O*-W: An Efficient O* algorithm to Process Optimal Sequence Traversal Query in a Complete Directed Graph

Qifeng Lu
MacroSys LLC.
Arlington, United States
qilu1@vt.edu

Kathleen Hancock
Center for Geospatial Information Technology, Virginia Polytechnic Institute and State University
Arlington, United States
hancockk@vt.edu

Abstract— **Optimal Sequence Traversal Query (OSTQ) is a graph based query that retrieves a minimum cost path that starts from a predefined vertex, traverses a set of given vertices, and ends at a predefined vertex. One of its applications is trip planning in the GeoProcessing domain in transportation. With sound theoretical support for optimally efficient, optimal, and greedy solutions, best first search in the artificial intelligence domain is effective to process such trip planning queries. O* is an existing bivariate best first search framework and its special case O*-SCDMST was proposed recently to retrieve optimal solutions for such a query. In this paper, we propose O*-W, a novel O* algorithm that uses a globally admissible heuristic to obtain optimal solutions for such a query. The performance of O*-W and O*-SCDMST in a complete directed graph whose vertices only contain the origin, the destination, and the vertices of interest, and is studied through a set of experiments, and the result demonstrates that when the number of vertices of interest is up to 15, on average, O*-W reduces computation time by more than one order of magnitude when compared to O*-SCDMST. More importantly, on average and in worst cases, O*-W increasingly outperforms O*-SCDMST when the number of points of interest increases.**

Keywords- Bivariate Best First Search, Heuristic, OSTQ, O*, O*-SCDMST, O*-W

I. INTRODUCTION AND BACKGROUND

Optimal Sequence Traversal Query (OSTQ) is a graph based query that retrieves a minimum cost path that starts from a predefined vertex, traverses a set of given vertices, and ends at a predefined vertex [1]. The graph, g-G, may include normal vertices that are neither vertices of interest, nor the given origin or destination. In the GeoProcessing domain in transportation, an OSTQ corresponds to a trip planning query that asks for a shortest or quickest path that traverses a set of locations with the given origin and destination. For example, a user may start from his/her office, go to a supermarket, a book store, a restaurant, a movie theater, and end at home.

Similar to a travelling salesman problem [2], OSTQ is NP hard [3][4]. The naïve method that retrieves the optimal solution would compute all possible order combinations of the given points of interest. The time complexity is O(n!), where n is the number of locations of interest.

O*, a bivariate best first algorithm that uses problem domain knowledge to guide the search for the optimal solution to an OSTQ in a g-G graph, was recently proposed [1]. As exact solutions, two special cases of O*, O*-SCDMST [1] and O*-Dijkstra [1], were proposed to process OSTQ in a fully connected directed graph whose vertices only contain the origin, the destination, and the vertices of interest. Such a graph is defined as *g-G*. The O*-SCDMST is proved to be optimal in a directed graph that obeys the triangle inequality and more efficient than O*-Dijkstra.

In a g-G graph, based on given rules, O* incrementally searches paths most likely to lead towards the goal state until it finds a path of minimum cost having traversed vertices of interest to the goal. O* uses a vertex's identification plus its VisitList, a sorted list that consists of a sorted sequence containing traversed vertices of interest along the path, to describe a state in the search.

A vertex in O* may have multiple states, and each state represents a different set of traversed vertices of interest. Different from states in a single-variate best first search such as A*, these states may not be compared to each other and removed accordingly unless some special state pruning rules unique to bivariate best search are followed [1].

At each step, O* uses a distance-plus-cost heuristic function, defined as *f(s)* for a state *s*, to determine the order in which the search visits states in the graph [1]:

$$f(s)=g(s)+h(s) \qquad (1)$$

where

g(s) is the cost function of the path from the initial state to the current state, and

h(s) is the global heuristic that estimates the distance from the current state to the goal state, traversing the remaining vertices of interest.

O* first takes the paths most likely to lead towards the goal state, which means the lower the *f(s)*, the higher is the priority for a state to be expanded.

Whenever an equal *f(s)* occurs, the state with a larger VisitList will be the next to expand. Otherwise, one is randomly selected. State A's VisitList, vl_A, is larger than State B's VisitList, vl_B, i.e., $vl_A > vl_B$, if the length of vl_A is longer. In other words, the path from the start state to A traverses more vertices of interest than that to B.

For an *N*-point traversal problem, O* first generates the source state that contains the given vertex and an empty VisitList. For all the states in the open list, the algorithm expands the state with the lowest f(s) value, and its children states are generated. A child state always inherits the VisitList of its parent whenever the child is not a vertex of interest; otherwise, the child's VisitList will be incremented by adding the vertex to it. The process continues until a goal state whose vertex is the final goal and VisitList contains all the vertices of interest or no solution is found. Once a goal state is reached, the algorithm will retrieve the obtained path using a data structure called backpointer, the combination of vertex identification and VisitList, to recursively obtain the parent until the origin state is reached.

Different heuristics in O* will result in different O* algorithms. The recently proposed O*-SCDMST is a special case of O* in that it uses a Semi-Connected Directed Minimum Spanning Tree (SCDMST) [1] to compute h(s) in a directed graph [1] and retrieves optimal paths.

In bivariate best first search, the global admissibility of a heuristic guarantees the solution be optimal. Global admissibility means that the global heuristic h(s) is admissible, i.e., its value is always smaller than or equal to the actual cost of the minimum-cost path from the current state to the goal state.

In this paper, O*-W, a novel O* algorithm that uses a globally admissible heuristic *H-W*, is proposed to retrieve optimal solutions for OSTQ query in a complete directed graph. Its performance is studied against O*-SCDMST, and the result shows that when the number of vertices of interest is up to 15, on average, O*-W reduces computation time by more than one order of magnitude when compared to O*-SCDMST. In addition, O* increasingly outperform O*-SCDMST on average and in worst cases.

The remaining of the paper is organized as follows. First, related work is introduced. Then, the O*-W algorithm is presented, followed by the algorithm to retrieve the heuristic for O*-W. Next, experiments and results are discussed. At last, the conclusion is provided.

II. RELATED WORK

This section provides a review of the state-of-the-art research on Traveling Salesman's Problem (TSP) and OSTQ. TSP is similar to OSTQ and the only difference is that TSP's origin and destination are the same while OSTQ's are not necessarily the same.

A. *Travelling Salesman Problem (TSP)*

The earliest TSP research is in Euclidean space that searches for a shortest round-trip route to traverse each city exactly once with all cities directly connected to each other. Solutions using dynamic programming [5], nearest neighbor [6], iterative algorithms such as 2-OPT, 3-OPT, and n-OPT [7], colony simulation [8], simulated annealing [9], Branch and Bound approach [10], were proposed to solve the problem, either exactly or approximately, and the result is a Hamiltonian cycle that visits each vertex exactly once and returns to the starting vertex. These algorithms may be adjusted to process OSTQ. However, none of these algorithms is within the best first search domain, which is the major interest of this paper.

B. *Bivariate Best First Search*

The concept of bivariate best first search was first proposed in [11] to address the deficiency of a single-variate best first search to process multiple categories of interest. It uses multiple variables to specify a state to be evaluated and expanded. L#, a generalized best first search that evaluates the promise upon a state in a similar form as A*, was proposed, together with a set of novel concepts in best first search, including local heuristic, global heuristic, local admissibility, and global admissibility [11]. As an instance of L#, the bivariate best-first-search C* was provided to processes Category Sequence Traversal Query (CSTQ) in a graph, which asks for a minimum cost route that starts from a given origin, traverses a set of ordered categories of interest that includes multiple objects in each category, with one selection from each category, and ends at a given destination [11].

As another instance of L#, O* was proposed to process OSTQ in a graph [1]. It is different from C* because their state definitions are different, and adopted data structures are different as well. O*-SCDMST and O*-Dijkstra are two special cases of O* to retrieve optimal solutions for fully connected directed graphs.

III. O*-W: AN EFFICIENT O* ALGORITHM TO PROCESS OSTQ IN A COMPLETE DIRECTED GRAPH

In this section, a global heuristic, *H-W*, as the estimate from the current state to the goal state is proposed. The O* algorithm uses *H-W* as the heuristic to retrieve optimal solutions for OSTQ is named *O*-W*.

Consider a complete directed graph, *G(V,E)*, where *V* and *E* are the set of vertices and edges, respectively, and a starting vertex *s* and an ending vertex *e* in *V*. *V* only contains *s*, *e* and the vertices of interest. Associated with each edge *(i,j)* from vertex V_i to vertex V_j in *V* is a cost *c(i,j)*. Let $|V|=n$ and $|E|=m$, *n* is larger than 2, and all the edge costs in *G(V,E)* obey the triangle inequality, which means for any triangle composed by three vertices in *V*, the sum of the costs of any two sides must be greater than or equal to the cost of the remaining side. *W*, an edge set, is defined to contain the following edges: 1) for any vertex *v* except *s* and *e* in *V*, *W* contains its incoming edge from a vertex v_x other than *e* and its outgoing edge to a vertex v_y other than *s* where v_x is not the same as v_y and the sum of the costs of these two edges are the least of all its incoming edge and outgoing edge combinations*; 2) for s, W* contains *s*'s least-cost outgoing edge, le_s, whose end vertex is not *e*; and 3) for *e*, *W* contains *e*'s least cost incoming edge, le_e, whose end vertex is not *s*. Accordingly, the total number of such edges in *W* is *2(n-1)*. Assume the total cost of these edges in *W* is *S*. Now *H-W* is defined as the following in such an environment:

For an OSTQ, assume all vertices except *s* and *e* in *V* from *G* are remaining vertices of interest, *e* is the goal state, and *s* is the current state, then the global heuristic *H-W* is *S/2*. When no vertices other than *s* and *e* exist in V, then *H-W* is *c(s,e),* the cost of the edge from *s* to *e*.

Theorem 1: *H-W* is globally admissible.

Proof:

1) When no vertices other than *s* and *e* exist in *V*, *H-W=c(s,e)*. Since *c(s,e)* is the same as the actual cost, then *H-W* is globally admissible.

2) The following is to prove that *H-W* is globally admissible when more than two vertices exist in *V*.

Assume an optimal path, *p*, is defined as $(v_0,v_1,\dots,v_{n-1})$ with the sequence of visited vertices of interest, its cost is *c(p)*, and $v_0=s$ and $v_{n-1}=e$. First, it is clear that 1) any vertex *v* other than *s* and *e* along *p* will have both an incoming edge and an outgoing edge, 2) the other end vertex of its incoming edge, v_a, is not *e*, 3) the other end vertex of its outgoing edge, v_b, is not *s*, and 4) v_a and v_b are not the same either because the graph is complete and obeys the triangle inequality. It is also true that the number of edges along such an optimal path is *n-1*. For any two consecutive edges *(j-1,j)* and *(j,j+1)* from vertex v_{j-1} to vertex v_j and then to v_{j+1} along *p*, based on the definition of the edges in *W*, the sum of the costs of both edges are not smaller than the sum of the cost of the incoming edge to *j* and the cost of the outgoing edge from *j* in *W*. Therefore, for any two consecutive edges starting from a vertex v_k on the partial path $(v_0, v_1,\dots,v_{n-3})$ of *p*, the sum of the two edges' costs is never smaller than the sum of the costs of two edges of the vertex v_{k+1} from *W* (one to v_{k+1} and the other from v_{k+1}).

Now assume all vertices on the partial path $(v_0, v_1,\dots,v_{n-3})$ of *p* are taken into account, then any edge *(j-1,j)* (*j>1* and *j<n-1*) along *p* is repeated once while each of edge *(0,1)*, i.e., $s\text{->}v_1$, and edge *(n-2,n-1)*, i.e., $v_{n-2}\text{->}e$, is counted only once. Correspondingly, all edges except edge le_s and le_e in *W* are taken into account and each is counted once. In addition, based on the definition of le_s and le_e, we know *c(0,1)*, the cost of edge *(0,1)*, is never smaller than $c(le_s)$, the cost of le_s; and *c(n-2,n-1)*, the cost of edge *(n-2,n-1)*, is never smaller than $c(le_e)$, the cost of le_e. Accordingly, assume the total weighted cost of the *n-1* edges along *p* is *c(ce)* (the cost of an edge that repeats once is doubled when to calculate the total cost),i.e., $c(ce)=c(0,1)+2c(1,2)+\dots+2c(n-3,n-2)+c(n-2,n-1)$, then the following inequations exist:

$c(0,1)>= c(le_s)$ (2)

$c(n-2,n-1)>=c(le_e)$ (3)

$c(ce)>=S- c(le_s)- c(le_e)$ (4)

since $c(0,1)+c(n-2,n-1)+c(ce)=2c(p)$ (5), based on inequations (2), (3), and (4) and equation (5),

$2c(p)>=S$, and thus

$c(p)>=S/2$.

Based on case 1) and case 2), the proof is complete.

Since *H-W* is globally admissible, based on Lemma 2 in [1], O*-W that uses globally admissible heuristics retrieves optimal solutions.

It is clear that *O*-W* is a special case of O*. Accordingly, it follows the same process as O* as discussed in Section 1. O*-W is proved globally admissible in a complete directed graph while O*-SCDMST in a fully-connected graph. Both Graphs obey the triangle inequality, and the first is a special case of the second. The other difference between O*-SCDMST and O*-W is that *O*-W* uses *W-H* as the global heuristic while O*-SCDMST uses the cost of the SCDMST tree.

IV. THE YUMEI ALGORITHM TO RETRIEVE *H-W*

The Yumei algorithm in Figure 1 is proposed to retrieve the edge set W for *H-W.* After W is obtained, the half of the total cost of its edges is *H-W*.

The time complexity of Yumei is $O(|V|^2)$ and spatial complexity is $O(|V|^2)$.

```
Input: A complete directed weighted graph with vertices V and edges E (V only contains the starting vertex s, and the ending vertex e, and vertices of interest).
Initialize: $V_{new}$ = {s}, W = {}
Calculate $le_s$ and $le_e$, $V_{new}$ = {s,e}, W = { $le_s$, $le_e$ }
Repeat until $V_{new}$ = V:
  Calculate edge (u, v) from E whose value is the minimum among all edges to v and u is not e.
  Calculate edge (v, w) from E whose value is the minimum among all edges from v and w is not s.
  If u=w:
       Calculate the edge (u',v) from E whose cost is the second least among all edges to v and u' is not e.
       Calculate the edge (v,w') from E whose cost is the second least among all edges from v and w' is not s.
       If cost(u',v)+cost(v,w)>cost(u,v)+cost(v,w'):
          Add v to $V_{new}$, add (u, v) and (v,w') to W
       Else:
          Add v to $V_{new}$, add (u', v) and (v,w) to W
  Else:
        Add v to $V_{new}$, add (u, v) and (v,w) to W
Output: $V_{new}$ and W
```

Figure 1: The pseudo code for Yumei algorithm

V. EXPERIMENT AND RESULT ANALYSIS

The purpose of the experiment is to test the performance of O*-W to retrieve optimal paths in complete directed graphs. O*-SCDMST is used as the baseline.

A. Data Set

An asymmetric TSP problem (Fischetti) with 34 points of interest [12], corresponding to vertices of interest in O*, is used as the data set for this experiment. The problem is a special case of Vehicle Routing Problem, and thus an asymmetric TSP [12]. The data set contains the edge costs between any two points in the generated complete directed graph that only contains the starting vertex, the ending vertex, and the vertices of interest. In this experiment, a set of sample OSTQ problems is generated from this data set. First, the number of points of interest consecutively changes from 2 to 15. Second, for each number of points of interest, 30 problem samples are randomly generated, i.e., the origin, the destination, and the points of interest in each problem sample are randomly selected from the 34 points. Consequently, a set of 420 problems is generated.

B. Performance Measures

To study the performances of the two algorithms, the following performance measures are identified.

Minimum Process Time (MinPT): the minimum time required to obtain a solution for each number of points of interest (seconds);

Maximum Process Time (MaxPT): the maximum time required to obtain a solution for each number of points of interest (seconds);

Average Process Time (APT): the average time required to process a query over all runs (seconds).

C. *Results and Discussion*

The results are presented in Table 1. *O*-S* represents O*-SCDMST, and *NPoI* represents the number of points of interest, i.e., the number of cities to traverse.

Figure 2 through Figure 4 are provided to visualize the performance measures provided in Table 1. Notice that the Y-Axis in Figure 3 and Figure 4 uses logarithmic scale.

In addition, it is observed that the cost of each optimal path retrieved by O*-W is the same as O*-SCDMST, which demonstrates that O*-W also retrieves optimal solutions for OSTQ processing.

Based on MinPT, O*-W can retrieve the optimal path within 2 seconds for an OSTQ of 15 NPoI. However, based on MaxPT, it may still require 871.88 seconds for another query with the same number of points of interest. This implies that O*-W's performance depends on how closely the selected H-W heuristic approaches to the actual cost.

Based on MinPT shown in Figure 2, O*-W outperforms O*-SCDMST over all runs.

Based on MaxPT shown in Figure 3, O*-W outperforms O*-SCDMST when NPoI is larger than 4. This is due to the fact that O*-W requires additional time to compute the H-W heuristic, and when NPoI becomes larger, this additional time is no longer a dominant factor, instead, the obtained heuristic expedites the overall search process. In addition, for these worst cases, O*-W increasingly outperforms O*-SCDMST when the number of points of interest increases.

Based on APT shown in Figure 4, O*-W increasingly outperforms O*-SCDMST when the number of points of interest increases. On average, O*-W can process OSTQ of up to 14 NPoI within 1 minute.

In Figure 3 and Figure 4, it is noticeable that O*-W is sub-exponential in time complexity. It is hard to decide from Figure 2 whose Y-Axis does not use logarithmic scale.

TABLE 1: PERFORMANCE RESULTS FOR O*-W AND O*-SCDMST (IN SECONDS)

	MinPT		*MaxPT*		*APT*	
NPoI	O*-S	O*-W	O*-S	O*-W	O*-S	O*-W
2	0.00	0.00	0.04	0.02	0.00	0.00
3	0.00	0.00	0.00	0.02	0.00	0.00
4	0.00	0.00	0.01	0.02	0.00	0.00
5	0.00	0.00	0.03	0.02	0.01	0.01
6	0.00	0.00	0.15	0.09	0.04	0.02
7	0.00	0.00	0.77	0.39	0.17	0.07
8	0.05	0.00	2.21	0.36	0.53	0.16
9	0.05	0.02	7.62	1.98	1.78	0.47
10	0.04	0.03	51.96	6.68	5.63	1.15
11	0.18	0.16	314.78	52.60	29.03	6.07
12	0.39	0.14	234.94	40.94	66.82	10.78
13	3.54	0.52	1109.16	199.05	245.07	30.78
14	2.06	0.87	3900.67	537.26	548.08	58.34
15	3.37	1.39	20857.75	871.88	2204.14	144.32

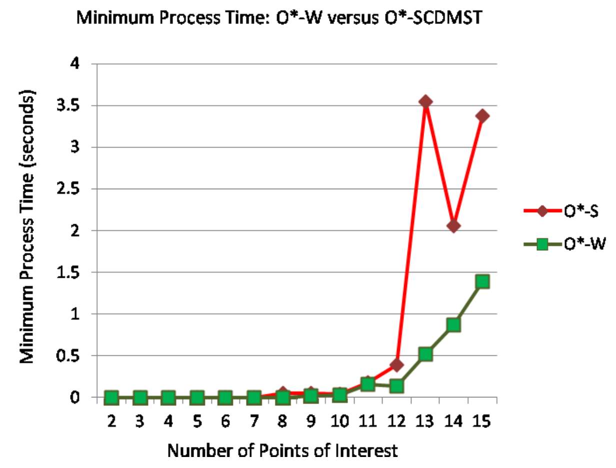

Figure 2: Minimum process time over different number of traversed points of interest: O*-W versus O*-SCDMST

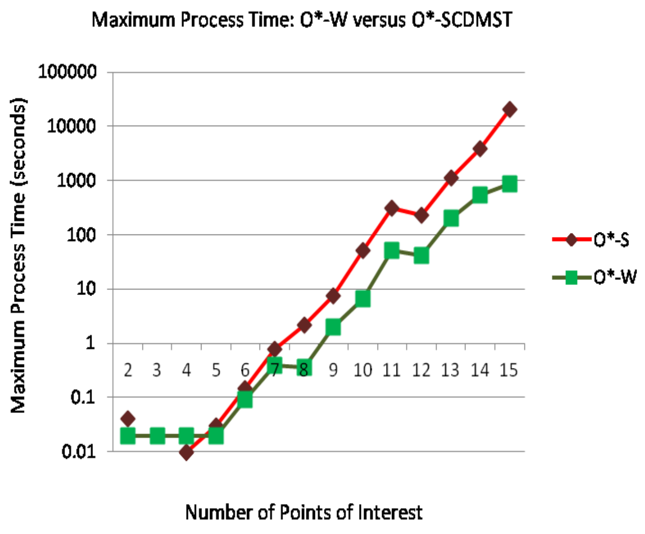

Figure 3: Maximum process time over different number of traversed points of interest: O*-W versus O*-SCDMST

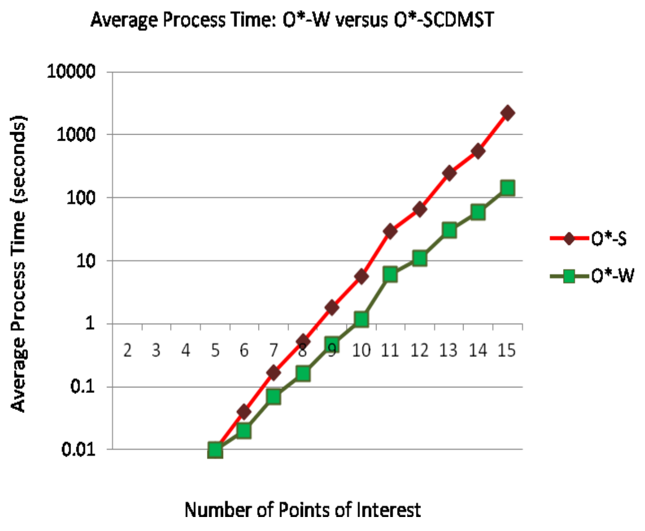

Figure 4: Average process time over different number of traversed points of interest: O*-W versus O*-SCDMST

VI. CONCLUSION

This paper proposed a novel O* algorithm, O*-W, to retrieve optimal solutions for OSTQ processing in a complete directed graph. It uses a globally admissible heuristic, H-W, to effectively and efficiently prune states. According to the experiment result, on average and in worst cases, O*-W increasingly outperforms O*-SCDMST. Even when the number of points of interest is not larger than 15, O*-W can still outperform O*-SCDMST by more than one order of magnitude, measured in either MaxPT or APT, which indicates O*-W works significantly better than O*-SCDMST.

The data size used in the experiments is rather small. To address OSTQ processing of a large data set, since it is known that the complexity to compute the pair-wise distances between all pairs of via-vertices is polynomial and much lower than the complexity of the corresponding OSTQ problem, in existing geoprocessing practices, typically these distances are pre-computed and stored in a database to expedite the process for querying any pair-wise distances.

REFERENCES

[1] Q. Lu and K. Hancock. O*: A Bivariate Best First Search Algorithm to Process Optimal Sequence Traversal Query in a Graph. geoprocessing, pp.53-61, 2011 Third International Conference on Advanced Geographic Information Systems, Applications, and Services, 2011.

[2] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms*. The MIT Press, 1997

[3] Michael R. Garey, Donald S. Johnson, and Larry Stockmeyer. *Some simplified NP-complete problems*. Proceedings of the sixth annual ACM symposium on Theory of computing, p.47-63. 1974.

[4] Wikipedia. *Travelling Salesman Problem*. http://en.wikipedia.org/wiki/Traveling_salesman_problem, last retrieved on September 20, 2011.

[5] M. Held and R. M. Karp. A Dynamic Programming Approach to Sequencing Problems, Journal of the Society for Industrial and Applied Mathematics 10(1) (1962): pp. 196–210.

[6] D. J. Rosenkrantz, R. E. Stearns, and P. M. Lewis II. An Analysis of Several Heuristics for the Traveling Salesman Problem. SIAM Journal on Computing. 6 (1977): pp. 563–581.

[7] J. L. Bentley. Fast Algorithms for Geometric Traveling Salesman Problems. ORSA Journal on Computing 4, (1992), pp. 387-411.

[8] Ma. Dorigo. Ant Colonies for the Traveling Salesman Problem. Université Libre de Bruxelles. IEEE Transactions on Evolutionary Computation, 1(1) (1997):pp. 53–66.

[9] E.H.L. Aarts and J. Korst. Simulated Annealing and Boltzmann Machines: A stochastic Approach to Combinatorial Optimization and Neural Computing. John Wiley & Sons, Chichester, 1989.

[10] J. Clausen and M. Perregaard, On the Best Search Strategy in Parallel Branch-and-Bound - Best-First-Search vs. Lazy Depth-First-Search, Proceedings of the Parallel Optimization Colloquium, (1996).

[11] Q. Lu and K. Hancock. C*: A Bivariate Best First Search to Process Category Sequence Traversal Queries in a Transportation Network. geoprocessing, pp.127-136, 2010 Second International Conference on Advanced Geographic Information Systems, Applications, and Services, 2010.

[12] Robert C. Prim: Shortest connection networks and some generalizations. In: Bell System Technical Journal, 36, pp. 1389–1401,1957