Indian Journal of Science and Technology Vol 8(11), DOI: 10.17485/ijst/2015/v8i11/71789, June 2015
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

# Finding Optimal Wiring for Hardware Components, using Fuzzy PSO and Traveling Salesperson Problem

**Alireza Rezaee[1*] and Fariba Jahandideh Shekealgourabi[2]**

[1]Department of System and Mechatronics Engineering, Faculty of New Sciences and Technologies, University of Tehran, Tehran-143951374, Iran; arrezaee@ut.ac.ir
[2]Computer of Science, University of Tehran, Tehran-143951374, Iran; fariba.jahandideh2014@gmail.com

## Abstract

This study aims to find a way for accessing the optimal connection network, using the Particle Swarm for problems optimization as well as fuzzy theory for expanding the problem space into a fuzzy space as a more real space in the decisions. Having been tested on the random data, this method has proved to enjoy a relative algorithm in solving the problems.

**Keywords:** Fuzzy, Genetic Algoritms, Matrice, Particle Swarm Optimization, Traveling Salesperson, Wirling

## 1. Introduction

Designing the optimal wiring prior to the actual wiring and links creation has caught the attention of experts in many areas. Due to problem intricacy, such a design is of utmost importance. The volume of the cable used by the network, in particular the big ones, is one of the components needed to be determined in the architecture and during the creation of computer wiring. Providing an optimal map indicative of the required wiring can result in the considerable savings in the organizational expenditures. However, as for computer wiring, this requires that the various components of the wiring be constant and clear[1–3].

The optimal wiring design plays an important role in the design of digital circuits and VLSI circuits, with its accuracy being measured on the nanometer scale. This shows the extent to which the accuracy and the sensitivity are essential for developing the optimal wiring map and the estimation required for the wiring volume[4–7]. The development of wiring map for a building which is under construction is another area in which the optimal wiring is very essential[8].

The applications mentioned above as well as the other possible applications will entail different requirements, depending on the problem space. Consequently, the process of finding optimal wiring will be subject to various components and limitations[9,10]. As a matter of fact, finding the optimal wiring as in the case of Traveling Salesperson Problem takes on different forms as the problem space varies. This study seeks to model optimal wiring finding using Traveling Salesperson Problem in standard state. This is followed by seeking an optimal response for the problem, using fuzzy method and bio auditing. It is assumed that the distances between connection pins are determined and the cost of creating a link is translated into the length of the wire[11–12].

### 1.1 Traveling Salesperson Problem: an Optimal Wiring Model

Although on the face of it Traveling Salesperson Problem (TSO) is simple; due to its difficulty and complexity it has caught the attention of mathematicians and computer researchers. This problem is stated as follow: having taken account of the cost C required for traveling from a city I to the city j, Traveling Salesperson Problem intends to pass m given cities only once and back to its own city. The desirable route is one imposing the least cost and this is what TSP aims to do. TSO is important as it is an instance of significant swarms of problems called Combinational Optimization Problems.

*Author for correspondence

Drawing on Graph theory, another expression of TSP is as follows:

In graph G = (V, E) V equals the combination of nodes and E equals the combination of crest (یال) as E = {(a, b):a, b ∈ v}

Dab indicates the Euaclidus distance between a and b as $D_{ab} + D_{ba}$ TSP aims to find a closed route with minimal length which passes each node only once. This closed route is the Hamiltonian cycle in Graph.

As one out of many problems, optimal wiring can simply be modeled by TSP. The problem is stated as below:

A given set of pins is assumed. The problem aims to find a wiring for all pins so that the minimum volume of wires is used and each pin is attached to only two wires. The cost in this case equals the length of wire used for wiring whose minimum volume is obtained, using optimization algorithm.

The wiring to be found out in the above-mentioned problem is the Hamiltian cycle in TSP.

TSPis a suitable criterion against which many available optimization methods can be compared including genetic algorithm, simulated annealing, neural and colonial networks. Considering as a model, this study uses PSO fuzzy to solve TSP.

## 2. Particle Swarm Optimization

Introduced by Kennedy and Eberhart in 1995[13] Particle Swarm Optimization (PSO) are used in almost every application and every discipline of engineering in recent years. PSO is an evolutionary computational algorithm inspired by the nature based on the iteration. In description of PSO algorithm, a swarm of given particles is formed, receiving the primary values as they are subject to a set of random solutions. Each particle will be defined in terms of position and velocity which are modeled by position vertex and velocity vertex respectively. These particles move recurrently in n-dimension space, searching for new solutions by calculating the level of fitness as an assessment criterion. The number of components available in the function used in optimization equals the problem space dimension[14]. PSO algorithm for solving combinatorial optimization problems is to have good performance.

A memory is allocated for saving the best position of each particle in the past and another memory is allocated to saving the best position occurring among all the particles. Drawing on the experiences built up by these memories, the particles make a decision as how to make the next movement. In each iteration, all particles move in n-dimension space of the problem and finally find the general optimal point.

The following equations yield the position and velocity of each particle:

$$v_i^{t+1} = wV_i^t + C_1 \times Rand()$$
$$+C_2 \times Rand() \times (p_g{}^t$$
$$X_i^{t+1} = X^t{}_i + T$$

Index i and index t yield the particle ith and the number of iterations of algorithm up to now respectively. $V_i$ and $X_i$ show the velocity vertex and th position vertex of particle ith respectively.

$$v_i = (v_{i1}, v_{i2}, \ldots, V_{in})$$
$$X_i = (X_{i1}, X_{i2}, \ldots, X_{in})$$

In Equation (1), w denotes intertia/weight and $R_1$ and $R_2$ are the random digits contributing to the diversity of particle and has an even distribution in range (0,1) for the dimension of ith particle. C1 and C2 are the positive constants called self recognition component and social component coefficients respectively. Combined, these constants are called cognitive confidence coefficients.

Pi is the best local position obtained by particle ith up to now and $R_g$ is the best position a particle among all particles have achieved. Function rand can generate a random digit between 0 and 1.

### 2.1 Fuzzy PSO

In fuzzy PSO model, the position and velocity of particles shown by real vertexes will be expanded and reflected by fuzzy matrices. This will be pointed out in the definition of problem. As a result of this change, new signs and Operators will be introduced so that the equations concerning PSO algorithm are implemented in fuzzy mode as well.

A fuzzy equation on the finite sets X and Y as equation domains can be expressed by a fuzzy metric. Assume sets X and Y are Y = {$Y_1$, $Y_2$ … $Y_m$}, X = {$X_1$, $X_2$ … $X_m$} respectively. The equation from X to Y is stated as

$$R = (r_{ij})_{m\times n} = \begin{pmatrix} r_{11} \cdots r_{1n} \\ \vdots \quad \ddots \quad \vdots \\ r_{m1} \cdots r_{mn} \end{pmatrix} \tag{5}$$

Here $r_{ij} \in [0,1]$ indicates the membership degree of i th component and jth component Y in equation R.

To model the problem, some definitions are necessary

# 3. Description of the Problem

## 3.1 Definition of Problem State Space

Definition 1: Set S is a cluster of cities or put it other way a cluster of pins as a solution for TSP.

If S = {$S_1$, $S_2$ … $S_n$}, n is the number of pins and if $S_1$(i ∈ {1, 2 … n}) it is the ith node of obtained pin wiring.

Defeintion 2: set N is actually the serial number of a solution for TSP, formed by convergence of the serial number of each set. If n, N = {$N_1$, $N_2$ … $N_n$} is the number of cities or pins and $N_I$(I ∈ {1, 2, … n}) indicates a city or a pin in the problem space. Through the convergence of pins numbers, the number of wiring network is obtained.

Each $S_i$ ε S indicates that the current path has undergone i-1 pin and ith pins will be met in the next step. The space of state (SS) is as follows:

$$SS = S * N = \{[\![(S]\!]_i, N_I)\}: S_i \in S, N_i \in N\}$$

The fuzzy equation R which is defined from S to N indicates that for each element in matrice R

$$r_{\ddot{v}} = \mu_R(S_i, N_J), (0\langle r_{\ddot{v}}\langle 1) \tag{6}$$

$\mu_r$ is the function of equation membership. The value of $r_{ij}$ is the degree of membership of this event in which ith pin ($S_i$) in the cluster of possible response S is the pin ith serial number $N_j$.

## 3.2 Definitions of Symbols

The best position of each particle since the beginning of algorithm implementation up to now is defined as follows:

$$p = \begin{pmatrix} p_{11} & p_{12} \cdots p_{1n} \\ \vdots & \ddots & \vdots \\ p_{n1} & \cdots & p_{nn} \end{pmatrix} \tag{7}$$

And the current position of the particle is as follows

$$X = \begin{pmatrix} x_{11} & x_{12} \cdots x_{1n} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{nn} \end{pmatrix} \tag{8}$$

The elements of the above-mentioned martices are defined and interpreted similarly to that of the elements of matric (6).

The velocity of the particles is defined as

$$V = \begin{pmatrix} v_{11} & v_{12} \cdots v_{1n} \\ \vdots & \ddots & \vdots \\ v_{n1} & \cdots & v_{nn} \end{pmatrix} \tag{9}$$

## 3.3 Definition of Operator

As the position and velocity of particles have been converted into the matrices, the agents equations (1) and (2) need to be redefined. Symbol ϕ is the modified operator of multiplier. Given that α is a real digit, the expression α ϕ $V$ or α ϕ $X$ equals the multiplication of all elements of matrices V or X by α. two symbols ϕ and Θ are the modified operators of addition and subtraction respectively. Given that A and B are two matrices of position and velocity, AØB and AΘB are the normal addition and subtractions operations between two matrices respectively.

Being so, the modified equations PSO of the problem is obtained as

$$\begin{aligned} v_i^{t+1} &= w \otimes v_i^{\,t} \oplus (c_1 \times Rand()) \otimes (p_i^{\,t} \Theta X_i^{\,t} \\ &\oplus (C_2 \times Rand()) \otimes (p_g^{\,t} \Theta X_g^{\,t}) \\ X_i^{t+1} &= X_I^{\,t} \oplus V_i^{t+1} \end{aligned} \tag{10}$$

# 4. Investigation of Algorithm

## 4.1 The Primary Value Allocation

The primary value allocated for position as well as the best position of each particle is

$$X^0 = P^0 = \begin{pmatrix} p_{11} & \cdots & p_{1n} \\ \vdots & \ddots & \vdots \\ p_{n1} & \cdots & p_{nn} \end{pmatrix} \tag{12}$$

The elements of above matric are randomly generated and should meet the following conditions

$$\text{1)} \quad \sum_{j=1}^{"} p_{\ddot{v}} = 1, i \in \{1, 2, \ldots, n\}$$

$$\text{2)} \quad p_{\ddot{v}} \in (0,1) \tag{13)–(14}$$

The same is true for the primary values allocated for velocity as follows

$$v^0 = \begin{pmatrix} v_{11} & \cdots & v_{1n} \\ \vdots & \ddots & \vdots \\ v_{n1} & \cdots & v_{nn} \end{pmatrix} \quad (15)$$

The elements of this matrice are also randomly generated and should meet the following conditions

$$\sum_{j=1}^{n} v_{ij} = 0, i \in \{1,2,\ldots,n\} \quad (16)$$

The following theory shows the necessity of the above equation.

If conditions (13) and (16) are met in equations (10) and (11), it follows that after several iterations, the velocity will continue to meet the condition (16) and position will continue to meet condition (13).

The above theory is proved as follows:

Initially 4 Lemma are stated:

Lemma 1: Given that μ is a real digit, if Velocity V meets condition (16), then the same condition would be met by μ ϕ V as well.

Lemma 2: If P1 and P2 meet the equation (13) then $P_1$ ϕ $P_2$ is met by this equation as well.

Lemma 3: If P meets equation (13) and V meets equation (16) then equation (13) will meet PØV as well.

Lemma 4: If V1 and V2 meet equation (16) , then $V_1$ ϕ $V_2$ is met by the same equation as well.

The above theory can be induced as follows:

In equations (10) and (11) and in state t=0

$$\begin{aligned} v_i^1 &= w \otimes v_i^0 \oplus \left(c_1 \times Rand(\ )\right) \otimes \left(p_i^0 \Theta X_i^0\right) \\ &\oplus \left(c_2 \times Rand(\ )\right) \otimes \left(p_g^0 \Theta X_i^o\right) \\ X_i^1 &= X_i^0 \oplus V_i^1 \end{aligned} \quad (17),(18)$$

Here $V_i$ and $X_i$ are the primary values of velocity and position.

Three values A, B and C are defined as follows:

$$B = \left(C_1 \times Rand(\ )\right) \otimes \left(p_i^0 \Theta X_i^0\right) A = w \otimes V_i^0$$
$$.C = \left(C_2 \times Rand(\ )\right) \otimes \left(P_g^0 \Theta X_i^0\right)$$

$V_i$ and $X_i$ meet equation (16) and (13) respectively. Given Lemma1, A meets equation (16) so ($P_i^0$ Θ $X_{\downarrow} i^{\uparrow}$ (0)) meets the same equation, using Lemma 2. So B meets equation (16) according to Lemma 1. Finally $V_i^1 = A\phi B\phi C$ meets (16), using Lemma 4. Moreover, given (18) it follows that $X_i$ meets Equation (13), using Lemma 3. So if t = 1 then velocity and position meet equations (16) and (13).

Given that t+g, (g>0) then velocity $V_i^g$ and position $X_i^g$ meet equations (16) and (13). Similarely given $t = 0, V_i^{g+1}$ and $X_i^{g+1}$ meet the same equation when t = g + 1.

Given the above theory it follows that if equations (13) and (16) are met in value allocation phase, except for normalization; there will be no need for resetting the values of velocity and position while the algorithm is being implemented. This renders the computations simple.

## 4.2 The Normalization of Position Matric

Position matric (7) may violate condition (14) after several iterations. This requires the matric to be normalized. To do so, initially all negative values are converted to 0. Then the matric is transformed provided equation (13) is not violated.

$$\begin{pmatrix} P_{11} / \sum_{i-1}^{n} p_{1i} & p_{12} / \sum_{i-1}^{n} p_{1i} \cdots & P_{1n} / \sum_{i-1}^{n} P_{1i} \\ \vdots & \ddots & \vdots \\ P_{n1} / \sum_{i-1}^{n} p_{ni} & \cdots & P_{nn} / \sum_{i-1}^{n} p_{ni} \end{pmatrix} \quad (19)$$

There is no maximal value of velocity in velocity matric. The previous experience shows that this algorithm requires no use of maximal value for velocity (V_Max).

## 4.3 Defuzzication

Since matric of position (7) shows the potential TSP response, this matric needs to be decoded so that possible response can be obtained. This is called defuzzication which is made possible by using Max Number method:

In this method a presentation of flags is used for saving the selection or non-selection of matric columns and another row which denotes the path is used for saving the response path. Initially, no column is selected. Then an element having the highest value and not selected by previous rows is selected for each row of matric. Afterwards, the flag of column relating to that element is marked as “selected”. In addition, the number of that column is saved in path presentation. The response wiring is obtained from the path presentation after all rows having been processed. The length of wire required is also calculated.

The cost imposed by the position matric is the cost of TSP path, obtained by using maximal number as well as defuzzication of the matric.

### 4.4 Description of Algorithm

Step 1: Primary value allocation

Step 1.1 the number of particles “Max_Num” and the highest number of iterations” Max_Iteration” are determined.

Step 1.2 for each particle, a random position matric $X_i^0$ and a random velocity matric $V_i^t$ are allocated, using a primary allocation method. The best local position $P_i = X_i^0$ and the best general position make up the best $P_i$.

Step 2: If the number of present iterations (t) equals Max-iteration, step 5 needs to be implemented.

Step 3: For all i(s) ranging from 0 to Max_Num-1, the position and velocity of ith needs to be computed.

Step 3.1. The new velocity should be computed, using equation (17).

Step 3.2. The new position needs to be computed, using equation (18) and position matric should be normalized, using equation (19).

Step 3.3 the new position should be defuzzicated and the position costs should be calculated.

If the cost imposed by new position is less than the cost imposed by the best local position of the particle, then the best local position should be replaced by the new position.

Step 4: If the length of the wire in the best local position of some particles is less that the wire length in the best general position, then the best general position should be replaced by the best local position.

Step 2 should be implemented.

Step 5: The best response wiring and its length should appear in the output.

## 5. Solving a Sample Problem

The data of sample problem with 14 pins and randomly distributed distances are given in file Fuzzy_PSO.m. The proposed trial was operated on a computer with these specifications: Premium IV, CPU 2 GHz, M RAM 512, Operating System WINDOWS XP, version 7 MATLAB.

The trail shows that

Given 100 particles in problem space and when the components of equations (10) and (11) are as follows: c1 = c2 = 2 and w = 1.5

Then, 1000 iterations of the algorithm will yield a rather optimal response.

## 6. Discussion

This study combined PSO method and fuzzy theory as a rapid method for solving optimization problems for purpose of finding the linking loop of shortest length. TSP was used to model the problem of finding a loop with the least consumption of wire. The evaluation of this method and trials showed that although this combined method is not much better than the algorithm offered by Lin-Kernighan, it is a starting point for solving the combined optimization problems. The algorithm used in this study can also be applied in various problems of routing and other combinational optimization problems.

## 7. References

1. Pang W, Wang K, Zhou C-G, Dong L-J. Fuzz discrete particle swarm optimization for solving traveling salesman problem. The 4th IEEE International Conference on Computer and Information Technology; 2004.
2. Kennedy J, Eberhart R. Particle swarm optimization. IEEE International Conference on Neural Networks (Perth, Australia); Piscataway, JN: IEEE Service Center; 1995. p. 1942–8.
3. Abraham A, Guo H, Liu H. Swarm intelligence: foundations, perspectives and applications. A Swarm chapter on the Internet. 2004.
4. Benjamin KBA. Genertic alogorithms and the traveling salesman problem [Thesis]. Havey Mudd University, Department of Mathematics; 2000.
5. Eberhart RC, Shi Y. Comparing inertia weights and constriction factors in particle swarm optimization. 2000 Congress on Evolutionary Computing. 2000; 1:84–8.
6. Mohan CK, Al-kazemi B. Discrete particle swarm optimization. Proceedings Workshop on Particle Swarm Optimization; Indian polis, IN: Purdue School of Engineering and Technology, IUPUI; 2001.
7. Lin S, Kernighan BW. An effective heuristic algorithm for the traveling salesman problem. MD, USA; Operations Res 21, Inst Operations Research Management Sciences; 1973. p. 498–516.

8. Hua Z, Huang F. A variable-grouping based genetic algorithm for large-scale integer programming. Inf Sci. 2006; 176(19):2869–85.
9. Acampora G, Gaeta M, Loia V. Combining multi agent paradigm and memetic computing for personalized and adaptive learning experiences. Comput Intell J. 2011; 27(2):141–65.
10. Guvenc U, Duman S, Saracoglu B, Ozturk A. A hybrid GA–PSO approach based on similarity for various types of economic dispatch problems. Electron Electr Eng Kaunas: Technologija. 2010; 2(108):109–14.
11. Yan XS, Wu QH, Hu CY, Liang QZ. Circuit design based on particle swarm optimization algorithms. Key Engineering Materials. 2011; 474-476:1093–8.
12. Marinakis Y, Marinaki M. A hybrid multi-swarm particle swarm optimization algorithm for the probabilistic traveling salesman problem. Comput Oper Res, 2010; 37(3):432–42. Doi:10.1016/j.cor.2009.03.004.
13. Nikoui HR, Semsari M. Digital circuit design using chaotic particle swarm optimization assisted by genetic algorithm. Indian Journal of Science and Technology. 2013 Sep; 6(9):5182–8.
14. Soleimanian GF, Ebrahimi L, Maleki I, Gourabi SJ. A novel PSO based approach with hybrid of fuzzy C-means and learning automata in software cost estimation. Indian Journal of Science and Technology. 2014 Jun; 7(6):795–803.