Università degli studi di Cagliari

Corso di laurea specialistica in ingegneria elettronica

Innovative Methods for analysis and evaluation of heuristic functions for global search algorithms

Filippo Ledda

Advisor:
Prof. Giuliano Armano

Advisor:
Prof. Carlos Linares López

Outline

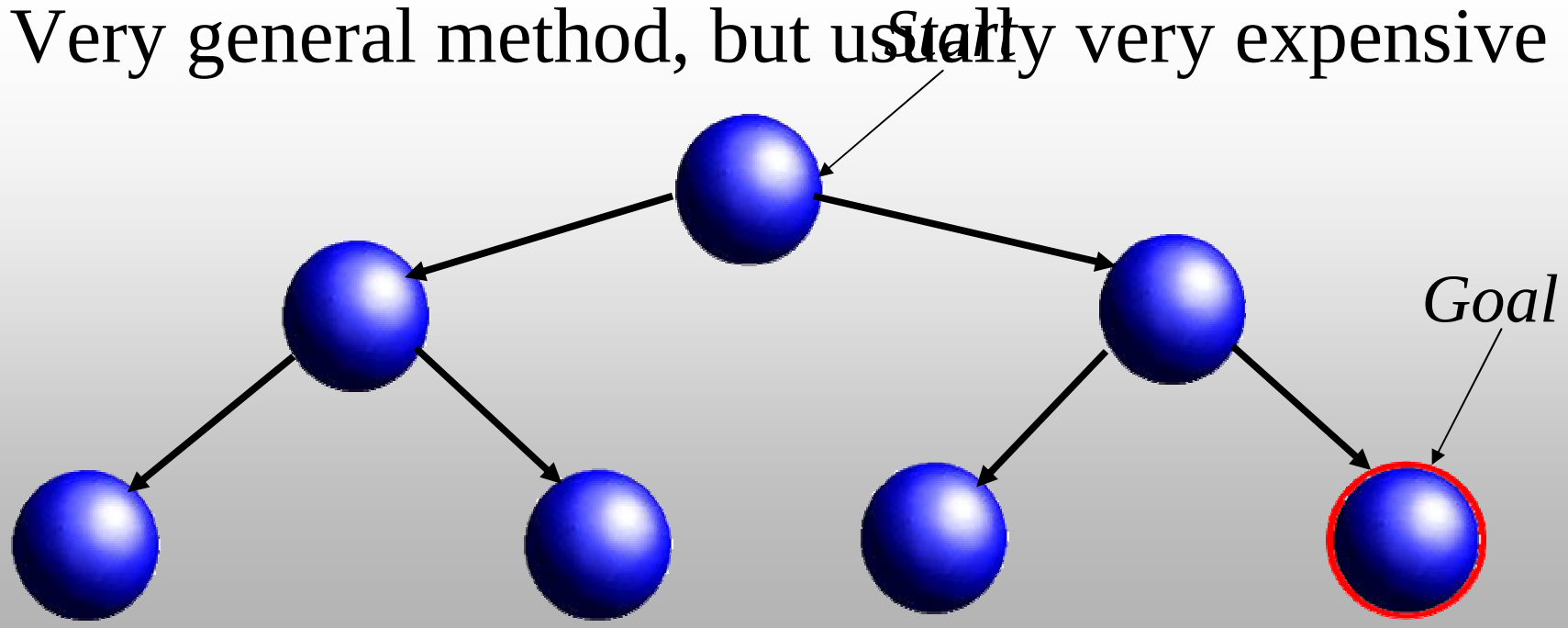
- Introduction to heuristic search
- State-of-the-art in the field of heuristic functions evaluation
- Innovative methods for evaluating heuristic functions
- Conclusions and future work

Introduction

- This thesis work is concerned with known and innovative methods for **evaluating heuristic functions** in the context of **global search** algorithms
  - **Search** is a general and effective method for problem solving in artificial intelligence
  - **Heuristic functions** improve effectiveness of search
- Not much work has been done so far

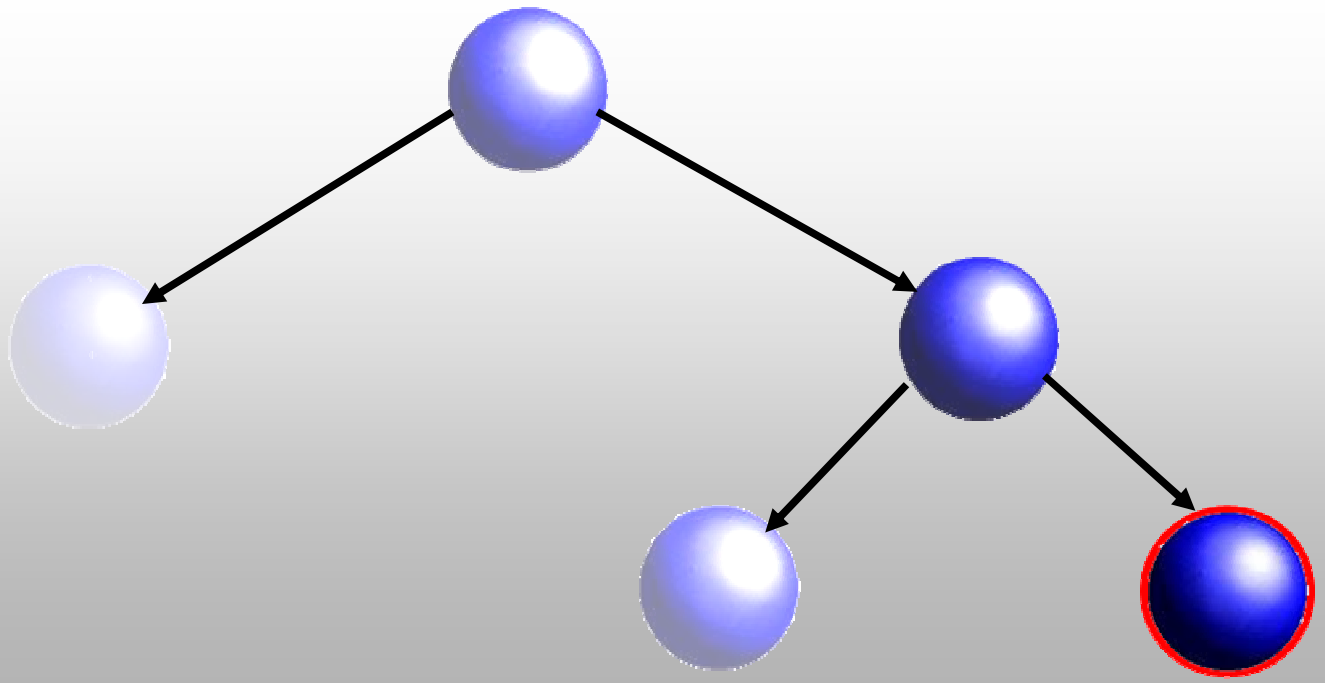
Problem solving with search

- The purpose of a search algorithm is to find a solution to a problem while navigating in a ***search space***
  - Global search: uses a tree structure to store search information
  - Very general method, but usually very expensive



Heuristic search

- Informed (heuristic) search
  - A function called ***heuristic*** guides the algorithm to the solution, usually giving an estimation about how far is a node from the solution



Heuristics evaluation

- Heuristics make **errors** in the estimation
  - how this errors affect the search?
- Evaluating heuristics is a very difficult task, and two important factors should be taken into account:
  - **Effectiveness**
  - **Efficiency**
- The task of evaluating the effectiveness of a heuristic along the search space is very hard
  - *Usually search spaces are very wide and heuristics locally behave unpredictably*

Heuristics evaluation: State-Of-The-Art (outline)

- *Admissible (optimistic, underestimating)* heuristics information comparison
  - ***Dominance***
  - ***Statistical Dominance***
- Estimating algorithms performance
  - ***Korf-Reid's Method***
    - Permits to estimate the number of expanded nodes by using the IDA* algorithm
  - ***Atomic tasks decomposition***

Admissible Heuristics Information Comparison: Dominance

- Admissible heuristics always underestimate the solution path cost
- An admissible heuristic h_1 is more informed than another admissible heuristic h_2 if:

$$h_1(n) \geq h_2(n) \forall n$$

- Drawbacks:
  - this condition is not usually met for real heuristics, even when one is clearly superior to the other
  - The condition must be verified for all nodes in the search space!

Stochastic Dominance: Probability Distribution Function

- A more useful definition can be obtained from statistical considerations
  - *Probability distribution function*

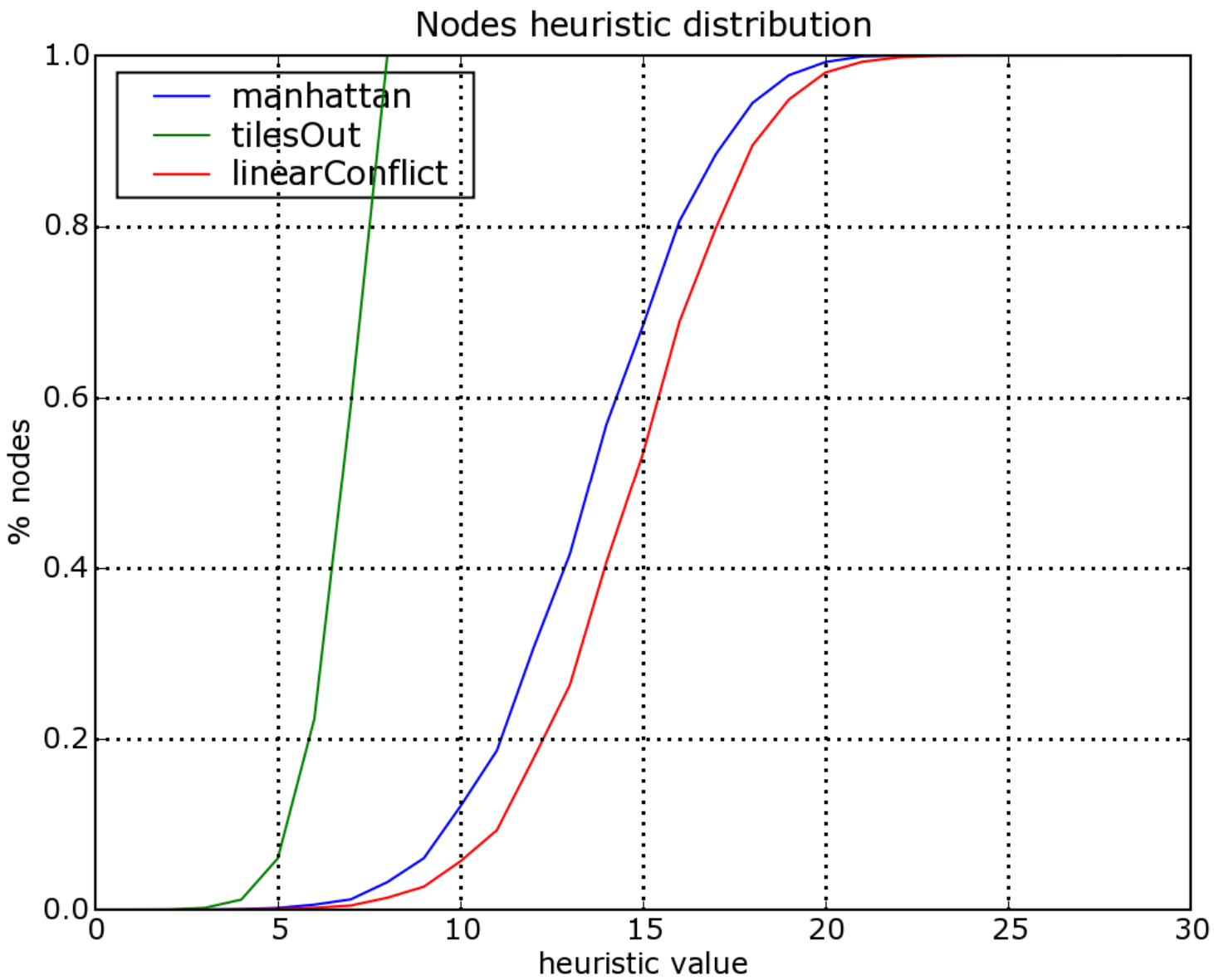
$$P(x) = P(h \leq x)$$

- A heuristic h_1 is stochastically more informed than another heuristic h_2 if both are admissible and:

$$P_{h_1}(x) \geq P_{h_2}(x) \forall x$$

- **The probability distribution can be approximated by sampling a fraction of the state space**

Example of Probability Distribution: 8-puzzle

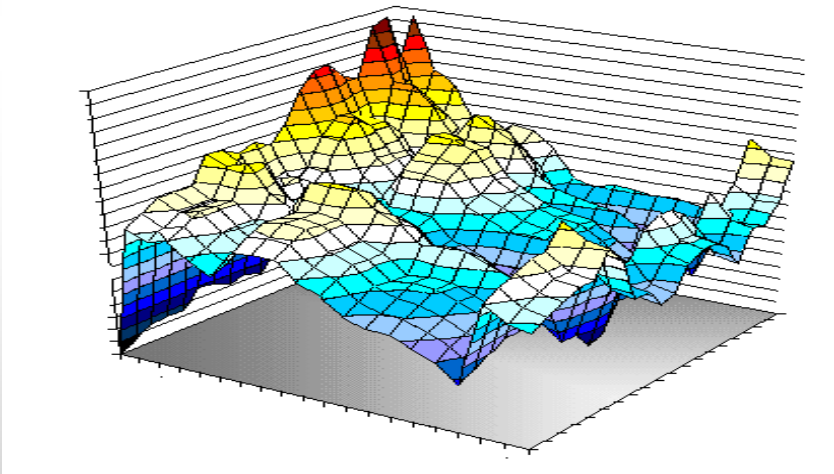


Innovative Methods For Heuristics Evaluation (Outline)

- Landscape analysis
  - Node types analysis
    - ***Minimums and plateaus density and distribution***
    - ***Estimate solution length using node types information***
  - ***Ruggedness Analysis***
- ***Fitness-Distance analysis***
- Transition Matrix based analysis
  - *Markov-chains based estimation of heuristic error for hill-climbing algorithm*
  - ***Estimate expanded nodes and solution distance***

The Landscape

- The distribution of heuristic values in the search space can be seen as a three-dimensional curve



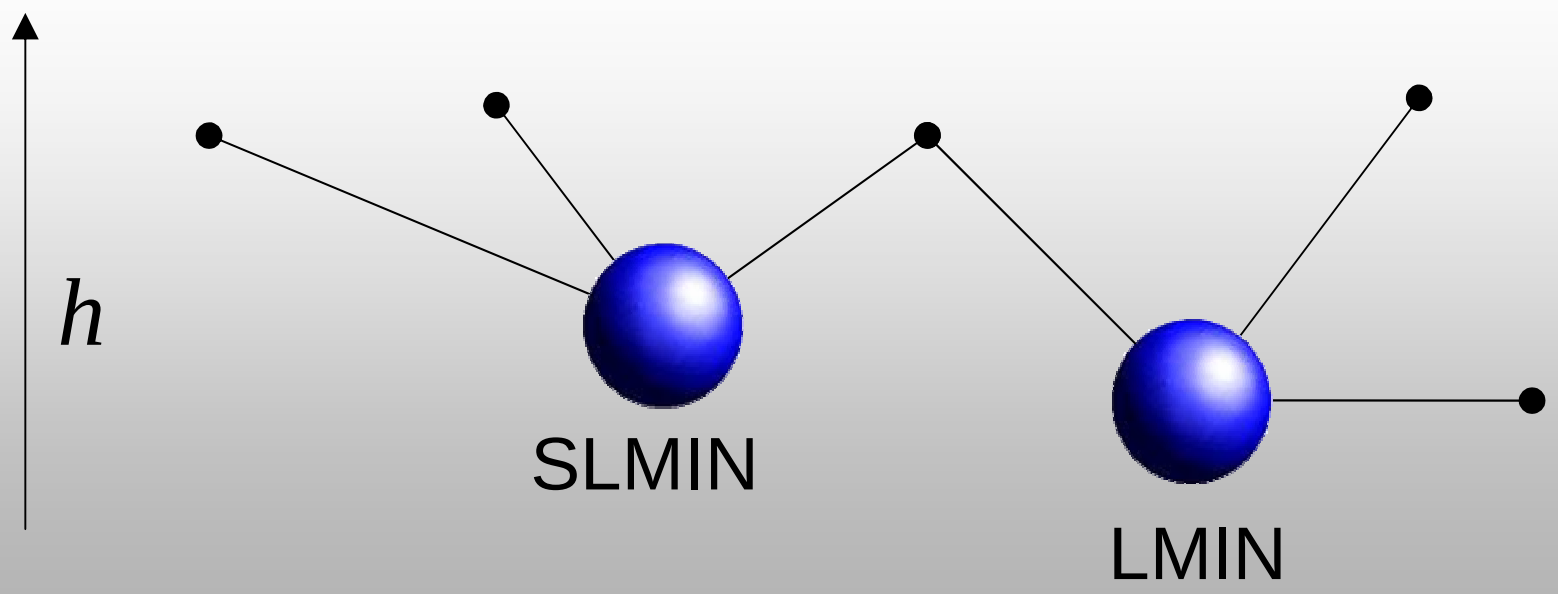
- Some landscape characteristics are correlated with heuristic errors

Local Landscape Analysis

- Classify a node by its heuristic value compared to the one of its children
  - A node can be:
    - local minimum
    - maximum
    - slope
    - edge
    - plateau
- **This method does not require the heuristic function to be admissible**

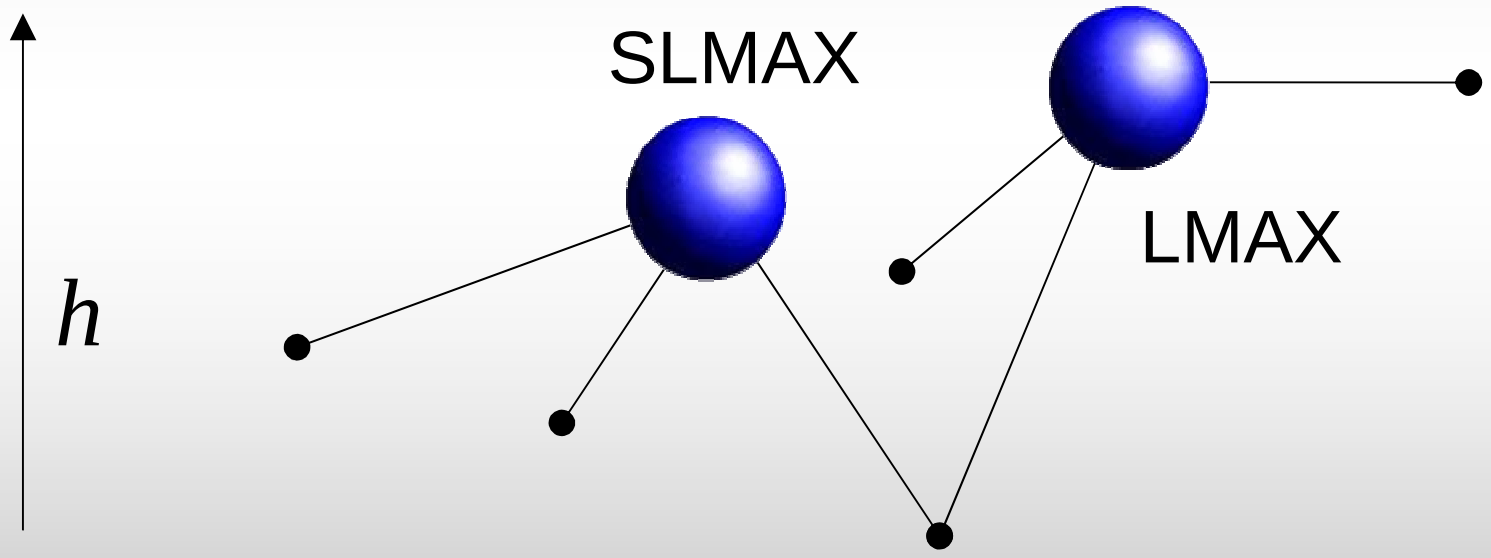
Node Types: SLMIN, LMIN

- *SLMIN* and *LMIN* are local minimums: all children have heuristic value greater (strictly for SLMIN) than the one of its parent
  - *A local minimum denotes a heuristic error*



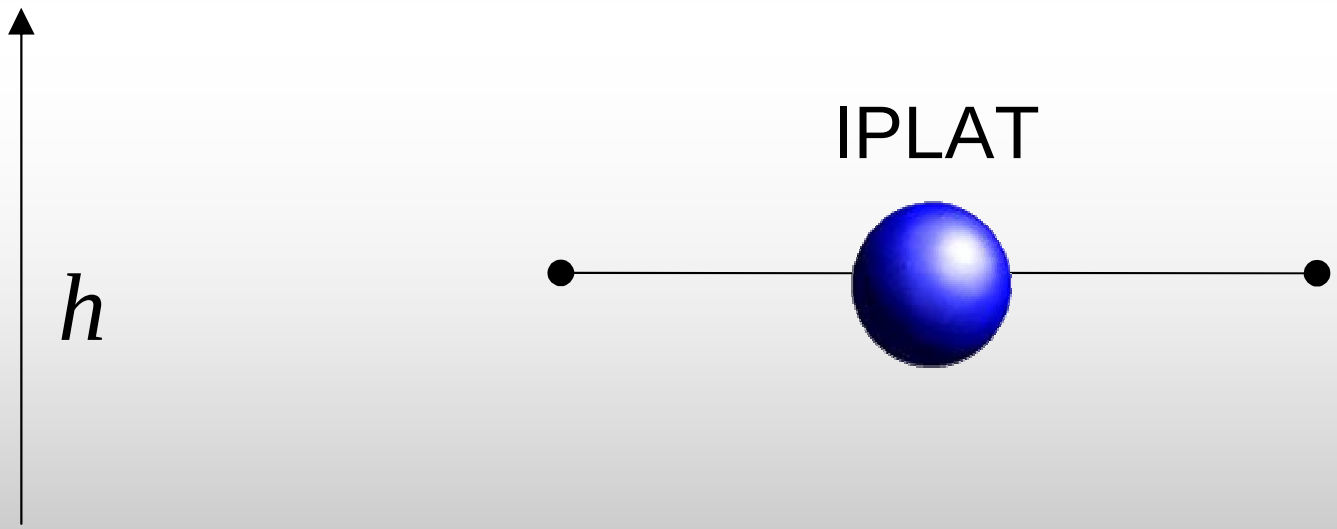
Node Types: SLMAX, LMAX

- A similar definition can be given for maximums



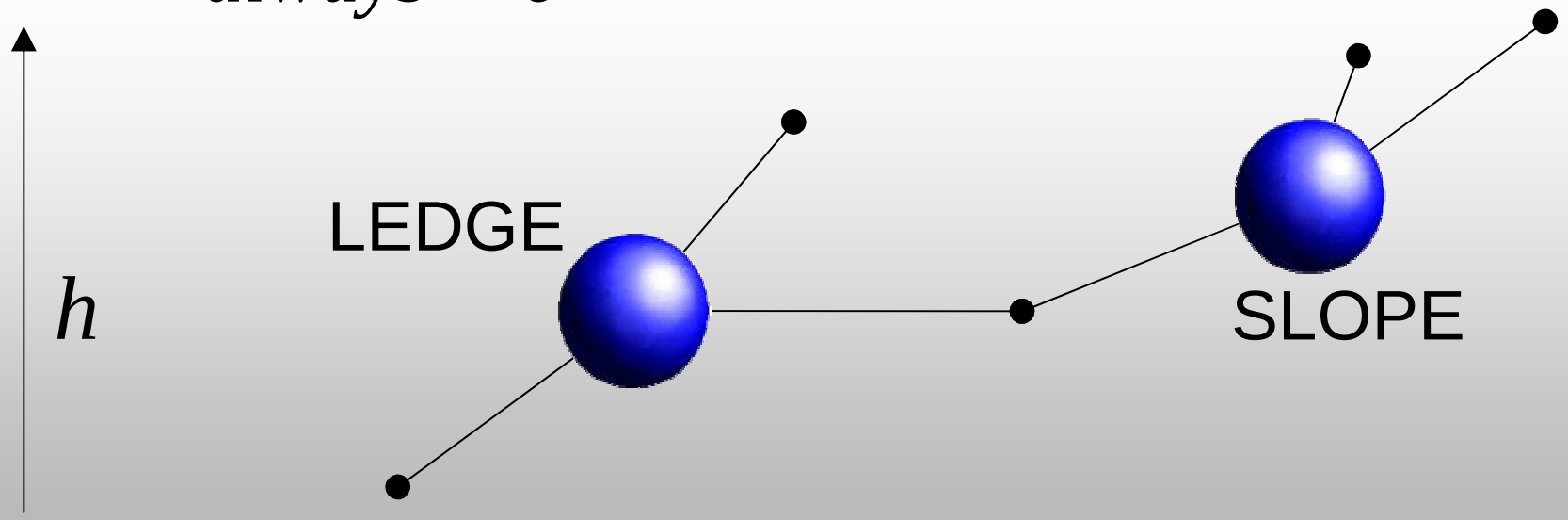
Node types: IPLAT

- *IPLAT* is a node whose heuristic value coincides with the one of all its children
  - *It is a symptom of heuristic local blindness*



Node types: LEDGE, SLOPE

- *LEDGE* and *SLOPE* are passage nodes; they denote good heuristic local behavior
  - *a perfect heuristic has only SLOPE nodes under the assumption of unique solution and edge cost always > 0*

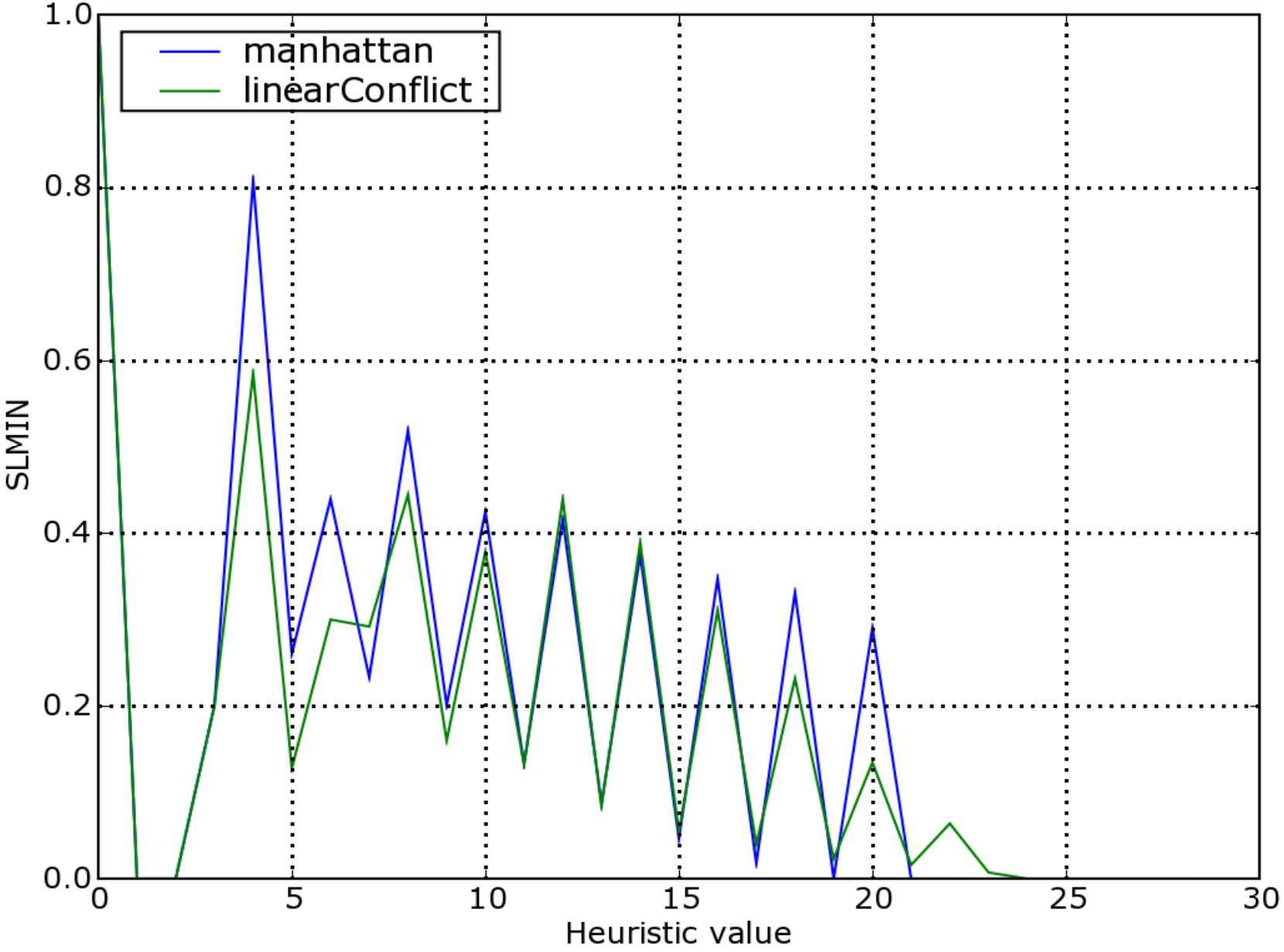


Density and distribution of local minimums and plateaus

- The **density** of minimums and plateaus in the search space may be reasonably taken as a direct measure of heuristic good behaviour

- The most important information is given by the **distribution** of minimums and plateaus

Distribution of SLMIN along heuristic values in 8-puzzle



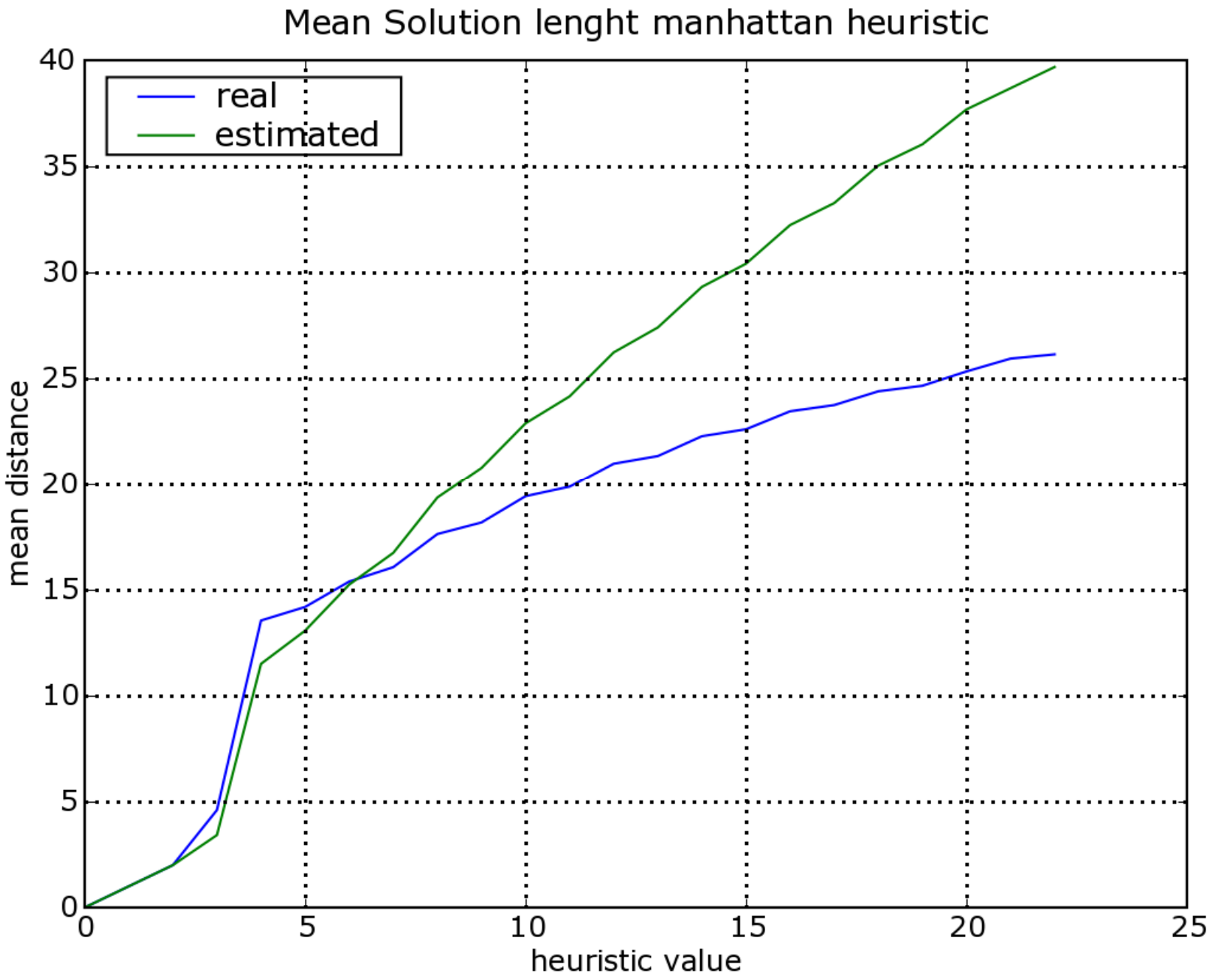
Solution length estimation based on local minimums and plateaus information

- Local minimums and plateaus denote heuristic errors
- We can measure this error:
  - *SLMIN*: equal or greater than 2 steps
  - *LMIN* and *IPLAT*: equal or greater than 1 step
- Estimate distance:

$$d\left(h, p_{SLMIN}\left(h\right)\right) = h + 2 \cdot k_{SLMIN}\left(h\right)$$

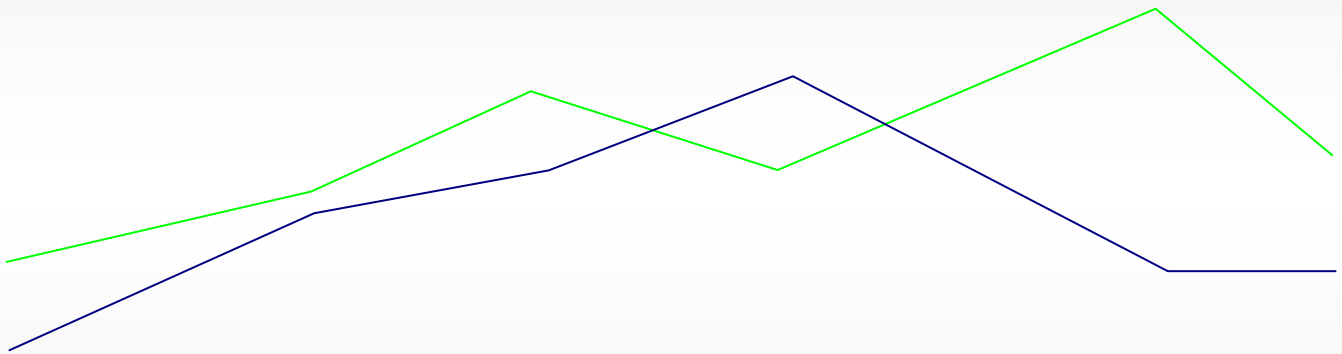
- *k*(*h*) *is a coefficient that depends on distribution of probability values*

Experiments: 8-puzzle



Fitness-Distance Analysis

- Fitness: the distribution on heuristic values
- Distance: the distribution of real solution distance



- Viewing heuristic as a solution distance estimator the ***correlation*** of fitness and distance vectors is a metric of its effectiveness
- It is an *a posteriori* method

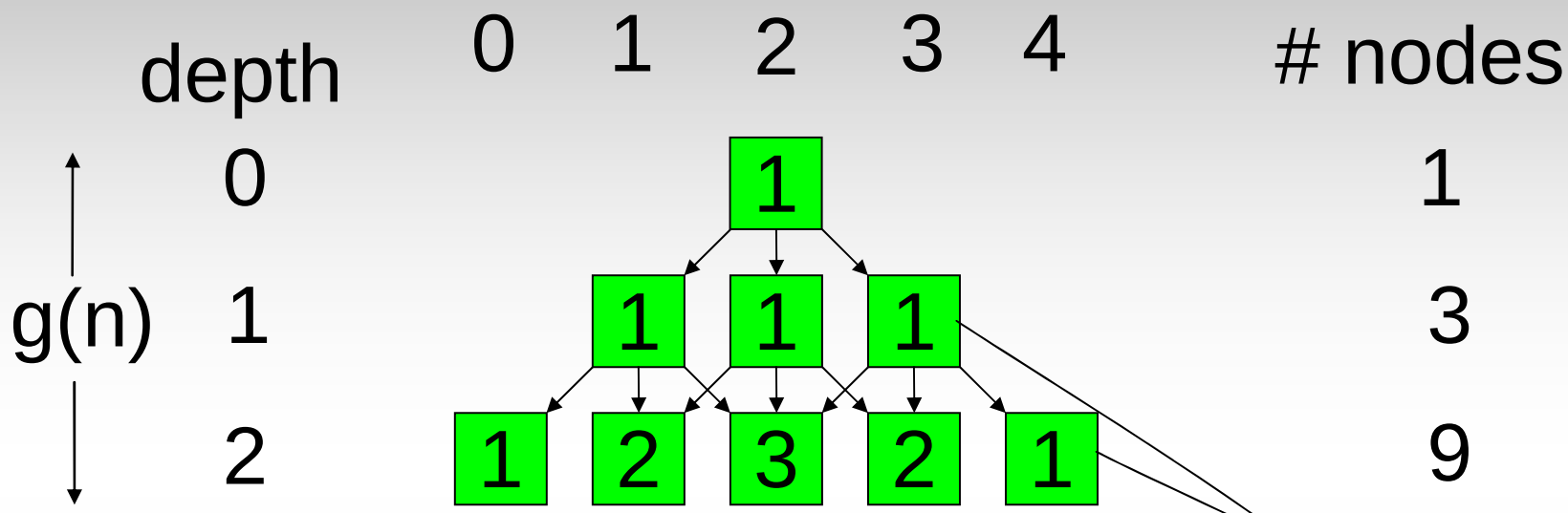
Ruggedness Analysis

- View landscape as a noisy surface
- Hypothesis: the more rugged the surface, the more errors the heuristic makes
- According to our experimental results, not very effective method
  - *Heuristic values distribution has an intrinsic ruggedness that can be confused with the one generated by errors*

Computing Expanded Nodes With Transition Matrix

- Definition: Transition matrix T
  - In T_{ij} the number of cluster j nodes generated by a node of a cluster i on average.
  - Hence, $\sum_j T_{ij}$ is the number of nodes expanded by a breadth-first algorithm at depth 1 by a node of cluster i.
  - Furthermore, $\sum_j T_{ij}^{(n)}$ is the number of nodes expanded at depth n by a node of cluster i

Example: brute-force search



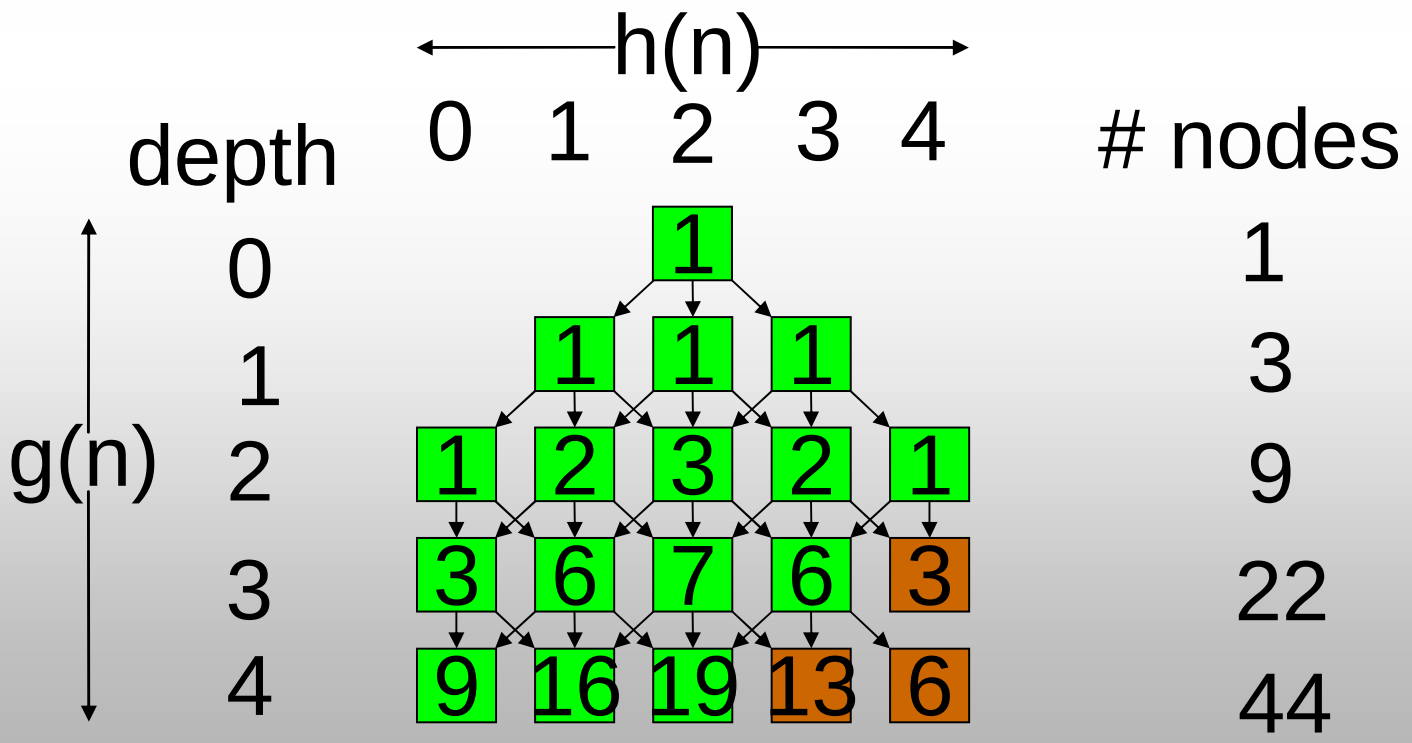
$T^2 =$

0	1	1	0	0	0
1	1	3	1	0	0
2	0	1	3	1	0
3	0	0	1	3	1
4	0	0	0	1	1

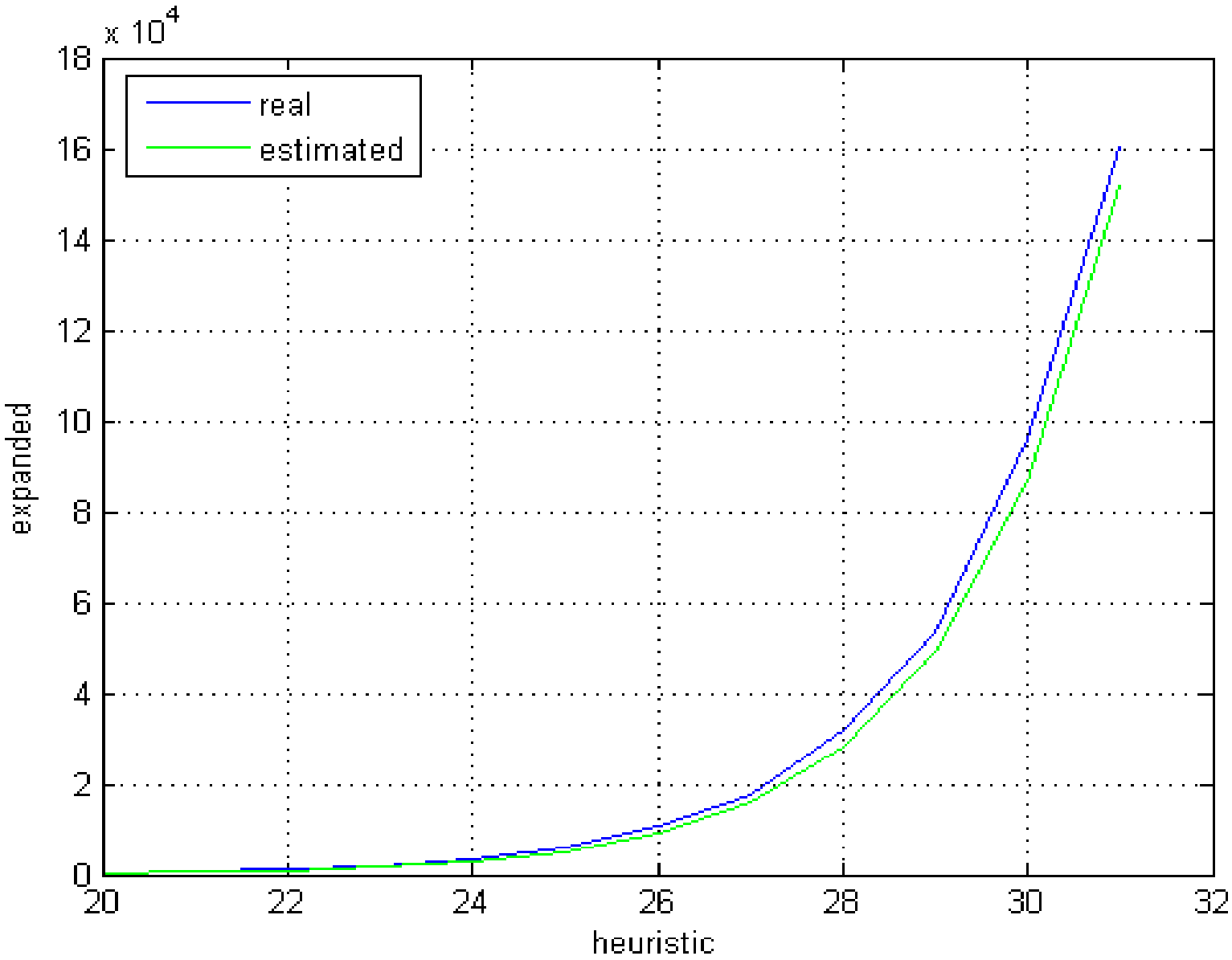
Computing expanded nodes by IDA*

$$N_{\exp}(i,d) = \sum_{k=1}^{d} \sum_{j} T_{ij}(k)$$

- IDA*: the sum does not include nodes s.t. $h+d>l$



Results 8-Puzzle: average expanded nodes



Estimate solution distance

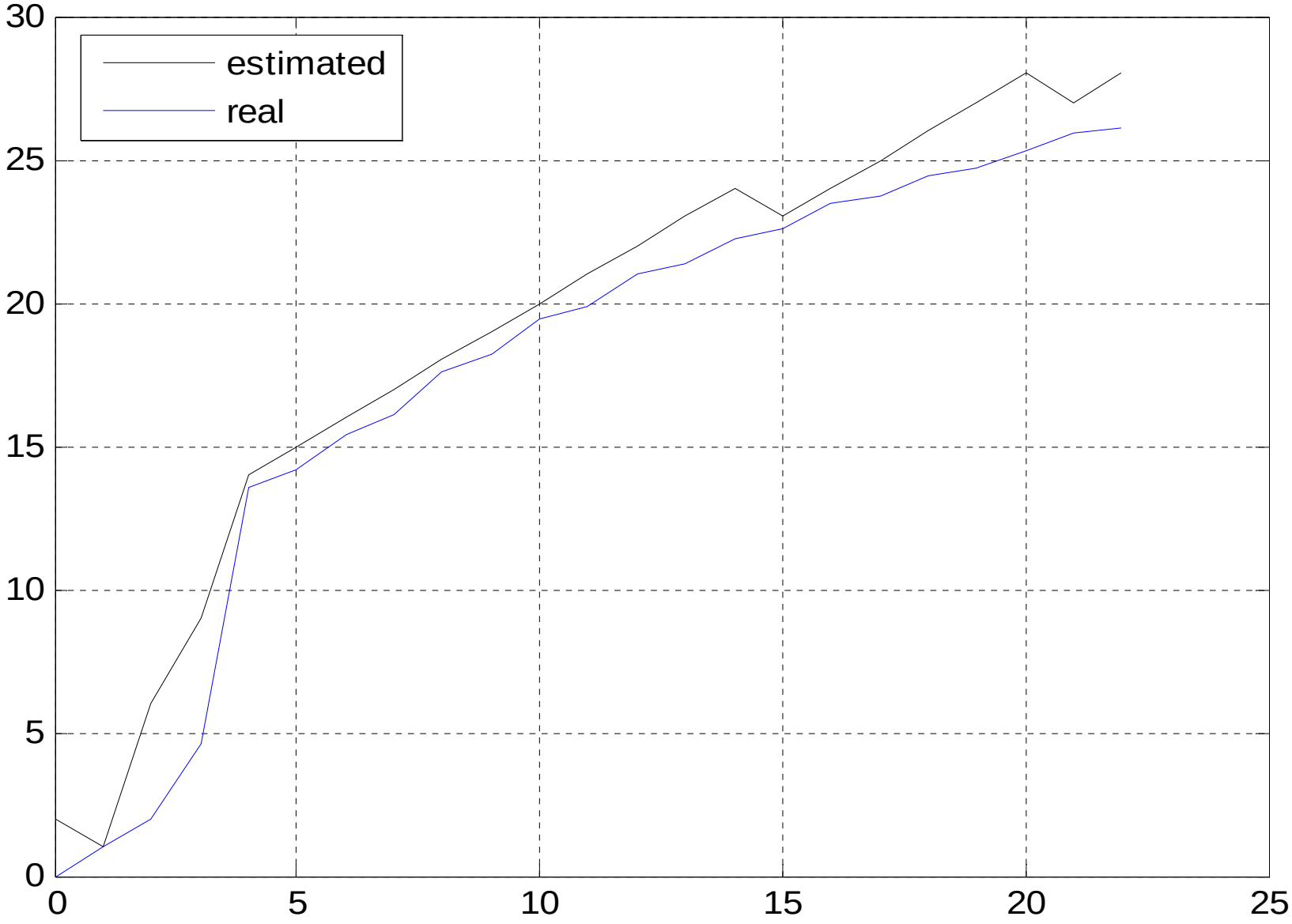
- When $T_{i0}(k) > 0$, we can assume that the goal node has been expanded (if the cluster 0 contains only the goal node)

- On average, one can suppose that a node has been expanded when

$$T_{i0} > 0.5$$

- The algorithm can be stopped when this condition is reached

Results for distance estimation: 8-puzzle



Conclusions

- Many different methods have been investigated, each one with its limitations and drawbacks
- Main questions:
  - What happens when changing domain?
  - Many methods are based on sampling the state space
    - What is the better way to perform sampling?
    - What percent of the search space are we expected to sample?

Future Work

- Testing methods in different domains
- **Practical application**: use heuristic evaluation in real problems for choosing better heuristic
  - *For example, the set of heuristic given by parametric perimeter or look-ahead search, or heuristics discovered by an automatic generator*