Heuristic Search Comes of Age*

Nathan R. Sturtevant
University of Denver
Denver, CO, USA
sturtevant@cs.du.edu

Ariel Felner
Ben-Gurion University
Beer-Sheva, Israel
felner@bgu.ac.il

Maxim Likhachev
Carnegie Mellon University
Pittsburgh, PA, USA
maxim@cs.cmu.edu

Wheeler Ruml
University of New Hampshire
Durham, NH, USA
ruml@cs.unh.edu

Abstract

In looking back on the last five to ten years of work in heuristic search a few trends emerge. First, there has been a broadening of research topics studied. Second, there has been a deepened understanding of the theoretical foundations of search. Third, and finally, there have been increased connections with work in other fields. This paper, corresponding to a AAAI 2012 invited talk on recent work in heuristic search, highlights these trends in a number of areas of heuristic search. It is our opinion that the sum of these trends reflects the growth in the field and the fact that heuristic search has come of age.

Introduction

Heuristic search is coming of age. The classical heuristic search problem is one where the environment is static, all actions have unit edge costs, the goal state is fixed, and significant computational resources are devoted to finding optimal solutions. There has been, and still is much to explore and understand in this setting, but in the last five to ten years research has branched out in significant ways beyond this foundation. This diversification, together with a better theoretical understanding of search, a new comprehensive textbook on the subject (Edelkamp and Schrödl 2012), and the formation of the Symposium on Combinatorial Search (SoCS), is indicative of the maturation of heuristic search as a field.

This paper provides a brief overview of recent work in heuristic search. It is not intended to be exhaustive, but represents some of the topics which are of interest to members of the heuristic search community.

We broadly divide the work into three sections. At the high level we look at theoretical advances. Then we turn to algorithmic advances, followed by advances which are more domain-specific.

Theory

One mark of a mature field is a the development of theoretical models that predict and explain behavior. Work in heuristic search is often focused on admissible heuristics: lower-bounds on the cost to reach the goal from a particular state in the state space. Despite this, the influence of heuristics was not well understood for many years. Initial models came out of work on Rubik's cube (Korf 1997). Such models have been continually improving (Korf, Reid, and Edelkamp 2001; Zahavi et al. 2010; Lelis, Zilles, and Holte 2011) and, in certain domains, can now predict the cost of solving a particular problem in a particular domain quite accurately.

Inconsistent heuristics also provide lower-bounds, but do not follow the triangle inequality. Early work in the field suggested that using inconsistent heuristics could incur costly overheads and should be avoided when possible (Mero 1984), although it was unclear when inconsistent heuristics would arise in practice. The understanding of inconsistent heuristics has been greatly increased (Felner et al. 2011). It is now understood that inconsistent heuristics are easy to build and can be quite beneficial in practice. The drawbacks previously studied do not apply to algorithms like IDA*, and in practice the problems in A* can be fixed with local propagation of heuristic values.

Algorithms

In this section we focus on general-purpose algorithms, instead of those motivated by specific problem domains.

Suboptimal Search

In many problems, time or memory constraints rule out the possibility of finding and proving an optimal solution. An optimal heuristic search is doomed to expand every node whose f-value is less than the optimal solution cost. But, under such constraints, suboptimal solutions are almost always acceptable.

For many years just weighting the heuristic (Pohl 1973), which creates an inadmissible heuristic, was considered to be state-of-art, as this provides a bound on the final solution suboptimality. Extensions of this idea include dynamically modifying the weights used (Likhachev, Gordon, and Thrun 2003) or continuing the search after the first solution is found (Hansen and Zhou 2007). But, there are several new and active directions for solving problems suboptimally.

One recent direction is to directly build effective inadmissible heuristics. This can be done off-line (Jabarri Arfaee, Zilles, and Holte 2011) or on-line during search (Thayer and

*This paper was invited as a "What's Hot" paper to the AAAI'12 Sub-Area Spotlights track.

Ruml 2011a). Another approach is to develop search methods explicitly designed to find solutions quickly. In domains with non-uniform edge costs, one can take into account both the *solution length* (the number of actions required) and *solution cost* (the cost of the actions) via two different heuristic functions. Using both heuristics together can reduce the time required to find a solution (Thayer and Ruml 2011b; Thayer et al. 2012) for a given suboptimality bound or reduce solution cost for a similar solving time. Another strategy for providing solutions more quickly is to replace tight suboptimality bounds with probabilistic bounds (Stern, Felner, and Holte 2011).

Real-Time Agent-Centered Search Algorithms

Real-time agent-centered problems (Koenig 2001) model a real-time agent trying to reach a goal with limited perception of the world and limited computational power. As such, agents are incapable of solving a problem in a single computation, but must incrementally learn about the world by interleaving search and movement.

Recent work in this area has shown that there are bounds on the minimal required learning (Sturtevant, Bulitko, and Björnsson 2010). Recent work has either tried to avoid learning by precomputation (Bulitko, Björnsson, and Lawrence 2010), specialized exploration rules (Hernández and Baier 2011), or new forms of alternate learning (Sturtevant and Bulitko 2011).

Other Novel Algorithms

Here we look at several other novel algorithms which are able to improve on existing approaches.

Enhanced Partial Expansions A* A* with a consistent heuristic is known to be optimal in node expansions, but the theory has said little about nodes that are generated but never expanded. Enhanced Partial Expansion A* (Felner et al. 2012) is a variant of A* which greatly reduces node generations given domain-specific knowledge.

Single-Frontier Bidirectional Search Bidirectional search has not been widely successful in heuristic search. Single-Frontier Bidirectional Search (Felner et al. 2010) converts traditional uni-directional search algorithms into bidirectional search algorithms. It also generalizes previous work that exploited dualities in pattern databases (Zahavi et al. 2006).

Monte-Carlo Search A majority of work in heuristic search has followed a best-first approach, for some metric of the best state to expand next. Monte-Carlo methods have out-performed best-first approaches in a number of solitaire games (Cazenave 2009; Schadd et al. 2008) and have promise as an alternate approach for suboptimal search.

Hardware-Centered Search Algorithms

A number of search approaches have sought to push the limits of hardware.

External Memory Search Work has continued in search on more traditional problems using external memory – that is, memory on a hard disk drive. Although hard drives are too slow to use for random access, carefully designed algorithms are able to use hard drives without a significant loss in performance (Korf 2008; Zhou and Hansen 2011), primarily by accessing data sequentially. Almost all work on external memory has also looked at the issue of parallel search.

Parallel Search Modern multi-core machines offer great potential for speeding up the running time of algorithms. While parallel search was addressed in the past, major advances were recently made by providing intelligent methods for distributing the search to different threads or cores. Such distributions are made by abstracting the entire state space into threads (Zhou and Hansen 2007; Burns et al. 2010), by classifying a state based on its g and h values (Jabbar and Edelkamp 2006) or by using a hash function on the description of the state (Kishimoto, Fukunaga, and Botea 2009). A simple approach which has been successful for suboptimal search is to search different configurations of the same algorithm in parallel (Valenzano et al. 2010).

GPU Search Algorithms Most modern machines are equipped with a graphics processing unit (GPU) with hundreds or thousands of SIMD processors. Researchers have been able to exploit this hardware for breadth-first search problems (Edelkamp, Sulewski, and Yücel 2010) and for domain-independent planning problems (Sulewski, Edelkamp, and Kissmann 2011).

Domains

Specific problem domains may have special properties that motivate specially-designed heuristics or algorithms. We look here at a broader class of domains, and then at two specific domains.

Explicitly Represented Domains

The traditional model of a problem in heuristic search has been an exponentially growing search tree with the task of finding an optimal path between the start state and a fixed goal state. This describes puzzles like Rubik's cube or the sliding-tile puzzle. A different model of a search problem is found in the video game industry. In this industry search is used to plan character movement in virtual worlds. The entire search graph representing the world usually fits in memory. Given both the real-time constraints of games and aesthetic considerations, the task is not to find an optimal solution, but a reasonable path between any two states in the state space in 1ms or less.

Overall, this forms a new class of heuristic-search problems: state spaces that are small enough to fit in memory, but too large for the all-pairs shortest-path data to be efficiently computed and stored in memory. Another application for such work is planning on real-world maps. This class of problems has required a novel set of solutions.

The process of building heuristics for these domains is significantly different than previous techniques (Sturtevant et al. 2009; Felner, Sturtevant, and Schaeffer 2009). There are many state spaces where the all-pairs shortest-path data can be compressed and used efficiently (Bast, Funke, and Matijevic 2006; Botea 2011; Goldenberg et al. 2010). These techniques rely on structure in the underlying state space. In road networks, for instance, this has been formalized by the notion of 'highway dimension' (Abraham et al. 2010).

Besides the use of heuristics to speed up search, abstraction and refinement approaches have been used to produce optimal paths in road networks (Geisberger et al. 2008) and suboptimal paths in commercial games (Sturtevant and Buro 2005; Sturtevant 2007; Sturtevant and Geisberger 2010).

Search in games is particularly interesting because of a number of particular constraints. First, the path found by a search algorithm is almost always post-processed in some way, which reduces the need for optimal paths. A search space, such as a grid world, is almost always a coarse abstraction of the way that characters move through the world. In virtual worlds the topology of the world can also change significantly over time, meaning that approaches should be robust to changing worlds. In most games, the majority of the CPU and memory budget is allocated to graphics, leaving relatively little time for planning. This time also has to be shared between multiple characters in the game, that may have interaction in their plans. These constraints have motivated work on robust world representations (Sturtevant 2011) and any-angle planning (Daniel et al. 2010), which reduces the need for path post processing. Grid maps from commercial games have also been extracted and made into standard benchmark problems (Sturtevant 2012).

Multi-Agent Path Finding

Multi-Agent Path Finding (MAPF) is a unique problem which spans the gap between exponential and in-memory domains. The goal is for a set of agents to travel between their respective start and goal positions without conflicting with other agents. MAPF has practical applications in robotics, video games, and vehicle routing. With just a single agent, the problem is equivalent to path planning in an in-memory domain. But, when the graph is full of agents, save a single empty location, the problem is equivalent to the sliding-tile puzzle, whose search space grows exponentially in the depth of search. Thus, by varying the number of agents, the problem smoothly transitions from single-agent search in an explicit graph to exponential search in an implicit graph. Work in this area has the potential to provide important new insights for search. Two main ideas have been taken to approach this problem: *decoupled* and *coupled search*.

In decoupled approaches paths are planned independently for each agent. Agents can share information about their plans either explicitly, using shared information about their locations (Silver 2005; Dresner and Stone 2008), or implicitly, by sharing rules for movement (Wang and Botea 2008; Jansen and Sturtevant 2008).

On the border between coupled and decoupled approaches is a set of algorithms which use both centralized and decentralized control of agents (Wang and Botea 2009; Khorshid, Holte, and Sturtevant 2011; Luna and Bekris 2011).

Work on coupled search has primarily focused on optimal solutions. One focus has been on optimizations to A*-like search (Standley 2010). An alternate approach has been on bi-level approaches which split the solving process into multiple parts which are solved independently (Sharon et al. 2011; 2012). One point of interest is that constraints between agents bear similarity to constraints in SAT or CSPs, and so the relationship between these domains needs more study (Rintanen 2011).

Search in Robotics

In robotics, heuristic searches have been highly successful for low-dimensional (e.g., 2D or 3D) path planning. But up until recently there has been a wide belief that these methods do not apply to higher dimensional path planning problems such as motion planning for robotic arms, foothold planning for legged robots, path planning for aerial vehicles and kinodynamic planning for cars. While this belief is probably correct for optimal heuristic searches such as A* search, allowing for solutions that are within even small suboptimality bounds proves to make a big difference.

Giving up optimality guarantees eliminates the need for a thorough expansion of all the states that may possibly belong to an optimal solution. Furthermore, the space of solutions within a given suboptimality bound is typically much larger than the space of provably optimal solutions and therefore it often includes solutions that are much easier to find. Recent research shows that these properties open up possibilities for developing novel suboptimal heuristic searches that exploit the specifics of motion planning in robotics such as common presence of obstacle-free, easy-to-solve and/or intrinsically low-dimensional subspaces of the high-dimensional search-space (Likhachev and Stentz 2008; Gochev et al. 2011; Gonzalez and Likhachev 2011). These and other suboptimal heuristic searches have been applied successfully to motion planning for various high-dimensional robotic systems while providing consistency in solutions and rigorous guarantees on completeness and bounds on suboptimality (Cohen, Chitta, and Likhachev 2012; Gochev, Safonova, and Likhachev 2012; Cohen et al. 2011; Vernaza et al. 2009; Likhachev and Ferguson 2009).

Search in Planning

The fields of heuristic search and planning are, in many ways, quite related, as the two fields employ many of the same techniques and have seen the transfer of ideas.

Work on pattern database heuristics (Cullberson and Schaeffer 1998), which originated in heuristic search, has been used in planning domains (Edelkamp 2001). What characterizes heuristic search in planning, by comparison to other applications of heuristic search, is the automatic extraction of heuristic functions from a declarative description of the problem domain. The last decade has seen a proliferation of different techniques for doing so. These techniques can be broadly characterized into four different classes of algorithms: critical-path heuristics (Haslum and Geffner 2000), delete-relaxation heuristics (Bonet and Geffner 2001), abstraction heuristics (Edelkamp 2001; Helmert, Haslum, and Hoffmann

2007), and landmarks heuristics (Karpas and Domshlak 2009). Recent research (Helmert and Domshlak 2009) has defined a formal framework for the comparison of classes of admissible heuristics, and has shown that there exist non-trivial connections across these major algorithm classes in planning.

Researchers in other fields have often used portfolios or restarts to improve problem coverage. These ideas are now gaining traction in both heuristic search (Valenzano et al. 2010) and planning (Richter, Thayer, and Ruml 2010). The parallel track of the 2011 International Planning Competition was won by the ArvandHerd planner (Valenzano et al. 2011) which has portfolio of algorithms, including one using Monte-Carlo search.

Conclusions

This paper has briefly highlighted some of the broad areas of research in heuristic search. Overall, heuristic search encompasses a broad range of topics in optimal and suboptimal search with solution time constraints that range from milliseconds to weeks. The breadth of advances in the field suggest that heuristic search is now coming of age.

Acknowledgements

Special thanks to Joerg Hoffman, Robert Holte, Rich Korf, Carlos Linares Lopez and Daniel Borrajo for discussions around the topic of this paper.

References

Abraham, I.; Fiat, A.; Goldberg, A. V.; and Werneck, R. F. 2010. Highway dimension, shortest paths, and provably efficient algorithms. In *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '10, 782–793. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics.

Bast, H.; Funke, S.; and Matijevic, D. 2006. Transit ultrafast shortest-path queries with linear-time preprocessing. In *In 9th DIMACS Implementation Challenge*.

Bonet, B., and Geffner, H. 2001. Planning as heuristic search. 129(1–2):5–33.

Botea, A. 2011. Ultra-fast optimal pathfinding without runtime search. In *AIIDE*.

Bulitko, V.; Björnsson, Y.; and Lawrence, R. 2010. Case-based subgoaling in real-time heuristic search for video game pathfinding. *J. Artif. Intell. Res. (JAIR)* 39:269–300.

Burns, E.; Lemons, S.; Ruml, W.; and Zhou, R. 2010. Best-first heuristic search for multicore machines. *J. Artif. Intell. Res. (JAIR)* 39:689–743.

Cazenave, T. 2009. Nested monte-carlo search. In *IJCAI*, 456–461.

Cohen, B.; Subramanian, G.; Chitta, S.; and Likhachev, M. 2011. Planning for manipulation with adaptive motion primitives. In *Proceedings of the International Conference on Robotics and Automation (ICRA)*.

Cohen, B.; Chitta, S.; and Likhachev, M. 2012. Search-based planning for dual-arm manipulation with upright orientation constraints. In *Proceedings of the International Conference on Robotics and Automation (ICRA)*.

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

Daniel, K.; Nash, A.; Koenig, S.; and Felner, A. 2010. Theta*: Any-angle path planning on grids. *J. Artif. Intell. Res. (JAIR)* 39:533–579.

Dresner, K., and Stone, P. 2008. A multiagent approach to autonomous intersection management. *JAIR* 31:591–656.

Edelkamp, S., and Schrödl, S. 2012. *Heuristic Search - Theory and Applications*. Academic Press.

Edelkamp, S.; Sulewski, D.; and Yücel, C. 2010. Perfect hashing for state space exploration on the gpu. In *ICAPS*, 57–64.

Edelkamp, S. 2001. Planning with pattern databases. 13–24.

Felner, A.; Moldenhauer, C.; Sturtevant, N. R.; and Schaeffer, J. 2010. Single-frontier bidirectional search. In *AAAI*.

Felner, A.; Zahavi, U.; Holte, R.; Schaeffer, J.; Sturtevant, N.; and Zhang, Z. 2011. Inconsistent heuristics in theory and practice. *Artificial Intelligence* 175(9-10):1570–1603.

Felner, A.; Goldenberg, M.; Sharon, G.; Stern, R.; Sturtevant, N.; Holte, R. C.; and Schaeffer, J. 2012. Partial-expansion a* with selective node generation. In *AAAI*.

Felner, A.; Sturtevant, N.; and Schaeffer, J. 2009. Abstraction-based heuristics with true distance computations. In *Symposium on Abstraction, Reformulation and Approximation (SARA-09)*.

Geisberger, R.; Sanders, P.; Schultes, D.; and Delling, D. 2008. Contraction hierarchies: Faster and simpler hierarchical routing in road networks. In *WEA*, 319–333.

Gochev, K.; Cohen, B.; Butzke, J.; Safonova, A.; and Likhachev, M. 2011. Path planning with adaptive dimensionality. In *Proceedings of the International Symposium on Combinatorial Search (SoCS)*.

Gochev, K.; Safonova, A.; and Likhachev, M. 2012. Planning with adaptive dimensionality for mobile manipulation. In *Proceedings of the International Conference on Robotics and Automation (ICRA)*.

Goldenberg, M.; Felner, A.; Sturtevant, N.; and Schaeffer, J. 2010. Portal-based true-distance heuristics for path finding. In *Third Annual Symposium on Combinatorial Search*.

Gonzalez, J. P., and Likhachev, M. 2011. Search-based planning with provable suboptimality bounds for continuous state spaces. In *Proceedings of the International Symposium on Combinatorial Search (SoCS)*.

Hansen, E. A., and Zhou, R. 2007. Anytime heuristic search. *J. Artif. Intell. Res. (JAIR)* 28:267–297.

Haslum, P., and Geffner, H. 2000. Admissible heuristics for optimal planning. 140–149.

Helmert, M., and Domshlak, C. 2009. Landmarks, critical paths and abstractions: What's the difference anyway? In *ICAPS*.

Helmert, M.; Haslum, P.; and Hoffmann, J. 2007. Flexible abstraction heuristics for optimal sequential planning. In *Proc. ICAPS*, 176–183.

Hernández, C., and Baier, J. A. 2011. Real-time heuristic search with depression avoidance. In *IJCAI*, 578–583.

Jabarri Arfaee, S.; Zilles, S.; and Holte, R. 2011. Learning heuristic functions for large state spaces. *Artificial Intelligence* 175(16–17):2075–2098.

Jabbar, S., and Edelkamp, S. 2006. Parallel external directed model checking with linear i/o. In *VMCAI*, 237–251.

Jansen, M., and Sturtevant, N. 2008. Direction maps for cooperative pathfinding. In *AIIDE*.

Karpas, E., and Domshlak, C. 2009. Cost-optimal planning with landmarks. In *Proc. IJCAI*, 1728–1733.

Khorshid, M. M.; Holte, R. C.; and Sturtevant, N. R. 2011. A polynomial-time algorithm for non-optimal multi-agent pathfinding. In *SOCS*.

Kishimoto, A.; Fukunaga, A. S.; and Botea, A. 2009. Scalable, parallel best-first search for optimal sequential planning. In *ICAPS*.

Koenig, S. 2001. Agent-centered search. *AI Magazine* 22(4):109–132.

Korf, R. E.; Reid, M.; and Edelkamp, S. 2001. Time complexity of iterative-deepening-A*. *Artificial Intelligence* 129(1–2):199–218.

Korf, R. E. 1997. Finding optimal solutions to Rubik's Cube using pattern databases. In *National Conference on Artificial Intelligence (AAAI-97)*, 700–705.

Korf, R. E. 2008. Minimizing disk i/o in two-bit breadth-first search. In *AAAI*, 317–324.

Lelis, L.; Zilles, S.; and Holte, R. C. 2011. Improved prediction of ida*'s performance via epsilon-truncation. In *SOCS*.

Likhachev, M., and Ferguson, D. 2009. Planning long dynamically-feasible maneuvers for autonomous vehicles. *International Journal of Robotics Research (IJRR)*.

Likhachev, M., and Stentz, A. 2008. R* search. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*.

Likhachev, M.; Gordon, G. J.; and Thrun, S. 2003. Ara*: Anytime a* with provable bounds on sub-optimality. In *NIPS*.

Luna, R., and Bekris, K. E. 2011. An efficient and complete approach for cooperative path-finding. In *AAAI*.

Mero, L. 1984. A heuristic search algorithm with modifiable estimate. *Artificial Intelligence* 23:13–27.

Pohl, I. 1973. The avoidance of (relative) catastrophe, heuristic competence, genuine dynamic weighting and computational issues in heuristic problem solving. In *International Joint Conference on Artificial Intelligence (IJCAI-73)*, 12–17.

Richter, S.; Thayer, J. T.; and Ruml, W. 2010. The joy of forgetting: Faster anytime search via restarting. In *ICAPS*, 137–144.

Rintanen, J. 2011. Planning with sat, admissible heuristics and a*. In *IJCAI*, 2015–2020.

Schadd, M. P. D.; Winands, M. H. M.; van den Herik, H. J.; Chaslot, G.; and Uiterwijk, J. W. H. M. 2008. Single-player monte-carlo tree search. In *Computers and Games*, 1–12.

Sharon, G.; Stern, R.; Goldenberg, M.; and Felner, A. 2011. The increasing cost tree search for optimal multi-agent pathfinding. In *IJCAI*, 662–667.

Sharon, G.; Stern, R.; Felner, A.; and Sturtevant, N. 2012. Conflict-based search for optimal multi-agent path finding. In *AAAI*.

Silver, D. 2005. Cooperative pathfinding. In *AIIDE*, 117–122.

Standley, T. 2010. Finding optimal solutions to cooperative pathfinding problems. In *AAAI*, 173–178.

Stern, R.; Felner, A.; and Holte, R. 2011. Probably approximately correct heuristic search. In *SOCS*.

Sturtevant, N., and Bulitko, V. 2011. Learning where you are going and from whence you came: h-and g-cost learning in real-time heuristic search. *International Joint Conference on Artificial Intelligence (IJCAI)* 365–370.

Sturtevant, N. R., and Buro, M. 2005. Partial pathfinding using map abstraction and refinement. In *AAAI*, 1392–1397.

Sturtevant, N., and Geisberger, R. 2010. A comparison of high-level approaches for speeding up pathfinding. In *Artificial Intelligence and Interactive Digital Entertainment (AIIDE)*, 76–82.

Sturtevant, N.; Felner, A.; Barer, M.; Schaeffer, J.; and Burch, N. 2009. Memory-based heuristics for explicit state spaces. In *International Joint Conference on Artificial Intelligence (IJCAI-09)*, 609–614.

Sturtevant, N.; Bulitko, V.; and Björnsson, Y. 2010. On learning in agent-centered search. In *Autonomous Agents and Multiagent Systems (AAMAS)*, 333–340. International Foundation for Autonomous Agents and Multiagent Systems.

Sturtevant, N. R. 2007. Memory-efficient abstractions for pathfinding. In *AIIDE*, 31–36.

Sturtevant, N. 2011. A sparse grid representation for dynamic three-dimensional worlds. In *Artificial Intelligence and Interactive Digital Entertainment (AIIDE)*, 73–78.

Sturtevant, N. 2012. Benchmarks for grid-based pathfinding. *Transactions on Computational Intelligence and AI in Games*.

Sulewski, D.; Edelkamp, S.; and Kissmann, P. 2011. Exploiting the computational power of the graphics card: Optimal state space planning on the gpu. In *ICAPS*.

Thayer, J. T., and Ruml, W. 2011a. Learning inadmissible heuristics during search. In *Proceedings of the Twenty-First International Conference on Automated Planning and Scheduling*.

Thayer, J. T., and Ruml, W. 2011b. Bounded suboptimal search: A direct approach using inadmissible estimates. In *IJCAI*, 674–679.

Thayer, J. T.; Stern, R.; Felner, A.; and Ruml, W. 2012. Faster bounded-cost search using inadmissible estimates. In

Proceedings of the Twenty-Second International Conference on Automated Planning and Scheduling.

Valenzano, R. A.; Sturtevant, N. R.; Schaeffer, J.; Buro, K.; and Kishimoto, A. 2010. Simultaneously searching with multiple settings: An alternative to parameter tuning for suboptimal single-agent search algorithms. In *International Conference on Automated Planning and Scheduling (ICAPS)*, 177–184.

Valenzano, R.; Nakhost, H.; Müller, M.; Schaeffer, J.; and Sturtevant, N. 2011. Arvandherd: Parallel planning with a portfolio. *Proc. 7th International Planning Competition (IPC 2011)*.

Vernaza, P.; Likhachev, M.; Bhattacharya, S.; Chitta, S.; Kushleyev, A.; and Lee, D. D. 2009. Search-based planning for a legged robot over rough terrain. In *Proceedings of the International Conference on Robotics and Automation (ICRA)*.

Wang, K. C., and Botea, A. 2008. Fast and memory-efficient multi-agent pathfinding. In *ICAPS*, 380–387.

Wang, K.-H. C., and Botea, A. 2009. Tractable multi-agent path planning on grid maps. In *IJCAI*, 1870–1875.

Zahavi, U.; Felner, A.; Holte, R. C.; and Schaeffer, J. 2006. Dual search in permutation state spaces. In *National Conference on Artificial Intelligence (AAAI-06)*, 1076–1081.

Zahavi, U.; Felner, A.; Burch, N.; and Holte, R. C. 2010. Predicting the performance of IDA* with conditional distributions. *Journal of Artificial Intelligence Research* 37:41–83.

Zhou, R., and Hansen, E. A. 2007. Parallel structured duplicate detection. In *AAAI*, 1217–. AAAI Press.

Zhou, R., and Hansen, E. A. 2011. Dynamic state-space partitioning in external-memory graph search. In *ICAPS*.