A Generalization of the Shortest Path Problem to Graphs with Multiple Edge-Cost Estimates (Student Abstract)

Eyal Weiss
Bar-Ilan University, Israel
eyal.weiss@biu.ac.il

Abstract

The shortest path problem in graphs is a cornerstone of AI theory and applications. Existing algorithms generally ignore edge weight computation time. In this paper we present a generalized framework for weighted directed graphs, where edge weight can be computed (estimated) multiple times, at increasing accuracy and run-time expense. This raises a generalized shortest path problem that optimizes different aspects of path cost and its uncertainty. We describe in high-level a complete anytime algorithm for the generalized problem and discuss possible future extensions.

Introduction

The problem of finding the shortest path in a directed, weighted graph is fundamental to artificial intelligence and its applications. The *cost* of a path in a weighted graph, is the sum of the weights of its edges. Informed and uninformed search algorithms for finding *shortest* (minimal-cost) paths are heavily used in planning, scheduling, machine learning, constraint satisfaction and optimization, and more.

A common assumption made by existing search algorithms is that the edge weights are determined in negligible time. However, recent advances challenge this assumption. This occurs when weights are determined by queries to remote sources, or when the graph is massive, and is stored in external memory (e.g., disk). In such cases, additional data-structures and algorithmic modifications are needed to optimize the order in which edges are visited, i.e., the access patterns (Vitter 2001; Hutchinson, Maheshwari, and Zeh 2003; Jabbar 2008; Korf 2008a,b, 2016; Sturtevant and Chen 2016). Similarly, when edge weights are computed dynamically using learned models, or external procedures, it is beneficial to delay weight evaluation until necessary (Dellin and Srinivasa 2016; Narayanan and Likhachev 2017; Mandalika, Salzman, and Srinivasa 2018; Mandalika et al. 2019).

We present a novel approach to handling expensive weight computation by allowing the search algorithms to incrementally use multiple *weight estimators*, that compute the edge weight with increasing accuracy, but also at increasing computation time. Specifically, we replace edge weights with an ordered set of estimators, each providing a lower and upper bound on the true weight. A search algorithm may quickly compute loose bounds on the edge weight, and invest more computation on tighter estimators later in the process, solely if it is deemed necessary.

Having multiple weight estimators for edges is a proper generalization of standard edge weights, and raises several shortest-path problem variants. The classic singular edge weight is a special case, of an estimator whose lower- and upper- bounds are equal. However, since the true weight may not be known (even applying the most expensive estimator), search algorithms should address finding paths whose bounds on the shortest-path cost are optimal in some aspect.

In this paper we formally define the *shortest path tightest lower-bound* (SLB) problem, which involves finding a path with the tightest lower bound on the optimal cost. We describe in high-level the mechanics of BEAUTY, an uninformed search algorithm based on uniform-cost search (UCS, a variant of *Dijksra's* algorithm). We then use it to construct an iterative complete anytime algorithm (A-BEAUTY) which is guaranteed to solve SLB problems.

Framework and Problem Definition

Definition 1. *A* **cost estimators function** *for a set of edges E, denoted as Θ, maps every edge $e \in E$ to a finite and non-empty sequence of* weight estimation procedures*,*

$$\Theta(e) := (\theta_e^1, \ldots, \theta_e^{k(e)}), k(e) \in \mathbb{N}, \tag{1}$$

where **estimator** *θ_e^i, if applied, returns lower- and upper-bounds (l_e^i, u_e^i) on $c(e)$, such that $0 \leq l_e^i \leq c(e) \leq u_e^i < \infty$). $\Theta(e)$ is ordered by the increasing running time of θ_e^i.*

This allows us to define *estimated weighted digraphs*:

Definition 2. *An* **estimated weighted digraph** *is a tuple $G = (V, E, \Theta)$, where V, E are a set of vertices and edges, resp., and Θ is a cost estimators function for E.*

We denote the *tightest* lower bound on the edge weight $c(e)$, over all estimators in $\Theta(e)$, as

$$l_{\Theta(e)} := \max\{l_e^i | \theta_e^i = (l_e^i, u_e^i) \in \Theta(e)\}.$$

Correspondingly, we denote the tightest lower bound on a path p with length n as $l_{\Theta(p)} := \sum_{i=1}^{n} l_{\Theta(e_i)}$.

Problem 1 (**SLB**, finding l^*)**.** *Let $P = (G, v_s, V_g)$, where G is an estimated weighted digraph with cost estimators functions Θ, $v_s \in V$ is the start (source) vertex and $V_g \subset V$*

is a set of goal vertices. The **Shortest-path tightest Lower Bound** *problem (SLB) is to find a path π from v_s to any goal vertex $v \in V_g$, such that π has the lowest tightest lower bound of any path from v_s to $v \in V_g$, w.r.t. Θ, i.e., $l(\pi) = l^*$,*

$$l^* := \min_{\pi'}\{l_{\Theta(\pi')} \mid \pi' \text{ is a path from } v_s \text{ to } v \in V_g\}. \quad (2)$$

Algorithms for Shortest Path Lower Bound

BEAUTY. The algorithm receives an SLB problem and two hyper-parameters l_{est}, l_{prune}, that can be used to take advantage of prior knowledge on the optimal lower bound l^*. In the base case that no such knowledge is apriori available, the two are set to ∞, and are then ignored completely. The primary modification w.r.t. UCS is the addition of an estimation loop that takes place for each successor: another (better) estimator is only applied for an edge if the lower bound of the path to that successor has the lowest (best) lower bound to that node. Namely, a better and more expensive estimate for an edge is asked only as long as the path that includes it is still the best candidate (in terms of tight lower bound).

In the general case that l_{est}, l_{prune} are different than ∞, the former serves as a threshold on the maximum allowed tight estimation for paths, so that after crossing it only the cheapest estimations are used, and the latter serves as a threshold for pruning paths that have lower bounds greater than it. The output of the algorithm in this case is a solution, indication of optimality, and lower and upper bounds for l^*.

Depending on the hyper-parameters l_{prune}, l_{est} and their relation to l^*, BEAUTY is *complete*, *sound*, and *optimal*.

A-BEAUTY. The algorithm iteratively calls BEAUTY with increasingly tightened l_{est} and l_{prune} around l^*, until the optimal solution is found. It starts with $l_{est} = 0$ and $l_{prune} = \infty$, and each time BEAUTY terminates it returns $\underline{l}^* > l_{est}$, which is used as l_{est} in the next call. Similarly, the returned $\bar{l}^*$ is a finite value (when a solution exists) that always is greater than, or equal to, l^*, so that by using the lowest value of $\bar{l}^*$ obtained, l_{prune} is monotonically non-increasing.

The full description of the algorithms, their theoretical guarantees and extensive experimental results are available at (Weiss, Felner, and Kaminka 2023).

Conclusions and Future Work

This paper presents a generalized framework for *estimated weighted directed graphs*, where the cost of each edge can be estimated by multiple estimators, where every estimator has its own run-time and returns lower and upper bounds on the edge weight. This allows to address new problems for optimizing different aspects of path uncertainty via cost bounds. We focus on the *shortest path tightest lower-bound* (SLB) problem, which we formally define. SLB problems involve finding a path with the tightest lower bound on the optimal (possibly unknown) cost. We then present two algorithms for solving SLB problems in a guaranteed manner.

The novel framework offers numerous directions for future research. Certainly, the algorithms presented are first steps, and their performance can probably be improved, e.g., by utilizing further available information on expected estimation times of estimators. Other algorithmic approaches can be tested as well. Extensions for undirected graphs and for informed search are also of significant interest, as are novel graph search problems that are based on estimated, rather than exact, costs.

Acknowledgements

Eyal Weiss is supported by the Adams Fellowship Program of the Israel Academy of Sciences and Humanities and Bar-Ilan University President's Scholarship. The research was partially funded by ISF Grant #2306/18 and BSF-NSF grant 2017764. Thanks to K. Ushi.

References

Dellin, C.; and Srinivasa, S. 2016. A unifying formalism for shortest path problems with expensive edge evaluations via lazy best-first search over paths with edge selectors. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 26.

Hutchinson, D.; Maheshwari, A.; and Zeh, N. 2003. An external memory data structure for shortest path queries. *Discrete Applied Mathematics*, 126(1): 55–82.

Jabbar, S. 2008. *External Memory Algorithms for State Space Exploration in Model Checking and Action Planning*. PhD Dissertation, Technical University of Dortmund, Dortmund, Germany.

Korf, R. 2008a. Linear-time disk-based implicit graph search. *Journal of the ACM*, 55(6): 26–1–26–40.

Korf, R. 2008b. Minimizing disk I/O in two-bit breadth-first search. In *AAAI*, 317–324.

Korf, R. 2016. Comparing Search Algorithms Using Sorting and Hashing on Disk and in Memory. In *IJCAI*, 610–616.

Mandalika, A.; Choudhury, S.; Salzman, O.; and Srinivasa, S. 2019. Generalized lazy search for robot motion planning: Interleaving search and edge evaluation via event-based toggles. In *Proceedings of the International Conference on Automated Planning and Scheduling*, 745–753.

Mandalika, A.; Salzman, O.; and Srinivasa, S. 2018. Lazy receding horizon A* for efficient path planning in graphs with expensive-to-evaluate edges. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 28.

Narayanan, V.; and Likhachev, M. 2017. Heuristic search on graphs with existence priors for expensive-to-evaluate edges. In *Twenty-Seventh International Conference on Automated Planning and Scheduling*.

Sturtevant, N. R.; and Chen, J. 2016. External Memory Bidirectional Search. In *IJCAI-16*.

Vitter, J. S. 2001. External memory algorithms and data structures: dealing with massive data. *ACM Computing Surveys*, 33(2): 209–271.

Weiss, E.; Felner, A.; and Kaminka, G. A. 2023. A Generalization of the Shortest Path Problem to Graphs with Multiple Edge-Cost Estimates. In *Proceedings of the 2nd ICAPS Workshop on Reliable Data-Driven Planning and Scheduling*. Available at https://arxiv.org/abs/2208.11489.