Missouri University of Science and Technology
Scholars' Mine

Opportunities for Undergraduate Research Experience Program (OURE)

Student Research & Creative Works

01 Apr 1991

Vehicle Routing using the Sweep Algorithm In Parallel

Jana G. Beattie

Carol L. Spelman

Follow this and additional works at: https://scholarsmine.mst.edu/oure

Part of the Computer Sciences Commons

Recommended Citation

Beattie, Jana G. and Spelman, Carol L., "Vehicle Routing using the Sweep Algorithm In Parallel" (1991). *Opportunities for Undergraduate Research Experience Program (OURE)*. 118.
https://scholarsmine.mst.edu/oure/118

This Presentation is brought to you for free and open access by Scholars' Mine. It has been accepted for inclusion in Opportunities for Undergraduate Research Experience Program (OURE) by an authorized administrator of Scholars' Mine. This work is protected by U. S. Copyright Law. Unauthorized use including reproduction for redistribution requires the permission of the copyright holder. For more information, please contact scholarsmine@mst.edu.

VEHICLE ROUTING USING THE SWEEP ALGORITHM IN PARALLEL

Jana G. Beattie (Hendrix College)
and
Carol L. Spelman (Principia College)

ABSTRACT

This paper presents a parallel version of the Sweep Algorithm, a heuristic solution to the single-terminal vehicle routing problem. The Sweep Algorithm uses a cluster-first, route-second approach in finding a near-optimal set of routes. The clusters of delivery points are formed by using the terminal as the center and moving around it in a sweeping fashion. After a cluster is defined, the route is found with a traveling salesperson algorithm.

The parallel version begins the sweep at different angles and performs both forward and backward sweeps. Each node handles a sweep and returns information concerning total distance traveled to the host. The host then decides which node has the best routes and requests the specific information.

The traveling salesperson algorithm is the nearest neighbor method. This starts the route at the point in the cluster nearest the terminal and proceeds by visiting the nearest unvisited point. Since this procedure is called many times in the course of finding all the routes, the quick but good results that this method yields were attractive.

Also discussed are other parallel implementations of this problem and ideas for further research.

Vehicle Routing Using the Sweep Algorithm in Parallel

Vehicle routing is a generalization of the infamous Traveling Salesman Problem. The main difference between the two is that the vehicle routing problem can have more than one "salesman." Also of concern in the vehicle routing problem is truck capacity, maximum traveling distance for the driver, simultaneous pickup and delivery, and seasonal demand Bott and Ballou point out several reasons for research in the area in their article "Research Perspectives in Vehicle Routing and Scheduling." The most telling of the reasons cited is that the "typical firm spends 10% of its sales dollar, and from one third to two thirds of its logistics budget on transportation."[1] Parallel processing provides a new arena to search for better and more efficient results.

Many algorithms exist for solving vehicle routing problems. These can be broken into three basic groups, including exact, interactive, and heuristic approaches. [1] The exact methods use mathematical programming which consumes computation time and storage space.[2] The interactive approaches require a person to formulate the routes and acts more as a "cost calculator." There are various ideas in the heuristics approaches.

First, in the heuristic category, are the insertion procedures which begin with one stop per route and combine routes until no gain is made by further recombination. Next are improvement procedures which work with routes to reduce costs while maintaining feasibility. Third is the route first, cluster second approach where a large, infeasible route of all the delivery sites is broken into clusters. Finally is the cluster first, route second approach in which the delivery sites are first clustered and then routed. [1] This is where the Sweep Algorithm falls.

The Sweep Algorithm

The Sweep Algorithm is an approach designed by Dr. Bill Gillett (University of Missouri-Rolla) and Dr. Leland Miller (Bowling Green State University) in 1970. Using the coordinates of the delivery sites and the depot, the demand at each site, the maximum distance for any one route, and the maximum capacity for a truck, the sites are broken into clusters and then routed. This is a generalized version that can be altered to meet the specific demands of a given situation. This might involve assigning distances rather than using the coordinate distances or making some other modifications to fit the situation. The algorithm generally lends itself to alterations for many variations. [1]

With the information above given, the first step is to order the sites by polar coordinates with the depot being the origin. With this ordering, the sweep begins. Sites are added to a cluster in this order until the next one cannot be added without exceeding the maximum capacity of the truck. The cluster is then sent to a TSP procedure for distance evaluation. If the distance returned from this procedure is greater than the maximum distance for a route, then sites are subtracted form the cluster until the distance constraint is satisfied. This cluster, now ordered by the TSP procedure, is considered a route.

---

[1] Bott, K. and Ballou, R. H., "Research Perspectives in Vehicle Routing and Scheduling," Transportation Research, May 1986: 239.

[2] Bott and Ballou actually cite an interesting example of a branch and bound exact method, which could provide an interesting parallel implementation. (Christofides, Mingozzi, and Toth. 1981)

Next begins an evaluation of the route. The points closest (CI) and next closest (NCI) to the route are defined, as well as the point in the route which is both closest to the depot and the next cluster (called KII). KII is chosen as such because it will be the easiest point to add to the next route. If the points CI and NCI are not in the first five sites after the current cluster, then the evaluation is complete and the route is saved as is. This is because adding either of these points to the route would only cause too much disturbance in the sweep. If the sites are within the five site limit, the route is re-evaluated including each of CI and NCI and excluding KII. If this re-evaluation provides a better route, then the better route is saved, and the sweep for the next cluster is begun. The sweep continues until all points have been routed.

Implementation

First, the algorithm was implemented sequentially in "C", as it was originally implemented in FORTRAN. To store the delivery site information, a doubly-linked list of structures was used. Each structure holds the x and y coordinates, the polar coordinates, the demand, and the list position, route and route order number for a delivery site. A doubly-linked list was implemented because of the ease of ordered insertion and dynamic storage.

Once the list is created and the elements of the structure assigned, the "sweep" is performed by proceeding down the list using the information stored in the structures. The TSP applied is a nearest neighbor algorithm. The algorithm requires starting at the site in the cluster nearest the depot and successively visiting the nearest unvisited site until all sites have been visited. This algorithm was chosen because of its ease of coding and good, fast results, as compared to several other simple TSP heuristics. [2] The latter is of the upmost importance, since the procedure is called many times in the course of the sweep. The results of the completed sweep are recorded into a file for the user.

The original version of the Sweep Algorithm was modified by Gillett and Miller to ask the user if another sweep was desired. By starting in a different position or sweeping a different direction, different routes are achieved and perhaps a better set found. This modification was implemented in the "C" version as well.

The parallel implementation involved many options. This program is in data parallel, meaning that the same program is run on each of the nodes with different data input. The algorithm had a natural inclination to this type of parallelization, because a different sweep could be done at each node. The host program creates the list from a file created by the user, and sends this list to the nodes. The host also tells each node where to begin its sweep and in which direction to proceed. The values of the starting angles and the directions are dependent on the number of nodes used. This is to say that if there are four nodes used there will be two forward sweeps from 0 and π and two backwards sweeps from π and 2π, etc. This is the obvious advantage to the parallel version, because in the time it took to run one sweep in serial many (dependant upon the number of nodes) sweeps could be performed by the parallel version.

Although the time sending messages is cause for concern, the serial program required user interaction and interpretation of the output. The parallel program makes its own decisions on the best set of routes according to specified criteria. In its current implementation the host program chooses the best set by the number of trucks used and the minimum total distance. Other criteria could be implemented as well by adding weights to the structures. The final action of the host is to create a file for the user that contains the necessary output

Conclusions

As previously mentioned, the parallel version has the obvious advantage of completing several sweeps in the time the serial version completes one sweep. One extension to the program which would provide even more benefit would be to have the nodes sweep from each site within their angle range. A node would keep track of the best set of routes internally then return that to the host. This would mean that a sweep, both forward and backward, would be done from each delivery site.

Although complete testing with the nearest neighbor TSP procedure has not been done, the results should prove to be favorable. The idea behind the TSP in this program is for it to be "quick and dirty." This means that the focus is not on optimization, but speed, because of the numerous times the procedure is called. A comparison between the program with this TSP procedure and another (perhaps the spacefilling curves heuristic discussed in [2]) would be interesting. A parallel TSP algorithm was considered, but the results were not quick or optimal. Simplicity is a key here.

Parallel implementation seems to be a good option for vehicle routing problems. The previously mentioned exact algorithm (see footnote 2) implemented in parallel would provide an interesting comparison to this algorithm. As well, a route first, cluster second approach could be organized in a parallel manner similar to the Sweep Algorithm. After each of these are implemented and tested, a recent article in the ORSA Journal on Computing gives an idea for using neural networks for choosing between different vehicle routing heuristics. [3] The article claims that vehicle routing problems with different characteristics are better solved by different heuristics. This is as well a further idea for research in parallel processing in the arena of vehicle routing.

Results

The following results were obtained using a consecutive point TSP procedure, which does not provide even close to optimal solutions. This TSP was used as a substitute due to the time constraints of the summer NSF program. Results with the nearest neighbor procedure have not been produced. The main objective of the Sweep Algorithm is the clustering and this output shows the different clusterings.

Problem

Number of Locations is 22.
Depot Coordinates are (145 , 215).
Maximum Load Capacity is 6000.
Maximum Distance is 200.

Position	X	Y	Demand
7	151	264	1100
5	159	261	700
10	130	254	800
11	128	252	1400
4	163	247	2100
8	146	246	400
3	161	242	800
9	142	239	100
2	163	236	500
6	148	232	600
12	128	231	1200
1	156	217	1300
13	129	214	1300
18	146	208	300
21	164	208	900
17	141	206	2100
15	147	193	1000
20	164	193	900
14	129	189	2500
19	155	185	1800
16	139	182	700

The clusters formed with one node are

Route 1
21 20 19 18 15 16
Route 2
17 14 13

Route 3
12 11 10 9 8 7 6
Route 4
5 4 3 2 1

Notice that the sweep was a backwards one and the switching of positions 15 and 17, which would have been accomplished in the KII re-evaluation.

The clusters formed with 4 nodes are

Route 1
1 2 3 4 5 6
Route 2
7 8 9 10 11 12 13
Route 3
14 15 16
Route 4
17 18 19 20 21

Note that the above clusters are the same ones that are formed with 16 nodes. Thus in this case the forward sweep from 0 provides the best results.

Looking at the best ratio of trucks/distance with 18 nodes gives the following clusters.

Route 1
16 15 14
Route 2
13 12 11 10 9 8 7 6
Route 3
5 4 3 2 1
Route 4
21 20 19 18 17

In this case the sweep that gave the best results was a backwards sweep starting at $3\pi/2$.

The above results show several of the scenarios that are possible from different sweeps.

References

1. Bott, K. and Ballou, R. H. "Research Perspectives in Vehicle Routing and Scheduling," Transportation Research. May 1986: 239 - 243.

2. Platzman, L. K. and Bartholdi, J. J. "Spacefilling Curves and the Planar Travelling Salesman Problem," Journal of the Association for Computing Machinery. 4 October 1989: 719 - 737,

3. Nygard, K. E., Juell, P., and Kadaba, N. "Neural Networks for Selecting Vehicle Routing Heuristics," ORSA Journal on Computing. Fall 1990: 353 - 364.