OMEGA, The Int. Jl of Mgmt Sci., Vol. 4, No. 6, 1976. Pergamon Press. Printed in Great Britain

Multi-Terminal Vehicle-Dispatch Algorithm

BILLY E GILLETT

JERRY G JOHNSON

(Received October 1975; in revised form June 1976)

This paper introduces the Multi-Terminal Sweep Algorithm, a heuristic algorithm for obtaining an approximate solution to the multiple terminal vehicle-dispatch problem. The procedure determines a set of routes by which vehicles from two or more terminals can service a collection of demand points so that the total distance traveled is kept near to the minimum. This solution also satisfies constraints on the vehicle load and on the length of each route. Application of the algorithm to eleven multiple terminal vehicle-dispatch problems shows that near-optimal solutions to large-scale problems can be found in a reasonable amount of computer time.

INTRODUCTION

THIS PAPER considers the following type of problem. A fleet of vehicles stationed at M terminals is to deliver specified quantities of a single type of product to N locations in such a manner that the total distance traveled by the vehicles is minimized. Each vehicle has the same load capacity C and the same maximum range D. Each demand point occupies a unique location and has a positive demand for goods that can be satisfied by one visit of a single vehicle. There is no limit on the number of vehicles used or the quantity of goods dispatched from a terminal.

A variety of heuristic approaches have been introduced for the solution of single-terminal vehicle-dispatch problems, but only those by Tillman and Cain [6], Wren and Holliday [7], and Golden, Magnanti and Nguyen [5] have been capable of solving a problem with more than one terminal.

THE MULTI-TERMINAL SWEEP ALGORITHM

The Multi-Terminal Sweep Algorithm employs the strategy of partitioning the multiple-terminal vehicle-dispatch problem into a collection of subproblems and solving each subproblem as a single-terminal problem. This approach reduces the amount of computation time required for obtaining a near-optimal solution by reducing the degree of complexity involved in solving each subproblem. However, an acceptable near-optimal solution usually requires that the procedure investigate a sequence of refinements or adjustments to the initial solution, because the partitioning process does not guarantee an optimal partition of the multi-terminal problem.

The requirement for the solution of many subproblems necessitates the use of a highly efficient single-terminal vehicle-dispatch algorithm. A modified version of the Sweep Algorithm, a vehicle-dispatch algorithm developed by Gillett and Miller [3], was chosen for this task. The Modified Sweep Algorithm, developed by Gillett and Johnson [4], exhibits computation times that are a linear function of the number of demand points when one considers problems having the same average number of locations per route. This property enables the Modified Sweep Algorithm to solve many types of large-scale vehicle-dispatch problems in reasonable amounts of time.

Since the Modified Sweep Algorithm is capable of producing very acceptable near-optimal solutions to single-terminal problems, the proper partitioning of demand points into clusters about individual terminals is the key to the solving of multiple-terminal problems. The optimal clustering pattern is a function of the spatial distribution of the demand points and of the load and distance constraints on the routes. The Multi-Terminal Sweep Algorithm places its principal emphasis on the spatial distribution criterion and attempts to assign demand points to terminals so that compact, widely separated clusters are formed.

Each cluster is composed of a single chain of demand points which encircles a single terminal. It is not intended that these chains of demand points represent feasible routes, but rather that they represent the arrangement of points along the periphery of the clusters. The clusters are built up by inserting unassigned demand points, one at a time, into the partially formed clusters. The Multi-Terminal Sweep Algorithm attempts to take into consideration the distance separating the clusters by minimizing, for each demand point, the extra distance that a vehicle would have to travel to include that point in the periphery about the selected terminal. This insertion cost, or the extra distance needed to include point j in the chain between points i and k, is calculated from the expression $d(i,j) + d(j,k) - d(i,k)$, where d represents the symmetric distance matrix for the problem.

The demand points are considered for assignment to terminals, so that the clusters are initially formed very close to the terminals and gradually expand outward. The minimum insertion cost is the criterion for determining the terminal to service each demand point. However, in order to resolve the assignment question in a region which is equidistant from two clusters, a second criterion is employed. If j is the demand point to be considered for assignment, the algorithm determines the unassigned demand point k which is nearest to it. An estimate of the cost of assigning both j and k to each of the two nearest clusters is then calculated. If the terminal with the minimum insertion cost for j alone also has a smaller estimated insertion cost for the two points j and k, then j is assigned to that terminal. Otherwise, the assignment of location j is deferred until after point k has again been considered for assignment. During this cluster forming process the distance constraint must be checked so that no demand point will be assigned to a terminal too distant to service it.

The initial solution, obtained by solving the single-terminal vehicle-dispatch problems defined by the initial assignment procedure, may be capable of further improvement. It is possible that the reassignment of several demand points from one terminal to another might permit a significant reduction in travel distance at one terminal, so that an overall improvement results. In order to maintain the principle of compact, widely separated clusters, a modification of the initial assignment procedure is used to perform the reassignment of demand points.

The procedure selects for possible reassignment a set of demand points that are situated in the regions between the terminals. Each terminal takes a turn as an 'attracting center' and attempts to draw these demand points away from adjacent terminals. The demand point which is a candidate for reassignment, as well as some of its neighbors in the same cluster, is declared to be unassigned and the cluster forming process begins again. If the candidate location is reassigned to the fully formed cluster about the attracting terminal, the assignment procedure continues to completion and two single-terminal vehicle-dispatch problems are solved again. If an improved solution results, the changes become part of the current best solution, and this becomes the basis for further refinement. Demand points whose reassignment results in an improved solution are removed from the list of candidates for further reassignment. Some candidate points whose reassignments do not show much promise of yielding any improvements are omitted from consideration. Candidate points are considered in an order that is determined by their bearing angles with respect to the attracting terminal. The solution which results after all terminals have served as attracting centers is a near-optimal solution to the multiple-terminal vehicle-dispatch problem.

COMPUTATIONAL RESULTS

The Multi-Terminal Sweep Algorithm was used to solve eleven multiple terminal vehicle-dispatch problems. These problems were derived from single-terminal problems found in the literature [1, 3].

Since none of the published papers that deal with multiple terminal procedures provided details of the specific problems that were solved, no exact comparison is possible. Tillman and Cain [6] provide some results for two problems with five terminals and fifty demand points. Their problems involved an average of 4·3 locations per route and an average execution time of 6·51 min on an IBM 360/50. This execution time can be compared with that of problem one, a four-terminal fifty-location problem with an average of 5·17 locations per route. Since the IBM 370/168 is roughly ten to fifteen times faster than the IBM 360/50, the 9·34 sec required to solve problem one is equivalent to about 1·5–2·4 min of IBM 360/50 time. Consequently, the multi-terminal procedure described in this paper is definitely competitive with Tillman and Cain's approach.

One important fact that emerges from the computational results is the relative insensitivity of computation time to the number of terminals in a multiple terminal problem. While a two-terminal problem requires considerably more time than a similar single-terminal problem, the computation time usually decreases as the number of terminals increase from two. Thus, all other parameters remaining the same, there appears to be no serious restriction associated with solving problems with a large number of terminals.

Problems with two terminals require a great deal of computation time because each refinement involves applying the Modified Sweep Algorithm to all N demand points. As more terminals are included in the problem, only those locations assigned to the two terminals associated with a given refinement are involved in the calculations. Thus, the more terminals in the problem the larger is the portion of locations that are not involved in the calculations. On the other hand, an increase in the number of terminals may result in an increase in the number of candidates for reassignment, and hence in an increase in execution time. The net effect observed in the eleven problems solved is that an increase in the number of terminals usually decreases the computation time.

CONCLUSIONS

The Multi-Terminal Sweep Algorithm has the following attributes:

(a) It is a new approach to the solving of the multi-terminal vehicle-dispatch problem, and can handle problems with many terminals and hundreds of demand points in a reasonable amount of time.

(b) When the average number of locations per route and the total number of locations remain constant, an increase in the number of terminals above two usually decreases the computation time.

(c) As with the single-terminal Sweep Algorithm, the computation time increases quadratically with an increase in the average number of locations per route as long as the total number of locations remains constant.

(d) If the number of terminals and the average number of locations per route remain constant, the computation time increases in proportion to the square of the total number of locations.

(e) The computation time required by the Multi-Terminal Sweep Algorithm compares favorably with that published for other multi-terminal procedures.

TABLE 1. LIST OF PROBLEMS

Problem No.	Source of parent problem	Number of terminals	Number of demand pt.	Maximum load	Maximum distance
1	Christofides and Eilon[1]	4	50	80	Unlimited
2	Christofides and Eilon[1]	4	50	160	Unlimited
3	Christofides and Eilon[1]	5	75	140	Unlimited
4	Christofides and Eilon[1]	2	100	100	Unlimited
5	Christofides and Eilon[1]	2	100	200	Unlimited
6	Christofides and Eilon[1]	3	100	100	Unlimited
7	Christofides and Eilon[1]	4	100	100	Unlimited
8	Gillett and Miller[3]	2	249	500	310
9	Gillett and Miller[3]	3	249	500	310
10	Gillett and Miller[3]	4	249	500	310
11	Gillett and Miller[3]	5	249	500	310

TABLE 2. SOLUTIONS OBTAINED BY THE MULTI-TERMINAL SWEEP ALGORITHM

Problem No.	Average No. of locations/route[1]	Number of iterations[2]	Solution	CPU sec IBM 370/168
1	5·17	1	593·1606	9·34
2	9·33	12	486·1902	13·23
3	7·82	24	652·3828	20·43
4	7·19	8	1066·651	51·24
5	13·50	10	778·8760	227·67
6	7·25	21	912·2253	53·05
7	7·25	25	939·4587	52·48
8	10·58	17	4832·039	457·61
9	10·58	26	4219·691	444·33
10	10·58	41	3821·981	344·04
11	10·22	48	3754·147	315·99

[1] Includes terminal.

[2] The initial solution and any refinement which utilized the Sweep Algorithm count as an iteration.

REFERENCES

1. CHRISTOFIDES N and EILON S (1969) An algorithm for the vehicle-dispatching problem. *Opl Res. Q.* **20**, 309–318.
2. DANTZIG GB and RAMSER JH (1959) The truck dispatching problem. *Ops. Res.* **12**, 80–91.
3. GILLETT BE and MILLER LE (1974) A heuristic algorithm for the vehicle-dispatch problem. *Ops. Res.* **22**, 340–349.
4. GILLETT BE and JOHNSON JG (1974) Sweep algorithm for the multiple depot vehicle-dispatch problem. 46th National Meeting of ORSA, San Juan, Puerto Rico.

5. GOLDEN BL, MAGNANTI TL and NGUYEN HQ (1975) Implementing vehicle routing algorithms. Tech. Report 115, O.R. Center, MIT.
6. TILLMAN FA and CAIN TM (1972) An upperbound algorithm for the single and multiple terminal delivery problem. *Mgmt Sci.* **18,** 664–682.
7. WREN A and HOLLIDAY A (1972) Computer scheduling of vehicles from one or more depots to a number of delivery points. *Opl. Res. Q.* **23,** 333–344.

ADDRESS FOR CORRESPONDENCE: *Dr. Billy E Gillett, Department of Computer Science, University of Missouri-Rolla, Rolla, MI 65401, USA.*

APPENDIX A

The data for problems 8–11 in Table 1 and the corresponding route for problem 11 using the Multi-Terminal Sweep Algorithm are given in Tables 3 and 4.

TABLE 3. PROBLEMS 8–11 OF TABLE 1

Problem	Depots	Depot coordinates
8	2	(−33, 33), (33, −33)
9	3	(70, 0), (−50, 60), (−50, −60)
10	4	(75, 0), (0, 75), (−75, 0), (0, −75)
11	5	(70, 0), (40, 80), (40, −80), (−60, 20), (−60, −20)

Loc. No.	*X*	*Y*	Demand	Loc. No.	*X*	*Y*	Demand	Loc. No.	*X*	*Y*	Demand
1	−99	−97	6	29	79	38	8	57	−28	73	100
2	−59	50	72	30	−62	−2	24	58	−76	55	62
3	0	14	93	31	−90	−68	53	59	41	42	90
4	−17	−66	28	32	52	66	13	60	92	40	57
5	−69	−19	5	33	−54	−50	47	61	−84	−29	44
6	31	12	43	34	8	−84	57	62	−12	42	37
7	5	−41	1	35	37	−90	9	63	51	−45	80
8	−12	10	36	36	−83	49	74	64	−37	46	60
9	−64	70	53	37	35	−1	83	65	−97	35	95
10	−12	85	63	38	7	59	96	66	14	89	56
11	−18	64	25	39	12	48	42	67	60	58	56
12	−77	−16	50	40	57	95	80	68	−63	−75	9
13	−53	88	57	41	92	28	22	69	−18	34	39
14	83	−24	1	42	−3	97	56	70	−46	−82	15
15	24	41	66	43	−7	52	43	71	−86	−79	4
16	17	21	37	44	42	−15	12	72	−43	−30	58
17	42	96	51	45	77	−43	73	73	−44	7	73
18	−65	0	47	46	59	−49	32	74	−3	−20	5
19	−47	−26	88	47	25	91	8	75	36	41	12
20	85	36	75	48	69	−19	79	76	−30	−94	3
21	−35	−54	48	49	−82	−14	79	77	79	−62	8
22	54	−21	40	50	74	−70	4	78	51	70	31
23	64	−17	8	51	69	59	14	79	−61	−26	48
24	55	89	69	52	29	33	17	80	6	94	3
25	17	−25	93	53	−97	9	19	81	−19	−62	52
26	−61	66	29	54	−58	9	44	82	−20	51	99
27	−61	26	5	55	28	93	5	83	−81	37	29
28	17	−72	53	56	7	73	37	84	7	31	12

TABLE 3—*contd*

Loc. No.	X	Y	Demand	Loc. No.	X	Y	Demand	Loc. No.	X	Y	Demand
85	52	12	50	140	−5	−39	55	195	−55	39	19
86	83	−91	98	141	8	−22	45	196	70	−14	90
87	−7	−92	4	142	−61	−76	100	197	0	95	35
88	82	−74	56	143	76	−22	38	198	−45	7	76
89	−70	85	24	144	49	−71	11	199	38	−24	3
90	−83	−30	33	145	−30	−68	82	200	50	−37	11
91	71	−61	45	146	1	34	50	201	59	71	98
92	85	11	98	147	77	79	39	202	−73	−96	92
93	66	−48	4	148	−58	64	6	203	−29	72	1
94	78	−87	36	149	82	−97	87	204	−47	12	2
95	9	−79	72	150	−80	55	83	205	−88	−61	63
96	−36	4	26	151	81	−86	22	206	−88	36	57
97	66	39	71	152	39	−49	24	207	−46	−3	50
98	92	−17	84	153	−67	72	69	208	26	−37	19
99	−46	−79	21	154	−25	−89	97	209	−39	−67	24
100	−30	−63	99	155	−44	−95	65	210	92	27	14
101	−42	63	33	156	32	−68	97	211	−80	−31	18
102	20	42	84	157	−17	49	79	212	93	−50	77
103	15	98	74	158	93	49	79	213	−20	−5	28
104	1	−17	93	159	99	81	46	214	−22	73	72
105	64	20	25	160	10	−49	52	215	−4	−7	49
106	−96	85	39	161	63	−41	39	216	54	−48	58
107	93	−29	42	162	38	39	94	217	−70	39	84
108	−40	−84	77	163	−28	39	97	218	54	−82	58
109	86	35	68	164	−2	−47	18	219	29	41	41
110	91	36	50	165	38	8	3	220	−87	51	98
111	62	−8	42	166	−42	−6	23	221	−96	−36	77
112	−24	4	71	167	−67	88	19	222	49	8	57
113	11	96	85	168	19	93	40	223	−5	54	39
114	−53	62	78	169	40	27	49	224	−26	43	99
115	−28	−71	64	170	−61	56	96	225	−11	60	83
116	7	−4	5	171	43	33	58	226	40	61	54
117	95	−9	93	172	−18	−39	15	227	82	35	86
118	−3	17	18	173	−69	19	21	228	−92	12	2
119	53	−90	38	174	75	−18	56	229	−93	−86	14
120	58	−19	29	175	31	85	67	230	−66	63	42
121	−83	84	81	176	25	58	10	231	−72	−87	14
122	−1	49	4	177	−16	36	36	232	−57	−84	55
123	−4	17	23	178	91	15	84	233	23	52	2
124	−82	−3	11	179	60	−39	59	234	−56	−62	18
125	−43	47	86	180	49	−47	85	235	−19	59	17
126	6	−6	2	181	42	33	60	236	63	−14	22
127	70	99	31	182	16	−81	33	237	−13	38	28
128	68	−29	54	183	−78	53	62	238	−19	87	3
129	−94	−30	87	184	53	−80	70	239	44	−84	96
130	−94	−20	17	185	−46	−26	79	240	98	−17	53
131	−21	77	81	186	−25	−54	98	241	−16	62	15
132	64	37	72	187	69	−46	99	242	3	66	36
133	−70	−19	10	188	0	−78	18	243	26	22	98
134	88	65	50	189	−84	74	55	244	−38	−81	78
135	2	29	25	190	−16	16	75	245	70	80	92
136	33	57	71	191	−63	−14	94	246	17	−35	65
137	−70	6	85	192	51	−77	89	247	96	−83	64
138	−38	−56	51	193	−39	61	13	248	−77	80	43
139	−80	−95	29	194	5	97	19	249	−14	44	50

TABLE 4. MULTI-TERMINAL SWEEP ALGORITHM SOLUTION OF PROBLEM ELEVEN OF TABLE 1

Terminal	Route No.	Route	Distance	Load
1	1	92–178–210–41–110–60–158–109	114·2307	472
1	2	20–227–29–97–132–59–171–105	140·6073	485
1	3	85–169–181–162–75–52–243–16–6–165	145·9844	463
1	4	222–37–116–126–215–74–104–141–25–199–44–111	179·0652	489
1	5	196–174–143–14–107–128–48–23–120–22–236	121·2726	459
1	6	98–240–117	68·91753	230
2	1	17–24–40–127–147–159–134–51–67–32	172·2452	449
2	2	176–233–39–102–15–219–136–226–78–201	131·6328	499
2	3	242–225–235–82–157–43–223–122–38	146·1150	496
2	4	197–42–10–238–131–214–57–203–11–241–56	166·0002	488
2	5	175–66–80–194–113–103–168–47–55	84·93602	357
3	1	245–88–247–149–86–151–94–119	136·0695	493
3	2	192–184–218–50–77–212–45–91–144	141·7722	435
3	3	216–46–93–187–161–179–200–63–180–152	124·4228	491
3	4	156–28–95–188–87–34–182–35–239	128·5520	439
4	1	69–177–237–249–62–146–84–135–3–118–123–190	163·5731	486
4	2	163–224–64–125–193–101–114–195	131·1170	485
4	3	2–170–148–26–13–167–89–153–9–230–27	157·7891	472
4	4	183–58–248–121–106–189–150–36	165·1048	499
4	5	217–83–220–206–65–53–228–137–173	148·6997	490
4	6	204–198–73–96–112–8–213–166–207–30–18–54	141·8514	500
5	1	19–185–140–7–246–208–160–164–172–72	187·5112	450
5	2	186–81–4–115–145–100–21	136·4537	471
5	3	138–209–244–108–154–76–155–70–99–234–33	186·4304	496
5	4	79–68–142–232–231–202–139–1–229–71–31–205	207·8954	487
5	5	211–90–61–221–129–130–124–49–12–133–5	112·4839	431
5	6	191	13·41641	94
		Total 3754·147		