

PAPER • OPEN ACCESS

SAEnet: an application for adjustment of horizontal network using particle swarm optimization algorithm

To cite this article: A Hafid *et al* 2019 *IOP Conf. Ser.: Earth Environ. Sci.* **389** 012048

View the [article online](#) for updates and enhancements.

SAEnet: an application for adjustment of horizontal network using particle swarm optimization algorithm

A Hafid, D N I Sari and F Sobatnu

Study Program of DIII Geodesy, Department of Civil Engineering, Politeknik Negeri Banjarmasin, Brigjen H. Hasan Basri St, Kayu Tangi, Banjarmasin, 70123, Indonesia

Abstract. The horizontal network of control stations plays an important role in geometric measurement activities, such as determining positions of point, distance, area, or even volume. One of the methods of determining horizontal network values is the least squares method. The method of least squares is one of the most frequently used methods to obtain unique estimates for a set of redundant measurements. The equations that are used in the horizontal network surveys are the nonlinear equations so it must be linearized in order to be computed using least squares method. The problem that arises when linearizing the nonlinear observation equation is the approximate coordinates of the network stations are needed for its computations. In order to handle this problem, the Particle Swarm Optimization (PSO) algorithm is applied as an algorithm for determining the approximate coordinate values of the network stations. The algorithm of PSO is a heuristic function that optimizes a problem by iteratively trying to improve a candidate solution toward the best solutions. The advantages of PSO are that PSO is designed for solving a problem more quickly when classic methods are too slow and finding an approximate solution when classic methods fail to find any exact solution. This paper presents the design of an application for horizontal network adjustment using PSO algorithm, namely SAEnet. The Algorithm was written in Matlab which is very effective in matrix manipulation and has been widely used in various fields of numerical computations. The computation of horizontal network adjustment using SAEnet application consists of determination of initial approximation values using PSO and the adjustment computation using the least squares method. SAEnet is also equipped with save menu to Microsoft Excel format and load menu from Microsoft Excel format. The test shows that the input and output of the program can run as expected. The results are then compared to other program, like STAR*NET, using a variety of data sets. The results of both computations show the maximum difference between both programs is 0.000040 while the average difference is 0.000020. The test shows that there is no significant difference between the results of the SAEnet and STAR*NET computation. Hence, SAEnet can be used for horizontal network adjustment computation.

1. Introduction

The horizontal network of control stations plays an important role in geometric measurement activities, such as determining positions of point, distance, area, or even volume. The horizontal control network comprises a series of points or positions which are spatially located for the purpose of topographic surveying, for the control of supplementary points, or dimensional control on site [1]. One of the methods of determining horizontal network station values is the least squares method. This method states



that *the most probable value for a quantity obtained from repeated observations is the value which renders the sum of the residuals squared a minimum* [2]. The method of least squares is one of the most frequently used methods to obtain unique estimates for a set of redundant measurements. The equation that is computed using least squares must be linear equation. In the case of nonlinear equation, that equation must be first linearized before the computation can be implemented.

The equations that are used in the horizontal network surveys are the nonlinear equations so it must be linearized in order to be computed using least squares method. The problem that arises when linearizing the nonlinear observation equation is the approximate coordinates of the network stations are needed for its computations. In order to handle this problem, the Particle Swarm Optimization (PSO) algorithm is applied as an algorithm for determining the approximate coordinates values of the network stations. The algorithm of PSO is a heuristic function that optimizes a problem by iteratively trying to improve a candidate solution toward the best solutions. The advantages of PSO are that PSO is designed for solving a problem more quickly when classic methods are too slow and finding an approximate solution when classic methods fail to find any exact solution.

This paper presents the design of an application for horizontal network adjustment using PSO algorithm, namely SAEnet. The Algorithm was written in Matlab which is very effective in matrix manipulation and has been widely used in various fields of numerical computations [3].

2. Method

The method of horizontal network adjustment consists of two sections: initial approximation values determination and the least squares adjustment. The initial approximations may be obtained by guessing or from observations. Certainly, the closer the initial approximations are to the final solution, the faster it will be obtained. As discussed above, the initial approximation values in this program are calculated using the help of Particle Swarm Optimization algorithm.

2.1. Particle swarm optimization

The Particle Swarm Optimization (PSO) was proposed by Kennedy and Eberhart. Particle Swarm Optimization as developed by the authors comprises a very simple concept. It requires only primitive mathematical operators, and is computationally inexpensive in terms of both memory requirements and speed [4]. The goal of the algorithm is to have all particles locate the optima in a multi-dimensional hyper-volume. This is achieved by assigning initially random positions to all particles in the space and small initial random velocities. The Particle Swarm Optimization algorithm is comprised of a collection of particles that move around the search space influenced by their own best past location and the best past location of the whole swarm or a close neighbour. Each iteration a particle's velocity is updated using [5]:

$$v_i(t+1) = v_i(t) + (c_1 \times \text{rand}() \times (p_i^{\text{best}} - p_i(t))) + (c_2 \times \text{rand}() \times (p_{g\text{best}} - p_i(t))) \quad (1)$$

where $v_i(t+1)$ is the new velocity for the i^{th} particle, c_1 and c_2 are the weighting coefficients for the personal best and global best positions respectively, $p_i(t)$ is the i^{th} particle's position at time t , p_i^{best} is the i^{th} particle's best position, and $p_{g\text{best}}$ is the best position known to the swarm. The $\text{rand}()$ function generate a uniformly random variable $\in [0,1]$. A particle's position is updated using:

$$p_i(t+1) = p_i(t) + v_i(t) \quad (2)$$

2.1.1. PSO for solving nonlinear equations

As discussed above, the equations that are used in the horizontal network surveys are the nonlinear equations. In order to solve systems of nonlinear equations using PSO, the nonlinear equations are transformed to an optimization problem. Let the form of a system of nonlinear equations be [6]:

$$\begin{aligned}
 f_1(x_1, x_2, \dots, x_n) &= 0 \\
 f_2(x_1, x_2, \dots, x_n) &= 0 \\
 &\vdots \\
 f_r(x_1, x_2, \dots, x_n) &= 0
 \end{aligned}
 \tag{3}$$

The equation (3) is then solved by using the auxiliary function:

$$F(x) = \sum_{i=1}^r f_i^2(x)
 \tag{4}$$

where $f_1(x^*), f_2(x^*), \dots, f_r(x^*)$ are the systems of nonlinear equations which have been modified so they are equal to zero, $F(x)$ is the sum of the square of the $f_i(x)$. The goal is to find the values of the unknowns x_1, x_2, \dots, x_n so the $f_1(x^*), f_2(x^*), \dots, f_r(x^*)$ are equal or close to zero.

2.2. Least squares adjustment

Many different methods have been derived for making adjustments in surveying. However, the method of least squares should be used because it has significant advantages over all other procedures. Its advantages can be summarized with the following four general statements: 1) it is the most rigorous of adjustments; 2) it can be applied with greater ease than other adjustments; 3) it enables rigorous post-adjustment analyses to be made; and 4) it can be used to perform pre-survey planning [2].

2.2.1. Least squares solution of nonlinear systems

A system of nonlinear equations that have been linearized can be written as:

$$JX = K + V
 \tag{5}$$

where the *Jacobian matrix* J contains the coefficients of the linearize observation equations. The individual matrices in equation (5) are:

$$J = \begin{bmatrix} \frac{\partial F_1}{\partial x_1} & \frac{\partial F_1}{\partial x_2} & \dots & \frac{\partial F_1}{\partial x_n} \\ \frac{\partial F_2}{\partial x_1} & \frac{\partial F_2}{\partial x_2} & \dots & \frac{\partial F_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial F_m}{\partial x_1} & \frac{\partial F_m}{\partial x_2} & \dots & \frac{\partial F_m}{\partial x_n} \end{bmatrix}$$

$$X = \begin{bmatrix} dx_1 \\ dx_2 \\ \vdots \\ dx_n \end{bmatrix} \quad K = \begin{bmatrix} l_1 - f_1(x_1, x_2, \dots, x_n) \\ l_2 - f_2(x_1, x_2, \dots, x_n) \\ \vdots \\ l_m - f_m(x_1, x_2, \dots, x_n) \end{bmatrix} \quad V = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_m \end{bmatrix}$$

The vector of least squares corrections in the equally weighted system of equation (5) is given by

$$X = (J^T J)^{-1} J^T K = N^{-1} J^T K
 \tag{6}$$

Similarly, the system of weighted equations is

$$WJX = WK \tag{7}$$

and its solution is

$$X = (J^T W J)^{-1} J^T W K = N^{-1} J^T W K \tag{8}$$

where X is the unknown corrections matrix, K the constants matrix, W the weight matrix, V the residuals matrix.

2.3. Nonlinear observation equations

For this network adjustment, there are three types of observation equation [7]:

2.3.1. Distance observation equation.

Distance between two points is related to the coordinates of the two points as follows:

$$l_{ij} + v_{l_{ij}} = ((x_j - x_i)^2 + (y_j - y_i)^2)^{1/2} \tag{9}$$

In equation (9), l_{ij} is the observed distance of a line between stations I and J , $v_{l_{ij}}$ the residual in the observation l_{ij} , x_i and y_i the most probable coordinate values for station I , and x_j and y_j the most probable coordinate values for station J .

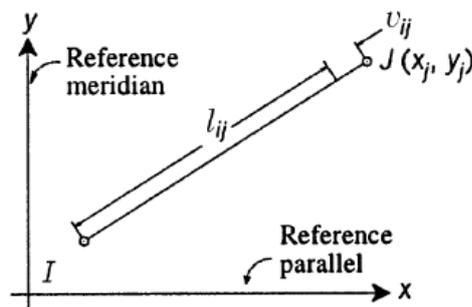


Figure 1. Observation of a distance. Adapted from “Observation of a distance” by Ghilani C D and Wolf P R 2006 *Adjustment Computations: Spatial Data Analysis* p 235

2.3.2 Azimuth observation equation

The complete observation equation for an observed azimuth of line IJ is

$$\tan^{-1} \frac{x_j - x_i}{y_j - y_i} + C = Az_{ij} + v_{Az_{ij}} \tag{10}$$

where Az_{ij} is the observed azimuth, $v_{Az_{ij}}$ the residual in the observed azimuth, x_i and y_i the most probable values for the coordinates of station I , x_j and y_j the most probable values for the coordinates of station J , and C a constant with value based on Table 1.

Table 1. Relationship between the Quadrant, C , and the Azimuth of the Line.

Quadrant	Sign($x_j - x_i$)	Sign($y_j - y_i$)	Sign α	C	Azimuth
I	+	+	+	0	α
II	+	-	-	180°	$\alpha + 180^\circ$
III	-	-	+	180°	$\alpha + 180^\circ$
IV	-	+	-	360°	$\alpha + 360^\circ$

2.3.3. Angle observation equation

Figure 2 illustrates the geometry for an angle observation. A horizontal angle observation equation can be written as the difference between two azimuth observations, and thus for clockwise angles,

$$\angle BIF = Az_{IF} - Az_{IB} = \tan^{-1} \frac{(x_f - x_i)}{(y_f - y_i)} - \tan^{-1} \frac{(x_b - x_i)}{(y_b - y_i)} + D = \theta_{bif} + v_{\theta bif} \quad (11)$$

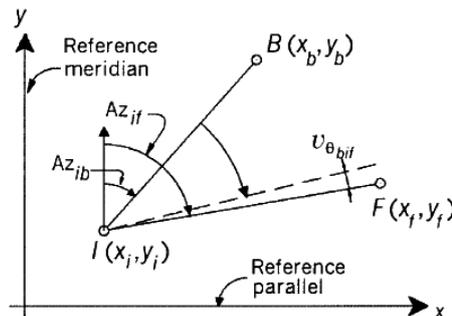


Figure 2. Relationship between an angle and two azimuths. Adapted from “Relationship between an angle and two azimuths” by Ghilani C D and Wolf P R 2006 *Adjustment Computations: Spatial Data Analysis* p 258

where θ_{bif} is the observed clockwise angle, $v_{\theta bif}$ the residual in the observed angle, x_b and y_b the most probable values for the coordinates of the backsighted station B , x_i and y_i the most probable values for the coordinates of the instrument station I , x_f and y_f the most probable values for the coordinates of the foresighted station F , and D a constant that depends on the quadrants in which the backsight and foresight occur. This term can be computed as the difference between the C terms from Equation (10) as applied to the backsight and foresight azimuths; that is,

$$D = C_{if} - C_{ib} \quad (12)$$

2.4 Linearized observation equations

For network adjustment computation, the above nonlinear observation equation has to be linearized, and an iterative solution approach is used. Assuming that the approximate coordinates of the network stations are available, the nonlinear observation equations can be approximated by the linear term as follows:

2.4.1 Horizontal Distance

The linearized form of equation (9) is

$$\left(\frac{x_i - x_j}{IJ}\right)_0 dx_i + \left(\frac{y_i - y_j}{IJ}\right)_0 dy_i + \left(\frac{x_j - x_i}{IJ}\right)_0 dx_j + \left(\frac{y_j - y_i}{IJ}\right)_0 dy_j = k_{l_{ij}} + v_{l_{ij}} \quad (13)$$

where $(\cdot)_0$ is evaluated at the approximate parameter values, $k_{l_{ij}} = l_{ij} - IJ_0$, and

$$IJ_0 = ((x_{j0} - x_{i0})^2 + (y_{j0} - y_{i0})^2)^{1/2}$$

2.4.2 Azimuth

The linearized form of equation (10) is

$$\left(\frac{y_i - y_j}{IJ^2}\right)_0 dx_i + \left(\frac{x_j - x_i}{IJ^2}\right)_0 dy_i + \left(\frac{y_j - y_i}{IJ^2}\right)_0 dx_j + \left(\frac{x_i - x_j}{IJ^2}\right)_0 dy_j = k_{Az_{ij}} + v_{Az_{ij}} \quad (14)$$

Both $k_{Az_{ij}} = Az_{ij} - \left[\tan^{-1} \left(\frac{x_j - x_i}{y_j - y_i} \right)_0 + C \right]$ and $IJ^2 = (x_j - x_i)_0^2 + (y_j - y_i)_0^2$ are evaluated using the approximate coordinate values of the unknown parameters.

2.4.3. Angle

Equation (11) expressed as a linearized expansion is

$$\begin{aligned} & \left(\frac{y_i - y_b}{IB^2} \right)_0 dx_b + \left(\frac{x_b - x_i}{IB^2} \right)_0 dy_b + \left(\frac{y_b - y_i}{IB^2} - \frac{y_f - y_i}{IF^2} \right)_0 dx_i \\ & + \left(\frac{x_i - x_b}{IB^2} - \frac{x_i - x_f}{IF^2} \right)_0 dy_i + \left(\frac{y_f - y_i}{IF^2} \right)_0 dx_f + \left(\frac{x_i - x_f}{IF^2} \right)_0 dy_f \\ & = k_{\theta_{bif}} + v_{\theta_{bif}} \end{aligned} \tag{15}$$

where

$$k_{\theta_{bif}} = \theta_{bif} - \theta_{bif_0} \quad \theta_{bif_0} = \tan^{-1} \left(\frac{x_f - x_i}{y_f - y_i} \right) - \tan^{-1} \left(\frac{x_b - x_i}{y_b - y_i} \right) + D$$

$$IB^2 = (x_b - x_i)^2 + (y_b - y_i)^2 \quad IF^2 = (x_f - x_i)^2 + (y_f - y_i)^2$$

are evaluated at the approximate values for the unknowns.

2.5 Flowchart of the program

The flowchart of how this program runs is illustrated in Figure 3.

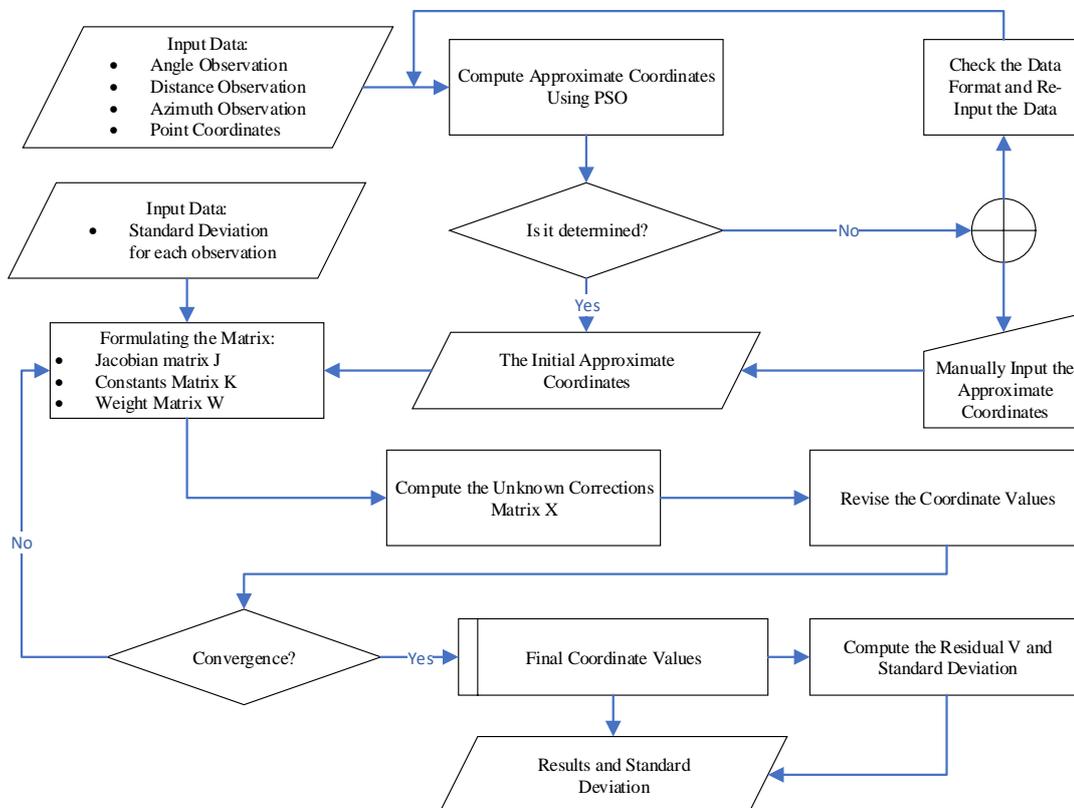


Figure 3. Flowchart of the running program.

The explanation of the flowchart are as follows:

1. Input the distance, angle, and/or azimuth observations, and the known point coordinates.
2. Compute the initial approximation of the unknown coordinates based on the input data using PSO algorithm.
3. Formulate the J and K matrix based on the observations, known points, and the approximation points data.
4. If there are any standard deviation from the observations data, input those data so the W matrix can be formulated.
5. Compute the unknown coordinate corrections (X matrix) using the least squares method.
6. Repeat the computation until the value of X becomes negligible.

3. Results and analysis

3.1. SAEnet user interface

The interface of SAEnet consists of 6 panels in only one window; those are *Distance Observation*, *Angle Observation*, *Azimuth Observation*, *Point Data Properties*, *Process*, and *Post-Adjustment Facts* panel. This Interface design has been able to compute any type of terrestrial horizontal surveys, i.e., trilateration, triangulation, resection, intersection, traverses, polygon, and network survey.

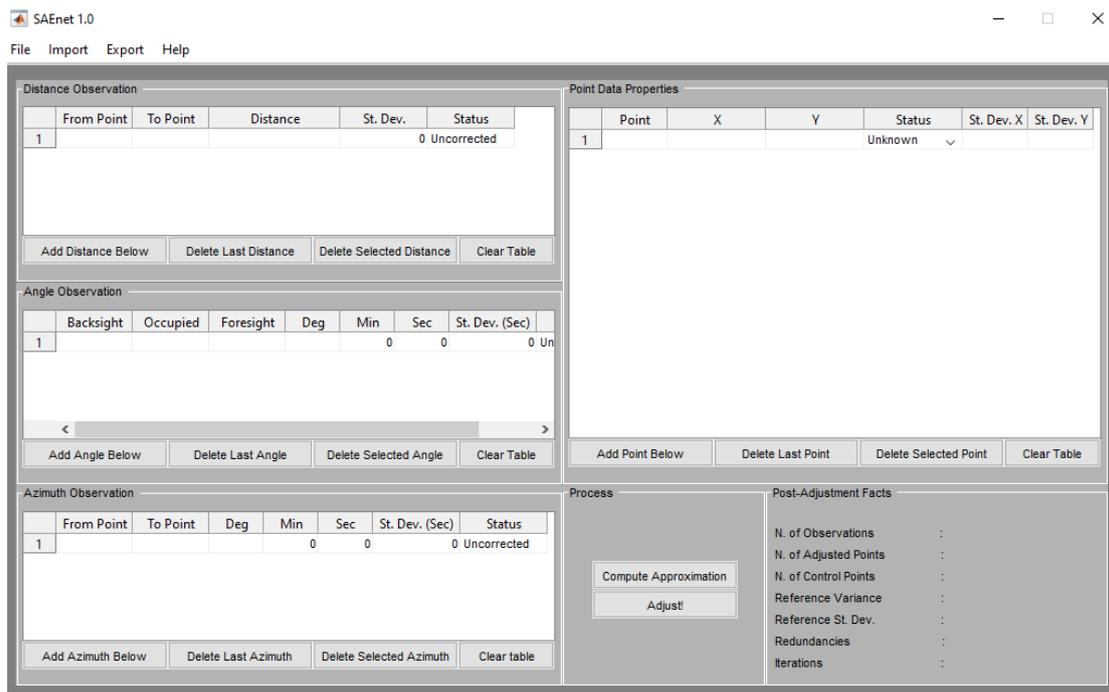


Figure 4. The main interface of SAEnet

3.2. SAEnet input data

This program can read data in *MS Excel* (*.xls, *.xlsx) format which facilitate the user to compile the input data in *MS Excel* program. User also can input the data directly into the program in case of there is no *MS Excel* Installed on the user’s PC/laptop. The input data of this program consists of *Distance Observation*, *Angle Observation*, *Azimuth Observation*, and *Point Data Properties*. The examples of input data are shown in Figure 5.

	A	B	C	D	E	F	G
1	Backsight	Occupied	Foresight	Deg	Min	Sec	St_Dev_(S
2	R	Q	S	38	48	50.7	4
3	S	Q	T	47	46	12.4	4
4	T	Q	R	273	24	56.5	4.4
5	Q	R	S	269	57	33.4	4.7

Figure 5a. Example of input angle data for SAEnet

	A	B	C
1	Point	X	Y
2	Q	1000	1000
3	R	1003.06	2640.01
4	S	2323.07	2638.47
5	T	2661.75	1096.07

Figure 5b. Example of input point data for SAEnet

	A	B	C	D
1	From_Poi	To_Point	Distance	St_Dev
2	Q	R	1640.016	0.026
3	R	S	1320.001	0.024
4	S	T	1579.123	0.025
5	T	Q	1664.524	0.026

Figure 5c. Example of input distance data for SAEnet

3.3. SAEnet processing data

The process of SAEnet program consists of two steps, *approximation values computation* and *adjustment computation*. The initial approximation values in SAEnet program is calculated by PSO algorithm and the adjustment process is executed by least squares method. The process panel for processing data in SAEnet program is shown in Figure 6.

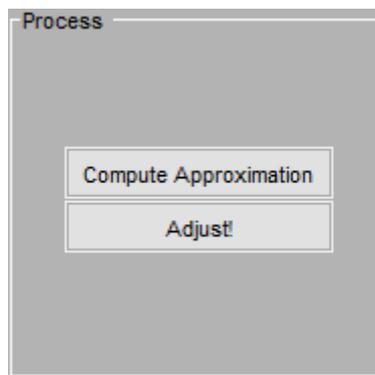


Figure 6. Two simple buttons for processing data in SAEnet

3.4. SAEnet output data

The main result of SAEnet program is the 2D adjusted coordinates of points. Furthermore, this program also shows the corrections of all the calculated observations, the standard deviation of points, and the post-adjustment facts which consists of number of observations, number of adjusted points, number of control points, number of iterations, redundancies, and reference of variance and standard deviation. The results of this program are then compared with another program named STAR*NET from MicroSurvey.

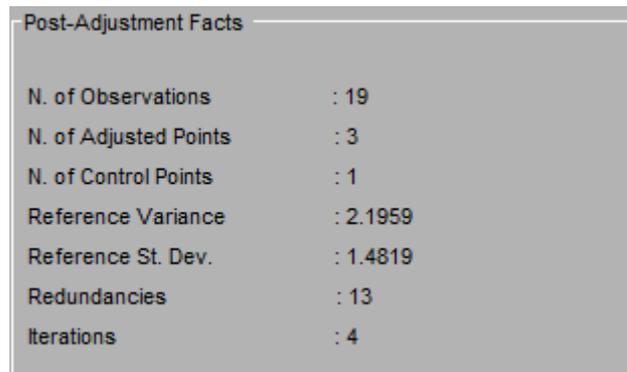


Figure 7. Example of post-adjustment facts results

*3.4.1 Comparison between SAEnet and STAR*NET results*

In order to check the validity of the SAEnet program, the computation results of the SAEnet are compared with STAR*NET program developed by MicroSurvey Software Inc. using a variety of data sets. The testing aspects include: the difference of adjusted coordinate values and the standard deviations of each point. The test shows that there is no significant difference between the results of the SAEnet and STAR*NET computation.

Table 2. Comparison of 2D adjusted coordinates between SAEnet and STAR*NET.

Point	SAEnet		STAR*NET		ΔX	ΔY
	X	Y	X	Y		
Q	1000	1000	1000	1000	0	0
R	1003.0571	2639.9747	1003.0571	2639.9747	5.44E-06	3.79E-05
S	2323.0748	2638.4481	2323.0748	2638.4481	7.18E-06	4.25E-05
T	2661.7540	1096.0556	2661.7540	1096.0556	4.7E-06	1.94E-05

One of data sets is presented in this paper to show its comparison. This data set includes of 4 stations with 1 station is set as fixed, 1 azimuth observation, 6 distance and 12 angle observations. The distribution of the points and its shape of network is shown in Figure. 8. The results of both computations show the maximum difference between both programs is 0.000040 while the average difference is 0.000020.

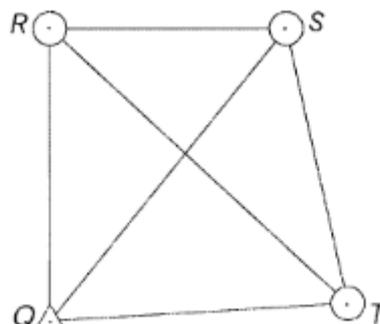


Figure 8. Example of horizontal network.

4. Conclusions

In this paper, a program for horizontal (2D) network adjustment using Particle Swarm Optimization, called SAEnet, has been presented. The minor difference either in 2D adjusted coordinates or the standard deviations of points compared with other program (STAR*NET), which are shown in the Table 2 and Table 3, shows that the SAEnet program can be used for the horizontal network adjustment computation.

Table 3. Comparison of standard deviations results between SAEnet and STAR*NET.

Point	SAEnet		STAR*NET		ΔX	ΔY
	St. Dev. X	St. Dev. Y	St. Dev. X	St. Dev. Y		
Q	0	0	0	0	0	0
R	0.000045	0.023566	0.000045	0.023566	4.82E-07	1.36E-07
S	0.022811	0.026711	0.022811	0.026711	1.65E-07	3.53E-07
T	0.024407	0.029249	0.024407	0.029249	1.76E-07	1.73E-07

5. References

- [1] Schofield W and Breach M 2007 *Engineering Surveying* 6th ed Elsevier Ltd p 6
- [2] Ghilani C D and Wolf P R 2006 *Adjustment Computations: Spatial Data Analysis* (New Jersey: John Wiley & Sons, Inc.)
- [3] Hidayat H 2014 K-Level: an Application for Adjustment of Leveling Network Based On Least Square Method *Jurnal Ilmiah Geomatika* Vol 20 pp 115-120
- [4] Kennedy J and Eberhart R 1995 Particle Swarm Optimization pp 1942-48
- [5] Brownlee J 2011 *Clever Algorithms: Nature-Inspired Programming Recipes* (Melbourne: LuLu) p 232
- [6] Jaberipour M, Khorram E and Karimi B 2011 Particle Swarm Algorithm for Solving Systems of Nonlinear Equations *Computers and Mathematics with Applications* vol 62 (Iran: Elsevier) pp 566-76
- [7] Olyazadeh R, Setan H and Fouladinejad N 2011 Network Adjustment Program using MATLAB *Geospatial World Forum*