Proceedings of the Ninth International
Symposium on Combinatorial Search (SoCS 2016)

Abstracts of Papers Presented at SoCS 2016 in the Previously Published Paper Track

Jorge Baier and **Adi Botea**
Editors

Abstract

This document gathers the abstracts for the papers that were presented as part of the *Previously Published Paper Track*.

Do People Think Like Computers?

Bas van Opheusden, Zahy Bnaya,
Gianni Galbiati, Wei Ji Ma
New York University

Although DeepBlue's triumph in chess and AlphaGo's recent triumph in Go are milestones in artificial intelligence, it is unclear whether their algorithms resemble human thinking. Laboratory tasks that are common in decision-making psychology rarely involve long sequences of decisions with many options at each step. To investigate how people explore such decision trees, we turned a challenging variant of tic-tac-toe into an experimental paradigm. People played against each other, against AI agents of different strengths, chose between two moves, and evaluated positions. We model human thinking with a heuristic search algorithm. The algorithm explores a decision tree using best-first search with selective pruning, guided by a heuristic function that evaluates board states with a weighted sum of features. To capture characteristic of human mistakes, we incorporate three types of noise. We estimate model parameters for individual subjects, and show that this model predicts subjects' choices better than a dozen alternative models. Moreover, the model can generalize from subjects's choices during games to predict their choices in similar tasks, response times and eye movements. Finally, the parameters inferred for different subjects suggest that stronger players build larger search trees and have less noise.

A Bayesian Effort Bias for Sampling-Based Motion Planning

Scott Kiesel, Wheeler Ruml
University of New Hampshire

Recent advances in sampling-based motion planning have exploited concepts similar to those used in the heuristic graph search community, such as computing heuristic cost-to-go estimates and using state-space abstractions to derive them. Following this trend, we explore how the concept of search effort can be exploited to find plans quickly. Previous work in motion planning attempts to find plans quickly by preferring states with low cost-to-go. Recent work in graph search suggests that following search-effort-to-go estimates can yield faster planning. In this paper, we demonstrate how this idea can be adapted to the context of motion planning. Our planner, Bayesian Effort Aided Search Trees (BEAST), uses on-line Bayesian estimates of effort to guide the expansion of a motion tree toward states through which a plan is estimated to be easy to find. We present results in five simulated domains (Kinematic and Dynamic Car, Hovercraft, Blimp and Quadrotor) indicating that BEAST is able to find solutions more quickly and has a higher success rate than previous methods. We see this work as further strengthening the algorithmic connections between motion planning and heuristic graph search.

This work was previously accepted at the Planning and Robotics Workshop at ICAPS 2016 (Kiesel and Ruml 2016).

Reusing Previously Found A* Paths for Fast Goal-Directed Navigation in Dynamic Terrain

Carlos Hernández, Universidad Andrés Bello
Roberto Asín,
Universidad Católica de la Santísima Concepción
Jorge A. Baier, Pontificia Universidad Católica de Chile

Goal-directed navigation, the problem of leading an autonomous agent from an initial location to a goal location, is an important problem in AI with recognized applications in robotics. In the most general of its variants—under the so-called *dynamic terrain*—one assumes the environment (i.e., the map) may change while the agent is moving.

Generalized Adaptive A* (GAA*) (Sun, Koenig, and Yeoh 2008) is an incremental heuristic search (IHS) (Koenig et al. 2004) algorithm, amenable for goal-directed navigation, that replans using A* when solving goal-directed navigation problems in dynamic terrain. Immediately after each A* search, it runs an efficient procedure that updates the heuristic values of states that were just expanded by A*, making them more informed. Those updates allow GAA* to speed up subsequent A* searches. Being based on A*, it is simple to describe and communicate; however, it is outperformed by other incremental algorithms like the state-of-the-art D* Lite (Koenig and Likhachev 2002) algorithm at goal-directed navigation.

In a AAAI15 paper (Hernández, Asín, and Baier 2015) we proposed Multipath Generalized Adaptive A*, a simple but powerful extension of GAA* that exploits more information from previous A* searches than GAA* does. In particular, MPGAA* reuses paths towards the goal which were computed by previous calls to A*. We prove that MPGAA* has the same properties of GAA*. In addition, our experimental evaluation shows that MPGAA* is superior to D* Lite in indoor navigation settings and usually faster than D* Lite in scenarios comparable to outdoor navigation. Reusing previously found A* paths is not a new idea. In IHS this idea has been used before in other algorithms, specifically in Tree-Adaptive A* (Hernández et al. 2011) (Hernández et al. 2015) and Multipath-Adaptive A* (Hernández, Baier, and Asín 2014). While none of these algorithms can be used in dynamic terrain settings, MPGAA* can be seen as an application of the same principle underlying Multipath-Adaptive A*. As such, the contribution of (Hernández, Asín, and Baier 2015) is to (1) show how to exactly integrate these ideas into GAA*, and (2) an experimental evaluation showing that MPGAA* is a superior algorithm for goal-directed navigation in dynamic terrain.

Optimally Solving Permutation Sorting Problems with Efficient Partial Expansion Bidirectional Heuristic Search

Marco Lippi, Università degli Studi di Bologna,
Marco Ernandes, Quest-IT
Ariel Felner, Ben Gurion University

Sorting integer permutations with a minimum number of moves is a task with many potential applications, ranging from computational biology to logistics (Hannenhalli and Pevzner 1999). Such a task can be easily formulated as a heuristic search problem, where different variants can be designed, according to different sets of allowed moves on a given permutation. Nevertheless, this domain has not yet captured the attention of the heuristic search community, with the exception of the pancake, the burnt pancake and the TopSpin puzzles. The intrinsic nature of permutation sorting problems causes a very large branching factor. We believe that this set of problems could represent challenging benchmarks for heuristic search algorithms as classic approaches such as A* and IDA* will quickly become inefficient or even infeasible as the problem dimension grows.

In a recent paper published in AI Communications, entitled "Optimally solving permutation sorting problems with efficient partial expansion bidirectional heuristic search" (Lippi, Ernandes, and Felner 2016), we introduced this challenging domain to the community, proposing heuristic functions for the considered variants. In addition, we also presented a novel technique for this category of problems, which combines two recently introduced paradigms that have been employed in difficult search domains with good performance: **(1)** *enhanced partial expansion* (EPE) (Felner et al. 2012; Goldenberg et al. 2014) and **(2)** *efficient single-frontier bidirectional search* (eSBS) (Felner et al. 2010; Moldenhauer et al. 2010; Lippi, Ernandes, and Felner 2012). In particular, we proposed a new class of algorithms combining the benefits of EPE and eSBS, named efficient Single-frontier Bidirectional Search with Enhanced Partial Expansion (eSBS-EPE): the result is a bidirectional search paradigm that employs *double-state* nodes (as in eSBS), while avoiding the generation of surplus nodes (as in EPE). Such a paradigm can be nicely coupled with classic A* or with IDA*. An extensive experimental evaluation was reported, showing that eSBS-EPE is a very effective approach for permutation sorting problems, often outperforming previous methods on large-size configurations. With the new eSBS-EPE class of methods we were able to push the limit and solve the largest size instances of some of the problem domains (85 and 28 for the pancake and the burnt pancake puzzles, respectively). This novel search paradigm hence provides a very promising framework also for other domains.

Multi-Agent Path Finding with Payload Transfers and the Package-Exchange Robot-Routing Problem

Hang Ma, University of Southern California
Craig Tovey, Georgia Institute of Technology
Guni Sharon, Ben Gurion University
T. K. Satish Kumar, Sven Koenig
University of Southern California

Payloads can be transferred in real-world applications such as ride-sharing (or taxis) with passenger transfers (Coltin and Veloso 2014) or package delivery with robots in offices (Veloso et al. 2015). The theoretical implications of allowing payload transfers are still poorly understood. In a paper published at AAAI 2016 (Ma et al. 2016a), we therefore studied transportation problems where robots have to deliver packages and can transfer the packages among each other. Specifically, we studied the package-exchange robot-routing problem (PERR), where each robot carries one package, any two robots in adjacent locations can exchange their packages, and each package needs to be delivered to a given destination. We proved that exchange operations make all PERR instances solvable. Yet, we also showed that PERR is NP-hard to approximate within any factor less than $4/3$ for makespan minimization (= the time when the latest package reaches its destination) and is NP-hard to solve for flowtime minimization (= the sum of the times when each package reaches its destination), even when there are only two types of packages. Our proof techniques (namely, reductions to versions of the satisfiability problem called $\leq$3,=3-SAT (Tovey 1984) and $2/\overline{2}/3$-SAT) also generate new insights into other transportation problems, for example, into the hardness of approximating optimal solutions to the standard multi-agent path-finding problem (MAPF), which does not permit exchange operations and also turns out to be NP-hard to approximate within any factor less than $4/3$ for makespan minimization (improving on (Yu and LaValle 2013b) and (Ratner and Warmuth 1990)). Finally,

Worst-Case Ratio	4-Neighbor Grids		8-Neighbor Grids	
	Vertices at Centers	Vertices at Corners	Vertices at Centers	Vertices at Corners
$\frac{SGP}{TSP}$	$6/(2+\sqrt{2}) \approx 1.757$	$\sqrt{2} \approx 1.414$	$3\sqrt{2}-3 \approx 1.243$	$\sqrt{4-2\sqrt{2}} \approx 1.082^*$
$\frac{SVP}{TSP}$	$3\sqrt{2}-3 \approx 1.243$	1	$3\sqrt{2}-3 \approx 1.243$	1
$\frac{SGP}{SVP}$	$\sqrt{2} \approx 1.414$	$\sqrt{2} \approx 1.414$	$\sqrt{4-2\sqrt{2}} \approx 1.082^*$	$\sqrt{4-2\sqrt{2}} \approx 1.082^*$

Table 1: Theoretical Results

we presented optimal and suboptimal PERR solvers that are inspired by MAPF solvers, namely a flow-based ILP formulation (Yu and LaValle 2013a) and an adaptation of conflict-based search (Sharon et al. 2015a). Our empirical results demonstrated that these solvers scale well and that PERR instances often have smaller makespans and flowtimes than the corresponding MAPF instances. See http://idm-lab.org/bib/abstracts/Koen16a.html for the full paper. We thank Jingjin Yu for making the code of their flow-based MAPF solver and Nathan Sturtevant for making game maps available to us. The research at USC was supported by NSF under grant numbers 1409987 and 1319966.

Path Planning on Grids: The Effect of Vertex Placement on Path Length

James Bailey, Craig Tovey, Georgia Institute of Technology
Tansel Uras, Sven Koenig,
University of Southern California
Alex Nash, Northrop Grumman

Video-game designers often tessellate continuous two-dimensional terrain into a grid of blocked and unblocked square cells (Tozour 2004). The three main ways to calculate short paths on such a grid are to determine truly shortest paths, shortest vertex (or, equivalently, any-angle) paths (Daniel et al. 2010a) and shortest grid paths, listed here in decreasing order of computation time and increasing order of resulting path length. In a paper published at AIIDE 2016 (Bailey et al. 2015), we quantified the advantage of placing vertices at cell corners rather than cell centers when finding shortest grid paths or shortest vertex paths on both 4-neighbor or 8-neighbor square grids, which provides valuable information to game designers when it comes to determining the placement of vertices in their games. We found the worst-case ratios between each possible pair of shortest grid paths (SGPs), shortest vertex paths (SVPs) and truly shortest paths (TSPs) for 4 and 8-neighbor grid graphs where vertices are placed at either corners or centers, extending earlier results (Ferguson and Stentz 2006) and (Nash 2012). Table 1 summarizes our theoretical results, which hold for all arrangements of blocked and unblocked cells. All worst-case ratios are tight (or asymptotically tight), that is, there exist path-planning problems for which the worst-case ratios have the given values (or are within any $\epsilon > 0$ of the given values) but no path-planning problems for which the worst-case ratios are larger. Values that are only asymptotically tight are indicated by an asterisk ($*$). For each possible instance, the worst-case ratio when placing vertices at centers is greater than or equal to the worst-case ratio when placing vertices at corners. For instance, for 4-neighbor grid graphs with vertices placed at centers (corners) the shortest grid path can be up to 1.757 (respectively 1.414) times as long as the truly shortest path. Our experimental results supported our theoretical results, suggesting again that shorter paths result from placing vertices at corners. See http://idm-lab.org/bib/abstracts/Koen15i.html for the full paper. The research at the University of Southern California (Georgia Institute of Technology) was supported by NSF under grant numbers 1409987 and 1319966 (grant number 1335301).

RRT-Based Nonholonomic Motion Planning Using Any-Angle Path Biasing

Luigi Palmieri, University of Freiburg
Sven Koenig, University of Southern California
Kai O. Arras, University of Freiburg

RRT (LaValle and Kuffner 2001) and RRT* (Karaman and Frazzoli 2011) have become popular motion-planning techniques, in particular for high-dimensional systems such as wheeled robots with complex nonholonomic constraints. Their planning times, however, can scale poorly for such robots, which has motivated researchers to study hierarchical motion-planning techniques that grow the RRT trees in more focused ways. In a paper published at ICRA 2016 (Palmieri, Koenig, and Arras 2016), we introduced Theta*-RRT that hierarchically combines (discrete) any-angle search with (continuous) RRT motion planning for nonholonomic wheeled robots. Theta*-RRT is a variant of RRT that generates a trajectory by expanding a tree of geodesics toward sampled states whose distribution summarizes geometric information of the any-angle path. Any-angle search is a family of discrete search techniques which, unlike A* or Dijkstra's algorithm, find paths that are not constrained to grid edges. Theta*-RRT initially generates a geometrically feasible any-angle path with Theta* (Daniel et al. 2010b), using only geometric information about the workspace. Then, it computes the trajectory by growing a tree of smooth local geodesics around the any-angle path (path-biasing heuristic), satisfying the nonholonomic constraints of the system. It repeatedly samples a state x_{rand} from a subspace of the free space centered on the any-angle path, makes x_{rand} a new tree vertex, and connects it to a tree vertex which is selected among several candidate tree vertices as the one that connects with minimum cost to x_{rand}. The cost depends on the length and smoothness of the trajectory from the candidate tree vertex to x_{rand} and the geodesic distance of both vertices to the any-angle path. We showed experimentally, for both a differential drive system and a high-dimensional truck-and-trailer system, that Theta*-RRT finds shorter trajectories significantly faster than four baseline planners (namely, RRT, A*-RRT, RRT* and A*-RRT*) without loss of smoothness, while A*-RRT* and RRT* (and thus also Informed RRT*) fail to generate a first trajectory sufficiently fast in environments with complex nonholonomic constraints. We also proved that Theta*-RRT retains the probabilistic completeness of RRT for all small-time controllable systems that use an analytical steer function. Thus, Theta* is useful not only for path planning but also for motion planning. See http://idm-lab.org/bib/abstracts/Koen16d.html for the full paper. The authors thank Aurelio Piazzi for providing access to the η^4 splines closed-form expressions. This work has been supported by the EC under contract number FP7-ICT-600877 (SPENCER). Sven Koenig's participation was supported by NSF under grant numbers 1409987 and 1319966 and a MURI under grant number N00014-09-1-1031.

Optimal Target Assignment and Path Finding for Teams of Agents

Hang Ma, Sven Koenig
University of Southern California

In a paper published at AAMAS 2016 (Ma and Koenig 2016), we studied the TAPF (combined target-assignment and path-finding) problem for teams of agents in known terrain, which generalizes both the anonymous and non-anonymous multi-agent path-finding problems which research so far had concentrated on. In the TAPF problem, each of the teams is given the same number of targets as there are agents in the team. Each agent has to move to exactly one target given to its team such that all targets are visited. The TAPF problem is to simultaneously assign agents to targets and plan collision-free paths for the agents to their targets in a way such that the makespan is minimized. The TAPF problem is NP-hard to approximate within any constant factor less than 4/3 if more than one team exists (Ma et al. 2016b). It is unclear how to generalize anonymous MAPF algorithms (where all agents form one big team) to solving the TAPF problem. Straightforward ways of generalizing non-anonymous MAPF algorithms (where each agent forms a team by itself) to solving the TAPF problem have difficulties with either scalability (due to the resulting large state spaces), such as searching over all assignments of agents to targets to find optimal solutions, or solution quality, such as assigning agents to targets with algorithms such as (Tovey et al. 2005; Zheng and Koenig 2009) and then planning collision-free paths for the agents with non-anonymous MAPF algorithms (perhaps followed by improving the assignment and iterating (Wagner, Choset, and Ayanian 2012)) to find sub-optimal solutions. We presented the CBM (Conflict-Based Min-Cost-Flow) algorithm, a hierarchical algorithm that solves TAPF instances optimally by combining ideas from anonymous and non-anonymous multi-agent path-finding algorithms. On the low level, CBM uses a min-cost max-flow algorithm (Goldberg and Tarjan 1987) on a time-expanded network (Yu and LaValle 2013c) to assign all agents in a single team to targets and plan their paths. On the high level, CBM uses conflict-based search (Sharon et al. 2015a) to resolve collisions among agents in different teams. Theoretically, we proved that CBM is correct, complete and optimal. Experimentally, we showed the scalability of CBM to TAPF instances with dozens of teams and hundreds of agents. See http://idm-lab.org/bib/abstracts/Koen16e.html for the full paper. We thank Jingjin Yu for making the code of their ILP-based MAPF solver and Guni Sharon for making the code of their CBS solver available to us. Our research was supported by NASA via Stinger Ghaffarian Technologies as well as NSF under grant numbers 1409987 and 1319966 and a MURI under grant number N00014-09-1-1031.

Improved Solvers for Bounded-Suboptimal Multi-Agent Path Finding

Liron Cohen, T. K. Satish Kumar, Tansel Uras, Hong Xu, Nora Ayanian, Sven Koenig
University of Southern California

Multi-Agent Path Finding (MAPF) with the objective to minimize the sum of the travel times of the agents along their paths is a hard combinatorial problem (Yu and LaValle 2013d). Optimal MAPF solvers, such as Conflict-Based Search or CBS for short (Sharon et al. 2015b), are thus often slow. Recent work has shown that w-suboptimal MAPF solvers run faster than optimal MAPF solvers at the cost of incurring a suboptimality factor $w \geq 1$. Enhanced Conflict-Based Search or ECBS (w_1) for short (Barer et al. 2014) is a w_1-suboptimal variant of CBS (for parameter $w_1 \geq 1$). Its suboptimality factor is due to it using focal search (Pearl and Kim 1982) with parameter w_1. Larger values of w_1 result in greedier searches. CBS+HWY(w_2) (Cohen, Uras, and Koenig 2015) is a w_2-suboptimal variant of CBS (for parameter $w_2 \geq 1$). Its suboptimality factor is due to it inflating some of the admissible heuristic values by factor w_2, making use of experience graphs (Phillips et al. 2012) in form of human-generated directed highways. Larger values of w_2 allow the paths of agents to conform more to the highways. ECBS(w_1)+HWY(w_2) (Cohen, Uras, and Koenig 2015) combines both of the above approaches by using focal searches with highways and has suboptimality factor $w_1 w_2$. In a first feasibility study published at IJCAI 2016 (Cohen et al. 2016), we developed a w_1-suboptimal variant of ECBS(w_1)+HWY(w_2), Improved-ECBS or iECBS(w_1) for short, that also uses focal searches with highways but has suboptimality factor w_1 rather than $w_1 w_2$. Thus, iECBS(w_1) has only one parameter, which simplifies parameter-tuning compared to ECBS(w_1)+HWY(w_2). This parameter is, like the parameter of ECBS(w_1), the suboptimality factor, which makes it easier to compare both MAPF solvers fairly for a given suboptimality factor. Our experimental results showed that iECBS(w_1) can run faster than ECBS(w_1) despite both MAPF solvers having the same suboptimality factor. So far, the highways used with MAPF solvers had been human-generated. We also developed two first approaches for automatically generating experience graphs for a given MAPF instance that can make ECBS(w_1)+HWY(w_2) and iECBS(w_1) about as fast as human-generated experience graphs in automated warehousing domains, thus potentially reducing the dependency on human expertise for highway generation. Finally, we observed heavy-tailed behavior in the runtimes of these MAPF solvers and developed a simple rapid randomized restart strategy that can increase the success rate of iECBS(w_1) within a given runtime limit. See http://idm-lab.org/bib/abstracts/Koen16h.html for the full paper. Our research was supported by NSF under grant numbers 1409987 and 1319966 and a MURI under grant number N00014-09-1-1031.

Personalized Route Planning in Road Networks

Stefan Funke, Universität Stuttgart
Sabine Storandt, University of Freiburg

When searching for an optimal route from A to B in a street network, the notion of optimality differs vastly from person to person. Some only care about reaching their destination as quickly as possible, others want to minimize fuel consumption, avoid tolls, or prefer scenic routes. Often, a fair trade-off between several such preferences leads to the envisioned route. If every single query comes with an individual preference set, we call this a *personalized route planning query*.

Formally, given a street network $G(V, E)$, we have for each edge $e \in E$ a d-dimensional non-negative cost vector $c(e) \in \mathbb{R}^d$ (e.g. c_1 corresponding to travel time, c_2 to gas price, and so on). A personal query consists not only of source and target $s, t \in V$ but also of non-negative weights $\alpha_1, \alpha_2, \cdots, \alpha_d$, where the α_i express the importance of edge cost component i for the user. The goal is to compute a path p from s to t in G minimizing $\sum_{e \in p} \alpha^T c(e)$

We present an abstract framework for answering personalized queries efficiently. In the preprocessing phase, first an overlay graph is constructed only based on the topology of the road network. In this overlay graph, edges represent (sets of) simple paths in the original network. In the second step, cost vectors are assigned to the edges in the overlay graph, one per corresponding simple path in the original network. In the query phase, the precomputed overlay graph is instrumented to determine the optimal route by relaxing overlay graph edges where possible instead of considering the respective complete paths in the original network.

The key challenge here is the pruning of sets of high-dimensional cost vectors assigned to an edge in the overlay graph. We develop several pruning algorithms that are used to reduce sets in an optimality preserving way.

We compare several incarnations of our framework (based on Contraction Hierarchies, Customizable Route Planning and k-Path Covers) in terms of preprocessing time, query time and space consumption. It turns out that a speed-up of more than 50 compared to the Dijkstra baseline even for high dimensions can be achieved.

The full paper was originally published at ACM SIGSPATIAL 2015 (Funke and Storandt 2015).

On the Completeness of Best-First Search Variants that Use Random Exploration

Richard Valenzano, University of Toronto
Fan Xie, University of Alberta

While suboptimal best-first search algorithms like Greedy Best-First Search (Doran and Michie 1966) are frequently used when building automated planning systems, their greedy nature can make them susceptible to being easily misled by flawed heuristics. This weakness has motivated the development of best-first search variants like ϵ-greedy node selection (Valenzano et al. 2014), type-based exploration (Xie et al. 2014), and diverse best-first search (Imai and Kishimoto 2011), all of which use random exploration to mitigate the impact of heuristic error. In this paper, we provide a theoretical justification for this increased robustness by formally analyzing how these algorithms behave on infinite graphs. In particular, we show that when using these approaches, the probability of not finding a solution can be made arbitrarily small given enough time. This result is shown to hold for a class of algorithms that includes the three mentioned above, regardless of how misleading the heuristic is. The full version of this paper can be found in the Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (Valenzano and Xie 2016).

External Memory Bidirectional Search

Nathan R. Sturtevant, Jingwei Chen
University of Denver

This abstract describes the key contribution of Parallel External-Memory Meet in the Middle (PEMM) (Sturtevant and Chen 2016), an external-memory bidirectional search algorithm. PEMM is an extension of the recent MM bidirectional search algorithm (Holte et al. 2016).

Traditional uni-directional best-first search algorithms work by iteratively expanding the best node on open until a path is found or all nodes on open are exhausted. MM is a bidirectional best-first search algorithm with a novel priority, $max(f, 2g)$, that guarantees the two search frontiers will meet in the middle of the shortest path.

External memory search algorithms use disk (with high latency and high throughput) to scale the size of solvable problems beyond what would fit in RAM. Since the access latency of external memory is much higher than RAM, a key to make external-memory algorithms efficient is to batch accesses to external memory to amortize the latency of external memory access.

We assume that we cannot fit all nodes into memory and only keep a small portion of nodes being expanded in memory. A bucket refers to a group of nodes that can be loaded into memory at the same time. For the purpose of expansion efficiency, nodes with the same priority should be placed into the same bucket. When a bucket is too big to fit in memory, smaller buckets can be produced by further dividing buckets according to metrics such as the g-cost and search direction of nodes in the bucket.

Main contribution of PEMM – DSD: In the original MM algorithm solution detection (checking if the frontiers have met) is performed as soon as a node is generated, which we call immediate solution detection (ISD). In PEMM, solution detection is performed at the time of expansion; we call this delayed solution detection (DSD). PEMM expands states from the same bucket together, but the successors of these states are not necessarily in the same bucket. Thus, performing ISD would inefficiently require loading new buckets into memory with every expansion. DSD amortizes the cost of solution detection by performing solution detection for all states in a bucket at the same time. A natural time for DSD is when a bucket is expanded, since all states in a bucket perform DSD against the same buckets on disk.

The full details of PEMM can be found in Sturtevant and Chen (2016).

Acknowledgements The material described in this section is based upon work supported by the National Science Foundation under Grant No. 1551406.

Comparing Search Algorithms Using Sorting and Hashing on Disk and in Memory

Richard E. Korf, University of California, Los Angeles

We compare sorting and hashing for implicit graph search using disk storage. We first describe efficient pipelined implementations of both algorithms, which reduce disk I/O. We then compare the two algorithms and find that hashing is faster, but that sorting requires less disk storage. We also compare disk-based with in-memory search, and surprisingly find that there is little or no time overhead associated with disk-based search. We present experimental results on the sliding-tile puzzles, Rubik's Cube, and the 4-peg Towers of Hanoi (Korf 2016).

References

Bailey, J.; Tovey, C.; Uras, T.; Koenig, S.; and Nash, A. 2015. Path planning on grids: The effect of vertex placement on path length. In *Proceedings of the Artificial Intelligence and Interactive Digital Entertainment Conference (AIIDE).*

Barer, M.; Sharon, G.; Stern, R.; and Felner, A. 2014. Suboptimal variants of the conflict-based search algorithm for the multi-agent pathfinding problem. In *Proceedings of the 7th Annual Symposium on Combinatorial Search.*

Cohen, L.; Uras, T.; Kumar, T. S.; Xu, H.; Ayanian, N.; and Koenig, S. 2016. Improved solvers for bounded-suboptimal multi-agent path finding. In *Proceedings of the International Joint Conference on Artificial Intelligence*.

Cohen, L.; Uras, T.; and Koenig, S. 2015. Feasibility study: Using highways for bounded-suboptimal multi-agent path finding. In *Proceedings of the 8th Annual Symposium on Combinatorial Search*.

Coltin, B., and Veloso, M. 2014. Scheduling for transfers in pickup and delivery problems with very large neighborhood search. In *AAAI Conference on Artificial Intelligence*, 2250–2256.

Daniel, K.; Nash, A.; Koenig, S.; and Felner, A. 2010a. Theta*: Any-angle path planning on grids. *Journal of Artificial Intelligence Research* 39:533–579.

Daniel, K.; Nash, A.; Koenig, S.; and Felner, A. 2010b. Theta*: Any-angle path planning on grids. *Journal of Artificial Intelligence Research* 39(1):533–579.

Doran, J. E., and Michie, D. 1966. Experiments with the Graph Traverser Program. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences* 294(1437):235–259.

Felner, A.; Moldenhauer, C.; Sturtevant, N. R.; and Schaeffer, J. 2010. Single-frontier bidirectional search. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence, Atlanta, USA, 2010*. AAAI Press.

Felner, A.; Goldenberg, M.; Sharon, G.; Stern, R.; Beja, T.; Sturtevant, N. R.; Schaeffer, J.; and Holte, R. 2012. Partial-expansion A* with selective node generation. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, Toronto, Canada*. AAAI Press.

Ferguson, D., and Stentz, A. 2006. Using interpolation to improve path planning: The Field D* algorithm. *Journal of Field Robotics* 23(2):79–101.

Funke, S., and Storandt, S. 2015. Personalized route planning in road networks. In *Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems, Bellevue, WA, USA, November 3-6, 2015*, 45:1–45:10.

Goldberg, A. V., and Tarjan, R. E. 1987. Solving minimum-cost flow problems by successive approximation. In *Annual ACM Symposium on Theory of Computing*, 7–18.

Goldenberg, M.; Felner, A.; Stern, R.; Sharon, G.; Sturtevant, N. R.; Holte, R. C.; and Schaeffer, J. 2014. Enhanced partial expansion A*. *Journal of Artificial Intelligence Research* 50:141–187.

Hannenhalli, S., and Pevzner, P. A. 1999. Transforming cabbage into turnip: Polynomial algorithm for sorting signed permutations by reversals. *Journal of ACM* 46(1):1–27.

Hernández, C.; Asín, R.; and Baier, J. A. 2015. Reusing previously found A* paths for fast goal-directed navigation in dynamic terrain. 1158–1164. AAAI Press.

Hernández, C.; Baier, J. A.; and Asín, R. 2014. Making A* Run Faster than D*-Lite for Path-Planning in Partially Known Terrain. In *Proceedings of the 24th International Conference on Automated Planning and Scheduling (ICAPS).*

Hernández, C.; Sun, X.; Koenig, S.; and Meseguer, P. 2011. Tree adaptive A*. In *Proceedings of the 10th International Joint Conference on Autonomous Agents and Multi Agent Systems (AAMAS).*

Hernández, C.; Uras, T.; Koenig, S.; Baier, J. A.; Sun, X.; and Meseguer, P. 2015. Reusing cost-minimal paths for goal-directed navigation in partially known terrains. *Autonomous Agents and Multi-Agent Systems* 29(5):850–895.

Holte, R. C.; Felner, A.; Sharon, G.; and Sturtevant, N. R. 2016. Bidirectional search that is guaranteed to meet in the middle. In *AAAI Conference on Artificial Intelligence*.

Imai, T., and Kishimoto, A. 2011. A Novel Technique for Avoiding Plateaus of Greedy Best-First Search in Satisficing Planning. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence*, 985–991.

Karaman, S., and Frazzoli, E. 2011. Sampling-based algorithms for optimal motion planning. *International Journal of Robotics Research* 30(7):846–894.

Kiesel, S., and Ruml, W. 2016. A bayesian effort bias for sampling-based motion planning. In *26th International Conference on Automated Planning and Scheduling: Planning and Robotics Workshop*.

Koenig, S., and Likhachev, M. 2002. D* lite. In *Proceedings of the 18th National Conference on Artificial Intelligence (AAAI)*, 476–483.

Koenig, S.; Likhachev, M.; Liu, Y.; and Furcy, D. 2004. Incremental heuristic search in AI. *AI Magazine* 25(2):99.

Korf, R. 2016. Comparing search algorithms using sorting and hashing on disk and in memory. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-2016)*.

LaValle, S., and Kuffner, J. 2001. Randomized kinodynamic planning. *International Journal of Robotics Research* 20(5):378–400.

Lippi, M.; Ernandes, M.; and Felner, A. 2012. Efficient single frontier bidirectional search. In *Proceedings of the 5th Annual Symposium on Combinatorial Search (SoCS)*. AAAI Press.

Lippi, M.; Ernandes, M.; and Felner, A. 2016. Optimally solving permutation sorting problems with efficient partial expansion bidirectional heuristic search. *AI Communications* to appear.

Ma, H., and Koenig, S. 2016. Optimal target assignment and path finding for teams of agents. In *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems*, 1144–1152.

Ma, H.; Tovey, C.; Sharon, G.; Kumar, S.; and Koenig, S. 2016a. Multi-agent path finding with payload transfers and the package-exchange robot-routing problem. In *AAAI Conference on Artificial Intelligence*, 3166–3173.

Ma, H.; Tovey, C.; Sharon, G.; Kumar, T. K. S.; and Koenig, S. 2016b. Multi-agent path finding with payload transfers and the package-exchange robot-routing problem. In *AAAI Conference on Artificial Intelligence*, 3166–3173.

Moldenhauer, C.; Felner, A.; Sturtevant, N. R.; and Schaeffer, J. 2010. Single-frontier bidirectional search. In *Proceedings of the 3rd Annual Symposium on Combinatorial Search (SoCS)*. AAAI Press.

Nash, A. 2012. *Any-Angle Path Planning*. Ph.D. Dissertation, University of Southern California. http://idm-lab.org/project-o.html.

Palmieri, L.; Koenig, S.; and Arras, K. 2016. RRT-based nonholonomic motion planning using any-angle path biasing. In *International Conference on Robotics and Automation*.

Pearl, J., and Kim, J. 1982. Studies in semi-admissible heuristics. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 4:392 –399.

Phillips, M.; Cohen, B.; Chitta, S.; and Likhachev, M. 2012. E-graphs: Bootstrapping planning with experience graphs. In *Proceedings of the 8th Robotics: Science and Systems Conference*.

Ratner, D., and Warmuth, M. 1990. The $(n^2 - 1)$-puzzle and related relocation problems. *J. of Symbolic Computation* 10(2):111 – 137.

Sharon, G.; Stern, R.; Felner, A.; and Sturtevant, N. 2015a. Conflict-based search for optimal multi-agent pathfinding. *Artificial Intelligence* 219:40–66.

Sharon, G.; Stern, R.; Felner, A.; and Sturtevant, N. 2015b. Conflict-based search for optimal multi-agent pathfinding. *Artificial Intelligence* 219:40–66.

Sturtevant, N., and Chen, J. 2016. External memory bidirectional search. *International Joint Conference on Artificial Intelligence (IJCAI)*.

Sun, X.; Koenig, S.; and Yeoh, W. 2008. Generalized Adaptive A*. In *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multi Agent Systems (AAMAS)*, 469–476.

Tovey, C.; Lagoudakis, M.; Jain, S.; and Koenig, S. 2005. The Generation of Bidding Rules for Auction-Based Robot Coordination. In Parker, L.; Schneider, F.; and Schultz, A., eds., *Multi-Robot Systems. From Swarms to Intelligent Automata*, volume 3. Springer. chapter 1, 3–14.

Tovey, C. 1984. A simplified NP-complete satisfiability problem. *Discrete Applied Mathematics* 8:85–90.

Tozour, P. 2004. Search space representations. In Rabin, S., ed., *AI Game Programming Wisdom 2*. Charles River Media. 85–102.

Valenzano, R., and Xie, F. 2016. On the Completeness of Best-First Search Variants that Use Random Exploration. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, 784–790.

Valenzano, R.; Sturtevant, N. R.; Schaeffer, J.; and Xie, F. 2014. A Comparison of Knowledge-Based GBFS Enhancements and Knowledge-Free Exploration. In *Proceedings of the Twenty-Fourth International Conference on Automated Planning and Scheduling*, 375–379.

Veloso, M.; Biswas, J.; Coltin, B.; and Rosenthal, S. 2015. Cobots: Robust symbiotic autonomous mobile service robots. In *International Joint Conference on Artificial Intelligence*, 4423.

Wagner, G.; Choset, H.; and Ayanian, N. 2012. Subdimensional expansion and optimal task reassignment. In *Annual Symposium on Combinatorial Search*, 177–178.

Xie, F.; Müller, M.; Holte, R. C.; and Imai, T. 2014. Type-Based Exploration with Multiple Search Queues for Satisficing Planning. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2395–2402.

Yu, J., and LaValle, S. 2013a. Planning optimal paths for multiple robots on graphs. In *IEEE International Conference on Robotics and Automation*, 3612–3617.

Yu, J., and LaValle, S. 2013b. Structure and intractability of optimal multi-robot path planning on graphs. In *AAAI Conference on Artificial Intelligence*, 1444–1449.

Yu, J., and LaValle, S. M. 2013c. Multi-agent path planning and network flow. In Frazzoli, E.; Lozano-Perez, T.; Roy, N.; and Rus, D., eds., *Algorithmic Foundations of Robotics X, Springer Tracts in Advanced Robotics*, volume 86. Springer. 157–173.

Yu, J., and LaValle, S. 2013d. Structure and intractability of optimal multi-robot path planning on graphs. In *Proceedings of the 27th AAAI Conference on Artificial Intelligence*, 1443–1449.

Zheng, X., and Koenig, S. 2009. K-swaps: Cooperative negotiation for solving task-allocation problems. In *International Joint Conference on Artifical Intelligence*, 373–378.