Computers & Industrial Engineering 70 (2014) 124–133

Contents lists available at ScienceDirect

Computers & Industrial Engineering

journal homepage: www.elsevier.com/locate/caie

The hybrid genetic algorithm with two local optimization strategies for traveling salesman problem ☆

Yong Wang *

School of Renewable Energy, North China Electric Power University, Beijing 102206, China

ARTICLE INFO

Article history:
Received 11 April 2013
Received in revised form 23 January 2014
Accepted 27 January 2014
Available online 1 February 2014

Keywords:
Traveling salesman problem
Genetic algorithm
Local optimization strategies

ABSTRACT

Traveling salesman problem (TSP) is proven to be NP-complete in most cases. The genetic algorithm (GA) is improved with two local optimization strategies for it. The first local optimization strategy is the four vertices and three lines inequality, which is applied to the local Hamiltonian paths to generate the shorter Hamiltonian circuits (HC). After the HCs are adjusted with the inequality, the second local optimization strategy is executed to reverse the local Hamiltonian paths with more than 2 vertices, which also generates the shorter HCs. It is necessary that the two optimization strategies coordinate with each other in the optimization process. The two optimization strategies are operated in two structural programs. The time complexity of the first and second local optimization strategies are $O(n)$ and $O(n^3)$, respectively. The two optimization strategies are merged into the traditional GA. The computation results show that the hybrid genetic algorithm (HGA) can find the better approximate solutions than the GA does within an acceptable computation time.

1. Introduction

The objective of traveling salesman problem (TSP) is to find the shortest tour visiting each city once and exactly once in a tourist map. The cities are connected with routes in the map. The tour including each city once is named as Hamiltonian circuit (HC) and the shortest Hamiltonian circuit is the optimal Hamiltonian circuit (OHC) for Euclidean TSP. As we know, TSP has been proven to be NP-complete (Zhang & Korf, 1996). TSP has been widely studied in the fields of combinatorial mathematics, graph theory and computer science due to its theoretical and practical values (Rodríguez & Ruiz, 2012; Wang & Liu, 2010). However, there is no polynomial algorithms for the NP-complete problems unless $NP = P$ (Berman & Karpinski, 2006). The research on the efficient algorithms for TSP is still one frontier subject.

Because of its hardness, TSP becomes one of the best platforms to test the performance of all kinds of algorithms. Exact algorithms, approximate algorithms and intelligent algorithms are extensively designed for TSP. Many literatures illustrated that the exact algorithms, such as the depth-first search graph algorithm (Douglas, 2006), the integer programming methods (Climer & Zhang, 2006) and dynamic programming methods, are feasible to tackle the TSP with less than 1000 cities (Johnson & McGeoch, 2004). They can find the OHC within an acceptable computation time using powerful Turing machines. When the scale of TSP becomes larger, the approximate algorithms demonstrate their good performance. The local search rules used by the approximate algorithms are efficient to allow them find the OHC or near OHC in a polynomial computation time. The experiments show that the *k*-opt algorithms (Verhoeven, Aarts, & Swinkels, 1995), the LKH algorithm has dealt with the large scale of TSP with thousands of cities, even more than 3,000,000 cities (Helsgaun, xxxx). On the other hand, the local search rules, such as the neighborhood information (Liu, 2008), will make the algorithms trap into the local minima. Therefore, the quality of the solutions cannot be evaluated due to the lack of the OHCs.

The intelligent algorithm is another resolution for TSP. They find the best or approximate solutions based on the evolutionary rules which are different from the local search rules. Almost all of the intelligent algorithms, including the anterior artificial neural network (Ghaziri & Osman, 2003) and the recent particle swarm optimization (Chen et al., 2010), have been applied to TSP. The genetic algorithm (GA) is one of the competitive intelligent algorithms for discrete optimization problems. It evolves to the optimal solution with the crossover operation and mutation operation (Schmitt & Amini, 1998). The crossover operation guarantees the better genes evolve to the offspring to obtain better solutions. The mutation operation maintains the diversity of the offspring to reproduce

☆ This manuscript was processed by Area Editor Mitsuo Gen.

* Tel.: +86 010 6177 2051.
E-mail address: wangyyong100@163.com

http://dx.doi.org/10.1016/j.cie.2014.01.015
0360-8352/

better solutions. Although it has many merits, the genetic algorithm usually detects an approximate OHC for TSP. It is always improved to acquire the better performance. The good initial solutions facilitate the genetic algorithm to evolve to the best solution. Liu (Liu, 2010) designed three kinds of initial solution generators under the GA framework for TSP. It is found that the various computation times are consumed and different results are obtained with different initial solutions. To enhance the performance of traditional GA, the local search rules are widely taken into consideration. The 1-shift and 2-opt local search rules are merged into the GA by Liu (2010). Also, the feasible search space is computed by Choi, Kim, & Kim (2003) through determining the infeasible solutions with sub-tours. The memetic algorithm is introduced by Bontoux, Artigues, & Feillet (2010). The cities are partitioned into a set of clusters first, and the minimum tour visiting each cluster is found with the crossover procedure based on a dynamic programming algorithm. The cluster method is also used by Ding, Cheng, & He (2007) in their two-level genetic algorithm, in which the first level is to generate the shortest sub-tours and the second level is to generate the shortest tour with the sub-tours. The immigration, local and global optimization strategies are defined and inserted into the GA (Xing et al., 2008). In addition, the other intelligent algorithms are combined with GA to reinforce its performance. The genetic algorithm is improved by Liu & Zeng (2009) with the reinforcement mutation which relies on the reinforcement learning. The genetic algorithm, simulated annealing, ant colony optimization and particle swarm optimization are integrated together to utilize their total advantages (Chen & Chien, 2011).

Different from the above researches, the genetic algorithm with two local optimization strategies is designed for TSP. The first local optimization strategy is the four vertices and three lines inequality. The four point conditions for symmetrical TSP has been summarized by Deineko, Klinz, & Woeginger (2006) under their assumptions. The approximate algorithms with the four point conditions are seldom regarded. The four vertices and three lines inequality is the extension of one of the fundamental four point conditions. It is the constraint of the local optimal Hamiltonian path connected by four vertices in the OHC or an approximate OHC. When the local Hamiltonian paths in the HCs are transformed into the local optimal Hamiltonian paths with the four vertices and three lines inequality, the HCs will become shorter. The computation complexity of the algorithm based on the local optimization strategy is $O(n)$. After the HCs are adjusted with the four vertices and three lines inequality, the adjacent vertices in the HCs cannot be exchanged to produce the shorter HCs. On the other hand, the nonadjacent vertices in the HCs are allowed to exchange to generate the next shorter HCs. The second optimization strategy is the reversion of a selected local Hamiltonian paths including over 2 vertices. The algorithm is designed according to the optimization strategy and the maximum computation complexity is $O(n^3)$.

The two strategies are merged into the genetic algorithm to construct a hybrid genetic algorithm (HGA). The first local optimization strategy is executed after the HCs are generated to produce the shorter HCs. The second optimization strategy is merged into the mutation operation to generate the shorter HCs. The HGA are designed and tested with the TSP instances downloaded from the TSPLIB. The results show that the OHCs of most of the small scale of TSP instances are found within an acceptable computation time. For the large scale of TSP, the errors of the detected approximate OHCs to the given OHCs are very small.

2. Basic knowledge on symmetrical TSP

In graph theory, it is to find the OHC in a weighted graph (WG) and only the simple graph G is considered for TSP. The cities and routes are mapped into the vertices and edges in the WG. For a graph G including n vertices, it is generally represented as $G = \langle V,E\rangle$, where $V = \langle v_1, v_2, \ldots, v_n\rangle$ are the vertices sets and $E = [e_{ij}]_{n\times n}$ are the edges sets. v_i $(1 \leqslant i \leqslant n)$ is the vertex and e_{ij} $(1 \leqslant i, j \leqslant n)$ is the edge connecting the two vertices v_i and v_j. The relationships of the vertices in graph G are represented as an adjacent matrix $A(G) = [a_{ij}]_{n\times n}$ $(1 \leqslant i, j \leqslant n)$, where $a_{ij} = 1$ if $(v_i, v_j) \in E(G)$ and v_i and v_j are adjacent in the graph G. Otherwise, $a_{ij} = 0$. If the edges are assigned with weights $W = [w_{ij}]_{n\times n}$, the graph G becomes one WG. The weight w_{ij} is often taken as distance, cost, etc. for various kinds of TSP. If $w_{ij} = w_{ji}$, the WG is symmetrical. Otherwise, it is asymmetrical. It is considered that the symmetrical TSP is more difficult than the asymmetrical TSP (Helsgaun, xxxx). The report said the asymmetrical TSP with 500,000 cities had been resolved whereas the symmetrical TSP with only 7397 cities was resolved at that time.

Given a WG including n vertices, there are total $(n-1)!/2$ HCs for the symmetrical TSP. The OHC is taken as the HC whose length is the shortest among all of the HCs for the Euclidean TSP. Given the HC including n vertices, it is represented as $HC^{n+1} = (v_1, v_2, v_3, \ldots, v_n, v_1)$. The end vertex v_1 emerges repeatedly once to form the HC. Given $l_{i\times j}$ is the distance between the two adjacent vertices v_i and v_j in the HC, the computation model of the symmetrical TSP is given as formula (1).

$$\left.\begin{array}{l} L_{\min} = \min(L(\mathrm{HC})) = \min \sum_{i,j=1}^{n} l_{i\times j} \\ \text{Subject to} \quad v_i \neq v_j \quad \text{and} \quad e_{i\times j} \in E(\mathrm{HC}) \end{array}\right\} \tag{1}$$

where $L(\mathrm{HC})$ is the length of the HC, $e_{i\times j}(1 \leqslant i, j \leqslant n)$ is the edge linking the two adjacent vertices v_i and v_j in the HC. For the simple WG, it is equal to the standard form of integer programming for TSP (Climer & Zhang, 2006). With the model, all the HCs will be traversed and evaluated to find the OHC.

The HC consists of the local Hamiltonian paths (LHP) and the OHC must be composed of the local optimal Hamiltonian paths (LOHP). The LHP or LOHP including i $(1 \leqslant i \leqslant n)$ vertices is represented as LHP^i or $\mathrm{LOHP}^i = (v_1, v_2, v_3, \ldots, v_i)$. The vertices v_1 and v_i are the end vertices, and the other vertices between them are the middle vertices. There are no identical vertices in the LOHPs or LHPs.

For general WG, the orders of the vertices are determined in the OHC. For an arbitrary LOHP in the OHC, the orders of these vertices are also concluded as well as its two end vertices. Its length is the minimum among those of the LHPs including the same vertices in condition that their two end vertices are identical. If i is big, it is also NP-complete to detect a LOHP^i. If i is small, it is relatively convenient to compute a LOHP^i. But not all of the LOHPs belong to the OHC.

3. The genetic algorithm with the two local optimization strategies

3.1. The traditional genetic algorithm

Genetic algorithm originates from the evolutionary rules of the nature population and it is one of the unconstrained optimization methods (Holland, 1975). The solution of optimization problem is encoded as the chromosome and the concrete parameters of solution are taken as the genes of the chromosome. For most of the optimization problems, the solutions are often encoded as the binary strings. However, it is believed that the binary string is not suitable to represent an HC (Michalewicz, 1994). The binary code of the cities will make the GA become complex for TSP. The literatures (Gen & Cheng, 1997; Michalewicz, 1994) summarized the current representations of the HCs, such as the adjacency