Topological Analysis of Admissible Heuristics in IDA*

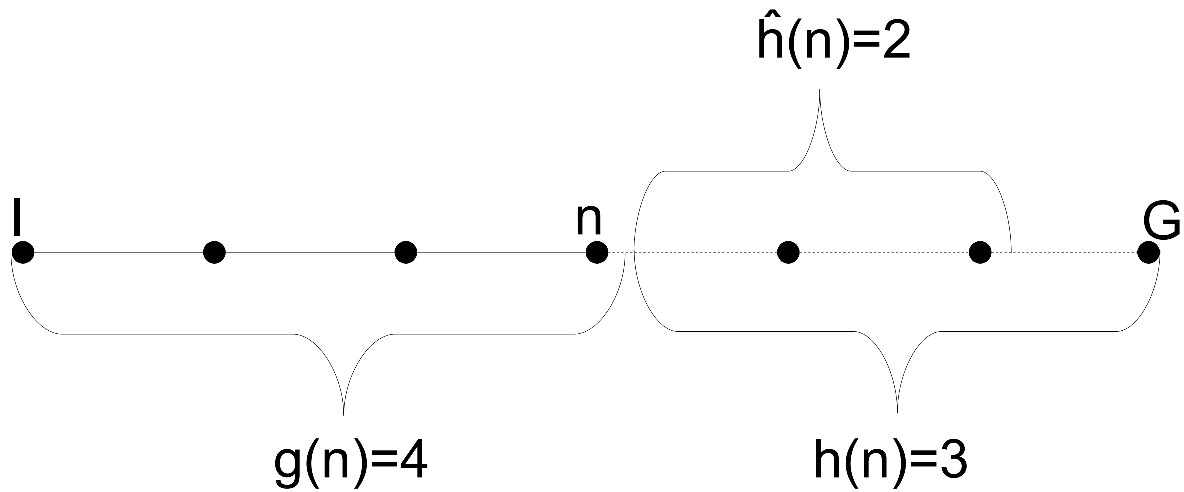
Heuristics

- Heuristics are evaluation functions, $\hat{f}$, used in state space search to decide which node is the best to expand next.
- $\hat{f}(n) = g(n) + \hat{h}(n) = estimated\ optimal\ distance$
- Admissible heuristics: $\forall n \rightarrow \hat{h}(n) \leq h(n)$
- Heuristic search algorithms using admissible heuristics are guaranteed to find the optimal solution (eventually).

Heuristic Informed search

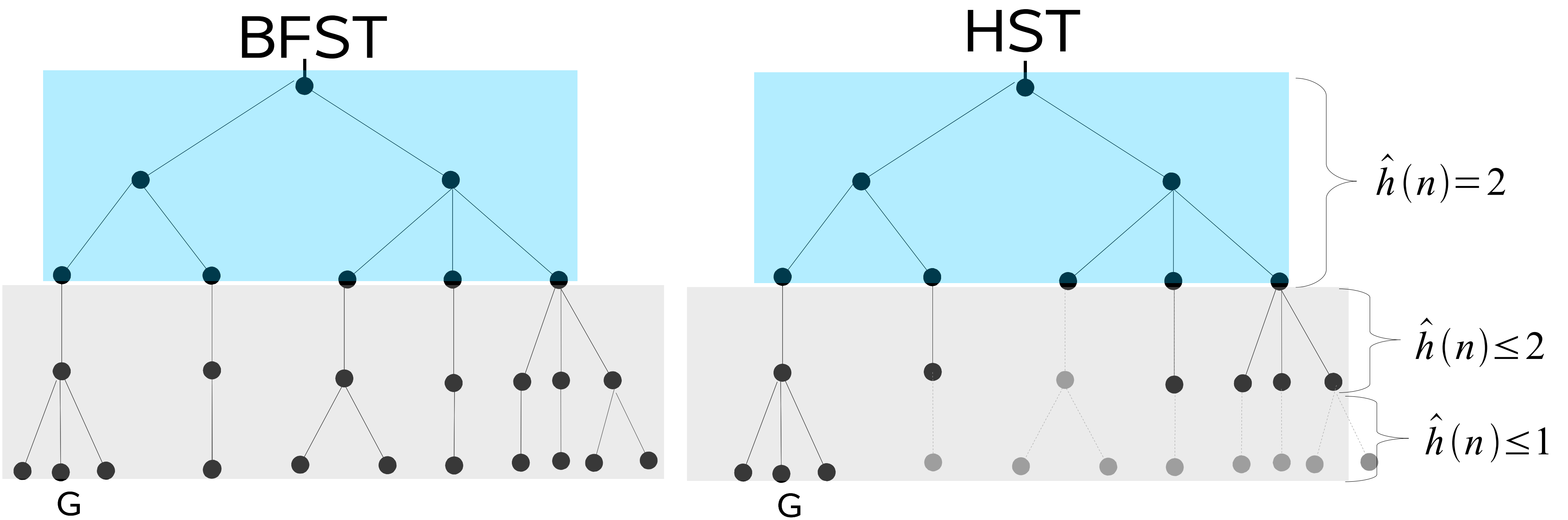
- Heuristic Search generates a Heuristic Search Tree(*HST*) from initial state(I) to goal state (G) using $\hat{f}(n)$.
- Each node in the HST represent a state in the domain, and each of its children represents the result of an action applied to the parent node.
- HST is a sub-tree of the Brute Force Search Tree(BFST) and thus smaller
- The smaller the HST the better is the heuristic.

Example 1



- I is initial state, G is Goal state, n is current state.
- g(n) is the Optimal Distance(*OD*) from I to n
- h(n) is the OD from n to G
- $\hat{h}$ is the heuristic estimate OD from n to G.
- f(n)=g(n)+h(n)=7=OD.
- $\hat{f}(n)=g(n)+\hat{h}(n)=6=estimated\ heuristic\ OD$

BFST vs HST

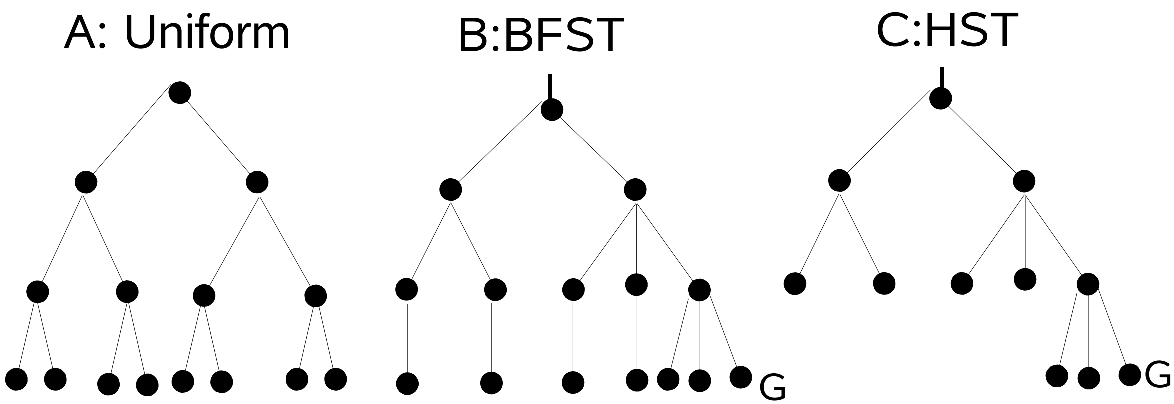


- $max(\hat{h}(n))=3$;OD=4; $\{\forall n | \hat{f}(n)>4\}$ *will be culled*
- Blue area represents "out of shadow". Same as BFST.
- Grey area represents "heuristic shadow" area. Some nodes culled.
- Heuristic shadow is a very common phenomenon.

Goals

- Given:
  - An admissible heuristic
  - A problem instance
  - A search based problem solver
- We want to be able to predict the size of the search tree generated to find the optimal solution.
- We need to determine how the HST grows.

Example 2



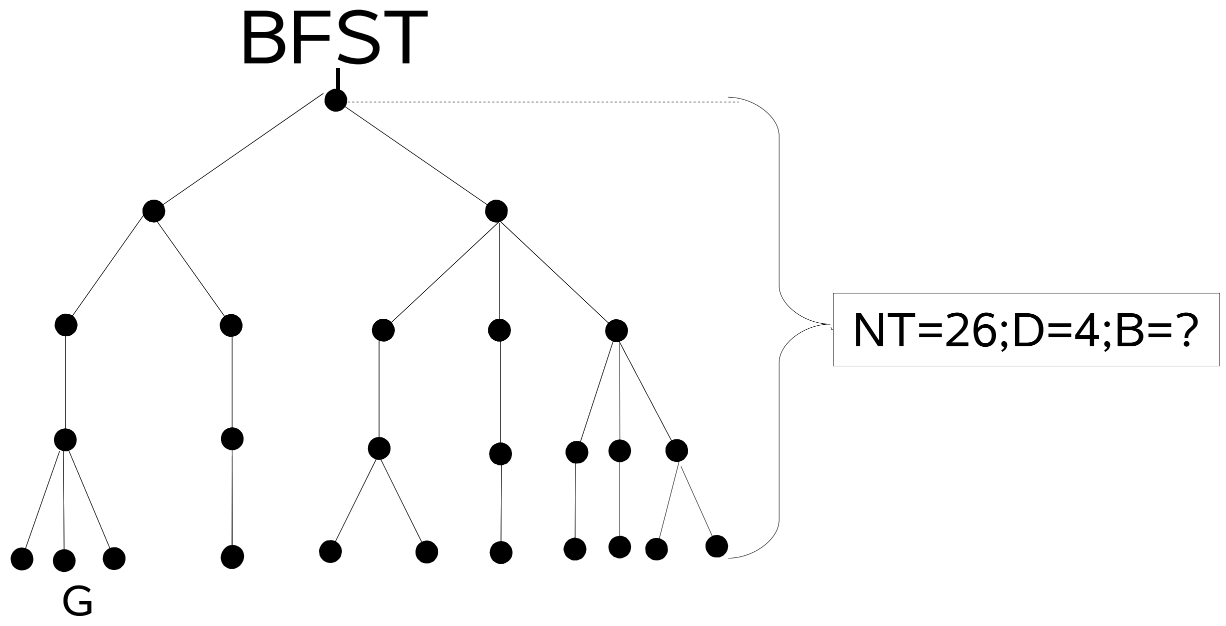
$$N_T = \frac{B^{D+1} - 1}{B - 1}$$

Eq. 1; N_T=nodes created;
B=branching factor;D=depth

Problem Evaluation

- To characterize the size reduction associated to a heuristic is not trivial.
- Uniform search tree model being used. Possible Variants are Depth and Branching Factor.
- Effective Branching Factor (EBF) is used on Eq 1 for size of BFST.

Effective Branching Factor



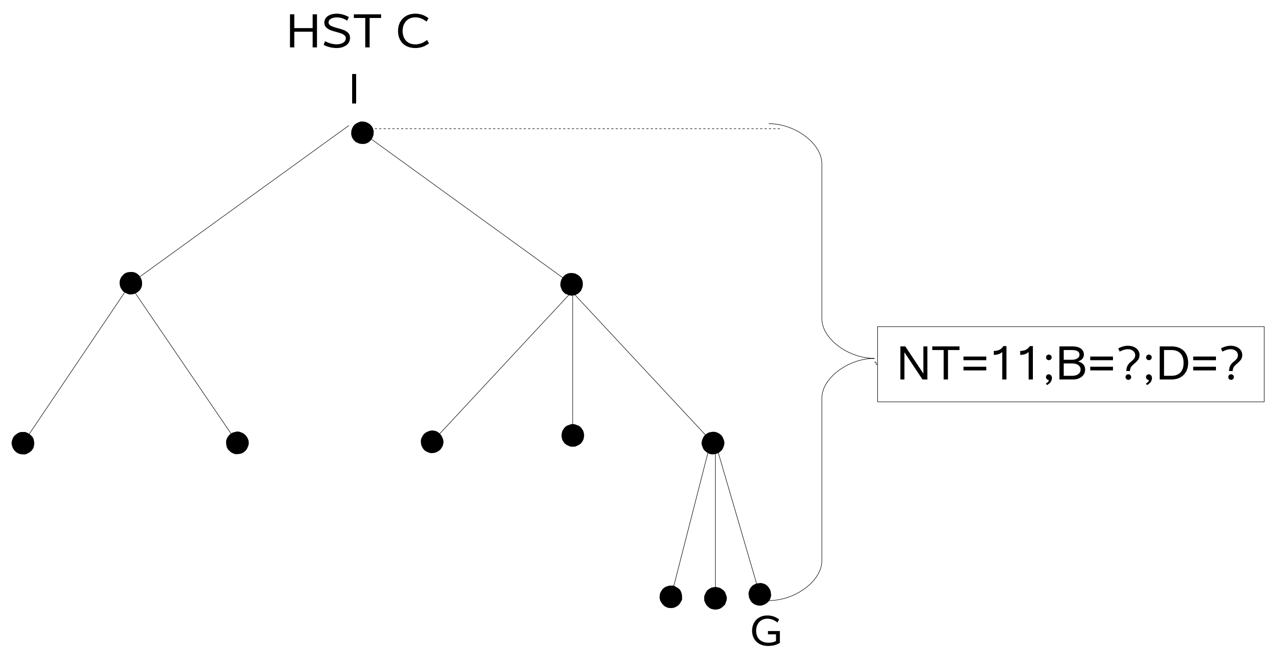
$$N_T = \frac{EBF^{D+1} - 1}{EBF - 1} \rightarrow EBF = 1.8625$$

Eq. 2; N_T=nodes created;
EBF=branching factor;D=depth

EBF Example

- EBF is an attempt at finding features which are stable as the tree grows.
- If EBF is stable then we should be able to predict the size of the BFST of any depth without expanding it.
- BFST are very expensive to grow pass a certain depth.

What about the HST?



$$N_T = \frac{B^{D+1} - 1}{B - 1}; N_T = 11; B = ?, D = ?$$

2 Schools on reduction approach

- Which is the variant from BFST to HST?
- Text-Book standard(Nilson, Russell):
  - $\mathbf{EBF_{HST} \leq EBF_{BFST.}}$
  - Depth is fixed to the Optimal Distance.
- Korf:
  - $\mathbf{EBF_{HST} = EBF_{BFST}}$
  - Effective Depth is reduced.
- Both of them gather statistics across the problem domain for different depths.

Text-Book school: issues

- In order to gather statistics for different problem instances we need to solve a significant amount of problem instances. We use heuristics because this problems are not easy to solve.
- Counter-intuitive to say depth is constant. Whenever a HST node is culled the path depth becomes smaller.
- Korf technique based on depth reduction more plausible.

Korf approach: issues

- Korf technique requires formula for BFST topology plus a Heuristic Value Distribution. That requires costly statistics on the domain again.
- Korf claims EBF does not change from BFST to HST. But node types frequency may change. EBF is still a potential variant.

Issues for Both Approaches

- Statistics on domain average out problem instances differences.
- Both claim that EBF can not increase from BFST to HST. We will proof this is not the case.

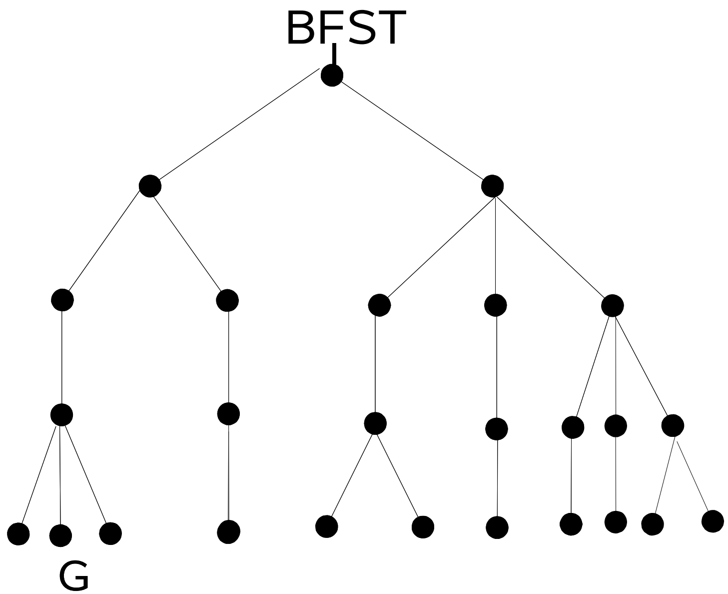
Goal is impossible

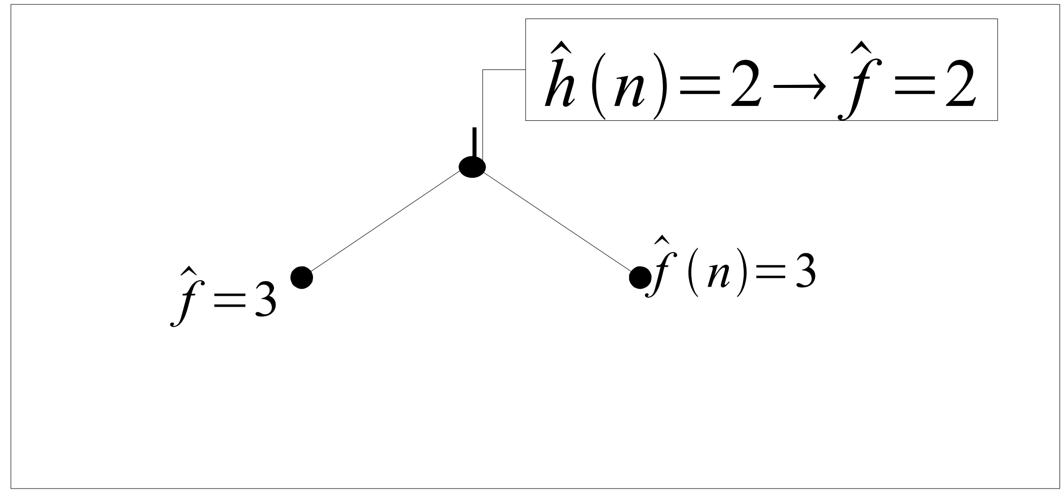
- Even if we have formula describing HST size, it is a function of Optimal Distance.
- We do not know the Optimal Distance until we solve the problem!
- Best next Goal: Predict the size of a depth bounded HST.
- IDA* iterative nature is the answer.

Korf's IDA*

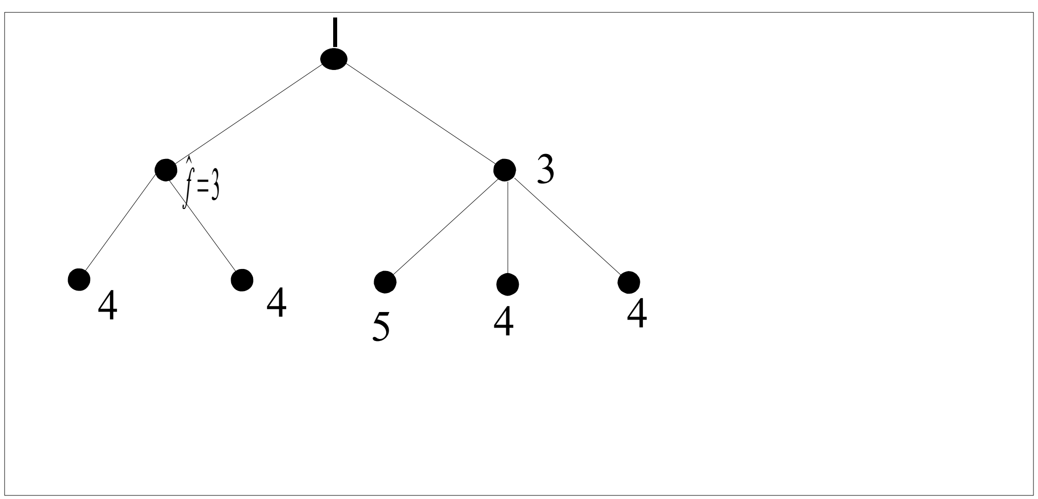
- IDA* is a linear-space version of A*.
- It performs a series of depth first searches, pruning a path and backtracking when the cost, $\hat{f}(n)$, of a node n on the path exceeds a bound C for that iteration.
- The initial bound C_o is set to the heuristic estimate of the initial state, and increases in each iteration to the lowest cost of all the nodes pruned on the last iteration, until a goal node is expanded.
- IDA* guarantees an optimal solution if the heuristic function is admissible.

IDA* iterations

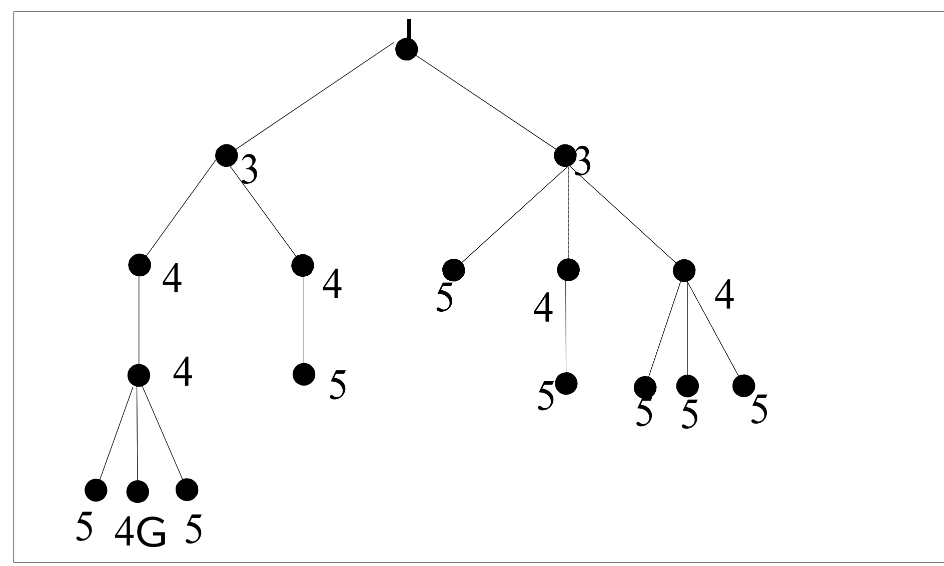




$$\{\forall n \mid \hat{f}(n)>2\}\ will\ be\ culled$$



$$\{\forall n \mid \hat{f}(n)>3\}\ will\ be\ culled$$



$$\{\forall n \mid \hat{f}(n)>4\}\ will\ be\ culled$$

New Goal

- Given:
  - An admissible heuristic
  - A problem instance
  - IDA*
- We want to be able to predict the depth-bounded size of the HST generated to find the optimal solution.

Proposed Model

- The HST for any iteration is a subtree of the corresponding BFST. So we can describe the size of the bounded HST as a function of the pruning of the BFST.
- EBF,ED are variants. First two iterations for solving system of eq:
- $N_T = \frac{(EBF_{BFST} - EBFR)^{\hat{f} - EDR + 1} - 1}{EBF_{BFST} - EBFR - 1}$
- $f(EBFR, EDR) = \frac{\ln(N_T * (EBF - EBFR - 1) + 1) - 1}{\ln(EBF - EBFR)} - \hat{f} - EDR + 1 = 0$

Example for Eight Puzzle, OD=20

- 3 heuristics
  - Out of place
  - Manhattan
  - Relaxed Adjacency
- Comparing:
  - EDR=0 -->Text-book approach
  - EBFR=0-->Korf-like approach
  - 2 iterations, EDR,EBFR unknown.

BFST for OD 20

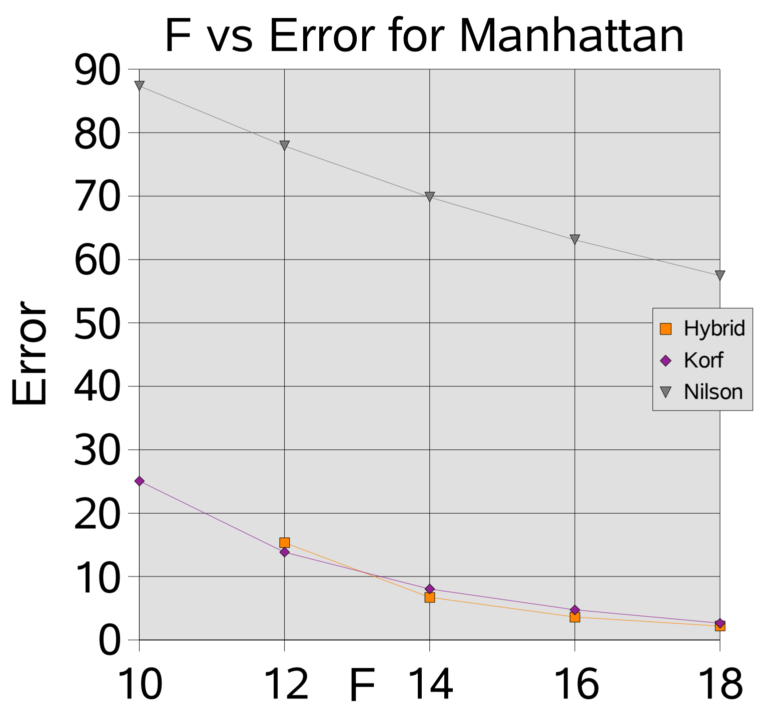
F	Existing	Predicted	Future	Error %	EBFR	EBF_BFST	ED_BFST
2	6	22.04	48	54.08	1.04	1.79	2
4	48	254.87	384	33.63	0.54	2.29	4
6	384	2353.45	3072	23.39	0.35	2.47	6
8	3072	20218.48	24576	17.73	0.26	2.57	8
10	24576	168601.16	196608	14.25	0.21	2.62	10
12	196611	1385682.63	1.57E+006	11.9	0.17	2.65	12
14	1572890	11297049.48	1.26E+007	10.22	0.15	2.68	14
16	12583100	91649952.24	1.01E+008	8.95	0.13	2.7	16
18	102679000	757696506.78	8.05E+008	5.91	0.11	2.72	18

$$EBF_{ASYMPTOTIC} = \sqrt{8} \approx 2.8284$$

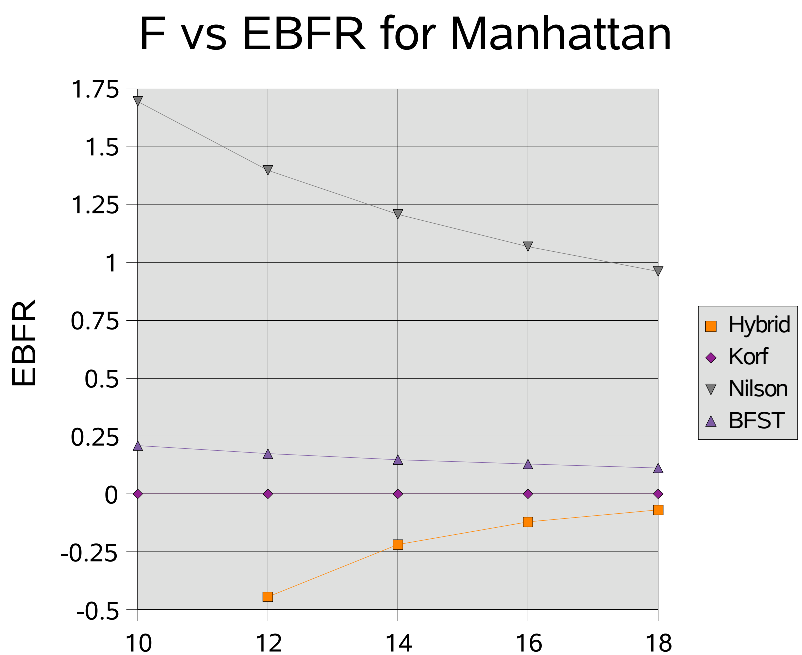
Manhattan

F	#Nodes Created IDA*
10	22
12	240
14	2233
16	19434
18	163202
20	1340980

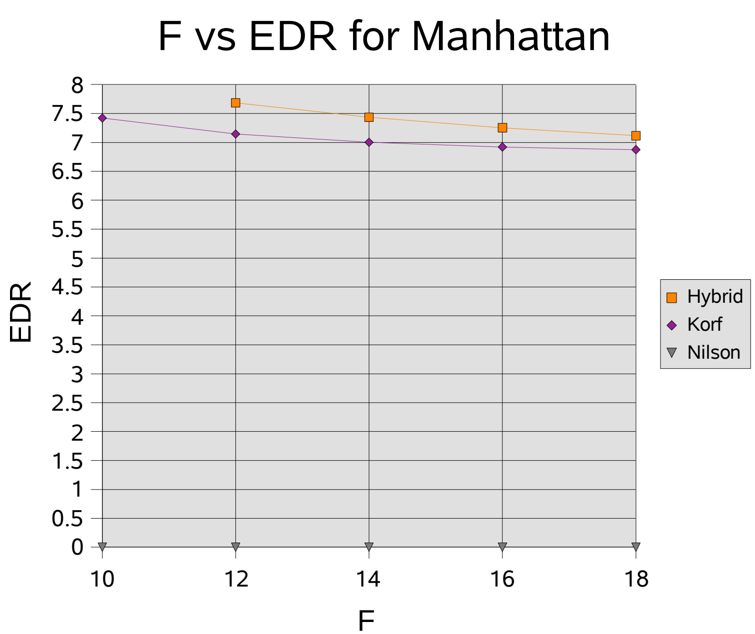
Manhattan



F vs Error for Manhattan
90
80
70
60
50
40
30
20
10
0
Error
10
12
F
14
16
18
Hybrid
Korf
Nilson



F vs EBFR for Manhattan
1.75
1.5
1.25
1
0.75
0.5
0.25
0
-0.25
-0.5
EBFR
10
12
14
16
18
Hybrid
Korf
Nilson
BFST



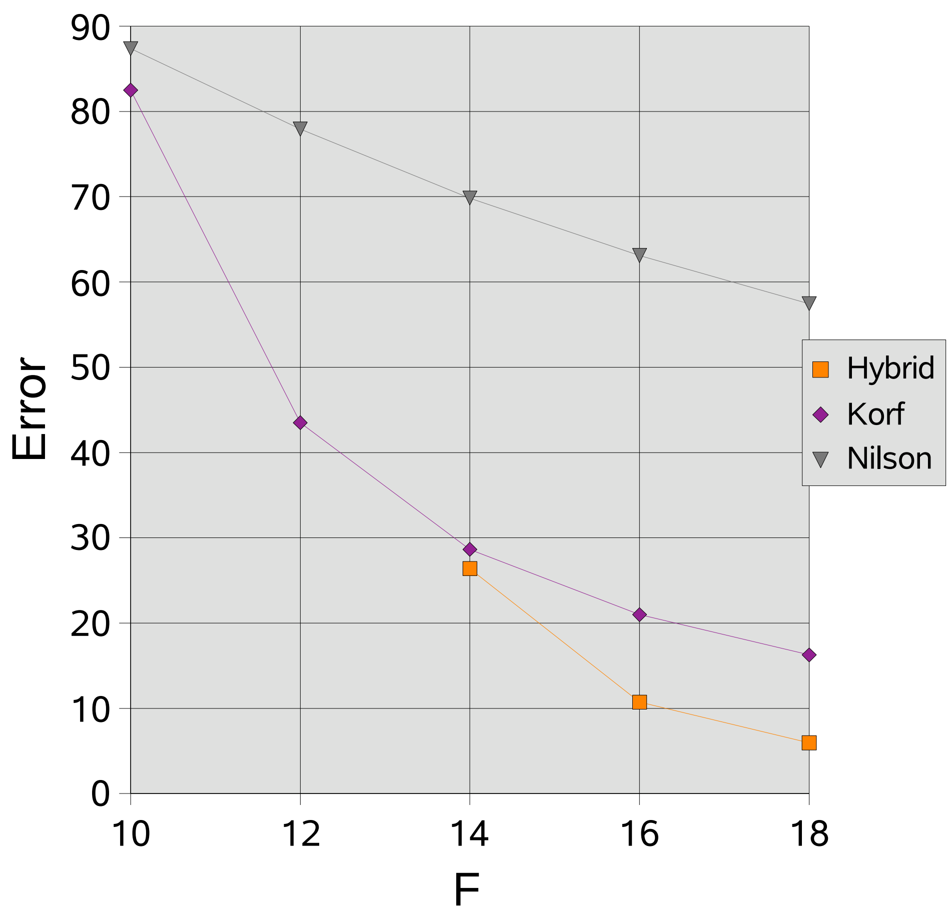
F vs EDR for Manhattan
8
7.5
7
6.5
6
5.5
5
4.5
4
3.5
3
2.5
2
1.5
1
0.5
0
EDR
10
12
14
16
18
F
Hybrid
Korf
Nilson

Relaxed Adjacency

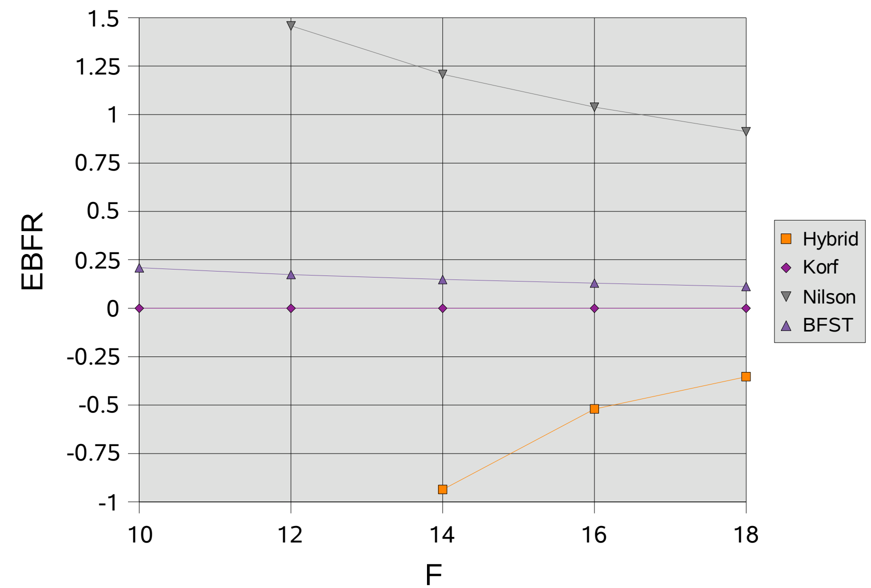
F	#Nodes Created IDA*
10	3
12	159
14	2258
16	25321
18	256379
20	2449791

Relaxed Adjacency

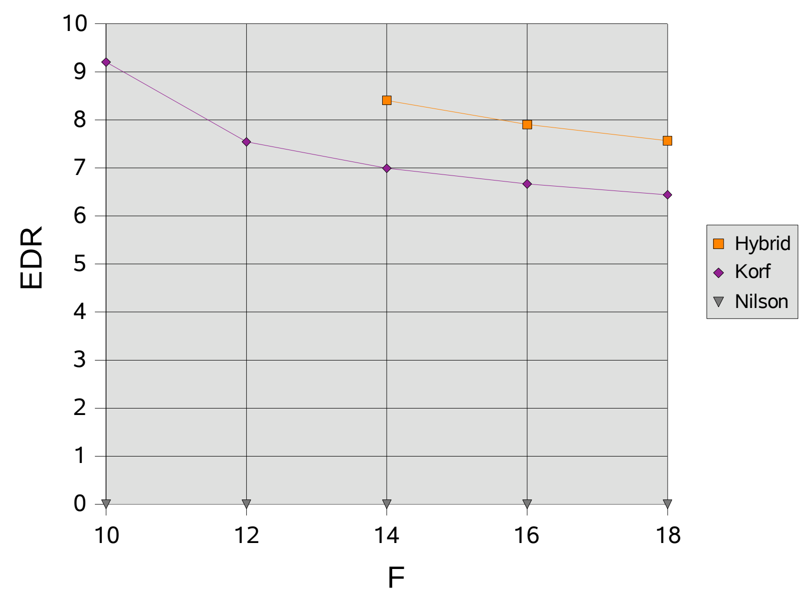
F vs Error for Relaxed Adjacency



F vs EBFR for Relaxed Adjacency



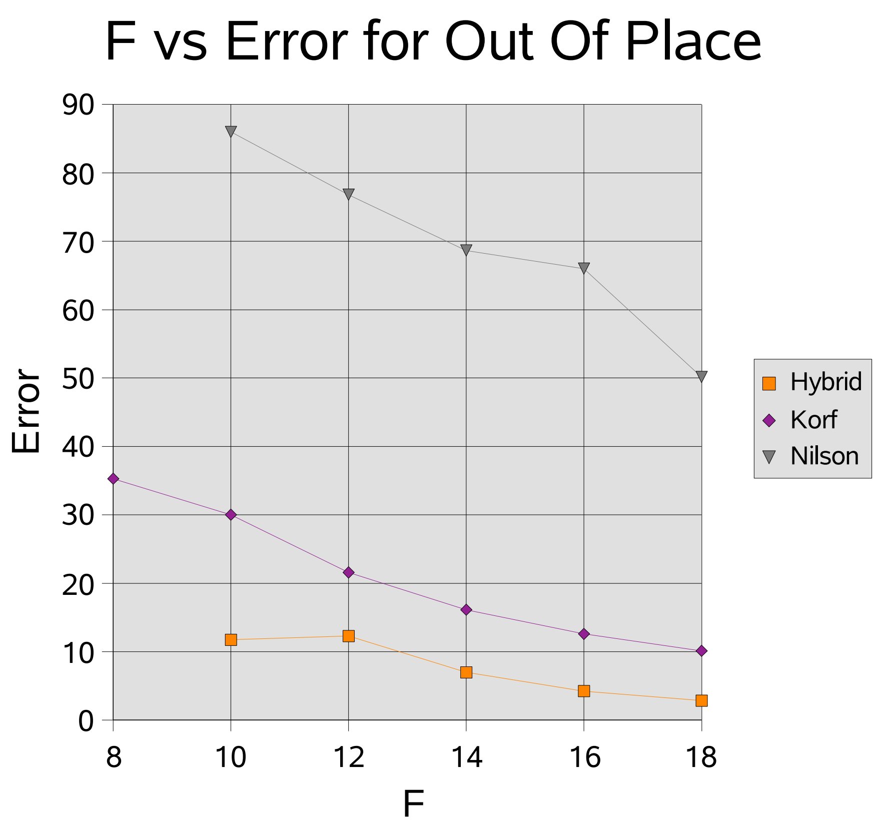
F vs EDR for Relaxed Adjacency

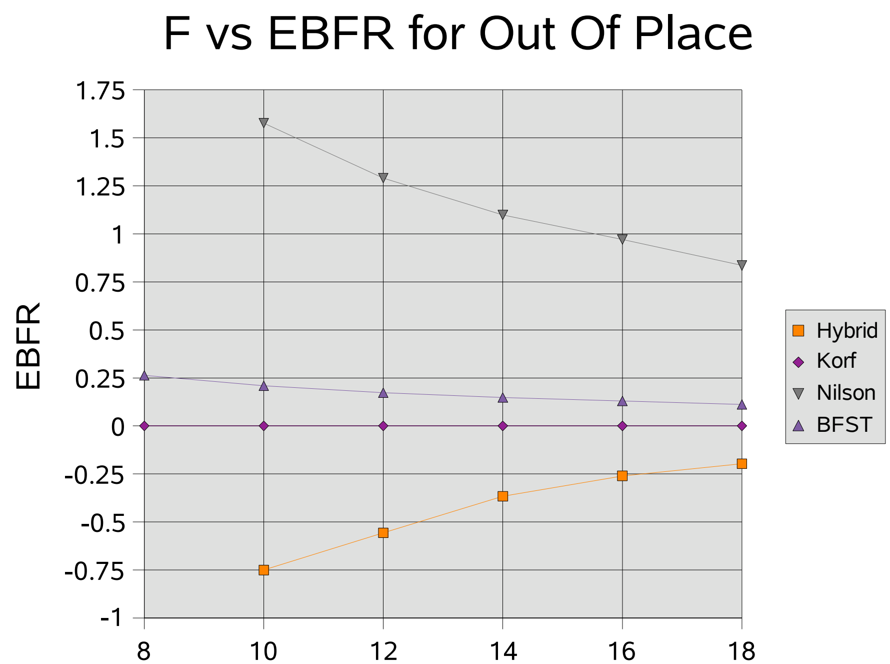


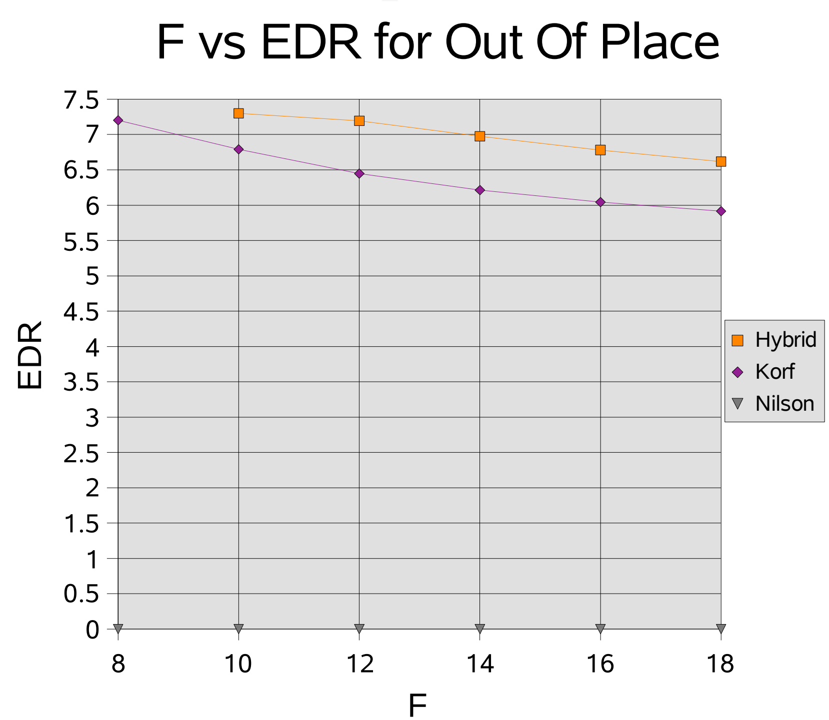
Out Of Place

F	#Nodes Created IDA*
8	3
10	43
12	497
14	5075
16	48410
18	443029
20	3943141

Out Of Place


F vs Error for Out Of Place
Error
F
Hybrid
Korf
Nilson
90
80
70
60
50
40
30
20
10
0
8
10
12
14
16
18


F vs EBFR for Out Of Place
EBFR
Hybrid
Korf
Nilson
BFST
1.75
1.5
1.25
1
0.75
0.5
0.25
0
-0.25
-0.5
-0.75
-1
8
10
12
14
16
18


F vs EDR for Out Of Place
EDR
F
Hybrid
Korf
Nilson
7.5
7
6.5
6
5.5
5
4.5
4
3.5
3
2.5
2
1.5
1
0.5
0
8
10
12
14
16
18

Statistic approach Avg BFST vs Instance BFST

F	Avg Russell	Instance	Error %
2	10	6	-66.67
4	112	48	-133.33
6	680	384	-77.08
8	6384	3072	-107.81
10	47127	24576	-91.76
12	364403	196608	-85.34

3 Claims

- Hybrid model makes better prediction of bounded HST than existing approaches.
- EBF can increase from BFST to HST.
- Statistical models deviate too much from individual problem instances to be useful.

Future

- Selection of generated heuristics for individual problem instances.
- Calculate time savings associated to the use of a heuristic on a individual instance.
- Extension of our approach to other search algorithms like A*.