# Iterative-deepening Bidirectional Heuristic Search with Restricted Memory

Shahaf S. Shperberg[1], Steven Danishevski[1], Ariel Felner[1], Nathan R. Sturtevant[2]

[1]Ben-Gurion University, Israel; [2]University of Alberta, Canada

1

**Abstract:** We introduce a new bidirectional memory-bounded algorithm, called IDBiHS, that can set the meeting point between the frontiers. The basic variant of IDBiHS uses memory linear in the search depth. In addition, we introduce two other variants of IDBiHS that use fixed amount of memory. Finally, we show that in many cases our new algorithms outperform pervious ones in terms of both node expansions and time.

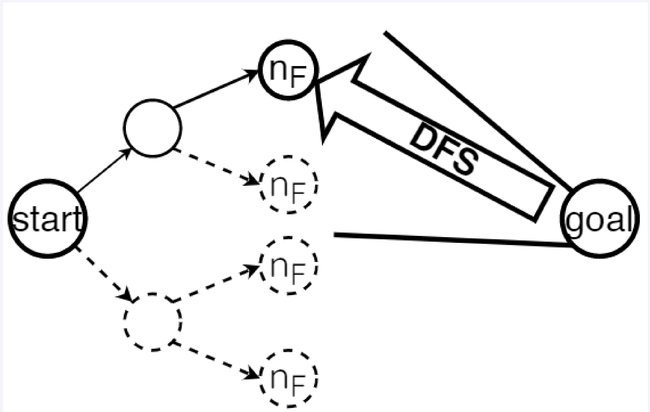
## Memory-based classification of algorithms

- **Unrestricted memory (UM)**: using memory proportional to the size of the explored search tree that they explored (e.g. A*).
- **Restricted Memory (RM)**
  - **Linear memory (LM)**: storing a single branch of the search tree (a path), memory linear in search depth.
  - **Fixed memory (FM)**: using fixed amount of memory M (as well as a single path).

2

## Iterative-deepening Bidirectional Heuristic Search (IDBiHS)

- Bidirectional search algorithm that uses linear memory
- Sets a meeting point using a split function
- Maintains a bound on the solution cost ($fT$)
- Split function chooses a forward g-value bound ($gT_F$) based on $fT$.
- Search nodes from *start* in a forward DFS (F_DFS) whose f-value doesn't exceed $fT$, and whose g-value doesn't exceed $gT_F$
- Once a node $n_F$ with $f_F(n_F) \leq fT$ and $g_F(n_F) > gT_F$ is found, suspend F_DFS and try to match $n_F$ by running a backward DFS (B_DFS)
- When trying to match a node $n_F$, B_DFS uses the f-bound $fT$ and a backward g-value bound defined as: $gT_B = fT - g_F(n_F) - \epsilon$
- If all forward nodes were search up to the threshold and a match (solution) was not found, increment $fT$ and update $gT_F$ using the split function.
- On each iteration there is **one F_DFS** run and **many B_DFS** runs

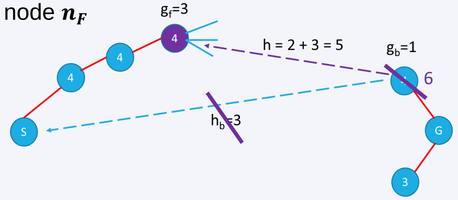



3

## Reducing the Number of Node Expansions

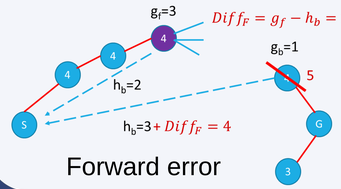
### Front-to-front Heuristic

- If a front-to-front (F2F) heuristic is available, it can be used by B_DFS when trying to match a node $n_F$

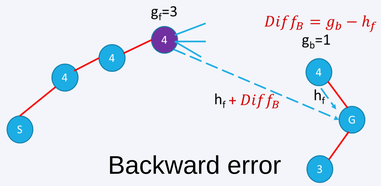



### Consistent Heuristic

- For consistent heuristics, the heuristic error along a path can be added to the heuristic of the other direction [Kaindl and Kainz, 1997]




Forward error



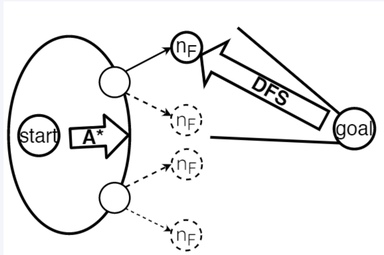

Backward error

4

## Fixed-memory Variants of IDBiHS

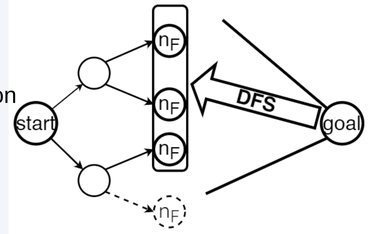
### A*+IDBiHS

- Inspired by A*+IDA*
- Run A* with available memory, then start F_DFS from open nodes




### IDBiHS-Trans

- Instead of matching very ***n_F*** node individually, store them in a transposition table, then match all nodes in the transposition table at once




5

## Empirical Evaluation

### LM algorithms

| Domain | H | Expanded: Best UM | IDA* | SFBDS JIL(1) | IDBiHS 0.5 | IDBiHS BW | Time(sec): Best UM | IDA* | SFBDS JIL(1) | IDBiHS 0.5 | IDBiHS BW |
|---|---|---|---|---|---|---|---|---|---|---|---|
| P[12] | G | 34(57%) | 95(76%) | 169(79%) | 85(76%) | **83**(77%) | 0.00 | 0.00 | 0.00 | 0.00 | |
| | G-1 | 1,256(14%) | 9,665(72%) | 8,622(74%) | 3,417(70%) | **2,857**(72%) | 0.00 | 0.01 | 0.01 | 0.00 | 0.00 |
| | G-2 | 13,089(10%) | 383,488(68%) | 205,469(71%) | 75,205(70%) | **72,939**(72%) | 0.04 | 0.25 | 0.15 | 0.06 | **0.05** |
| | G-3 | 38,545 (6%) | 8,601,396(66%) | 4,161,889(69%) | 1,540,325(71%) | **1,390,806**(69%) | 0.13 | 5.44 | 2.90 | 1.14 | **1.03** |
| P[45] | G | 45,822(50%) | 140,574(79%) | 184,809(74%) | 128,912(78%) | **127,743**(78%) | 2.32 | 0.94 | 1.25 | 0.85 | **0.84** |
| STP | MD | 8,143,628 (2%) | 242,460,834(50%) | 139,225,772(46%) | 174,414,968(40%) | **137,331,587**(48%) | 24.76 | 47.85 | 35.90 | 36.44 | **29.29** |
| Grid | Octile | 383(70%) | **47,060**(75%) | 131,488(78%) | 210,353(78%) | 122,304(79%) | 0.00 | **0.00** | 0.01 | 0.01 | 0.01 |

- IDBiHS is better on average in exponential domains and worse in polynomial domains (see paper for analysis)

### FM algorithms

| Domain | H | Mem | Expanded: Max-BAI | BIDA* | A*+IDA* | A*+IDBiHS | IDBiHS-Trans | Time(sec): Max-BAI | BIDA* | A*+IDA* | A*+IDBiHS | IDBiHS-Trans |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| P[12] | G | 50% | 84(74%) | 69(68%) | 40(65%) | **36**(73%) | 125(90%) | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| | | 10% | 84(78%) | 75(78%) | 56(75%) | **48**(77%) | 65(82%) | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| | | 1% | 93(76%) | 96(75%) | 62(78%) | **54**(75%) | 59(78%) | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| | G-1 | 50% | **858**(47%) | 2,743(43%) | 932(44%) | 1,129(64%) | 1,630(68%) | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| | | 10% | 2,987(80%) | **950**(72%) | 3,435(75%) | 1,021(69%) | 998(64%) | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| | | 1% | 5,525(80%) | 3,854(81%) | 4,607(77%) | 1,345(70%) | **1,203**(67%) | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| | G-2 | 50% | 20,662(62%) | 16,654(50%) | 96,891(55%) | 18,041(61%) | **8,243**(74%) | **0.03** | 0.12 | 0.09 | 0.06 | 0.05 |
| | | 10% | 45,113(83%) | 16,352(66%) | 135,907(80%) | 21,509(65%) | **6,670**(67%) | 0.04 | 0.10 | 0.09 | **0.03** | 0.05 |
| | | 1% | 113,607(79%) | 27,665(74%) | 207,447(82%) | 31,866(70%) | **11,924**(74%) | 0.09 | 0.08 | 0.25 | **0.03** | 0.06 |
| | G-3 | 50% | 223,490(78%) | 130,167(60%) | 2,259,163(80%) | 289,218(72%) | **35,915**(67%) | **0.26** | 1.96 | 1.58 | 0.43 | 0.70 |
| | | 10% | 578,231(79%) | 212,442(68%) | 3,293,269(75%) | 381,136(71%) | **41,020**(68%) | 0.51 | 2.01 | 2.13 | **0.36** | 0.81 |
| | | 1% | 1,631,356(75%) | 318,838(68%) | 4,910,121(74%) | 622,505(68%) | **104,516**(72%) | 1.35 | 1.71 | 3.10 | **0.50** | 0.91 |
| P[45] | G | 50% | 145,410(74%) | 97,663(63%) | 108,176(65%) | **87,563**(72%) | 634,304(93%) | 1.99 | 1.74 | 2.56 | **1.61** | 4.92 |
| | | 10% | 134,226(78%) | 128,601(75%) | 144,863(67%) | **120,340**(73%) | 412,355(81%) | 1.10 | 1.01 | 1.32 | **0.87** | 2.97 |
| | | 1% | 161,791(79%) | 162,385(77%) | 170,279(65%) | **143,120**(75%) | 128,204(77%) | 0.95 | 1.00 | 1.54 | **0.84** | 1.66 |
| STP | MD | 50% | 7,998,667(46%) | 7,654,126(40%) | 38,347,881(65%) | 15,489,868(42%) | **4,738,575**(50%) | 20.83 | 29.16 | 27.94 | 24.02 | **11.69** |
| | | 10% | 13,804,939(64%) | 5,183,115(51%) | 71,470,823(74%) | 24,322,900(41%) | **4,447,875**(47%) | 9.50 | 13.17 | 17.04 | 12.22 | **8.79** |
| | | 1% | 55,391,222(66%) | 6,026,913(56%) | 118,002,945(68%) | 46,075,139(39%) | **3,941,356**(35%) | 24.83 | 13.18 | 23.41 | 13.59 | **9.08** |
| Grid | Octile | 50% | **8,362**(75%) | 14,911(78%) | 16,503(72%) | 19,880(76%) | 12,833(78%) | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| | | 10% | 31,439(76%) | 36,417(77%) | 42,914(74%) | 55,964(75%) | **14,388**(78%) | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| | | 1% | 45,826(75%) | 46,632(79%) | 46,518(74%) | 106,818(77%) | **40,545**(76%) | 0.00 | 0.00 | 0.00 | 0.01 | 0.00 |

- new algorithms are very competitive and often outperform existing methods.