Weighted Hamiltonian Augmentation Method Involving Insertions

Roy W. Dahl,* Karen Keating,† and Laurence Levy‡
DISTINCT Management Consultants, Columbia, Maryland
Daryl J. Salamone§
University of Maryland, College Park, Maryland
Barindra Nag¶
Towson State University, Towson, Maryland
and
Joan A. Sanborn**
NASA Goddard Space Flight Center, Greenbelt, Maryland

This paper presents an algorithm designed to solve the AIAA Artificial Intelligence Design Challenge. The problem under consideration is a stochastic generalization of the traveling salesman problem in which travel costs can incur a penalty with a given probability. The variability in travel costs leads to a probability constraint with respect to violating the budget allocation. Given the small size of the problem (11 cities), we consider an approach that combines partial tour enumeration with a heuristic city insertion procedure. For computational efficiency during both the enumeration and insertion procedures, precalculated binomial probabilities are used to determine an upper bound on the actual probability of violating the budget constraint for each tour. The actual probability is calculated for the final best tour, and additional insertions are attempted until the actual probability exceeds the bound.

Prior Work

THE problem presented by the Design Challenge is a generalization of the well-known traveling salesman problem. Without the stochastic constraint and the special side constraint on ordering of cities, the problem is a time-constrained traveling salesman problem (TCTSP). The TCTSP has been studied by Golden et al.[1] They use an insertion procedure that trades off the increase in travel cost against the increase in profit.

Golden et al.[2] studied a similar problem known as the orienteering problem. Their procedure determines the center of gravity of the cities currently on the tour and inserts additional cities relative to the center of gravity. Their results are limited to tests on Euclidean space problems. Additional work on the orienteering problem was done by Golden et al.[3] They relaxed the time-constraint subject to a penalty. This work uses an insertion procedure and a deletion procedure after the center of gravity tour is constructed.

An in-depth discussion of the traveling salesman problem can be found in Ref. 4. Heuristic as well as exact approaches are discussed. Exact solution procedures use integer programming techniques, including Lagrange multipliers. Heuristic approaches include worst-case bounds and computational analysis.

Received June 26, 1987; presented as Paper 87-2333 at the AIAA Guidance, Navigation, and Control Conference, Monterey, CA, Aug. 17–19, 1987; revision received Oct. 29, 1987.

*Principal Analyst.
†Senior Programmer/Analyst.
‡Senior Analyst.
§Graduate Student.
¶Assistant Professor, Department of Management, School of Business and Economics.
**Aerospace Engineer, Guidance and Control Branch. Member AIAA.

Contribution of Paper

The solution procedure presented in this paper incorporates a unique combination of operations research techniques. These include partial enumeration to create a starting tour, an insertion procedure to expand this tour, and the use of the binomial distribution to define an upper bound for the probability constraint. This approach is designed specifically for small problems. In particular, enumeration techniques would not be appropriate for a 100-city problem. We were also not concerned with the execution time of the algorithm as long as it met the 20-min time limit.

Description of the Algorithm

Our approach to the design challenge consists of the five steps listed below.

1) Calculate upper bounds for the stochastic constraint using the binomial distribution.

2) Enumerate tours of six or fewer cities that satisfy the upper bounds. Keep the best tour of each size.

3) Insert cities into each of the enumerated tours based on the "biggest bang for the buck" (best value per cost). Choose the best tour from the expanded tours based on total city value.

4) For the best expanded tour, calculate the exact probability of violating the stochastic constraint.

5) Perform final insertions into the best tour until the stochastic constraint is violated.

These steps are described in detail below.

Calculate Binomial Bounds

A critical constraint in this problem is the global probability constraint. It would become computationally prohibitive to compute the actual probability of violating the budget constraint within the enumeration routine. Each arc in the tour (a trip from one city to another) will incur the first-class premium with a fixed probability. If each of these penalties were equal, then the actual probabilities could be calculated using a binomial distribution. However, we observe that the binomial distribution can be used as a conservative estimate of the probability of violating the budget. The advantage of this observation is that a set of "critical values" can be computed for any size tour using the binomial distribution and the probability of incurring the premium, prior to doing any of the enumeration. We describe below precisely how these critical values are computed.

For every possible tour length ($n = 2$ through 11), we calculate a critical value r, which represents the minimum number of penalties that must occur to possibly violate the global probability contraint. We determine r such that

$$\sum_{k=r}^{n} P(k) \leq (1 - \text{global probability constant})$$

where $P(k)$ is the binomial probability distribution. In the enumeration routine we use the critical value as follows. The penalties associated with cities in the tour are sorted from largest to smallest. We know that in the worst case we would incur the largest penalties. So, if the sum of the r largest penalties does not exceed the budget, no other combinations of r penalties will exceed the budget. Since r represents the minimum number of penalties that must occur to possibly violate the global probability constraint, we can be sure that the current tour satisfies this constraint.

As an example, assume we have a five-city tour. The probability of incurring a first-class premium penalty is 0.2, and the global probability constant is 0.95. The binomial probabilities for incurring k premiums ($k = 0,1, \ldots ,5$) are:

k	0	1	2	3	4	5
$P(k)$	0.3277	0.4096	0.2048	0.0512	0.0064	0.0003

We sum $P(k)$ starting with $k = 5$ (in reverse order), stopping just before the sum is greater than 0.05 (1—global probability constant). In this case, $P(5) + P(4) < 0.05$, but $P(5) + P(4) + P(3) > 0.05$. Thus, the critical value r is equal to 4. Let the total cost of the tour be 630. The budget constraint is 675 and, therefore, a gap of 45 remains. Assume the ordered list of premium penalties for the five arcs in this tour are 20, 15, 6, 3, and 2. The sum of the four largest penalties is 44, and we are assured that the only way of violating the budget is to incur a penalty on more than four arcs. However, the critical value of 4 guarantees that the cumulative probability of four or more arcs being penalized is less than the global probability constant of 0.95. If, on the other hand, the third penalty were 7 rather than 6, the total cost of the current tour would be 635 and the gap would be reduced to 40. The sum of the four largest penalties would be 45. This violates the budget and, therefore, we cannot be sure that the probability of violating the budget constraint will be smaller than the acceptable probability.

Because these critical values are computed at the beginning of the algorithm, the probability checks for any n-city tour considered in the enumeration will consist of simply summing the ordered list of premium penalties until the budget is violated. The number of penalties summed, not the actual penalty, is then compared to the critical value for an n-city tour. If the critical value is not exceeded, we know the probability is satisfied, and so we can accept this tour and leave the actual probability calculation for later in the algorithm.

The only possible drawback to using the binomial distribution is that the probabilities may be too conservative. In practice, this does not appear to be the case. Also, because an insertion routine based on the actual probabilities is used in the last step of the algorithm, any missed opportunities due to being overly conservative may be overcome.

Enumerate Tours

We enumerate all possible two-city, three-city, and up to six-city tours, provided that the sum of the r largest penalties does not exceed the budget. (Recall that r is the critical value derived from the binomial distribution.) The tour with the highest value of each size is retained for the next step.

Each potential tour must be checked against the binomial upper bound. Before investing the time to check a tour for violation of the global probability constraint, we perform a "quick check." The quick check sums all the premium penalties to see if the worst case (all penalties were incurred) violates the budget limit. If the quick check satisfies the budget limit, there is no need to evaluate this tour for violation of the global probability constraint.

In another effort to save computational time in the enumeration step, we do not continue to expand a tour that violates the global probability constraint. When a tour of less than six cities violates the global probability constraint, it is clear that adding more cities would also violate this constraint. This results in a reduction of the number of tours to be enumerated and evaluated with respect to the global probability constraint, which in turn saves on running time.

To deal with the local constraint that the city value for visiting Los Angeles (LAX) is lost unless Boston (BOS) is included at some later position in the tour, we exclude any tours in which LAX occurs without BOS appearing after it. For the special case in which LAX is the home city, we add the value of LAX to the value of BOS and set the value of LAX to zero.

Insert Cities

Starting with the best six-city tour, we consider for insertion all cities not in the tour. For each city we identify the "cheapest" place in the tour to insert that city. The cheapest insertion is that position in the tour that causes the least increase in the cost of the tour for that city. Of these unused cities, we choose the one that produces the best value per cost and insert that city into the tour in the cheapest place. If the resultant tour has a larger value than the previous "best" tour, the tour is retained. We continue inserting cities into the tour using this procedure until the probability upper bound is violated. At this point we perform the same procedure on the other tours. We choose the tour with the highest value from the resulting tours.

Regarding the LAX-BOS constraint, we consider three cases in the insertion step. If neither LAX nor BOS are in the current tour, in order to insert LAX, we force BOS into the tour at a later position. In this case, the insertion cost and the value are set to the totals for both LAX and BOS. If BOS is in the tour and BOS is not the home city, in order to insert LAX it must precede BOS on the tour. If LAX and BOS are already in the tour, or if BOS is the home city and LAX is not in the tour, we perform an unrestricted insertion.

Calculate Exact Probabilities

For the highest-value tour from the previous step, the exact probability of violating the global probability constraint is determined using the following procedure. For each arc in the tour, we consider the possibility that the penalty on the arc is incurred in the cost of the tour. Using this procedure, we can track those combinations of penalties that cause the tour budget to be violated. For each case where a violation occurs, we can determine the probability for that combination of penalties. By summing the probabilities for those combinations where the budget is exceeded, we calculate the actual probability of exceeding the budget.

Perform Final City Insertion

At this point we have a tour that satisfies all of the constraints of the problem along with the actual probability that the tour will exceed the budget. We now use a final insertion procedure that operates as follows. We determine the cheapest insertion for all of the cities not in the tour and add the city with the cheapest insertion to the tour. The exact probability of exceeding the budget is then recalculated. If the new tour is feasible, we repeat the process. Otherwise, we conclude that we have already found the best tour. The LAX-BOS restriction is handled as in the original insertion procedure.

Conclusions

The algorithm was coded in Microsoft FORTRAN on an IBM-AT and was tested on the sample data provided in the Design Challenge. The enumeration procedure, which consumed most of the running time of the algorithm, identified the following tour of length 6 with a value of 70 and a cost of 1880: "DTT-CHI-LAX-PHX-DFW-BOS-DTT." The insertion routine added two additional cities and increased the value of the tour to 82 with an expected cost of 2447: "DTT-CHI-LAX-PHX-DFW-MSY-ATL-BOS-DTT." The final insertion routine added another city to the tour, so that the final tour has a value of 88 with an expected cost of 2704. The probability of exceeding the global probability constraint is 0.005. The final tour is: "DTT-CHI-LAX-PHX-DEN-DFW-MSY-ATL-BOS-DTT." This tour was generated within the time constraints and includes all of the high-value cities. We tested our algorithm on some randomly generated problems of the same size (11 cities) and obtained similar results.

For larger problem sets, we would replace the current enumeration routine with a heuristic routine to generate a starting tour. The insertion routine and the binomial bound would still be appropriate for large data sets.

References

[1]Golden, B., Levy, L., Dahl, R., "Two Generalizations of the Traveling Salesman Problem," *Omega*, Vol. 9, No. 4, 1981, pp. 439–441.

[2]Golden, B., Levy, L., Vohra, R., "Orienteering Problem," *Naval Research Logistics*, Vol. 34, pp. 307–318.

[3]Golden, B., Storchi, G., Levy, L., "Time-Relaxed Version of the Orienteering Problem," *Proceedings of the S.E. TIMS Conference*, Old Dominion Univ., Norfolk, VA, 1986, pp. 35–37.

[4]Bodin, L., Golden, B., Assad, A., Ball, M., "Routing and Scheduling of Vehicles and Crews—The State of the Art," *Computers and Operations Research*, Vol. 10, No. 2 1983, pp. 65–211.