This document gives full references to the papers cited in the slides for Robert Holte's AAAI 2005 tutorial "Where Do Heuristics Come From ? (The Use of Abstraction to Speed Up Search)", plus a few extra references.

The list below follows exactly the same order as the slides.

1 PART 1

General introduction to search: [73].

Dijkstra's algorithm: [13].
A*: [29].
IDA*: [46].
Breadth-first Heuristic Search: [79].
Algorithm "C": [2].
Limited Discrepancy Search: [30].
RBFS: [48].
SMA*: [72].
Weighted A*: [67].
RTA*: [47].
BULB (Beam search with limited discrepancy backtracking): [24].

ABSTRIPS: [74].
ALPINE: [45].
Bacchus and Yang: [1].
Refinement Search for Explicit Graphs: [37].

Reinforcement Learning: [76].
Learning Heuristics: [18].
Speedup Learning for planning: [23,78,81].

Lookahead search: [4,5,47].

Metalevel Reasoning: [8].
GPS (the General Problem Solver): [65].
Multi-Tac: [62].

Puzzles - solving Rubik's Cube optimally: [49].
Parsing: [44].
Dynamic Programming - State Space Relaxation (SSR): [7].
Coarse-to-Fine Dynamic Programming - CFDP: [69].
Weighted logic programs: [22].
Quality-of-Service Routing in Networks: [57].

Sequential Ordering Problem: [34].
Co-operative Pathfinding: [75].
Vertex Cover: [19].
Multiple Sequence Alignment: [55,60,80].
Building Macro-Tables with optimal entries: [33].
Planning: [3,17,31].
Constrained Optimization (Mini-bucket elimination): [43].

Branch-and-bound: [9,16,71].
The Graph Traverser: [15].
A*: [29].

Minsky's early description of refinement: [61].
ABSTRIPS: [74].
Somalvico and colleagues: [27,58].
Gaschnig: [25].
Valtorta: [77].
Pearl: [66].
ABSOLVER: [63,68].
Multi-level version of Valtorta's theorem: [28].

Using memory to speed up search.
IDA*: [46].
MA*: [6].
IE, SMA*: [72].
Perimeter Search: [14,59].
ITS: [26].
Memory Enhanced IDA* (transposition tables): [70].

Pattern Databases: [10–12].
Hierarchical Heuristic Search (Hierarchical A*): [39,38].

2 PART 2

General definitions of domain abstraction and several basic results, including "size matters" and non-surjective abstractions: [32,35,36,41].

Korf and Reid formula for using the distribution of heuristic values to predict search performance: [54,53,41].

Pattern Databases: [10–12].
Hierarchical A* (HA*): [38].
Hierarchical IDA* (HIDA*): [40].

Delayed Duplicate Detection (DDD): [50,51].
Frontier Search: [55].

Efficient perfect hash functions for permutations: [64,56].

Optimal path caching and P-g caching: [38].

3 PART 3

Max'ing: [42].
Pattern Database Compression: [20].
Dovetailing: [10–12].

Additive Pattern Databases: [19,52].
Max'ing after adding: [42].

Space efficient pattern databases: [80].
Hierarchical IDA*: [40].
Reverse Resumable A*: [75].

Super-customization: [19,34,35].

Multiple PDB lookups based on symmetries: [10–12].
"Dual" PDB lookups: [21].

Coarse-to-Fine Dynamic Programming - CFDP: [69].

References

[1] F. Bacchus and Q. Yang. Downward refinement and the efficiency of hierarchical problem solving. *Artificial Intelligence*, 71(1):43–101, 1994.

[2] A. Bagchi and A. Mahanti. Search algorithms under different kinds of heuristics – a comparative study. *J. ACM*, 30(1):1–21, 1983.

[3] Blai Bonet and Hector Geffner. Planning as heuristic search. *Artificial Intelligence*, 129(1-2):5–33, 2001.

[4] Vadim Bulitko, Lihong Li, Russell Greiner, and Ilya Levner. Lookahead pathologies for single agent search. In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI), poster section*, pages 1531–1533, Acapulco, Mexico, 2003.

[5] Vadim Bulitko, Nathan Sturtevant, and Maryia Kazakevich. Speeding up learning in real-time search via automatic state abstraction. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, 2005.

[6] P. P. Chakrabarti, S. Ghose, A. Acharya, and S. C. de Sarkar. Heuristic search in restricted memory. *Artificial Intelligence*, 41(2):197–221, 1989.

[7] Nicos Christofides, Aristide Mingozzi, and Paolo Toth. State-space relaxation procedures for the computation of bounds to routing problems. *Networks*, 11(2):145–164, 1981.

[8] Michael T. Cox and Ashwin Ram. On the intersection of story understanding and learning. In A. Ram and K. Moorman, editors, *Understanding language understanding: Computational models of reading and understanding*, pages 397–434. MIT Press, 1999.

[9] G.A. Croes. A method for solving traveling-salesman problems. *Operations Research*, 6:791–812, 1958.

[10] J. C. Culberson and J. Schaeffer. Efficiently searching the 15-puzzle. Technical report, Department of Computer Science, University of Alberta, 1994.

[11] J. C. Culberson and J. Schaeffer. Searching with pattern databases. *Advances in Artificial Intelligence (Lecture Notes in Artificial Intelligence 1081)*, pages 402–416, 1996.

[12] J. C. Culberson and J. Schaeffer. Pattern databases. *Computational Intelligence*, 14(3):318–334, 1998.

[13] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271, 1959.

[14] J. F. Dillenburg and P. C. Nelson. Perimeter search. *Artificial Intelligence*, 65:165–178, 1994.

[15] J. E. Doran and D. Michie. Experiments with the graph traverser program. In *Proceedings of the Royal Society A*, volume 294, pages 235–259, 1966.

[16] W.L. Eastman. *Linear programming with pattern constraints.* PhD thesis, Harvard University, Cambridge, MA, 1958.

[17] S. Edelkamp. Planning with pattern databases. *Proceedings of the 6th European Conference on Planning (ECP-01)*, pages 13–34, 2001.

[18] Marco Ernandes and Marco Gori. Likely-admissible and sub-symbolic heuristics. In *Proceedings of the 16th Eureopean Conference on Artificial Intelligence, ECAI'2004, including Prestigious Applicants of Intelligent Systems, PAIS 2004*, pages 613–617, 2004.

[19] A. Felner, R. E. Korf, and Sarit Hanan. Additive pattern database heuristics. *Journal of Artificial Intelligence Research (JAIR)*, 22:279–318, 2004.

[20] Ariel Felner, Ram Meshulam, Robert C. Holte, and Richard E. Korf. Compressing pattern databases. *Proceedings of the Nineteenth National Conference on Artificial Intelligence (AAAI-04)*, pages 638–643, 2004.

[21] Ariel Felner, Uzi Zahavi, Jonathan Schaeffer, and Robert Holte. Dual lookups in pattern databases. *Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence (IJCAI-05)*, 2005.

[22] Pedro Felzenszwalb and David McAllester. A* beats dynamic programming, (unpublished). http://ttic.uchicaga.edu/∼dmcallester/astar.pdf.

[23] Lev Finkelstein and Shaul Markovitch. A selective macro-learning algorithm and its application to the nxn sliding-tile puzzle. *Journal of Artificial Intelligence Research*, 8:223–263, 1998.

[24] David Furcy and Sven Koenig. Limited discrepancy beam search. In *Proceedings of Nineteenth International Joint Conference on Artificial Intelligence (IJCAI)*, 2005.

[25] J. Gaschnig. A problem similarity approach to devising heuristics: First results. *IJCAI*, pages 301–307, 1979.

[26] Subrata Ghosh, Ambuj Mahanti, and Dana S. Nau. ITS: An efficient limited-memory heuristic tree search algorithm. In *AAAI*, pages 1353–1358, 1994.

[27] G. Guida and M. Somalvico. A method for computing heuristics in problem solving. *Information Sciences*, 19:251–259, 1979.

[28] O. Hansson, A. Mayer, and M. Valtorta. A new result on the complexity of heuristic estimates for the A* algorithm. *Artificial Intelligence*, 55:129–143, 1992.

[29] P.E. Hart, N. J. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4:100–107, 1968.

[30] W. D. Harvey and M. L. Ginsberg. Limited discrepancy search. In *Proc of IJCAI-95*, pages 607–613, Montreal Canada, 1995.

[31] Patrik Haslum and Hector Geffner. Admissible heuristics for optimal planning. In *Artificial Intelligence Planning Systems*, pages 140–149, 2000.

[32] I. Hernádvölgyi and R. C. Holte. Experiments with automatically created memory-based heuristics. *Proc. SARA-2000, Lecture Notes in Artificial Intelligence*, 1864:281–290, 2000.

[33] I. T. Hernádvölgyi. Searching for macro operators with automatically generated heuristics. *Advances in Artificial Intelligence - Proceedings of the Fourteenth Biennial Conference of the Canadian Society for Computational Studies of Intelligence (LNAI 2056)*, pages 194–203, 2001.

[34] Istvan T. Hernádvölgyi. Solving the sequential ordering problem with automatically generated lower bounds. In *Proceedings of Operations Research 2003 (Heidelberg, Germany)*, pages 355–362, 2003.

[35] Istvan T. Hernádvölgyi. *Automatically Generated Lower Bounds for Search*. PhD thesis, School of Information Technology and Engineering, University of Ottawa, Canada, 2004.

[36] R. C. Holte and I. T. Hernádvölgyi. A space-time tradeoff for memory-based heuristics. *Proceedings of the Sixteenth National Conference on Artificial Intelligence (AAAI-99)*, pages 704–709, 1999.

[37] R. C. Holte, T. Mkadmi, R. M. Zimmer, and A. J. MacDonald. Speeding up problem-solving by abstraction: A graph-oriented approach. *Artificial Intelligence*, 85:321–361, 1996.

[38] R. C. Holte, M. B. Perez, R. M. Zimmer, and A. J. MacDonald. Hierarchical A*: Searching abstraction hierarchies efficiently. *Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI-96)*, pages 530–535, 1996.

[39] Robert C. Holte, Chris Drummond, M.B. Perez, R.M. Zimmer, and Alan J. MacDonald. Searching with abstractions: A unifying framework and new high-performance algorithm. *Proceedings of the 10th Canadian Conference on Artificial Intelligence (AI'94)*, pages 263–270, 1994.

[40] Robert C. Holte, Jeffery Grajkowski, and Brian Tanner. Hierarchical heuristic search revisited. *Proc. SARA-2005, Lecture Notes in Artificial Intelligence*, 2005.

[41] Robert C. Holte and Istvan T. Hernádvölgyi. Steps towards the automatic creation of search heuristics. Technical Report TR04-02, Computing Science Department, University of Alberta, Edmonton, Canada T6G 2E8, 2004.

[42] Robert C. Holte, Jack Newton, Ariel Felner, and Ram Meshulam. Mutiple pattern databases. *Proceedings of the Fourteenth International Conference on Automated Planning and Scheduling (ICAPS-04)*, pages 122–131, 2004.

[43] Kalev Kask and Rina Dechter. A general scheme for automatic generation of search heuristics from specification dependencies. *Artificial Intelligence*, 129(1-2):91–131, 2001.

[44] Dan Klein and Christopher D. Manning. A* parsing: Fast exact Viterbi parse selection. In *Proceedings of HLT-NAACL 03*, 2003.

[45] Craig A. Knoblock. Automatically generating abstractions for planning. *Artificial Intelligence*, 68(2):243–302, 1994.

[46] R. E. Korf. Depth-first iterative-deepening: An optimal admissible tree search. *Artificial Intelligence*, 27:97–109, 1985.

[47] R. E. Korf. Real-time heuristic search. *Artificial Intelligence*, 42(3):189–211, 1990.

[48] R. E. Korf. Linear-space best-first search. *Artificial Intelligence*, 62(1):41–78, 1993.

[49] R. E. Korf. Finding optimal solutions to Rubik's Cube using pattern databases. *Proceedings of the Fourteenth National Conference on Artificial Intelligence (AAAI-97)*, pages 700–705, 1997.

[50] R. E. Korf. Delayed duplicate detection. *Extended abstract, Proc. IJCAI03*, pages 1539–1541, 2003.

[51] R. E. Korf. Best-first frontier search with delayed duplicate detection. In *Proceedings of the Nineteenth National Conference on Artificial Intelligence (AAAI-04)*, pages 650–657, San Jose, CA, July 2004.

[52] R. E. Korf and A. Felner. Disjoint pattern database heuristics. *Artificial Intelligence*, 134(1-2):9–22, Jan 2002.

[53] R. E. Korf and M. Reid. Complexity analysis of admissible heuristic search. *Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAI-98)*, pages 305–310, 1998.

[54] R. E. Korf, M. Reid, and S. Edelkamp. Time complexity of Iterative Deepening A*. *Artificial Intelligence*, 129(1-2):199–218, 2001.

[55] R. E. Korf and W. Zhang. Divide-and-conquer frontier search applied to optimal sequence alignment,. In *Proceedings of AAAI-2000*, pages 910–916, Austin, TX, 2000.

[56] Rich E. Korf and Peter Schultze. Large-scale parallel breadth-first search. In *National Conference on Artificial Intelligence (AAAI'05)*, pages 1380–1386, 2005.

[57] Yuxi Li, J. Harms, and Robert C. Holte. IDA*_MCSP: A fast exact MCSP algorithm. In *Proceedings ICC'05*, 2005.

[58] D. Mandrioli, A. Sangiovanni-Vincentelli, and M. Somalvico. Toward a theory of problem solving. In A. Marzollo, editor, *Topics in Articial Intelligence*, pages 48–167. Springer-Verlag, 1976.

[59] G. Manzini. BIDA*: an improved perimeter search algorithm. *Artificial Intelligence*, 75:347–360, 1995.

[60] Matthew McNaughton, Paul Lu, Jonathan Schaeffer, and Duane Szafron. Memory-efficient A* heuristics for multiple sequence alignment. In *Eighteenth National Conference on Artificial Intelligence*, pages 737–743, 2002.

[61] Marvin Minsky. Steps toward artificial intelligence. In E. Feigenbaum and J. Feldman, editors, *Computers and Thought*, pages 406–452. McGraw-Hill, 1963.

[62] Steven Minton. Integrating heuristics for constraint satisfaction problems: A case study. In *National Conference on Artificial Intelligence (AAAI'93)*, pages 120–126, 1993.

[63] J. Mostow and A. Prieditis. Discovering admissible heuristics by abstracting and optimizing: A transformational approach. *IJCAI*, pages 701–707, 1989.

[64] Wendy J. Myrvold and Frank Ruskey. Ranking and unranking permutations in linear time. *Information Processing Letters*, 79(6):281–284, 2001.

[65] A. Newell, J.C. Shaw, and H.A. Simon. A variety of intelligent learning in a general problem solver. In M.C. Yovits and S. Cameron, editors, *Self-organizing systems: Proceedings of an interdisciplinary conference*, pages 153–189. Pergamon Press, 1960.

[66] J. Pearl. *Heuristics: Intelligent Search Strategies for Computer Problem Solving*. Addison & Wesley, 1984.

[67] Ira Pohl. Heuristic search viewed as path finding in a graph. *Artificial Intelligence*, 1:193–204, 1970.

[68] A. E. Prieditis. Machine discovery of effective admissible heuristics. *Machine Learning*, 12:117–141, 1993.

[69] Christopher Raphael. Coarse-to-fine dynamic programming. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 23(12):1379–1390, 2001.

[70] A. Reinefeld and T. Marsland. Enhanced iterative-deepening search. *PAMI*, 16:701–710, 1994.

[71] M.J. Rossman and R.J. Twery. A solution to the travelling salesman problem (abstract E3.1.3). *Operations Research*, 6:687, 1958.

[72] S. Russell. Efficient memory-bounded search methods. *Proceedings of the Tenth European Conference on Artificial Intelligence (ECAI-92)*, pages 1–5, 1992.

[73] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, 1995.

[74] E. Sacerdoti. Planning in a hierarchy of abstraction spaces. *Artificial Intelligence*, 5(2):115–135, 1974.

[75] David Silver. Cooperative pathfinding. *Proceedings of the First Annual Conference on Artificial Intelligence and Interactive Entertainment (AIIDE-05)*, 2005.

[76] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998.

[77] M. Valtorta. A result on the computational complexity of heuristic estimates for the A* algorithm. *Information Sciences*, pages 47–59, 1984.

[78] M. Veloso and D. Borrajo. Learning strategy knowledge incrementally. In *Proceedings of the 6th International Conference on Tools with Artificial Intelligence*, pages 484–490, 1994.

[79] R. Zhou and E. Hansen. Breadth-first heuristic search. *Proc. Fourteenth International Conference on Automated Planning and Scheduling (ICAPS-04)*, pages 92–100, 2004.

[80] R. Zhou and E. A. Hansen. Space-efficient memory-based heuristics. *Proceedings of the Nineteenth National Conference on Artificial Intelligence (AAAI-04)*, pages 677–682, 2004.

[81] Terry Zimmerman and Subbarao Kambhampati. Learning-assisted automated planning: looking back, taking stock, going forward. In *AI Magazine*, pages 73–96, Summer, 2003.