The Traveling Salesman and the Quadratic Assignment Problems: Integration, Modeling and Genetic Algorithm

Ping Ji* William Ho

Department of Industrial and Systems Engineering
The Hong Kong Polytechnic University
Hung Hom, Kowloon, Hong Kong

Abstract *The traveling salesman problem and the quadratic assignment problem are the two of the most commonly studied optimization problems in Operations Research because of their wide applicability. Due to their NP-hard nature, the individual problems are already complex and difficult to solve. In this paper, the two hard problems are integrated together first, that is called the integrated problem of which the complexity is absolutely much higher than that of the individual ones. Not only a complete mathematical model which integrates both the traveling salesman and the quadratic assignment problems together is built, but also a genetic algorithm hybridized with several improved heuristics is developed to tackle the problem.*

1 Introduction

Mathematical modeling is a powerful tool in today's life. Without applying it, the optimal solution to a particular problem cannot be obtained. Although heuristic methods and simulation are alternative tools for solving the problem, no one can assure that the solution generated using these tools is optimal or even no one knows how good the solution is before the optimal solution has been found.

The focus of this paper is confined to integrating the traveling salesman problem (TSP) and the quadratic assignment problem (QAP) together, and solving the problems simultaneously. Actually, the individual problems are already very complex and hard to solve [1-2]. Hence, the integrated problem to be studied in this paper is extremely complicated and very challenging. To optimize the integrated problem, a genetic algorithm (GA) is developed. Nevertheless, genetic algorithms are heuristic, and in general, a GA can only produce a near-optimal solution to the problem, or even the GA generates an optimal solution, but we cannot know or claim it is the optimal

* Corresponding author. Email: mfpji@inet.polyu.edu.hk

solution. So, this paper will not only develop a GA but also build a mathematical model so that the optimal solution can be obtained together with the effectiveness of the GA can be estimated, too.

2 The Traveling Salesman Problem

The TSP is one of the most widely studied integer programming problems. The TSP can be easily stated as follows. A salesman wants to visit *m* distinct cities and then returns home. He wants to determine the sequence of the travel so that the overall traveling distance is minimized while visiting each city not more than once. Although the TSP is conceptually simple, it is difficult to obtain an optimal solution. In an *m*-city situation, any permutation of *m* cities yields a possible solution. As a consequence, *m*! possible tours must be evaluated in the search space. By introducing variables x_{ij} to represent the tour of the salesman travels from city *i* to city *j*, one of the common integer programming formulations for the TSP can be written as [3]:

$$\text{Minimize } z = \sum_{i=1}^{m} \sum_{\substack{j=1 \\ j \neq i}}^{m} d_{ij} x_{ij} \quad (1)$$

subject to

$$\sum_{i=1}^{m} x_{ij} = 1 \qquad j = 1, 2, .., m;\ i \neq j. \quad (2)$$

$$\sum_{j=1}^{m} x_{ij} = 1 \qquad i = 1, 2, .., m;\ i \neq j. \quad (3)$$

$$u_i - u_j + m x_{ij} \leq m - 1 \qquad i, j = 2, 3, .., m;\ i \neq j. \quad (4)$$

All $x_{ij} = 0$ or 1, All $u_i \geq 0$ and is a set of integers (M1)

The distance between city *i* and city *j* is denoted as d_{ij}. The objective function (1) is simply to minimize the total distance traveled in a tour. Constraint set (2) ensures that the salesman arrives once at each city. Constraint set (3) ensures that the salesman leaves each city once. Constraint set (4) is to avoid the presence of sub-tour.

Generally, the TSP formulated is known as the Euclidean TSP, in which the distance matrix *d* is expected to be symmetric, that is $d_{ij} = d_{ji}$ for all *i*, *j*, and to satisfy the triangle inequality, that is $d_{ik} \leq d_{ij} + d_{jk}$ for all distinct *i*, *j*, *k*.

3 The Quadratic Assignment Problem

Although the linear programming problems are very common and cover a wide range of problems, the objective function may not be a linear function, or some of the constraints may not be linear in a real-life situation. Such an optimization problem is called a nonlinear programming problem.

The QAP is a generalization of the linear assignment problem. The major difference between them is that the objective function of the QAP is in nonlinear expression. Therefore, it is comparatively difficult to solve. The QAP can be described as follows. Consider a set of facilities (p, r = 1, 2, .., n) is placed uniquely in a set of locations (q, s = 1, 2, .., n). The workflow intensity between each pair of facilities is apr while the distance between each pair of locations is bqs. Also, a fixed cost cpq associated with the placement of facility p in location q is specified. The formulation of the QAP can be written as [2, 4]:

$$\text{Minimize } z = \sum_{p=1}^{n}\sum_{q=1}^{n}\sum_{\substack{r=1\\ r\neq p}}^{n}\sum_{\substack{s=1\\ s\neq q}}^{n} a_{pr} b_{qs} y_{pq} y_{rs} + \sum_{p=1}^{n}\sum_{q=1}^{n} c_{pq} y_{pq} \qquad (5)$$

subject to

$$\sum_{p=1}^{n} y_{pq} = 1 \qquad q = 1, 2, .., n. \qquad (6)$$

$$\sum_{q=1}^{n} y_{pq} = 1 \qquad p = 1, 2, .., n. \qquad (7)$$

All ypq = 0 or 1 (M2)

The decision variables ypq represent the placement of facility p in location q. The objective function (5) is to assign facilities to locations so that the traveling distance of material flow is minimized, while assuming that the cost of assigning a facility does not depend upon the location, that is cpq = 0. Constraint set (6) ensures that each location must only be placed by one facility. Constraint set (7) ensures that each facility must be assigned to one location only.

4 The Integrated Problem

In the former two sections, the descriptions together with the formulations of the TSP and the QAP are carried out thoroughly. Briefly, the TSP and the QAP are the sequencing and the assignment problems, respectively. Since a wide variety of planning problems arising in the real-life situations belongs to these two types of problems, they are used prevalently as mentioned earlier.

In cases where the problems involve both sequencing and assignment, the TSP and the QAP should be integrated together. Consider a group of customers (i, j = 1, 2, .., m) with a unique order or product (p = 1, 2, .., n) is delivered by a salesman. Each of the n product types must be stored in a warehouse, say q. But, a warehouse can only store a particular type of products. Since a product type must be assigned to a warehouse, n warehouses are needed to store n types of products. The problem faced can be stated as follows. At the beginning, the salesman starts from his original location or starting point, moves to a warehouse that stores products, picks up a product from the warehouse, then moves to the desired location of the customer, and delivers it there. After that, the salesman moves back to the previous warehouse if the next product is the same type with the previous one or moves to another warehouse if it is different from the previous one to pick up the next product and repeats the operation procedure. After delivering all products to the customers, the salesman returns to its original location. In order to minimize the total distance traveled by the salesman for serving all the customers, it is necessary to determine not only the sequence of deliveries (i.e., xij) but also the assignment of product types to warehouses (i.e., $y_{p_j q}$) in which pj refers to the product type of customer j.

The objective of the problem is to minimize the total traveling distance of the salesman, which includes the distance from the starting point to a warehouse at the beginning (i.e., d0q), the distances from a warehouse to a customer's location (i.e., dqj), the distances from a customer's location to a warehouse (i.e., diq), and the distance from the last customer's location to the starting point (i.e., di0). It is noted that the starting point can be referred as customer 0 (i.e., i, j = 0). A pure integer nonlinear programming model can be formulated as follows:

Minimize z = $\sum_{i=0}^{m} \sum_{\substack{j=1 \\ j \neq i}}^{m} \sum_{p=1}^{n} \sum_{q=1}^{n} \left(d_{iq} + d_{qj}\right) x_{ij}\, y_{p_j q} + \sum_{i=1}^{m} d_{i0}\, x_{i0}$ (8)

subject to

$\sum_{i=0}^{m} x_{ij} = 1$ for j = 0, 1, .., m; i? j. (9)

$\sum_{j=0}^{m} x_{ij} = 1$ for i = 0, 1, .., m; i? j.

(10)

$u_i - u_j + m x_{ij} \le m - 1$ for i, j = 1, 2, .., m; i? j. (11)

$\sum_{p=1}^{n} y_{pq} = 1$ for q = 1, 2, .., n. (12)

$\sum_{q=1}^{n} y_{pq} = 1$ for p = 1, 2, .., n. (13)

All xij and ypq = 0 or 1, All ui ≥ 0 and is a set of integers (M3)

Since there is a nonlinear term xij $y_{p_j q}$ in the objective function and the model contains both binary variables (i.e., xij and ypq = 0 or 1) as well as integer variables (i.e., ui = 1, 2, .., m), M3 can be regarded as a pure integer nonlinear programming model.

Actually, M3 is somewhat similar to the integration of both M1 and M2 except the objective function. The objective function (8) calculates the total traveling distance of the salesman. Constraint set (9) guarantees that exactly one customer must be served immediately before customer j. Constraint set (10) guarantees that exactly one customer must be served immediately after customer i. Constraint set (11) is to eliminate sub-tours. Constraint set (12) ensures that exactly one product type is stored in one warehouse. Constraint set (13) ensures that exactly one warehouse stores one product type.

The model is very complex, and very challenging since both the TSP and the QAP are considered simultaneously. It will therefore require a lot of computing time to solve the model, and to verify the model. However, it was found that many binary decision variables are defined. So, it is worth of considering how to transfer the complete model into a linear one, or the possibility of this transformation. If this transformation is possible and successful, the model will be an integer linear program, which can be solved by a commercial package, like CPLEX or LINDO.

5 A Genetic Algorithm Approach

Apart from building a mathematical model for finding the optimal solution to the integrated problem, a heuristic method is developed to solve the problem efficiently, but a heuristic may not achieve the optimal solution. In this paper, the GA technique will be selected as a tool to solve the problem. The reason is that GAs have been applied successfully to a wide variety of optimization problems such as the TSP, the QAP, and the minimum spanning tree problem [5-6]. In addition, the merits of GA including simplicity, ease of operation, and flexibility are the encouraging factors for applying it.

In recent years, many researchers discovered that a simple GA was not desirable for solving the TSP and the QAP with a large problem size [7-10]. Besides, the problem we are facing can be regarded as the combination of two hard optimization problems. So, a simple GA must not perform well in this situation. The GA developed is therefore hybridized with several heuristics in order to improve the solution further.

The flowchart of the HGA for the integrated problem is illustrated in Fig. 1. The HGA is very similar to the general GA except that the initial solutions or chromosomes (i.e., parents) together with the chromosomes generated from the genetic operators (i.e., offspring) are improved by several heuristics. Since two problems are considered simultaneously, a chromosome is represented by the two-link representation, in which the first link is denoted as the sequence of deliveries, whereas the second link refers to the assignment of product types to warehouses. In the following, the improved heuristics adopted in the HGA are described briefly.

5.1 The Nearest Neighbor Heuristic

First, the nearest neighbor heuristic (NNH) is applied to generating the first link, which is the sequence of deliveries, in the initial chromosomes. On the other hand, the second link, which represents the product assignment, in the initial chromosomes is generated randomly. The principle of the NNH is to start with the first customer randomly, then to select the next customer as close as possible to the previous one from those unselected customers to form the delivering sequence until all customers are selected.

5.2 The 2-opt Local Search Heuristic

Comparing with the total number of customers, the number of different product types is much fewer. Therefore, it is desirable to perform the 2-opt local search heuristic for the second link only of each initial chromosome as well as each offspring generated by the genetic operators. The principle of this heuristic is very straightforward. For one parent, all possible 2 swaps are examined to generate offspring, and the best offspring will replace the parent if the offspring has a shorter traveling distance than the parent. The process will be repeated unless there is no further improvement in the offspring in terms of the total traveling distance.

5.3 The Iterated Swap Procedure

The computational effort will be high if the 2-opt local search is performed for the first link, which is the sequence of deliveries, because the number of customers is quite large. As a consequence, a 'fast' improved heuristic is developed, which is called the iterated swap procedure (ISP). The ISP is performed for the first link of each initial chromosome generated by the NNH as well as each offspring generated by the genetic operators. The principle of the ISP is very similar to that of the 2-opt local search heuristic, except that some instead of all 2 swaps are examined. As a result, the computational effort will be lowered.

6 Conclusions

The traveling salesman problem and the quadratic assignment problem are two of the most widely adopted optimization problems. Each of them belongs to the NP class, which means that they cannot be solved efficiently in polynomial time. Instead, they are solved in exponential time. In this paper, the definitions and the formulations of the individual problems were provided in advance. It was then followed by the complete mathematical model which integrates both individual problems together. Since the formulation is in the nonlinear form, it is quite difficult to obtain the global optimum. So, it is desirable to convert it into simpler form. Certainly, the integrated problem is extremely complex since the individual ones are already hard to solve. To solve the problem efficiently and effectively, a hybrid genetic algorithm approach was developed. The algorithm is so called because three heuristics are incorporated to improve the solution further.

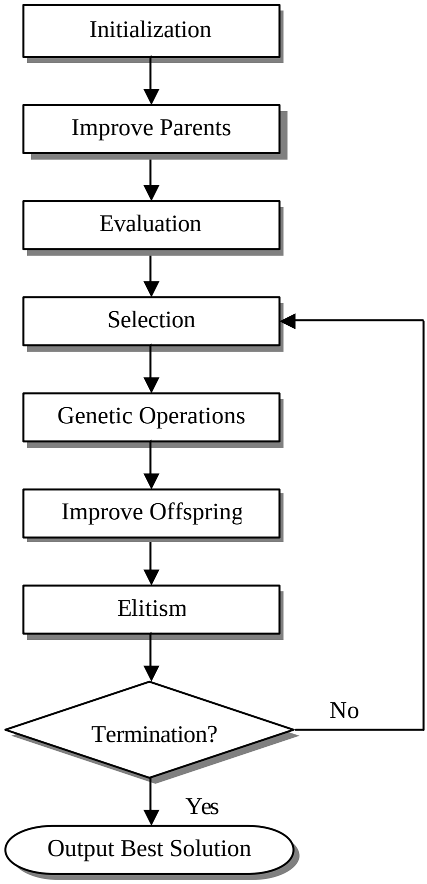

Figure 1: The flowchart of the HGA

Acknowledgements

The work described in this paper has been supported by the Research Grants Council of Hong Kong, China (Project no.: PolyU 5259/04E).

References

[1] Reinelt, G. *The Traveling Salesman: Computational Solutions for TSP Applications*. Springer, New York (1994)

[2] Burkard, R. E., Karisch, S. E. and Rendl, F. "QAPLIB – A quadratic assignment problem library." *European Journal of Operational Research*, Vol. 55, pp. 115-119 (1991)

[3] Winston, W. L. and Venkataramanan, M. *Introduction to Mathematical Programming: Operations Research*. Brooks/Cole-Thomson Learning, California (2003)

[4] Williams, H. P. *Model Building in Mathematical Programming*. John Wiley & Sons, New York (1999)

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

[6] Gen, M. and Cheng, R., *Genetic Algorithms and Engineering Design*, Wiley, New York (1997)

[7] Freisleben, B. and Merz, P. " A genetic local search algorithm for solving symmetric and asymmetric traveling salesman problems". In *Proceedings of the 1996 IEEE International Conference on Evolutionary Computation*, Nagoya, Japan, 1996, pp. 616-621 (1996)

[8] Freisleben, B. and Merz, P. "New genetic local search operators for the traveling salesman problem". In *Proceedings of the 4th Conference on Parallel Problems Solving from Nature*, Springer, pp. 890-900 (1996)

[9] Huntley, C. L. and Brown, D. E. "Parallel genetic algorithms with local search". *Computers & Operations Research*, Vol. 23, No. 6, pp. 559-571 (1996)

[10] Ahuja, R. K., Orlin, J. B. and Tiwari, A. " A greedy genetic algorithm for the quadratic assignment problem". *Computers & Operations Research*, Vol. 27, pp. 917-934 (2000)