Bidirectional Depth Bounded A* Search with Linear Additive Metric Memoization

Brief Problem Statement

As a solution to the problem in discussion, we need to find an optimal unicast route for the central controller in MPLS networks so that the controller can keep each Label Switching Path (LSP) on the suggested route that satisfies a list of constraints/conditions on additive linear metrics. According to the problem statement, if we map the MPLS network onto a graph, two nodes may have a bidirectional, unidirectional, or disjoint association, which means that the path for any two LSPs can go over the same nodes and links in both forward and backward directions in case of a bidirectional association, in only forward/backward direction in case of a unidirectional association, or the LSPs may not share the same node and link in case of a disjoint association. So, we can project the MPLS network on a directed and weighted graph from which we need to find an optimal path from a source node to a destination node.

To ensure optimal utilization of network resources, our designed controller may reroute LSPs to suboptimal paths either to balance uniform resource allocation over whole network or to avoid overload on certain network areas where a predefined threshold is set on monitored metrics.

Solution Components

A* Search Algorithm:

If we convert our MPLS network into a weighted graph by converting additive constraints into edge weights, we can use A* search algorithm to solve MCP and MCOP problems over the network. A* is a graph traversal and path search algorithm, which we chose due to its completeness, admissibility, optimality, and efficiency. A* is a particular case of generalizing a geometric goal-directed branch and bound approach that stores all generated nodes in memory for the use of heuristics that guide its search to achieve better performance. This informed search algorithm aims to find a path from source node to the goal node with the most negligible cost, which in our case can be available BW or Boolean ability to support specific protocol, or cumulative maximum hop limit or maximum allowable latency over the edge-path. A* can find the most optimal path by maintaining a tree of paths and extending those paths one edge at a time until it meets its termination criteria. A* decides which edges to expand at each step based on two estimates: one is the cost incurred to travel from the source node to the current node and the probable cost that will ensue if the algorithm extends the current edge all the way to the final node.

The A* algorithm has found applications in route planning in transportation engineering, path search in video games, parsing using stochastic grammars, and informational search with online learning, but I have not found any recent applications of the algorithm in MPLS Network route-finding problems [4].

Bidirectional Search:

Nicholson [1] and Dantzig [2] first proposed a bi-directional search idea for finding the shortest route between two points in a network by building two shortest-path trees alternating between the source and the destination. This method leads to significant savings in time. Kuipers and Mieghem [3] elaborately evaluated the application of bi-directional search on Quality of Service (QoS) routing when it comes to the Multi-Constrained Path (MCP) and Multi-Constrained Optimal Path (MCOP) problems.

The standard A* algorithm examines a large number of unnecessary nodes as it traverses a graph or a tree, particularly when the shortest (sub)path expands towards the destination, or the target node can be found among the leaves of a search tree. To reduce the number of unnecessary scans, we propose to deploy the algorithm over the network graph with a bidirectional scanning approach, where the algorithm starts scanning from the source node as well as from the destination node in order to reduce the complexity substantially by avoiding the scan of a large part of the network topology, resulting in higher efficiency.

Depth Bounded Search:

As we are using a bidirectional search, the A* algorithm does not need to traverse the whole tree and reach all the way to the target node from the starting node either way. We can terminate the search as soon as both instances of the A* algorithm (one starting from the source node and another starting from the destination node) reaches the same level in the tree generated from a network graph. The terminating depth will be dynamically decided during the execution-time.

State Memoization or Tabulation:

Our algorithm must try to accumulate various network states during its online run so that the controller can instantaneously move a minimum number of LSPs to alleviate scenarios where parts of the network are subject to undesired constraints. We suggest a simultaneous memoization approach to speed up the decision-making process (while putting LSPs on sub-optimal paths) by storing the metric information, path characteristics, node/edge states, constraint triggers, and flags over the A* algorithms' itinerary. When the network reaches the saved states again, the cached results can also help the controller stabilize the LSP paths, so they do not oscillate and reroute too often.

These approaches together can provide a holistic solution to our “Reroute Candidate Selection Algorithm for Network Optimization” problem. We need to determine the theoretical complexity and run some simulations by varying constraints, network size etc. to generate results and determine how our proposed approach evaluates against other state-of-the-art approaches.

Related Works

The suggested study done by Zhang et al. [5] proposed a cost-effective bidirectional K-shortest path algorithm, which served as a building block of our study to select MCOP route(s) in constraints-based MPLS networks. I relied on Delling et al. [4] for selecting A* search algorithm for this problem and Kuipers et al. [3], Bemten et al. [6] to come up with Bidirectional bounded approach for this problem.

References

1. Nicholson, T. Alastair J. "Finding the shortest route between two points in a network." *The computer journal* 9.3 (1966): 275-280.
2. Dantzig, George B. "On the shortest route through a network." *Management Science* 6.2 (1960): 187-190.
3. Kuipers, Fernando A., and Piet Van Mieghem. "Bi-directional search in QoS routing." *International Workshop on Quality of Future Internet Services*. Springer, Berlin, Heidelberg, 2003.
4. Delling, D., Sanders, P., Schultes, D., Wagner, D. (2009). Engineering Route Planning Algorithms. In: Lerner, J., Wagner, D., Zweig, K.A. (eds) Algorithmics of Large and Complex Networks. Lecture Notes in Computer Science, vol 5515. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-02094-0_7
5. Zhang, Baoxian, Jie Hao, and Hussein T. Mouftah. "Bidirectional multi-constrained routing algorithms." IEEE Transactions on Computers 63.9 (2013): 2174-2186.
6. Van Bemten, Amaury, et al. "Bounded Dijkstra (BD): Search Space Reduction for Expediting Shortest Path Subroutines." arXiv preprint arXiv:1903.00436 (2019).