Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16)

What's Hot in Heuristic Search?*

Roni Stern[1] and **Levi H. S. Lelis**[2]
[1] Ben Gurion University of the Negev, Israel
e-mail: roni.stern@gmail.com
[2] Universidade Federal de Vicosa, Brazil
e-mail: levi.lelis@ufv.br

Abstract

Search in general, and heuristic search in particular, is at the heart of many Artificial Intelligence algorithms and applications. There is now a growing and active community devoted to the empirical and theoretical study of heuristic search algorithms, thanks to the successful application of search-based algorithms to areas such as robotics, domain-independent planning, optimization, and computer games. In this extended abstract we highlight recent efforts in understanding suboptimal search algorithms, as well as ensembles of heuristics and algorithms. The result of these efforts are meta-reasoning methods which are applied to orchestrate the different components of modern search algorithms. Finally, we mention recent innovative applications of search that demonstrate the relevance of the field to general AI.

Introduction

Most approaches to Artificial Intelligence (AI) applications involve searching in large state spaces. Therefore, most AI algorithms and applications include heuristic search algorithms. We observed in the past years an influx in interest in the field of heuristic search, mostly due to the success of search-based planning algorithms in robotics (Likhachev and Ferguson 2009; Butzke et al. 2014), domain independent planning (Bonet and Geffner 2001; Helmert 2006), optimization (Marinescu and Dechter 2009), and commercial computer games (Churchill and Buro 2015).

In this extended abstract we highlight recent progress in: 1) the development and understanding of suboptimal search algorithms, 2) the design of ensembles of search algorithms and heuristics, and 3) the use of meta-reasoning to orchestrate autonomously different search algorithms components. Also, we point to several recent and innovative applications of heuristic search. This extended abstract is by no means an exhaustive review of heuristic search research, but a highlight of recent research in the field.

Understanding Suboptimal Heuristic Search

Much work has been done by the heuristic search community on studying algorithms and heuristics designed to find optimal solutions to search problems. However, it has been acknowledged that in many cases suboptimal solutions are also acceptable, especially since search problems can be too hard to solve optimally even with "almost perfect heuristic estimators, whose heuristic error is bounded by a small additive constant" (Helmert, Röger, and others 2008). Consequently, there are numerous suboptimal search algorithms, including beam search variants (Furcy and Koenig 2005; Lelis, Zilles, and Holte), and search algorithms that include random elements (Nakhost and Müller ; Xie, Valenzano, and Müller 2013; Valenzano et al. 2014). Unlike the theory of optimal search (Dechter and Pearl 1985; Goldenberg et al. 2014), the theory of suboptimal search algorithms is relatively under-developed, and hence developing deeper understanding of their behavior is an important current line of research.

Recent papers studied the behavior of Greedy Best First Search (GBFS), a popular suboptimal heuristic search algorithm that systematically expands states in an order dictated by a heuristic function that assigns merit to states. While simple to implement and often quite successful, GBFS is also known to fail on some domains. GBFS is especially ineffective in state spaces that have large regions in which the heuristic is uninformative: either giving equal value to all states in the region (known as a plateau) or misleading the search away from a goal (known as a local minimum).

Xie et al. explain theoretically this weakness of GBFS, showing that multiple small uninformative heuristic regions (UHR) – i.e., plateaus or local minima – can cause GBFS to "become stuck in the union of many distinct UHRs from different parts of the search space", resulting in a large virtual UHR over the open list (Xie, Müller, and Holte 2015). To partially resolve this, they proposed to perform periodic local searches. We expect that future work will study ways to detect such UHRs early in the search and more effective methods to escape these UHRs.

Another key factor that affects GBFS's performance is the heuristic being used. In optimal search, the use of a more accurate heuristic usually results in better search performance. Wilt and Ruml (2015) showed that this is not the case for GBFS, where using the more accurate heuristics may result in substantially poorer performance in terms of search running time. This highlights the challenge faced when developing an effective heuristic for suboptimal search: what function to optimize when developing a heuristic for suboptimal search? Wilt and Ruml proposed to measure the rank correlation between the used heuristic and the shortest distance to a goal. This correlation, named Goal Distance Rank Correlation (GDRC), was able to predict in some cases GBFS's performance, but we expect further research along this line.

Ensembles and Portfolios

It is known that heuristic search algorithms can benefit from the information provided by multiple heuristic functions. Traditional methods for using multiple heuristic functions aggregate the values returned by all the heuristics to a single value, e.g., by summing the values (Felner, Korf, and Hanan 2004), taking their max (Korf 1997; Holte et al. 2006), or more sophisticated mechanisms for aggregating multiple heuristics (Katz and Domshlak 2010; Pommerening et al. 2015). Recent work (Röger and Helmert 2010) has observed, especially when finding optimal solutions is not needed, that it is often more helpful to preserve the ordering induced by each heuristic function instead of aggregating their absolute value.

One way of preserving the ordering induced by several heuristic functions is to maintain several open lists, each ordered by a different heuristic function, and in every expansion cycle choose to expand a node from a different open list. This extends traditional best-first search algorithms such as A* (Hart, Nilsson, and Raphael 1968), which operates a single open list sorted according to the information provided by a single heuristic. Current state-of-the-art planners use this multiple open list approach (Röger and Helmert 2010). Recently, the idea was introduced and extended in search-based robotic planners, providing also theoretical guarantees on solution quality and search complexity even when using an arbitrarily large set of admissible and inadmissible heuristics (Phillips et al. 2015; Narayanan, Aine, and Likhachev 2015).

This direction of research opens the way for applying a range of techniques used in other sub-fields of AI when considering multiple sources of information. Prior work has already used genetic algorithms to choose the best combination of heuristics (Elyasaf and Sipper 2013), but one can consider adapting any feature selection methods from the machine learning literature, viewing heuristics as features and search effort as classification error. Alternatively, one may consider applying boosting methods, which are popular in the construction of ensemble of classifiers. A preliminary example of this was done by Richter and Helmert (2009).

In addition to combining multiple heuristics, another effective strategy is to use a set of different search algorithms in what is known as a search portfolio. For example, Fast Downward Stone Soup is system that sequentially runs different instances of the Fast Downward planner, with each instance being run for a fixed amount of time (Helmert, Röger, and Karpas 2011). As another example, ArvandHerd runs multiple search algorithms with different strategies in parallel (Valenzano et al. 2011). Recently, Seipp et al. (2015) presented an algorithm that automatically chooses the configuration of a portfolio planner, i.e., how much time to allocate to each of the planners in the portfolio.

Decision Making in Heuristic Search

As search algorithms become more complex, using ensembles of heuristics, operator orderings, and other algorithmic variations, there is a growing importance in deciding when to apply which algorithmic component.

Recent work apply rational meta-reasoning to select which set of heuristic to apply to evaluate a given state for A* (Tolpin et al. 2013) and for IDA* (Betzalel, Felner, and Shimony 2015; Tolpin et al.), showing significant gains. A different application of meta-reasoning is for cases where the searching agent can interleave planning and execution. The meta-reasoning here is to decide how much searching to do before committing to the execution of an action, aiming to minimize the *goal achievement time* – the overall computation and execution time of reaching a goal (O'Ceallaigh and Ruml 2015). In a similar line of research, Barley et al. (2014) developed a meta-reasoning system for selecting a subset of heuristic functions while minimizing the A* running time in domain-independent planning problems. Rayner et al. (2013) showed how to make near-optimal selection of heuristic subsets on map-based pathfinding problems. Recent advancements in predicting the running time of search algorithms developed by Lelis et al. (2013; 2013; 2014) can provide the facilitative technologies for meta-reasoning tasks such as deciding which search strategy will find the goal the quickest. Generally, we believe that applying AI decision making algorithms to develop more intelligent search algorithms is a promising research direction.

Novel Applications

With the dissemination of AI into everyday life, recent years have shown many new applications of heuristic search algorithms to novel domains. These domains include feature selection for clustering algorithms (Mariño and Lelis 2015), anomaly detection for cyber-security (Mirsky et al. 2015), finding error-correction codes (Palombo et al. 2015), a Kiva-like domain for multi-agent pathfinding (Cohen, Uras, and Koenig 2015), Maximum a Posteriori Estimation in Probabilistic Programs (Tolpin and Wood 2015), an AI player of a commercial video game which accounts for the player's enjoyment (Churchill and Buro 2015), and automated discovery of chemical compounds (Heifets and Jurisica 2012).

These applications of heuristic search theory and algorithms provides yet another demonstration of the impact of researching heuristic search methods.

Conclusions

While heuristic search is a mature sub-field of AI, there are many fundamental research challenges being addressed these days. In this brief review we covered recent progress in the understanding of suboptimal search algorithms, the design and use of heuristic ensembles, and more generally the application of meta-reasoning in guiding search algorithms.

Acknowledgments

We thank Ariel Felner, Robert Holte, Wheeler Ruml, and Malte Helmert for their helpful comments.

References

Barley, M.; Franco, S.; and Riddle, P. 2014. Overcoming the utility problem in heuristic generation: Why time matters. In *ICAPS*, 38–46.

Betzalel, O.; Felner, A.; and Shimony, S. E. 2015. Type system based rational lazy IDA*. In *Symposium on Combinatorial Search (SOCS)*.

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

Butzke, J.; Sapkota, K.; Prasad, K.; MacAllister, B.; and Likhachev, M. 2014. State lattice with controllers: Augmenting lattice-based path planning with controller-based motion primitives. In *International Conference on Intelligent Robots and Systems (IROS)*, 258–265. IEEE.

Churchill, D., and Buro, M. 2015. Hierarchical portfolio search: Prismata's robust ai architecture for games with large search spaces. In *AIIDE*.

Cohen, L.; Uras, T.; and Koenig, S. 2015. Feasibility study: Using highways for bounded-suboptimal multi-agent path finding. In *Symposium on Combinatorial Search (SOCS)*, 2–8.

Dechter, R., and Pearl, J. 1985. Generalized best-first search strategies and the optimality of A*. *Journal of the ACM* 32(3):505–536.

Elyasaf, A., and Sipper, M. 2013. HH-evolver: a system for domain-specific, hyper-heuristic evolution. In *conference companion on genetic and evolutionary computation (GECCO)*.

Felner, A.; Korf, R. E.; and Hanan, S. 2004. Additive pattern database heuristics. *J. Artif. Intell. Res. (JAIR)* 22:279–318.

Furcy, D., and Koenig, S. 2005. Limited discrepancy beam search. In *IJCAI*, 125–131.

Goldenberg, M.; Felner, A.; Stern, R.; Sharon, G.; Sturtevant, N. R.; Holte, R. C.; and Schaeffer, J. 2014. Enhanced partial expansion A. *J. Artif. Intell. Res. (JAIR)* 50:141–187.

Hart, P. E.; Nilsson, N. J.; and Raphael, B. 1968. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics* SCC-4(2):100–107.

Heifets, A., and Jurisica, I. 2012. Construction of new medicines via game proof search. In *AAAI*, 1564–1570.

Helmert, M.; Röger, G.; and Karpas, E. 2011. Fast downward stone soup: A baseline for building planner portfolios. In *Workshop on Planning and Learning, ICAPS*, 28–35.

Helmert, M.; Röger, G.; et al. 2008. How good is almost perfect? In *AAAI*, volume 8, 944–949.

Helmert, M. 2006. The fast downward planning system. *J. Artif. Intell. Res. (JAIR)* 26:191–246.

Holte, R. C.; Felner, A.; Newton, J.; Meshulam, R.; and Furcy, D. 2006. Maximizing over multiple pattern databases speeds up heuristic search. *Artificial Intelligence* 170:1123–1136.

Katz, M., and Domshlak, C. 2010. Optimal admissible composition of abstraction heuristics. *Artif. Intell.* 174(12-13).

Korf, R. E. 1997. Finding optimal solutions to rubik's cube using pattern databases. In *AAAI/IAAI*, 700–705.

Lelis, L. H.; Otten, L.; and Dechter, R. 2013. Predicting the size of depth-first branch and bound search trees. In *IJCAI*, 594–600.

Lelis, L. H.; Otten, L.; and Dechter, R. 2014. Memory-efficient tree size prediction for depth-first search in graphical models. In *Principles and Practice of Constraint Programming (CP)*, 481–496.

Lelis, L. H.; Zilles, S.; and Holte, R. C. Stratified tree search: a novel suboptimal heuristic search algorithm. In *AAMAS*.

Lelis, L. H.; Zilles, S.; and Holte, R. C. 2013. Predicting the size of IDA* 's search tree. *Artificial Intelligence* 196:53–76.

Likhachev, M., and Ferguson, D. 2009. Planning long dynamically feasible maneuvers for autonomous vehicles. *International Journal of Robotics Research* 28(8):933–945.

Marinescu, R., and Dechter, R. 2009. Memory intensive and/or search for combinatorial optimization in graphical models. *Artif. Intell.* 173(16-17):1492–1524.

Mariño, J. R., and Lelis, L. H. 2015. Feature selection as state-space search: An empirical study in clustering problems. In *Symposium on Combinatorial Search (SoCS)*.

Mirsky, Y.; Cohen, A.; Stern, R.; Felner, A.; Rokach, L.; Elovici, Y.; and Shapira, B. 2015. Search problems in the domain of multiplication: Case study on anomaly detection using markov chains. In *Symposium on Combinatorial Search (SOCS)*, 70–77.

Nakhost, H., and Müller, M. Towards a second generation random walk planner: an experimental exploration. In *IJCAI*.

Narayanan, V.; Aine, S.; and Likhachev, M. 2015. Improved multi-heuristic a* for searching with uncalibrated heuristics. In *Symposium on Combinatorial Search (SoCS)*.

O'Ceallaigh, D., and Ruml, W. 2015. Metareasoning in real-time heuristic search. In *Symposium on Combinatorial Search (SOCS)*.

Palombo, A.; Stern, R.; Puzis, R.; Felner, A.; Kiesel, S.; and Ruml, W. 2015. Solving the snake in the box problem with heuristic search: First results. In *Symposium on Combinatorial Search (SOCS)*, 96–104.

Phillips, M.; Narayanan, V.; Aine, S.; and Likhachev, M. 2015. Efficient search with an ensemble of heuristics. *IJCAI*.

Pommerening, F.; Helmert, M.; Röger, G.; and Seipp, J. 2015. From non-negative to general operator cost partitioning. *AAAI*.

Rayner, D. C.; Sturtevant, N. R.; and Bowling, M. 2013. Subset selection of search heuristics. In *IJCAI*, 637–643.

Richter, S., and Helmert, M. 2009. Preferred operators and deferred evaluation in satisficing planning. In *ICAPS*, 273–280.

Röger, G., and Helmert, M. 2010. The more, the merrier: Combining heuristic estimators for satisficing planning. In *ICAPS*.

Seipp, J.; Sievers, S.; Helmert, M.; and Hutter, F. 2015. Automatic configuration of sequential planning portfolios. In *AAAI*.

Tolpin, D., and Wood, F. 2015. Maximum a posteriori estimation by search in probabilistic programs. In *Symposium on Combinatorial Search (SOCS)*, 201–205.

Tolpin, D.; Betzalel, O.; Felner, A.; and Shimony, S. E. Rational deployment of multiple heuristics in IDA*. In *ECAI*.

Tolpin, D.; Beja, T.; Shimony, S. E.; Felner, A.; and Karpas, E. 2013. Toward rational deployment of multiple heuristics in A*. In *IJCAI*.

Valenzano, R.; Nakhost, H.; Müller, M.; Schaeffer, J.; and Sturtevant, N. 2011. Arvandherd: Parallel planning with a portfolio. In *IPC 2011 Deterministic Track*, 113–116.

Valenzano, R.; Schaeffer, J.; Sturtevant, N.; and Xie, F. 2014. A comparison of knowledge-based gbfs enhancements and knowledge-free exploration. In *ICAPS*, 375–379.

Wilt, C. M., and Ruml, W. 2015. Building a heuristic for greedy search. In *Symposium on Combinatorial Search (SoCS)*.

Xie, F.; Müller, M.; and Holte, R. 2015. Understanding and improving local exploration for gbfs. In *ICAPS*.

Xie, F.; Valenzano, R. A.; and Müller, M. 2013. Better time constrained search via randomization and postprocessing. In *ICAPS*.