JOURNAL OF AEROSPACE COMPUTING, INFORMATION, AND COMMUNICATION
Vol. 4, October 2007

Minimum-Time Dispatch Problem: Formulation and Solution

Ayman H. Kassem*
King Fahd University of Petroleum and Minerals, Dhahran 31261, Saudi Arabia

DOI: 10.2514/1.32742

This paper presents the formulation and the solution procedure for a minimum-time dispatch problem. The problem can be summarized as finding the minimum time and shortest path to transfer time-sensitive materials from a dispatch center to specific targets or vice versa. It is a famous problem with many civil and military applications. The formulation consists of two stages. The first stage is an iterative k-mean algorithm to cluster targets to lie within a specific distance. The second stage is a shortest-path traveling salesman problem, solved using a genetic algorithm. Test cases confirm the efficiency and applicability of the proposed approach.

Nomenclature

$d(i, j)$	distance between i and j
D	The total length of a tour
K	Number of stops in k-means algorithm
n	Number of targets
v	stop, station or strategic point
x	problem size
Y	Number of iterations

I. Introduction

THE problem, which we call the Minimum Time Dispatch Problem (MTDP), can be summarized as follows: "*find the minimum time and the shortest path to transfer time-sensitive materials from a dispatch center to specific targets or vice versa*." This problem has many varieties and applications. Civil applications include designing the path and location of train stations of a railway which is the closest to a specific number of cities. Another application is for food or fuel industries, where it is required to find the collecting-points which are close to the raw material producers to guarantee transportation safety of hazardous materials in case of fuel industry, or food reservation in case of food industry. It is also required to find the shortest path for the collecting vehicle to save time and money. The distribution of mailboxes and the mail-collection route is another good application of the proposed problem.

For military applications, the dispatch and collection of troops are famous problems. It is safer and faster to have a small number of rendezvous points which lie within specific ranges from troops-operations than visiting all troops-operations points as shown in Fig. 1. Visiting all points, will increase the vulnerability of the troops-carrier during repeatedly takeoff and landing. All these applications and many others lie in the same mainstream of the following proposed procedure.

This problem can be formulated as two-stage procedure. The first stage is an iterative k-means algorithm. The k-means algorithm is used to cluster targets, depending on their spatial distribution, to a specific number of

Received 11 June 2007; revision received 10 September 2007; accepted for publication 10 September 2007.

* Assistant Professor, Aerospace Engineering Department, AIAA Member.

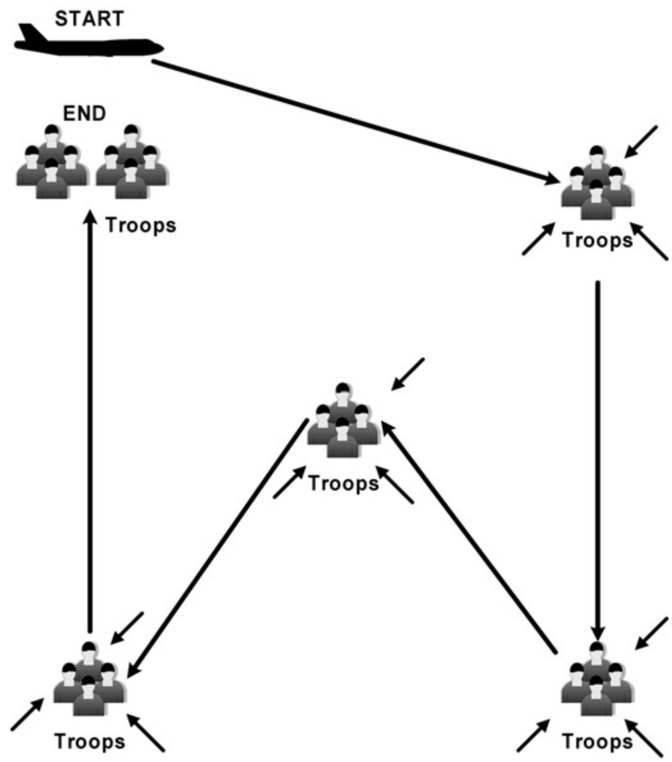

Fig. 1 Troops dispatch.

stops/stations, so that each target is attached to its closest station. The iterative k-mean solves the k-mean problem when the number of stops is unknown, but instead, the maximum distance allowed between a stop and a target is given. In this case, the k-means algorithm is iterated while increasing the number of stops till the maximum allowed distance-constraint is satisfied. The second stage is a traveling salesman problem (TSP), solved with a genetic algorithm (GA), to find the shortest path between the final number of stops results from stage-one. Section two gives the formal introduction to k-means algorithm, TSP, and MTDP respectively. Section 3 discusses the use of GA for solving TSP. The results obtained and the test cases are documented in section 4, followed by section 5, which concludes and provides future work.

II. Basic Definitions

A. The K-means Problem

K-means algorithm is a post-clustering technique, which is widely used in data mining, image processing and pattern recognition.[1,2] For spatial problem, it can be summerized as follows:

Problem: Given a set of n locations (targets), arrange this set into K subsets according to its spatial closeness. That is, each element in the same subset shares similar proximity characteristics as every other element in the subset. The steps for k-means algorithm are given in Table 1.

B. The Traveling Salesman Problem (TSP)

The idea of TSP is simple. Given a graph, travel along the edges of the graph in such a way that each vertex is visited exactly once and end up back to the starting vertex. This is a Hamiltonian cycle of the graph. If each graph edge has a weight or distance, then the sum of the weights along the cycle is the price (cost) of this particular tour. There is a brute-force solution, where we just look at all possible paths around the vertices of the graph, that costs $O((n-1)!)$. For general graphs, this problem is NP complete. The TSP can be formulated as follows:[3]

Table 1 Steps in k-means algorithm.

1. Pick K places (stops/stations) randomly as k means.
2. Compute the distance of n targets to each mean.
3. Quick sort the list of the resulting $K * n$ distances with respect to the distance.
4. Arrange each target according to its closest mean stop.
5. Compute another k means from the K subsets from Step 4.
6. If all k means are not stable (i.e., it did not reach a settling value), go to step 2. Otherwise, halt and return solution.

Table 2 MTDP algorithm.

1. n-targets are given with their spatial coordinates and the maximum travel-distance constraint.
2. The k-means algorithm is initialized and executed with n-targets and K-stops till convergence.
3. The distance constraint is checked. If it is ok, then go to Step 5.
4. Increment the number of stops (K) and go to Step 2.
5. Given the stops/stations spatial locations, run the traveling salesman problem to find the shortest path.
6. Return the solution (location of optimum stops and the shortest path).

TSP Instance: A finite set V of stops/stations $(v_1, v_2,\ldots,v_m)$ and a "distance" $d(v_i, v_j) \in R+$ for each pair of strategic points $v_i, v_j \in V$.

Question: What is the ordering of all strategic points in $V \langle v_{\pi(1)}, v_{\pi(1)}, \ldots, v_{\pi(m)} \rangle$ which minimizes the total length of the tour given by:

$$D = \sum_{i=1}^{m-1} d(v_{\pi(i)}, v_{\pi(i+1)}) + d(v_{\pi(m)}, v_{\pi(1)}) \tag{1}$$

C. The Minimum Time Dispatch Problem Formulation

The problem can be formulated as solving an iterative k-means problem followed by a traveling salesman problem as described in the following algorithm given in Table 2.

III. Genetic Algorithms

In 1975, Holland[4] introduced genetic algorithms (GAs). Genetic Algorithms are stochastic global search techniques based on the mechanics of genetics. Roughly, a genetic algorithm works as in Table 3. Further description of genetic algorithms can be found in Goldberg.[5]

Table 3 The pseudo-code for GA.

BEGIN GA
Make initial population at random.
WHILE NOT (*stopping condition*) DO
BEGIN
Select parents from the population.
Produce offspring from the selected parents (crossover).
Mutate the individuals.
Extend the population adding the offspring to it.
Reduce the extended population.
END
Output the best individual found.
END GA

IV. Genetic Algorithms for TSP

The first researcher to tackle the Traveling Salesman Problem with genetic algorithms was Brady.[6] His work was followed by Grefenstette et al.[7] and Oliver et al.[8] There are many different representations for solving the TSP problem using genetic algorithms. For example, binary representation and matrix representation, which use binary alphabets for tour representation. Although binary alphabets constitute the standard way of the representation of genetic algorithms in the TSP problem, the crossover and mutation operators do not constitute closed operations, which means that the results obtained using the above mentioned operators are not valid tours (i.e., some cities are visited more than once). This is the reason why repair operators must be used in such representations.

The most natural representation of one-tour is denominated by path representation. In this representation, the n cities that should be visited are put in order according to a list of n elements, so that if the city i is the j-th element of the list, city i is the j-th city to be visited. Hence, the tour: city 3 $\Rightarrow$ city 2 $\Rightarrow$ city 4 $\Rightarrow$ city 1 $\Rightarrow$ city 7 $\Rightarrow$ city 5 $\Rightarrow$ city 8 $\Rightarrow$ city 6 is simply represented as (3 2 4 1 7 5 8 6). This representation has allowed a great number of crossover and mutation operators to be developed. We can affirm nowadays that most of the TSP approximations using genetic algorithms are realized with this representation. The fundamental reasons lie in its intuitive representation, as well as in the quality of the results obtained.

Since TSP is using path representation, the classical operator crossover and mutation operators are not suitable. Other crossover and mutation operators have to be defined. One of these methods, the partially-mapped crossover (PMX) operator, was suggested by Goldberg and Lingle.[9] It passes visiting-order and city-number information from the parent tours to the offspring tours. A portion of one parent's string is mapped onto a portion of the other parent's string and the remaining information is exchanged. A correction procedure is then required because it might be the case that the child has duplicate cities. If a city is represented twice in the child (one originally, and one from the swapped sub-string), simply replace the first occurrence with a city that got swapped away.

We have implemented a variation of PMX which uses a modified crossover. It can be considered a combination of PMX and three-point crossover (three adjacent points) done sequentially (i.e. one crossover position at a time), which is implemented by doing one-point crossover randomly and repeating the process for three times. It was found that this method is better than both one-point crossover and three-point crossover.

Consider the following example. For the two parents, assume parent 1 is (1 2 3 4 5 6), and parent 2 is (6 4 2 5 3 1). Select a position randomly (say 3); then switch the two cities in both parents. The result would be (1 2 2 4 5 6) and (6 4 3 5 3 1). Now we have repeated cities (city 2 in child one and city 3 in child two). A correction procedure, modified from Goldberg[9] version, is then used to correct this and the results are (1 3 2 4 5 6) and (6 4 3 5 2 1). This procedure is repeated for three times. Similarly, the mutation operator can be performed in the same way but the switching and the correction is done only once.

The MTDP was implemented using MATLAB. Different GA parameters were studied as shown in the next section.

V. Case Study

GA algorithm was tested for the TSP problem on a number of graphs with different types, such as regular graphs and random graphs with much more difficult structures; and satisfactory results were obtained. The area is selected to be 100 units * 100 units.

The result for the case of 20-stops (population size $\mu = 1000$, number of generations $= 10^6$) is shown in Fig. 2. It has been tested for 100 times and the minimum tour was reached in all test cases (i.e., success rate of 100%).

The full MTDP procedure is tested using a special graph as shown in Fig. 3. The targets (diamond shapes) are randomly distributed on the circumference of small circles (circles with double lines) and the centers of these circles are located on the circumference of a larger one. The optimal solution is the dotted polygon connecting the centers of small circles.

Different problem sizes were tested and the results agreed with the optimal solution. Figures 4, 5 represent two test cases, one with 10 stops and 200 targets and the other with 20 stops and 800 targets. In both cases the procedure was able to capture the optimal path (the dotted line) and the optimal stops (black circles).

From the test problems, it was clear that TSP dominates the execution time for MTDP. So a thorough study was done to find a good estimate for the required number of iterations per problem size for the TSP. This number could

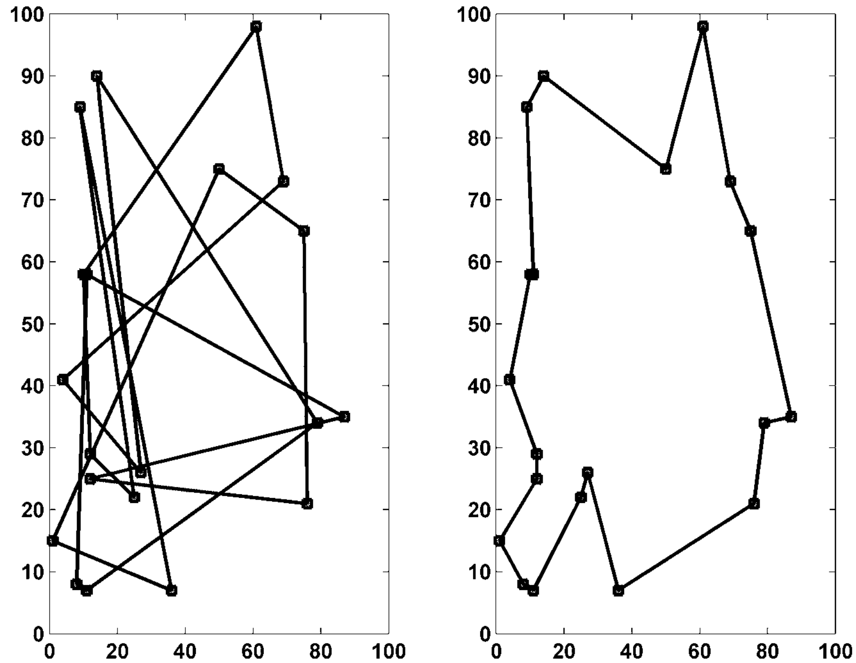

Fig. 2 GA results for TSP problem with 20 stops.

be used as a guide for users instead of trial and error. The relation is given by Eq. 2

$$Y = e^{(1.12x-2.17)} \tag{2}$$

where Y is the number of iterations = (population size $*$ number of generations), and x is the problem size (i.e., number of cities in the TSP). Using Y from Eq. 2 will guarantee that the GA will find a tour within 10% error from the minimum tour.

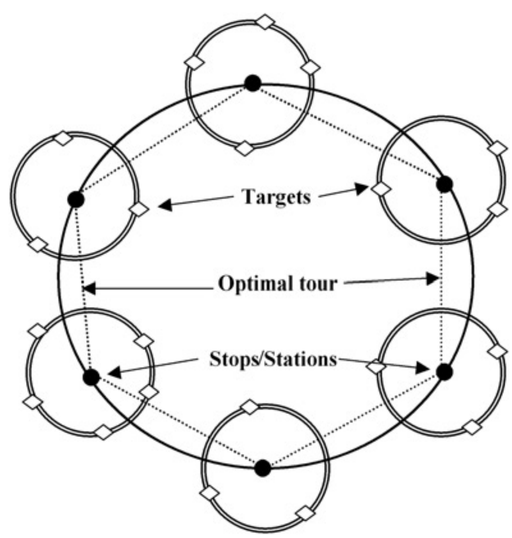

Fig. 3 A special graph for testing MTDP.

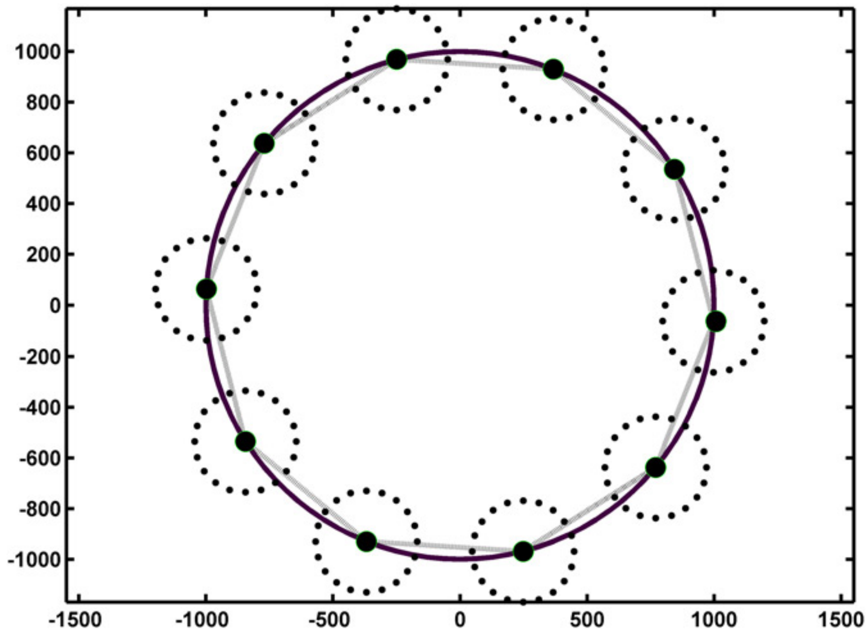

Fig. 4 Test Case 1 (10 stops and 200 targets).

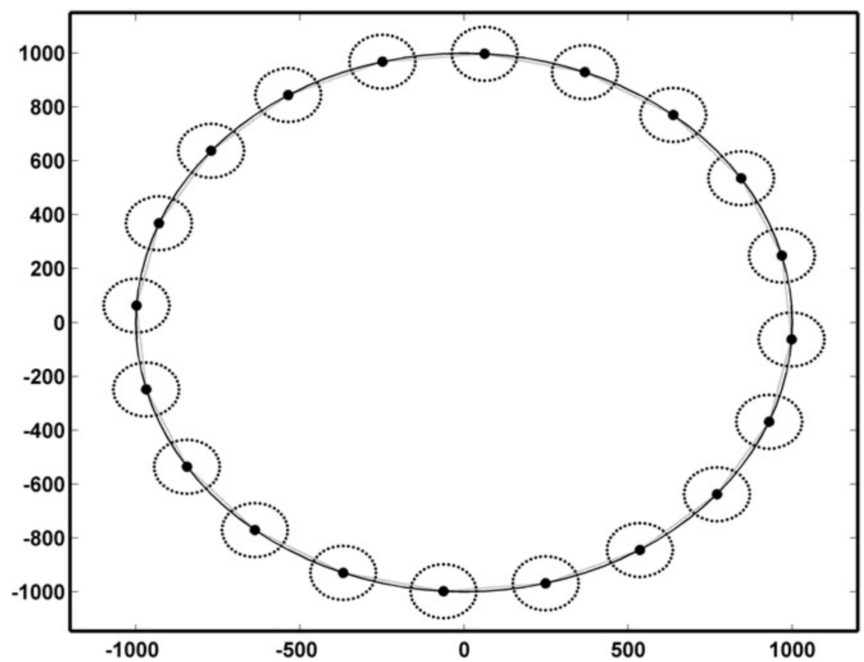

Fig. 5 Test Case 2 (20 stops and 800 targets).

VI. Conclusions

In this paper, MTDP was presented. This problem could be formulated as an iterative k-means clustering problem followed by a TSP. MTDP has many civil and military applications and the proposed procedure can be easily modified to suite different problems and variations. Test cases showed the efficiency and reliability of the proposed solution procedure. Future research will extend the formulation to include more applications and will examine different optimization techniques to speed up the solution process.

Acknowledgments

The author acknowledges the support of King Fahd University of Petroleum and Minerals, Dhahran, Sauid Arabia, for this work.

References

[1]Ralambondrainy, H., "A Conceptual Version of the K-means Algorithm," *Pattern Recognition Letters*, Vol. 16, 1995, pp. 1147–1157.

[2]Tapas Kanungo, David M. Mount, Nathan S. Netanyahu, Christine D. Piatko, Silverman, R., and Angela Y. Wu, "An Efficient K-Means Clustering Algorithm: Analysis and Implementation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 24, No. 7, July 2002, pp. 881–892.

[3]Stinson, D. R., *An Introduction to the Design and Analysis of Algorithms*, The Charles Babbage Research Center, Winnipeg, Manitoba, Canada, 2nd ed, 1987.

[4]Holland, J., "*Adaptation in natural and artificial systems*," The University of Michigan Press, 1975.

[5]Goldberg, D., *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, 1989.

[6]Brady, R., "Optimization Strategies Gleaned from Biological Evolution," *Nature*, Vol. 317, 1985, pp. 804–806.
doi: 10.1038/317804a0

[7]Grefenstette, J., Gopal, R., Rosmaita, B., and Van Gucht, D., "Genetic Algorithms for the TSP," *Proceedings of the First International Conference on Genetic Algorithms and Their Applications*, Lawrence Erlbaum, Hillsdale, New Jersey, 1985, pp. 160–165.

[8]Oliver, I.M., Smith, D.J., and Holland, J.R.C., "A Study of Permutation Crossover Operators on the TSP," in *Genetic Algorithms and Their Applications: Proceedings of the Second International Conference*, New Jersey, 1987, pp. 224–230.

[9]Goldberg D., and Lingle, R., "Alleles, Loci, and the Traveling Salesman Problem," *International Conference on Genetic Algorithms*, 1985.

Christopher Rouff
Associate Editor