Survey of Traveling Salesman Problem

金 昌 恩*

ABSTRACT

Two different algorithms for traveling salesman problem(TSP) will be discussed. One is the engineering approach to the TSP. The other one is Branch-and-Bound algorithm to take advantage of the special structure of combinational problems. Also a generalization of TSP will be presented.

1. INTRODUCTION

The problem of finding the shortest route which visits each of agiven collection of "cities", finally returning to the city from which it began, is traditionally called the Traveling Salesman Problem(TSP). Such problems occur in a variety of contexts, e.g., delivery routes for a product to different stores in a city, security guard inspections of locations in a factory collecting the money from coin telephones, etc.. The objective of TSP is to find the optimal value of the "decison variables"; that is, those variables which can be controlled within the problem structure. This approach is called multistage problem solving, and dynamic programming is a systematic technique for reaching an answer in problems of this nature. Many techniques exist for solving various optimization problems. Numerous algorithms have been developed to solve both linear and anonlinear objective functions subject to various constraint configurations.

The man primarilly responsible for the current popularity of dynamic programming is Richard Bellman. Bellman first developed the concepts of dynamic programming in the late 1940's and early 1950's while working as a member of the Rand Corporation. As Bellman and his associates began to proliferate the techniques and methodologies of dynamic programming, important contributions were made by other authors. Aris, Nemhauser, Wiled, Mitten, Denardo, Dreyfus, and Beightler all independently contributed to the mathematical properties of the dynamic programming approach.

We will discuss two different algorithms for TSP. One is the Engineering Approach to the TSP.[3] The other one is the Branch-and-Bound Algorithm to take advantage of the special structure of combinatorial problems.[5] Also a generalization of TSP will be presented.[4]

2. AN ENGINEERING APPROACH TO THE TRAVELING SALESMAN PROBLEM

2.1 Summary of Engineering Approach

By an engineering approach to the TSP is intuitively to the non-mathematician, and takes advantage of and builds on any knowledge we may have of the problem. The engineering approach method is not a single algorithm, but rather a sequence of steps and programs designed to generate successively smaller cost circuits.

*명지대학교 산업공학과 조교수

접수일 : 1990. 11. 5.

Briefly, an engineering approach procedure is first to order the cost matrix and its companion row index matrix to generate the good start or nominal feasible trial arrays are not random or abitrary, but, rather, possess the property of having many low cost adjecent pairs of cities. As an improvement procedure(step 2), each nominal feasible array is permuted to generate a "nominal feasible least cost" array consistent with the nominal feasible array and the permutation algorithm. A statistical analysis of frequency of pairings of the cities in the set of lower cost "nominal feasible least cost" arrays determines a number of pairs of cities that we conjecture would most likely appear in the grobal minimum array. With certain cities paired off, the number of combinatorial possibilities is so reduced that the graph of the network can be generated exhaustively and rapidly.

2.2 Statement of problem

Given a set of integers(cities) from 1 to n, $[1, i_2, i_3, \cdots, i_n]$ which minimizes the sum

$$C_{1.\ i_2}+C_{i2,\ i3}+C_{i3.\ i4}+\cdots+C_{IN.1}.$$

A feasible permutation or path begins and ends with the same city. All other cities in the set must appear once and only once. The cost matrix is symmetric $C_{ij}=C_{ji}$.

2.3 Permutation Algorithm

The permutation algorithm consists of pair-wise interchanges of elements of an array. While the algorithm is a kind of optimization in policy space of dynamic programming, it is not necessary to understand dynamic programming to perform the algorithm.[3]

If we define
f_i=cost of the i'th array, $[A_i]$
$[P_j]$=set of possible interchanges(set of policies)
of the elements of the array$[A_i]$,

the algorithm may be described mathematically by the recursive formula

(1) $f_{i+1} \leqslant M_{in[PJ]}f$ where f_0=nominal feasible array.

The algorithm states that given array $[A_{i+1}]$ will have a cost f_{i+1} which is $[A_{i+1}]$ is found, the recursive formula(1) is reapplied to generate the array $[A_{i+2}]$, etc., The FORTRAN program executes the algorithm very quikly to get a feasible solution.

Consistent with the nominal trial array and the pair-wise interchange rules, the permutation algorithm finds the least cost array, which is in general a local minimum. This approach is to apply the pair-wise interchange algorithm to a variety of nominal trial arrays to generate the conjectured grobal minimum.

2.4 Discussion

The main idea of the engineering approach to the TSP is to reduce the number of combinatorial branching possibilities at each city. The programs work automatically whithout manual intervention and can handle problems of virtually any size. The decision making and analysis step include the selection of the number of nominal feasible arrays; the selection of the number set of smallest "nominal feasible least cost" arrays; the frequency level above which pairs of cities are and the smallest "nominal feasible least" array. The engineer can make these decisons simply and quickly.

3. BRANCH AND BOUND APPROACH TO THE TRAVELING SALESMAN PROBLEMS

3.1 Summary of Branch and Bound Method

The Branch and Bound Algorithm is the most widely used method for solving bolth pure and mixed integer programming problems in practice. Most computer codes for solving integer programs are based on this approach. Basically the Branch and Bound Algorithm is just an efficient enumeration procedure for examining all possible integer feasible solutions. The efficiency of computations can be enhanced by introducing the concept of bounding. This concept indicates that if the continuous optimum solution of a subproblem yields a worse objective value than the one associated with the best available integer solution, it does not pay to explore the subproblem any further. In other words, once a feasible integer solution is found, its associated objective can be used as a bound to discard inferior subproblems.

But there is no definite "best" way for selecting the branching variable or the specific sequence in which the subproblem must be scanned.

3.2 Formulation

Using Branch and Bound Method, we can constract the formulation as follows to characterize the traveling salesman problem.

(1) $\text{Min} \sum_{i=1}^{n} \sum_{j=1}^{n} C_{ij} X_{ij}$ where $C_{ii}=\infty$

subject to

(2) $\sum_{j=1}^{n} X_{ij}=1$ (departure)

(3) $\sum_{i=1}^{n} X_{ij}=1$ (arrival)

(4) X_{ij} nonnegative integer for all i and j

(5) solution is a tour.

$X_{ij}=1$ indicates that the salesman travels from city i directly to city j, and $C_{ij}>0$ is the corresponding distance.

3.3 Brand and Bound Algorithm

At the beginning of an interation t, we have an upper bound X_o^t on the optimal value of the objective function.

We can let X_o^t be any suitable large number, such as the sum $(C_{12}+C_{23}+\cdots\cdots+C_{n1})$ corresponding to tour City 1 to City 2 to …… City 1. In addition, we have a master list containing a number of asignment models.

All of these are of the form (1) through (4), but differ from each other in that various C_{ij} values have been revised to equal ∞

The procedure at interaction t is

Step 1. Terminate the computations if the master list is empty. Otherwise, remove a problem from the master list.

Step 2. Solve the chosen assignment model. If the optimal value of the objective function is greater than or equal to X_o^t, then let $X_o^{t+1}=X_o^t$, and return to Step 1. Otherwise, proceed to Step 3.

Step 3. If the obtained optimal solution to the chosen assignment model is a tour, then record it, let X_o^{t+1} be the associated optimal value of the objective function, and return to Step 1. Otherwise, proceed to Step 4.

Step 4. Select in the obtained optimal solution of the chosen assignment model a subtour that contains the smallest number of cities. For each of the $X_{ij}=1$ in the selected subtour, add a problem to the master list, and set the corresponding $C_{ij}=\infty$;leave all the other costs the same as in the problem chosen in Step 1. Let $X_o^{t+1}=X_o^t$, and return to Step 1. [4]

3.4 Discussion

As we can easily discover by making bad choices in step 1 and step 4 of the procedure of Branch and Bound Algorithm, the computational burden depends critically on how well we resolve the arbitray selections. We will often find it worthwhile in branch-and-bound procedures to perform extra calculations in step 2 to obtain a truly good bound on the optimal value of the objective function. Your effort will be rewarded by having to explore fewer branches. Also, we can usually take advantage in Step 2 of the calculations already done on the problem that gave rise to the current one. The practical success of applying a Branch and Bound approach to solve an actual combinatorial problem depends considerably on exploiting the special structure of a model in order to implement the algorithm.

4. A GENERALIZATION OF THE TRAVELING SALESMAN PROBLEM

The traveling salesman problem is a difficult combinatorial problem of wide applicability. We take a different approach. We describe a heuristic solution strategy and provide an example which illustrates the procedure. Given a domicile denoted by S, a set I=[1, 2, ⋯, m] of market, and a set K=[1, 2, ⋯, n] of items, the traveling purchaser problem is to generate a cycle through a subset of the m markets and the domicile and purchase each n specific items at one of these markets in such a way that the total of travel and purchase costs is minimized.

It is assumed that each item is available in at least one market, that the traveler may pass through a market any number of items without purchasing an item there, that a traveler may purchase as many items as there are available at each market, and that no items are available at the domicile. Also, we are given the matrices D and C where

d(i, K)=the cost of items K at market i, and

c(i, j)=the cost of travel from i to j.

If item K is not available in market i, we define d(i, K)=M where $M \gg \max([\max d(i, K)], \max [c(i, j)]$. In this case where m=n and each market carries only one item, the traveling purchaser problem reduces to TSP. Therefore, the Traveling Purchaser Problem(TPP) is more general than TSP.

Procedure TPP.[4]

Step 1: Solve the all-pairs shortest-path problem to obtain the shortest distance matrix C=[c(i, j)].

Step 2. Find the market i^*which sells more items than any other market at the cheapest price. Resolve ties by choosing

i^* to min $\sum_{i=1}^{n}$ d(i, K)

from initial cycle S-i^*-S. Call this α

Step 3. Compute $f(\alpha, L)=\min[d(i, L)]$for all L, and $g(\alpha, p, L)$
$=\max[f(\alpha, L)]^{iE}-d(p, L), 0]$ for all L and $p \notin \alpha$.

Step 4. Find the maket $p^* \notin \alpha$ and adjacent markets i^*, $2^* \in \alpha$ such that

$$S(i, j, p)=C(i, j)-C(i, P)-C(p, j)+\sum_{i=1}^{n} g(\alpha, p, L)$$

is maximized. Suppose $S(i^*, j^*, p^*)=\max[S(i, j, p)]$

Step 5. If $S(i^*, j^*, p^*)>0$, insert p^* between i^* and j^* Update α, $f(\alpha, L)$ for all L, and $g(\alpha, p, L)$ for all L and $p \notin \alpha$. Go to step 4. If $S(i^*, j^*, p^*)>0$, terminate.

The algorithm is straightforward to perform; it is also easy to see why it should work well. In step 1 the shortest distance matrix is computed. An initial cycle is chosen in Step 2. In Step 3, $f(\alpha, L)$ is the cost of item L given the current α and $g(\alpha, p, L)$ is the decrease in the cost of item L if market p is next inserted into the cycle. Saving or total decreases in travel and purchase costs when market p is inserted between markets i and j are determined in Step 4. If there exists a positive savings, the largest one defines the next insertion. Then the cycle is updated and we search to see if another market can be added to the cycle.

5. SUMMARY

An engineering approach to the traveling salesman problem is a method which is intuitively "reasonable" to the nonmathematician. It consists of a sequence of operations which develops good starting circuits, improve these circuits, extract sufficient information from the improved circuits to determine pairs or chains of cities likely to appear in the conjectured global minimum cost circuit. The weak point of this approach is that pair-wise interchanges are often not adequate to generate the global minimum, we must turn to supplementary methods.

Branch and Bound is also general purpose strategy for TSP. Formulation of TSP contains zero-one variable can be solved by Branch and Bound Algorithm. This method is practically taking advantage of a model's special structure is that we then can handle moderately large problems. But this also has a disadvantage With bad choices in step 1 and Step 4 of the procedure of Branch and Bound Agorithm, the computational burden depends critically on how well we resolve the arbitrary selection. In other words, exhaustive pursuit of the branching tree would be equivalent to complete enumeration of all sequences.

Finally a generalization of the traveling salesman problem can handle two variables; $d(i, K)$=the cost of item K at market i, and $C(i, j)$=the cost of travel from i to j. If each market carries only one item, the traveling purchaser problem (TPP) can be exactly the same as the traveling salesman problem(TSP). TPP is a more advanced practical approach than any other method.

REFERENCES

1. Baker, Kenneth R., *Introduction to Sequencing and Scheduling*, John Wiley & Sons, 1974.
2. Flood, M. M., "The Traveling Salesman Problem," *Operations Research, 4*(1956), pp. 61-75.
3. Flores, B., and Roberts, S. M., "An Engineering Approach to the Traveling Salesman Problem," *Management Science, 13*(1966), pp. 269-288.
4. Golden, B., and Levy, L., and Dahl, R., "Two generalizations of the Traveling Salesman Problem," *Omega, 9*(1981), pp. 439-441.
5. Mitten, L. G., "Branch and Bound Methods : General Fomulation and Properties," *Operation Research, 18(4).*