# Solving traveling salesman problems via artificial intelligent search techniques

**Conference Paper** · February 2012



2 authors:

Supaporn Suwannarongsri
King Mongkut's University of Technology North Bangkok


Deacha Puangdownreong
Southeast Asia University

# Solving Traveling Salesman Problems via Artificial Intelligent Search Techniques

Supaporn SUWANNARONGSRI[1,*] and Deacha PUANGDOWNREONG[2]
[1] Department of Industrial Engineering,
Faculty of Engineering, South-East Asia University
19/1 Petchakasem Rd., Nongkhaem, Bangkok, THAILAND
[2] Department of Electrical Engineering,
Faculty of Engineering, South-East Asia University
19/1 Petchakasem Rd., Nongkhaem, Bangkok, THAILAND
[*] corresponding author: supaporns@sau.ac.th http://www.sau.ac.th

*Abstract:* - The traveling salesman problem (TSP) is one of the most intensively studied problems in computational mathematics and combinatorial optimization. It is also considered as the class of the NP-complete combinatorial optimization problems. By literatures, many algorithms and approaches have been launched to solve such the TSP. However, no current algorithms that can provide the exactly optimal solution of the TSP problem are available. This paper proposes the application of AI search techniques to solve the TSP problems. Three AI search methods, i.e. genetic algorithms (GA), tabu search (TS), and adaptive tabu search (ATS), are conducted. They are tested against ten benchmark real-world TSP problems. As results compared with the exactly optimal solutions, the AI search techniques can provide very satisfactory solutions for all TSP problems.

*Key-Words:* - Traveling Salesman Problem, Genetic Algorithm, Tabu Search, Adaptive Tabu Search

## 1 Introduction

The traveling salesman problem (TSP) has been firstly proposed as one of the mathematical problems for optimization in 1930s [1]. The problem is to find an optimal tour for a traveling salesman wishing to visit each of a list of $n$ cities exactly once and then return to the home city. Such optimal tour is defined to be a tour whose total distance (cost) is minimized. This problem can be considered as the class of combinatorial optimization problems known as NP-complete [1], [2]. By literature, many algorithms and approaches have been launched to solve the TSP problems. Those algorithms and approaches can be classified into exact and heuristic approaches [3], [4], [5].

The TSP problems possessing no longer than 20 cities can be optimally solved by exact methods. Some are dynamic programming [6], branch and bound [7], and linear programming [2]. The heuristic methods could provide very satisfactory solutions of the TSP problems possessing large amount number of cities. However, the optimum solution can not be guaranteed. To date, the artificial intelligent (AI) search techniques have been applied to solve the TPS problems, for example, simulated annealing (SA) [8], artificial neural network [9], tabu search [10], and genetic algorithms (GA) [11].

In this paper, the artificial intelligent (AI) search techniques are applied to solve the TSP problems. Three AI search methods, i.e. genetic algorithms (GA), tabu search (TS), and adaptive tabu search (ATS), are conducted against ten benchmark real-world TSP problems collected in TSPLIB95 [12]. Results obtained by those AI search techniques will be compared with exactly optimal solutions. This paper consists of five sections. The problem formulation of TSP problem optimization, AI search technique algorithms, AI-based TSP solving, and conclusions, are provided in Section 2, 3, 4, and 5, respectively.

## 2 TSP Problem Formulation

By theory, the traveling salesman problem (TSP) has been firstly proposed as one of the mathematical problems in 1800s by Harmilton and Kirkman. However, the general formulation of TSP has been firstly lunched based on the graph theory in 1930s [1].

Let $G = (V, E)$ be a complete undirected graph with vertices $V, |V| = n$, where $n$ is the number of cities, and edges $E$ with edge length $c_{ij}$ for the-ij city $(i, j)$. Our work focus on the symmetric TSP case in which $c_{ij} = c_{ji}$, for all cities $(i, j)$. TSP problem formulations for minimization as expressed in (1) – (5) [2]. Equation (1) is the objective function, which minimizes the total distance to be traveled. Constraints (2) and (3) define a regular assignment problem, where (2) ensures that each city is entered from only one other city, while (3) ensures that each city is only departed to on other city. Constraint (4) eliminates subtours. Constraint (5) is a binary constraint, where $x_{ij} = 1$ if edge $(i, j)$ in the solution and $x_{ij} = 0$, otherwise.

$$\text{Min} \qquad \sum_{i \in V} \sum_{j \in V} c_{ij} x_{ij} \tag{1}$$

$$\text{Subject to} \qquad \sum_{\substack{j \in V \\ j \neq i}} x_{ij} = 1, \quad i \in V \tag{2}$$

$$\sum_{\substack{i \in V \\ i \neq j}} x_{ij} = 1, \quad j \in V \tag{3}$$

$$\sum_{i \in S} \sum_{j \in S} x_{ij} \leq |S| - 1, \quad \forall S \subset V, S = \varnothing \tag{4}$$

$$x_{ij} = 0 \; or \; 1, \quad i, j \in V \tag{5}$$

However, the difficulty of solving TSP is that subtour constraints will grow exponentially as the number of city grows large, so it is not possible to generate or store these constraints. Many applications in real world do not demand optimal solutions. Therefore, many researchers proposed several heuristic algorithms, which are fast and easy to implement.

# 3 AI Search Algorithms

The artificial intelligent (AI) search techniques used to solve the TSP problems in this work are genetic algorithms (GA), tabu search (TS), and adaptive tabu search (ATS). Their algorithms are briefly reviewed as follows.

## 3.1 Genetic Algorithm

The genetic algorithm or GA is one of AI search optimization techniques. GA has natural selection mechanism and genetic operation, i.e. crossover and mutation techniques to find optimum solution. The GA algorithms can be briefly summarized as follows [13], [14].

**Step 1.** Randomly generate the populations.
**Step 2.** Evaluate all population chromosomes via the objective function.
**Step 3.** Select some chromosomes and set them to be parents.
**Step 4.** Generate next generation of population by crossover and mutation.
**Step 5.** Evaluate the (fitness) objective function of new populations.
**Step 6.** Replace old population by new ones that more fit.
**Step 7.** Once termination criteria are met, terminate search process; otherwise go back to Step 2.

GA will stop the search process when the termination criteria are satisfied. Generally, we use the preset maximum generation set as the stopping criteria. The optimum solution is the best chromosome found in current population.

## 3.2 Tabu Search

The tabu search or TS is proposed by Glover [15], [16]. The TS is also one of AI search optimization techniques. Based on the neighborhood search, the TS has the tabu list (TS) used to store the visited solutions and to conduct as an aspiration criteria when the local entrapment occurs. The TS algorithms can be briefly described as follows [17], [18].

**Step 1.** Initialize a search space ($\Omega$), TL = $\varnothing$, search radius ($R$), *count*, and $count_{\max}$.
**Step 2.** Randomly select an initial solution $S_0$ from a certain search space $\Omega$. Let $S_0$ be a current local minimum.
**Step 3.** Randomly generate $N$ solutions around $S_0$ within a search radius $R$. Store the $N$ solutions, called neighborhood, in a set X.
**Step 4.** Evaluate the objective value of each member in X via objective functions. Set $S_1$ as a member giving the minimum cost.
**Step 5.** If $f(S_1) < f(S_0)$, put $S_0$ into the TL and set $S_0$=$S_1$, otherwise, store $S_1$ in the TL instead.
**Step 6.** If the termination criteria: *count*= $count_{\max}$ or desired specification are met, then stop the search process. $S_0$ is the best solution, otherwise Update *count*= *count*+1, and go back to Step 2.

### 3.3 Adaptive Tabu Search

The adaptive tabu search or ATS is the modified vertion of the tabu search. The ATS was launched in 2004 [19]. The ATS possesses two distinctive mechanisms denoted as back-tracking (BT) regarded as one of the diversification strategies and adaptive radius (AR) considered as one of the intensification strategies. The ATS can be regarded as one of the most powerful AI search techniques. Convergence proof and performance evaluation of the ATS have been reported [19], [20]. The ATS algorithm is summarized step-by-step as follows.

**Step 1.** Initialize a search space ($\Omega$), TL = $\varnothing$, search radius ($R$), *count*, and $count_{\max}$.
**Step 2.** Randomly select an initial solution $S_0$ from a certain search space $\Omega$. Let $S_0$ be a current local minimum.
**Step 3.** Randomly generate $N$ solutions around $S_0$ within a search radius $R$. Store the $N$ solutions, called neighborhood, in a set X.
**Step 4.** Evaluate the objective value of each member in X via objective functions. Set $S_1$ as a member giving the minimum cost.
**Step 5.** If $f(S_1) < f(S_0)$, put $S_0$ into the TL and set $S_0$=$S_1$, otherwise, store $S_1$ in the TL instead.
**Step 6.** Activate the BT mechanism, when a local entrapment occurs.
**Step 7.** If the termination criteria: *count*= $count_{\max}$ or desired specification are met, then stop the search process. $S_0$ is the best solution, otherwise go to Step 8.
**Step 8.** Invoke the AR mechanism, once the search approaches the local or the global solution to refine searching accuracy.
**Step 9.** Update *count*= *count*+1, and go back to Step 2.

## 4 AI-Based TSP Solving

In this work, algorithms of GA, TS, and ATS are coded by MATLAB running on Pentium (R), 2.00 GHz CPU, 1 GB RAM, to solve ten benchmark real-world TSP problems collected in TSPLIB95 [12]. Details of selected benchmark TSP problems are summarized in Table 1.

Table 1 Selected real-world TSP problems.

| TSP Problems | No. of Cities | Opt. distance (km.) |
|---|---|---|
| Eil51 | 51 | 426 |
| Berlin52 | 52 | 7,542 |
| St70 | 70 | 675 |
| Pr76 | 76 | 108,159 |
| Eil76 | 76 | 538 |
| Rat99 | 99 | 1,211 |
| Rd100 | 100 | 7,910 |
| KroA100 | 100 | 21,282 |
| KroB100 | 100 | 22,141 |
| Ch150 | 150 | 6,528 |

Each problem will be tested over 20 times to calculate the average optimum distance and the average search time. The search parameters of AI algorithms will be a priory set as follows.

For GA,
- number of population = 100
- crossover = 70%
- mutation = 4.5%
- replacement with 1 point
- TC : maximum generation = 2,000

For TS,
- $\Omega$ = [1, 2, …, No. of cities]
- number of neighborhood members $N$ = 40
- search radius $R$ = 20% of $\Omega$
- TC : $count_{\max}$ = 1,000

For ATS,
- $\Omega$ = [1, 2, …, No. of cities]
- number of neighborhood members $N$ = 40
- search radius $R$ = 20% of $\Omega$
- Activate BT, when local entrapment occurs.
- invoke AR once 20, 40, 60, and 80 times of solution cannot be improved
- TC : $count_{\max}$ = 1,000

Results obtained are summarized in Table 2. Referring to Table 2, we found that GA, TS, and ATS can fine the optimum distant of all problems. From the average optimum distance in Table 2, it was fount that the ATS outperforms other algorithms. The second is the TS, and the third is GA, respectively. This may because the search process of the TS and the GA hit many local entrapments, while the ATS with BT mechanism can efficiently escape such the local entrapments. Figs. 1 – 4 depict the results of the Eil51 problem as an example. Fig. 1 shows 51 city locations of the Eil51 problem, while results obtained by the GA, TS, and ATS are shown in Fig. 2, 3, and 4, respectively. Optimum distance of the Eil51 problem obtained by the GA, TS, and ATS, are 437.85 km., 442.11 km., and 431.17 km., respectively.

Table 2 Results obtained by AI search techniques.

| TSP problems | Opt. distance (km.) | Obtained solutions (average distance (Km.)) by AI | | | | | | average search time (sec.) by AI | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | GA | %Err | TS | %Err | ATS | %Err | GA | TS | ATS |
| Eil51 | 426 | 441.46 | 3.63 | 445.05 | 4.47 | 438.12 | 2.85 | 2.72 | 2.53 | 2.91 |
| Berlin52 | 7,542 | 7,833.26 | 3.86 | 8,152.79 | 8.10 | 7,702.16 | 2.12 | 3.58 | 2.15 | 4.14 |
| St70 | 675 | 695.44 | 3.03 | 738.78 | 9.45 | 684.62 | 1.43 | 3.84 | 3.19 | 4.22 |
| Pr76 | 108,159 | 116,273.12 | 7.50 | 123,240.57 | 13.94 | 110,478.35 | 2.14 | 4.13 | 4.05 | 4.57 |
| Eil76 | 538 | 548.26 | 1.91 | 582.62 | 8.29 | 540.17 | 0.40 | 4.05 | 3.86 | 4.43 |
| Rat99 | 1,211 | 1,274.84 | 5.27 | 1,295.42 | 6.97 | 1,213.50 | 0.21 | 6.28 | 5.83 | 7.35 |
| Rd100 | 7,910 | 8,506.51 | 7.54 | 8,788.39 | 11.10 | 8,412.62 | 6.35 | 10.68 | 9.12 | 13.46 |
| KroA100 | 21,282 | 22,875.42 | 7.49 | 23,644.18 | 11.10 | 21,525.56 | 1.14 | 11.97 | 10.89 | 15.13 |
| KroB100 | 22,141 | 24,765.94 | 11.86 | 25,349.36 | 14.49 | 22,568.14 | 1.93 | 12.52 | 11.18 | 15.46 |
| Ch150 | 6,528 | 6,986.35 | 7.02 | 7,012.21 | 7.42 | 6,612.74 | 1.30 | 14.84 | 12.57 | 16.73 |

**Note**： %Err stands for percentage of solution errors compared with optimum distances.

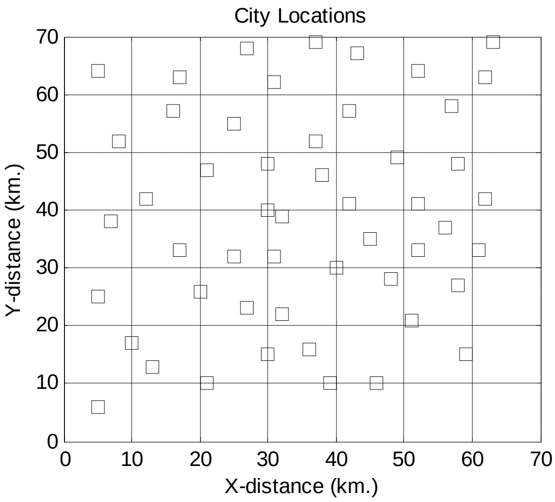


Fig. 1 City locations of Eil51 problem.

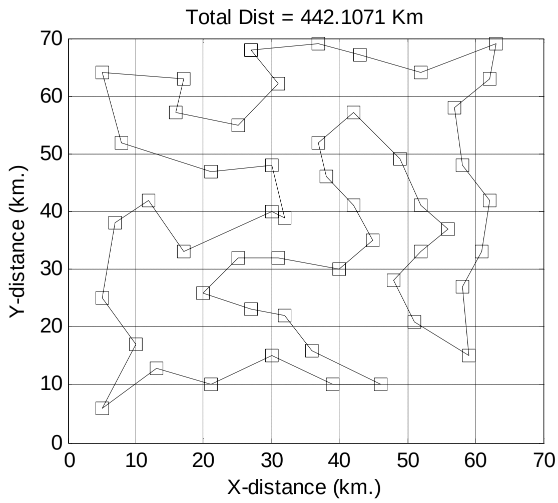


Fig. 3 Result of Eil51 problem by TS.

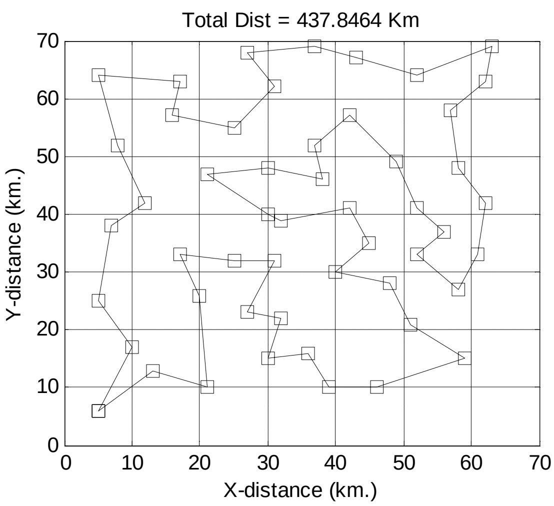


Fig. 2 Result of Eil51 problem by GA.

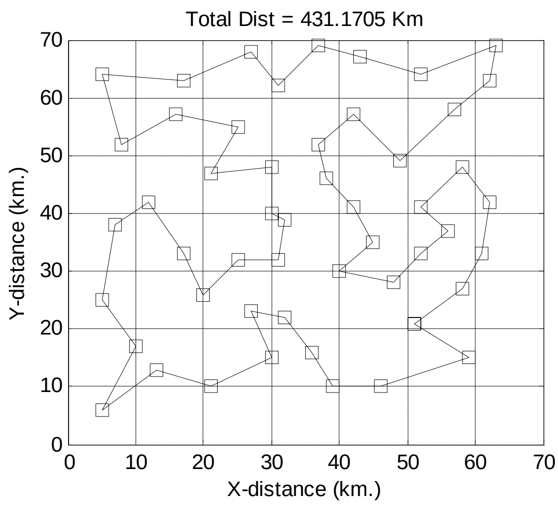


Fig. 4 Result of Eil51 problem by ATS.

## 5 Conclusion

Solving the traveling salesman problem (TSP) by AI search techniques has been proposed in this article. The genetic algorithms (GA), the tabu search (TS), and the adaptive tabu search (ATS), are most popular and powerful optimization methods. They have been conducted to be tested against ten benchmark real-world TSP problems collected in TSPLIB95. Compared with the exactly optimal solutions, the AI search techniques can provide very satisfactory solutions for all TSP problems. However, the ATS outperforms other algorithms with most optimum solution found with reasonable time consumed.

*References:*


[1] M. Bellmore and G.L. Nemhauser, The Traveling Salesman Problem: a Survey, *Operation Research*, Vol. 16, 1986, pp.538-558.

[2] G.B. Dantzig, D.R. Fulkerson, and S.M. Johnson, Solution of a Large-scale Traveling Salesman Problem, *Operation Research*, Vol. 2, 1954, pp.393.

[3] G. Reinelt, *The Traveling Salesman : Computational Solutions for TSP Applications*, Springer-Verlag, 1994.

[4] E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan, and D.B. Shmoys, *The Traveling Salesman Problem : A Guided Tour of Combinatorial Optimization*, John-Wiley & Sons, 1986.

[5] D.S. Johnson and L.A. McGeoch, The Traveling Salesman Problem: a Case Study in Local Optimization, *Local Search in Combinatorial Optimization*, Wiley, 1997, pp.215-310.

[6] M. Held and R. Karp, A Dynamic Programming Approach to Sequencing Problems, *SIAM J*, 1962, pp.196.

[7] J. Little, K. Murty, D. Sweeney, and C. Karel, An Algorithm for the Traveling Salesman Problem, *Operation Research*, Vol. 12, 1963, pp.972.

[8] E.H.L. Aarts, J.H.M. Korst, and P.J.M. Laarhoven, A Quantitative Analysis of the Simulated Annealing Algorithm: a Case Study for the Traveling Salesman Problem, *J of Stats Phys*, Vol. 50, 1988, pp.189-206.

[9] J.V. Potvin, The Traveling Salesman Problem: a Neural Network Perspective, *INFORMS J. of Computing*, Vol. 5, 1993, pp. 328-348.

[10] C.N. Fiechter, A Parallel Tabu Search Algorithm for Large Scale Traveling Salesman Problems, *Discrete Applied Mathematics*, Vol. 51(3), 1994, pp. 243-267.

[11] J.V. Potvin, Genetic Algorithms for the Traveling Salesman Problem, *Annuals of Operation Research*, Vol. 63, 1996, pp.339-370.

[12] TSPLIB95, *Symmetric traveling salesman problem*, http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/. 5 February, 2010.

[13] H. Holland, *Adaptation in Natural and Artificial Systems*, Ann Arbor, The University of Michigan Press, 1975.

[14] D.E. Goldberg, *Genetic Algorithm in Search, Optimization, and Machine Learning*, Addison-Wesley Publishing, 1989.

[15] F. Glover, Future Paths for Integer Programming and Links to Artificial Intelligence, *Computers and Operations Research*, Vol. 13, 1986, pp.533-549.

[16] F. Glover, Tabu Search – part I, *ORSA Journal on Computing*, Vol.1, 1989, pp.190-206.

[17] F. Glover, Tabu Search – part II, *ORSA Journal on Computing*, Vol.2, 1990, pp.4-32.

[18] F. Glover, Tabu Search for Nonlinear and Parametric Optimization (with Links to Genetic Algorithms), *Discrete Applied Mathematics*, Vol. 49, 1994, pp.231-255.

[19] D. Puangdownreong, T. Kulworawanichpong, and S. Sujitjorn, Finite convergence and performance evaluation of adaptive tabu search, *LNCS* (*Lecture Notes in Computer Science*), Springer-Verlag Heidelberg, Vol. 3215, 2004, pp.710-717.

[20] S. Sujitjorn, T. Kulworawanichpong, D. Puangdownreong, and K-N. Areerak, Adaptive tabu search and applications in engineering design, *Book Chapter in Integrated Intelligent Systems for Engineering Design* (*ed. X.F. Zha and R.J. Howlett*), IOS Press, Netherlands, 2006, pp.233-257.


ISBN: 978-1-61804-068-8