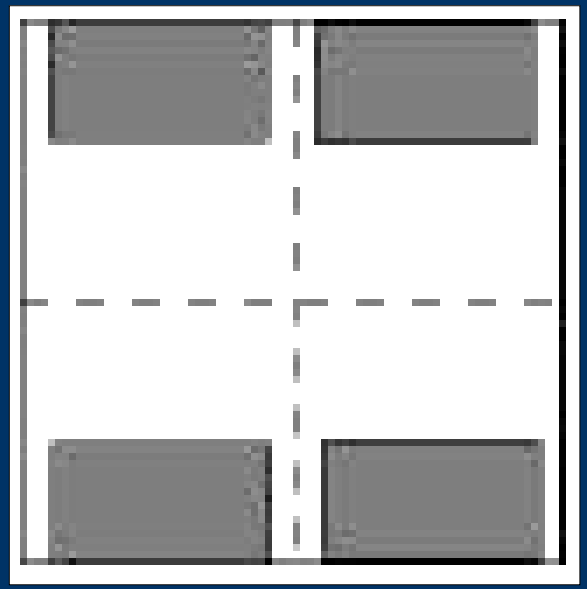


# Path Planning

- Skeletonization
  - Visibility graphs
  - Voroni diagrams
  - Problem:dynamic environments
- Local approaches
  - Potential fields
  - Gradient based method
  - Problem: local minima
- Cell decomposition
  - Quadtree decomposition
  - Recursively subdivide environment
  - Problem: sub-optimal partitioning

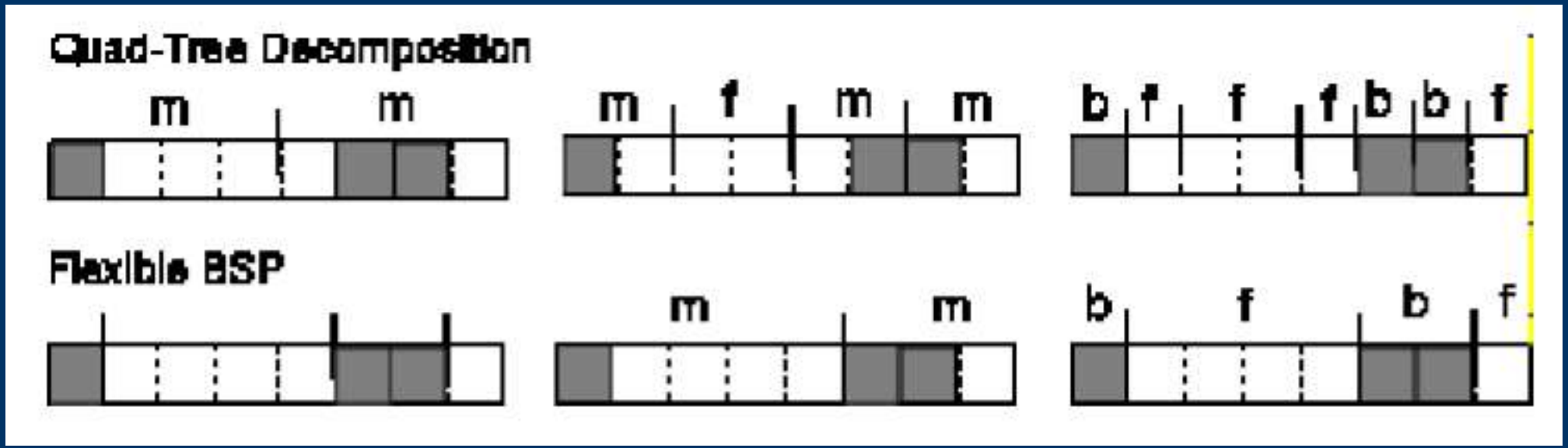
# Quadtree Decomposition

- All cells are broken into four equal sized regions
- Position of cells is fixed
- Computational cost, adjacency lists
- Example: all mixed cells



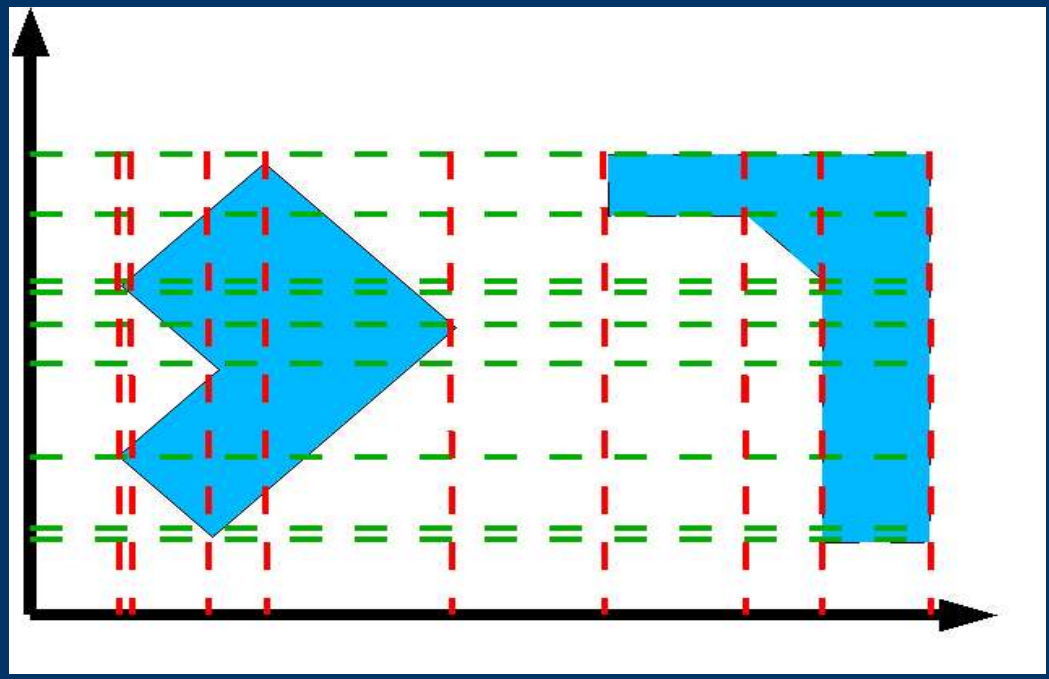
# *Flexible Binary space partitioning*

- Binary space paritioning subsumes quadtree decomposition
- Extend to allow partitioning at an arbitrary point
- One dimensional example of flexible BSP:



# *Finding partitioning points*

- Need to select
  - partitioning dimension
  - partitioning point
- Only check projection points
- Large jump in info. gain
- Linear complexity



# Entropy

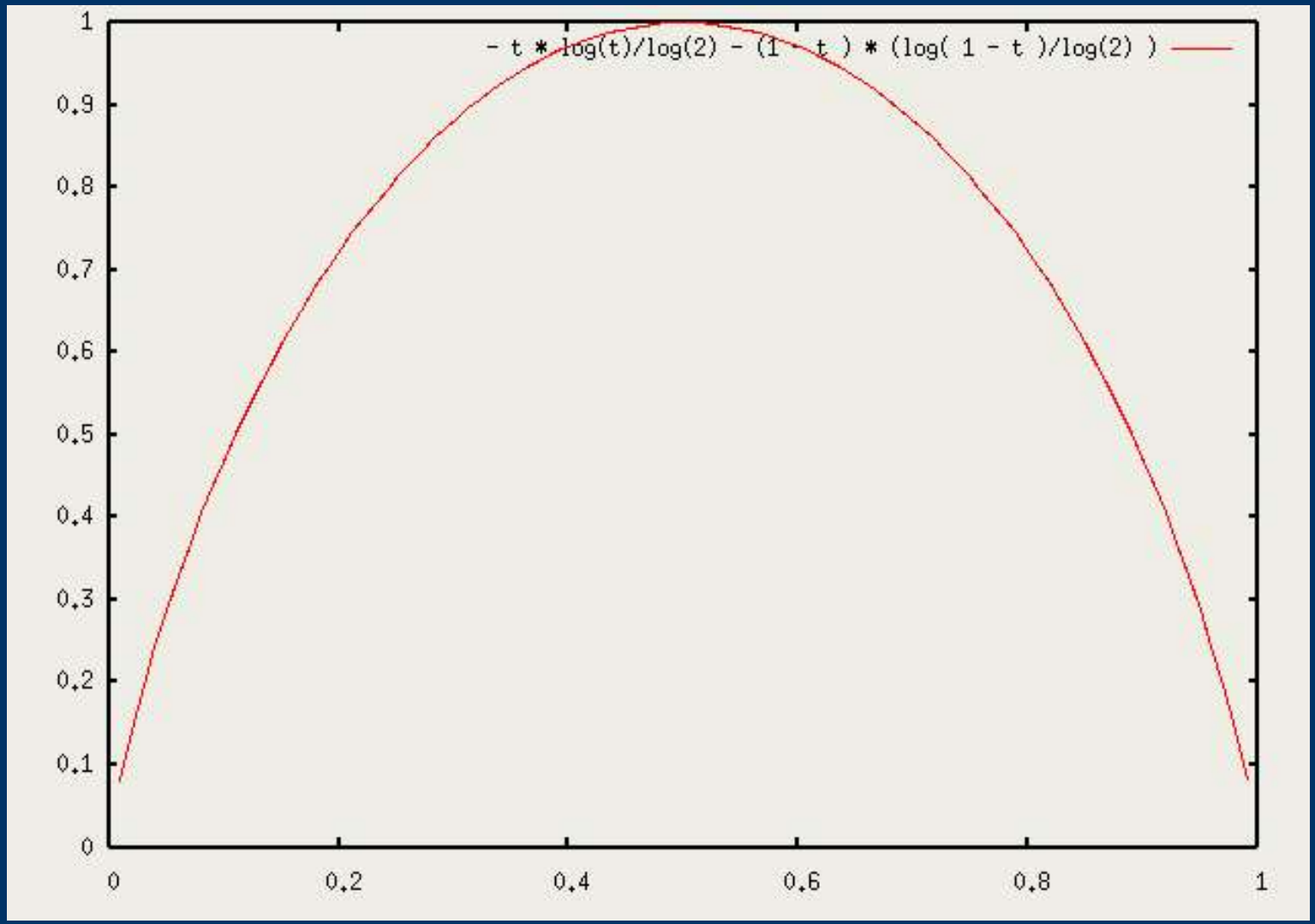
- Minimum length to encode area

$$\text{Entropy}(A) = -p_f \log_2 p_f - p_b \log_2 p_b$$

Percentage of free area $$p_f = \frac{|f|}{|A|}$$

Percentage of blocked area $$p_b = \frac{|b|}{|A|}$$

# Entropy

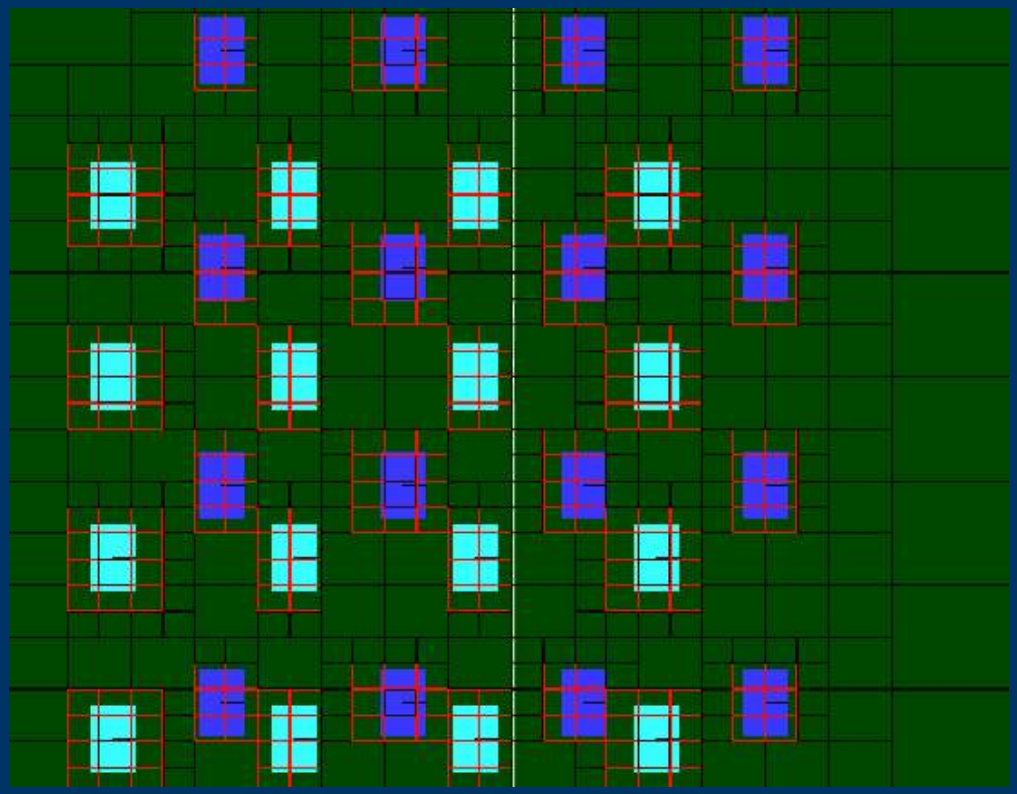

- t * log(t)/log(2) - (1 - t ) * (log( 1 - t )/log(2) )
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.2
0.4
0.6
0.8
1

# Information Gain Heuristic

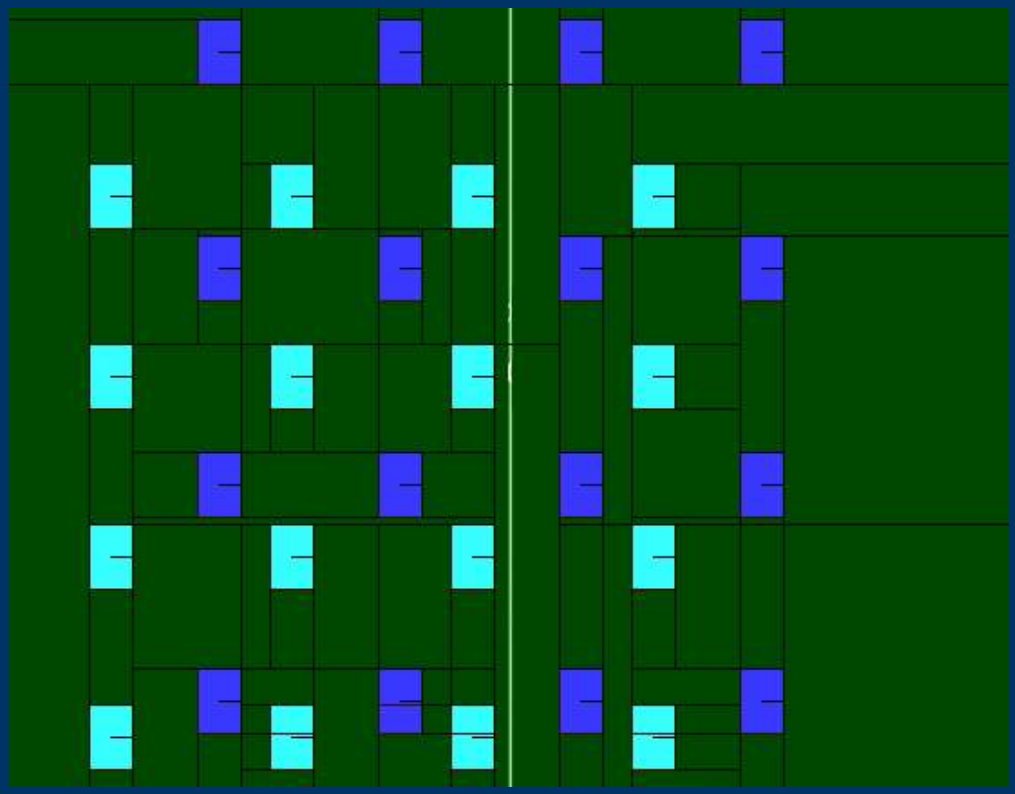
- Heurisitic used by ID3 and other machine learning algorithms
  - Create a minimum depth decision tree
  - Entropy
- Pick partition point $p$ that maximizes products of entropy and cell size

$$\text{gain}(p_i, A) = \text{E}(A) - \sum \frac{|P|}{|A|} \text{E}(|P|)$$

# *Example: 32 Obstacles*



Quadtree



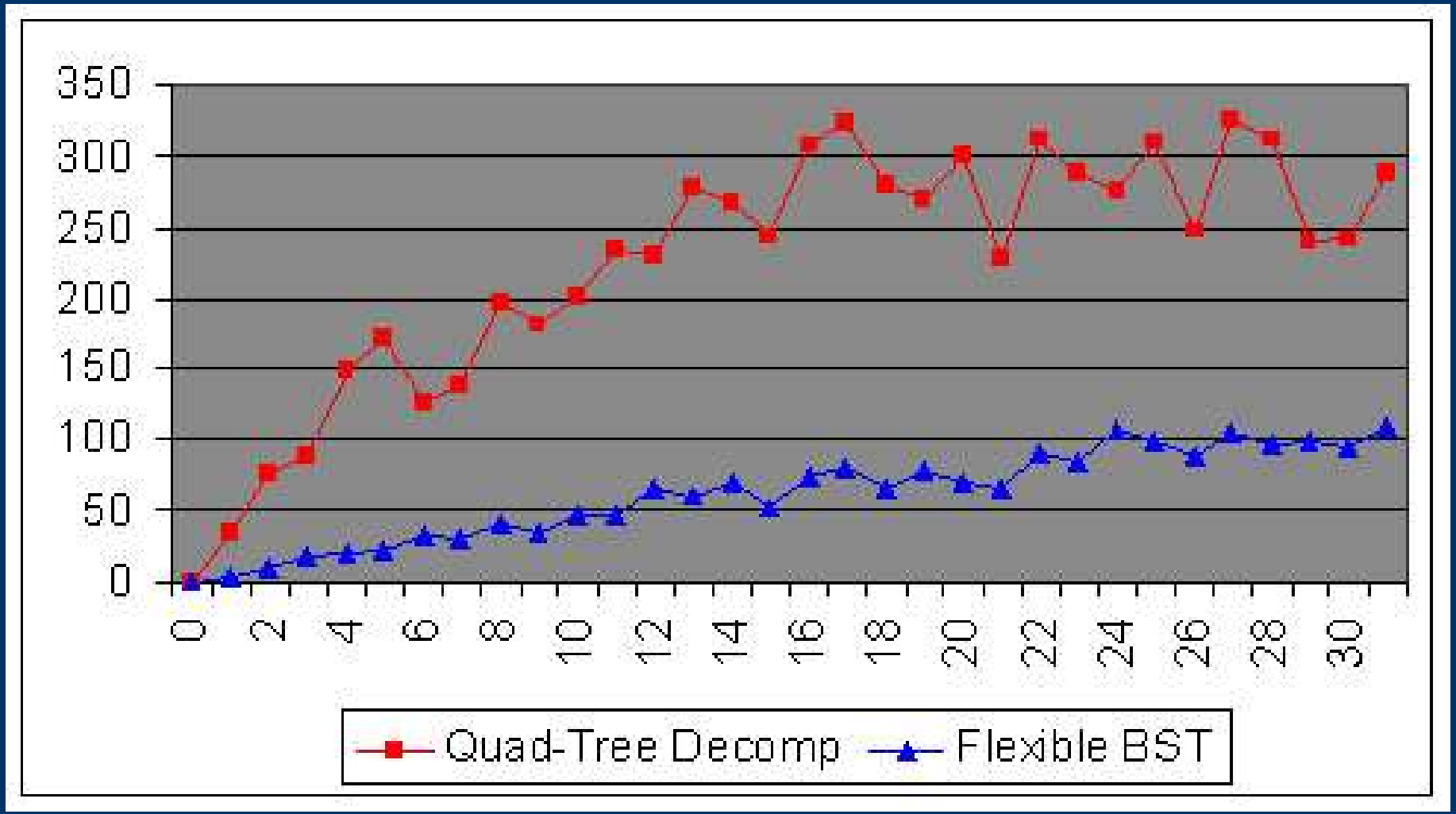
Flexible BSP

# *Empirical Evaluation*

- 2.8m by 2.3m environment
- Number of obstacles 1 – 32
- Size of obstacles randomized
- Repeated over a number of iterations

- Compared the number of cells generated
- Flexible BPS generated only 27% of the number of cells of standard Quadtree decomposition
- Even better results possible (11 vs. 47 cells)

# Empirical Evaluation



# *Conclusion*

- Flexible BSP produces only 27% of the number of cells in a decomposition
- The entropy heuristic leads naturally to an anytime algorithm
- Probabilistic algorithm suited for highly dynamic, uncertain environments
- Future Work: extend to 3D environments and quadtree decomposition