GBFHS: A generalized breadth-first heuristic search algorithm

Artificial Intelligence

Sam Chen

Introduction

- AI Navigation System
- Path Problems
- Bidirectional Heuristic Algorithm
- MM, MMe, NBS
- What is GBFHS ?
- Distinguish from associated topics

Objective & The General Problem

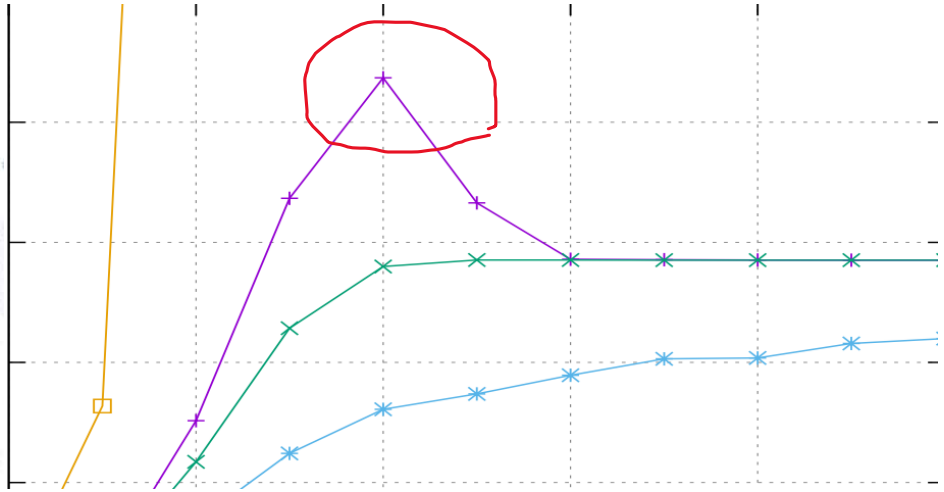
"Hump-In-The-Middle Phenomenon"[1]

"Ill-Behaved & Well-Behaved"[1]

Previous Works

- Nicholson's
- MM&Mme&MM0
- NBS

Figure 1: "Hump-In-the-Middle" Phenomenon
Source: Adapted from [1]



As demonstraded in [1],pr(n) = max{2g(n), f(n)}

The GBFHS Algorithm

Algorithm 1 GBFHS(I,G,ϵ, $split$)→ optimalSolutionCost

```
1:  if trivially solved then
2:      return(0)
3:  best ← unsolvable
4:  open_F ← {(I,0)}, open_B ← {(G, 0)}
5:  closed_F ← closed_B ← ∅
6:  for fLim from max(h_F(I), h_B(G), ε) up by 1 until
    open_F = ∅ ∨ open_B = ∅ do
7:      if best = fLim then
8:          return(best)
9:      gLSum ← fLim − ε + 1
10:     gLim_F, gLim_B ← split(gLSum, gLim_F, gLim_B)
11:     expandLevel(gLim_F, gLim_B, fLim)
12:     if best = fLim then
13:         return(best)
14: return(best)
```

Figure2: Pseudo-code of GBFHS
Source: Adapted from [1]

Algorithm 2 expandLevel($gLim_F$, $gLim_B$, $fLim$)

```
1:  expandable_F ← {n | n ∈ open_F ∧ isExpandable(n,F)}
2:  expandable_B ← {n | n ∈ open_B ∧ isExpandable(n,B)}
3:  while expandable_F ≠ ∅ ∧ expandable_B ≠ ∅ do
4:      n ← pick(expandable_F ∪ expandable_B)
5:      dir ← direction(n)
6:      expandable_dir ← expandable_dir \ {n}
7:      move(n, open_dir, closed_dir)
8:      for all c ∈ expand(n, dir) do
9:          if c ∈ open_dir ∪ closed_dir ∧ g_dir(n) + cost(n, c) ≥
            g_dir(c) then
10:             continue
11:         if c ∈ open_dir ∪ closed_dir then
12:             remove(c, open_dir ∪ closed_dir)
13:         g_dir(c) ← g_dir(n) + cost_dir(n, c)
14:         add(c, open_dir)
15:         if isExpandable(c, dir) then
16:             add(c, expandable_dir)
17:         if c ∈ open_opp(dir) then
18:             best ← min(best, g_dir(c) + g_opp(dir)(c))
19:             if best ≤ fLim then
20:                 return
21: return
```

Figure3: Pseudo-code of expandLevel
Source: Adapted from [1]

Theorems Behind GBFHS

Sufficient Condition for GBFHS Node Expansion

Sufficient Condition for GBFHS(F) Node Expansion

In Unit-Cost Domains, GBFHS can stop on First Collision

GBFHS is Admissible

GBFHS is well-behaved

Experiments

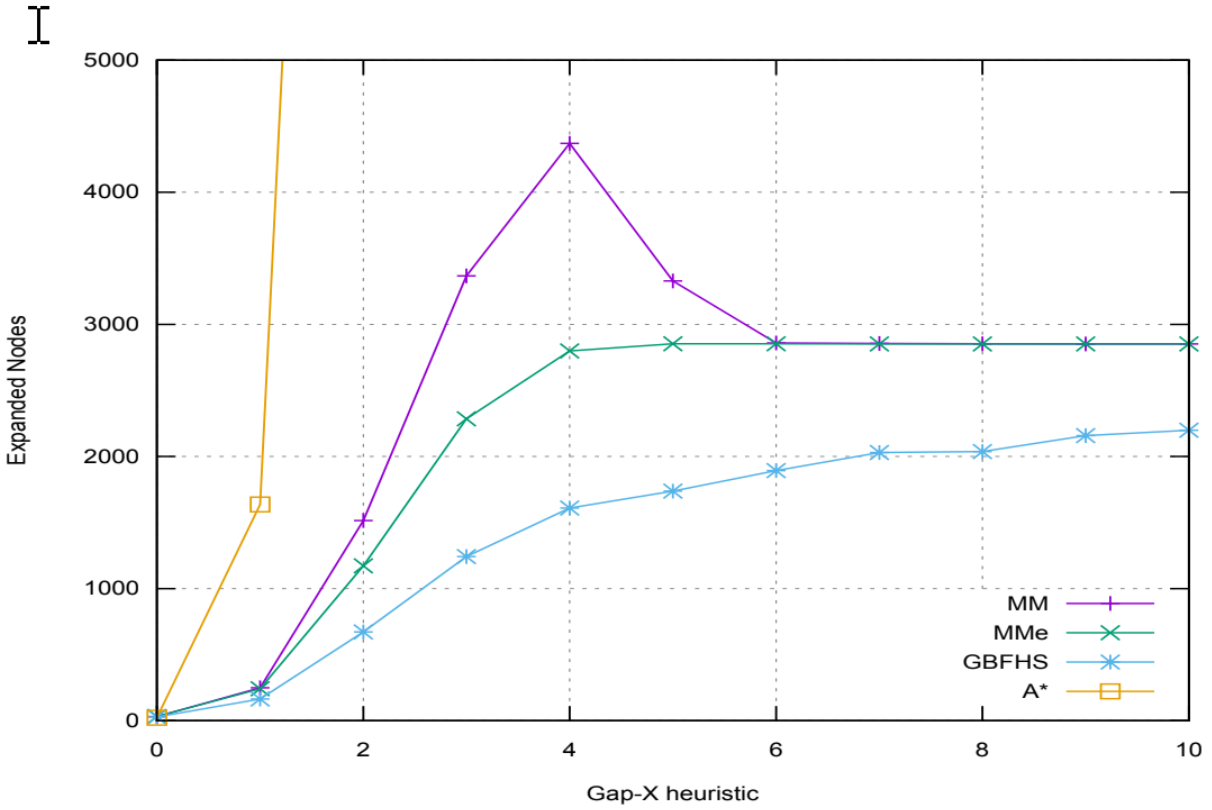


Figure 4: Average number of nodes expanded for A*, MM, MMe, and GBFHS on 50 random 10-pancake problems as the heuristic accuracy degrades
Source: Adapted from [1]

Analysis 1: GBFHS vs A*

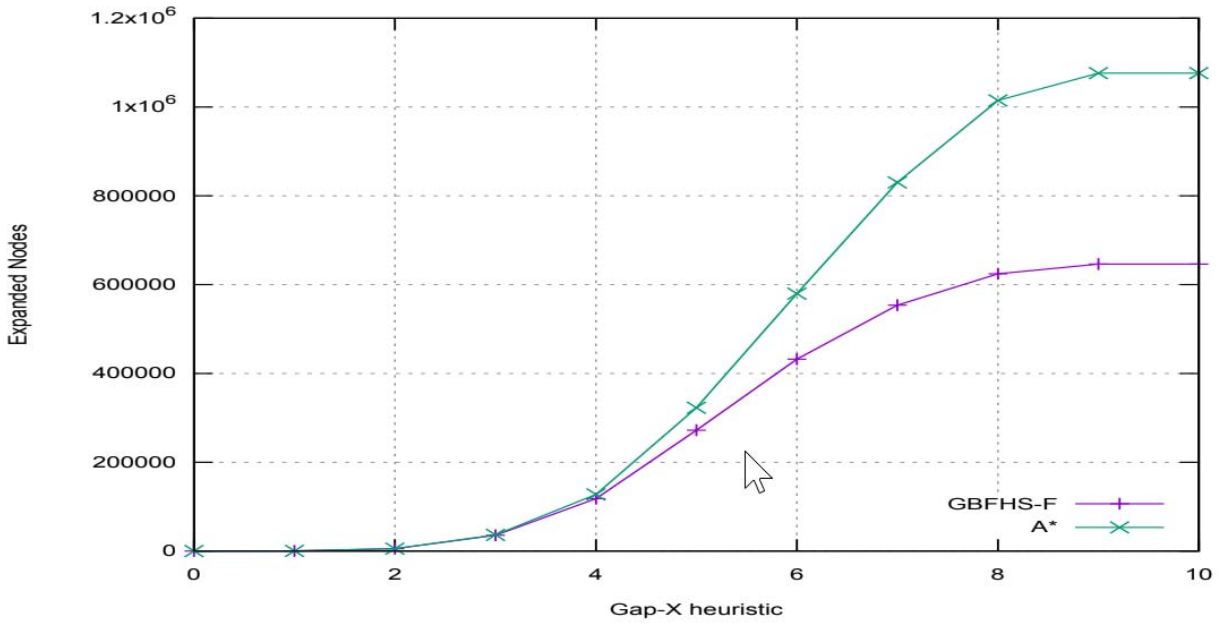


Figure 5: Average Nodes Expanded for A8 and GBFHS on 50 Random 10-pancake problems as the heuristic accuracy degrades
Source: Adapted from [1]

Analysis 2: GBFHS vs MM/MMe

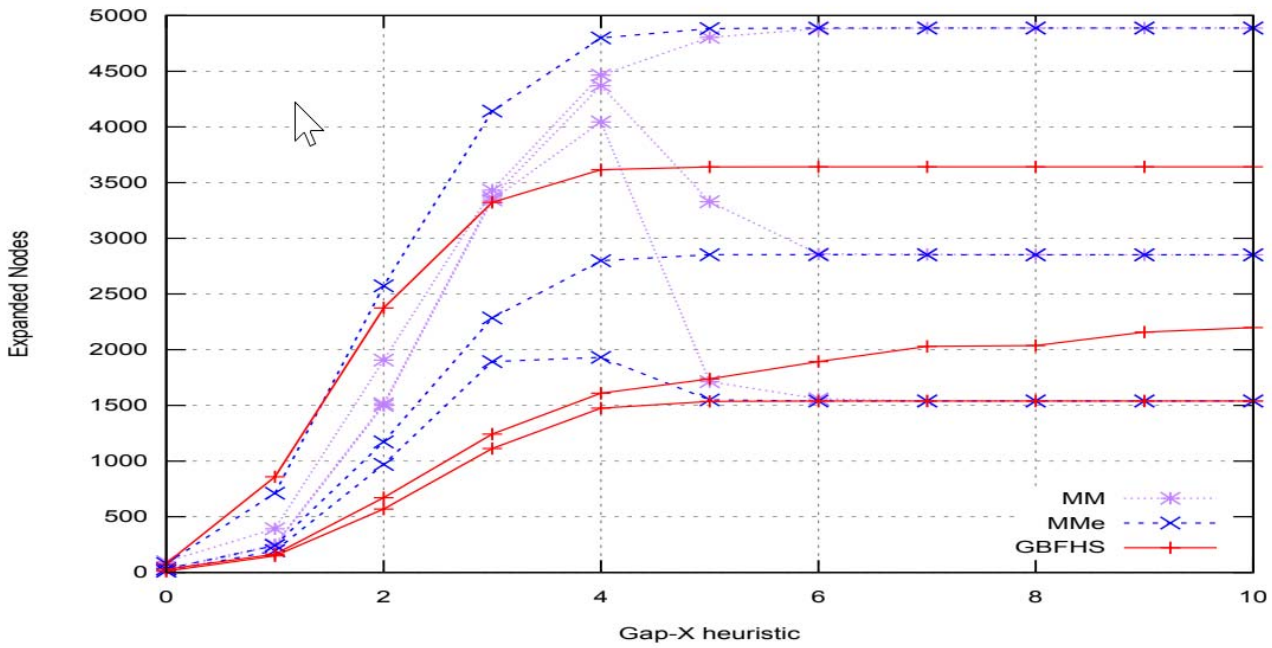


Figure 6: Actuals, Lower and Upper Bounds for MM, MMe, and GBFHS on 50 random 10-pancake problems as the heuristic degrades

Source: Adapted from [1]

Analysis 3: GBFHS vs NBS

domain	heuristic	GBFHS	A*	BS*	MMe	NBS
16 pancake	GAP	279	**125**	339	283	335
16 pancake	GAP-2	**250,941**	1,254,082	947,545	587,283	625,900
16 pancake	GAP-3	**2,140,718**	unsolvable	29,040,138	7,100,998	6,682,497
15 puzzle	MD	12,507,393	15,549,689	**12,001,024**	13,162,312	12,851,889

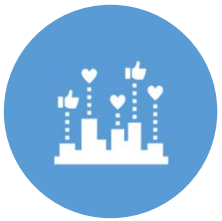
Table 1: Comparison between NBS, A*, GBFHS, MMe, and BS*
Source: Adapted from [1]

Pros of GBFHS

Good Flexibility

Stop on First Collision and guaranteed optimal solution

Better than NBS,MM, MMe and A* (when running unidirectionally)



Well-Behaved

Cons of GBFHS

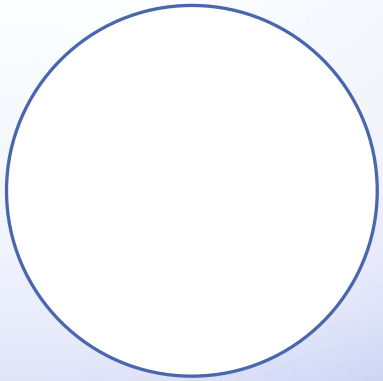
- Further improvements on inputs
- Cannot stop on first collision with arbitrary-cost domains
- Can be dominated by others
- Only work on domain with non-negative edge costs

Future Steps

- Automatically determine the value for the minimum action cost of the domain
- Automatically determine a good split and use it.
- Deep investigation on arbitrary-cost domains

Conclusion

Question Time



Reference

- [1] Barley, M., Riddle, P., Linares López C, Dobson, S., & Pohl, I. (2018). GBFHS: A Generalized Breadth-First Heuristic Search Algorithm. Presented at The Eleventh Annual Symposium on Combinatorial Search, Stockholm, Sweden. 14 July - 15 July 2018. Proceedings of the Eleventh Annual Symposium on Combinatorial Search (SOCS 2018). (pp. 9).[Online].Avaiable:https://researchspace.auckland.ac.nz/handle/2292/45475>
- [2] Shahaf S. Shperberg, Ariel Felner.(2020).On the Differences and Similarities of fMM and GBFHS. Presented at The Thirteenth International Symposium on Combinatorial Search (SoCS 2020).[Online].Avaiable:https://www.semanticscholar.org/paper/On-the-Differences-and-Similarities-of-fMM-and-Shperberg-Felner/b63ed5131f89010199f217eb7239d256c2b31bf0#paper-header.