Artificial Intelligence Approaches for GPS GDOP Classification

Nadali Zarei
Department of Electrical Engineering,
University of Yazd

ABSTRACT
Geometrical dilution of precision (GDOP) concept is a powerful and widespread quantify for determining the errors resulting from satellite configuration geometry. GDOP computation is based on the complicated transformation and inversion of measurement matrices that has a time and power burden. Also, basic back propagation neural network (BPNN) is easy to fall into local minima. To overcome this problem, in this study we propose an approach based on neural network (NN) and evolutionary algorithms (EAs) for GPS GDOP classification. In this article we use a number of EAs such as genetic algorithm (GA), particle swarm optimization (PSO), new PSO (NPSO), and imperialist competitive algorithm (ICA) to train an NN. Simulation results illustrate that the proposed methods have superiority performance.

General Terms
Signal Processing, and Pattern Recognition, Artificial Intelligence.

Keywords
GPS GDOP classification, neural network (NN), genetic algorithm (GA), particle swarm optimization (PSO), new PSO (NPSO), and imperialist competitive algorithm (ICA).

1. INTRODUCTION
Global positioning system (GPS) is a satellite based positioning system funded by the U.S. Department of Defense (DOD) in 1973. There are at least 24 satellites orbiting the earth in 6 planes transmitting signals which are received by a GPS receiver [1,2]. In order to verify the accuracy of the position, the user equivalent range error, the error of a component in the distance from receiver to a satellite, is multiplied by a factor which depends on the geometry of satellites constellation. This factor is called the geometric dilution of precision (GDOP) [2]. There are basically two main approaches employed to justify the satellite geometry based on the GPS GDOP including approximation and classification. Unlike the GPS GDOP approximate methods which compute the values in order to select the optimal subsets of the satellites, the GPS GDOP classification methods are employed to choose one of the acceptable subsets (four optimum satellites from 24 existing satellites) of satellites for navigation uses [1-3].

A subset of satellites with the value of GPS GDOP less than 2 is ideal, i.e. the attained measure of location and time by these satellites are trustworthy, but a higher GPS GDOP indicates poorer satellites positioning and an inferior measurement configuration [4]. The GPS GDOP ratings are represented in Table 1 [5].

Table 1: GPS GDOP ratings

<table>
<thead>
<tr>
<th>Class number</th>
<th>GPS GDOP value</th>
<th>Ratings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class 1</td>
<td>1</td>
<td>Ideal</td>
</tr>
<tr>
<td>Class 2</td>
<td>2-3</td>
<td>Excellent</td>
</tr>
<tr>
<td>Class 3</td>
<td>4-6</td>
<td>Good</td>
</tr>
<tr>
<td>Class 4</td>
<td>7-8</td>
<td>Moderate</td>
</tr>
<tr>
<td>Class 5</td>
<td>9-20</td>
<td>Fair</td>
</tr>
<tr>
<td>Class 6</td>
<td>21-50</td>
<td>Poor</td>
</tr>
</tbody>
</table>

To overcome this problem, in this study we propose an approach based on neural network (NN) and evolutionary algorithms (EAs) for GPS GDOP classification. In this article we use a number of EAs including genetic algorithm (GA), particle swarm optimization (PSO), new PSO (NPSO), and imperialist competitive algorithm (ICA) to train an NN.

The rest of the paper is organized as follows: Section 2 presents GPS GDOP briefly. In Section 3, EAs including genetic algorithm (GA), particle swarm optimization (PSO), new PSO (NPSO), and imperialist competitive algorithm (ICA) are explained. Section 3 describes the proposed methods. Simulation results and comparing with existing well-known methods are discussed in Section 5. Finally, Section 6 includes the conclusion of the study.

2. THE CONCEPT OF GPS GDOP
Basically, the GPS accuracy and performance are dependant on the GDOP. Figure 1 illustrates the geometry of the satellites and its affect on the GDOP values [1].
We assume the definitions of GDOP computation, useful for the sequel of the document and help understanding our contributions. The absolute distance between a satellite and a user is defined as follows [8,9]:

\[ R_i = \sqrt{(X_i - X_u)^2 + (Y_i - Y_u)^2 + (Z_i - Z_u)^2} - \delta t_u \]  

(1)

where:

\[ \Delta \rho_{\text{trop},i} \] and \[ \Delta \rho_{\text{iono},i} \] which are the errors induced by the ionospheric and the tropospheric propagation, are calculated from a model, \( (X_u, Y_u, Z_u, \delta t_u) \) are the four system unknowns and \( \delta t_u \) is the correction the receiver has to apply to its own clock. To resolve this system we need four equations which mean four pseudo-ranges from four different satellites. The pseudo-ranges can be approximated by a Taylor expansion. We obtain:

\[ \rho_i = \sqrt{(X_i - X_u)^2 + (Y_i - Y_u)^2 + (Z_i - Z_u)^2} \]

(2)

\[ -\delta t_u \]

The Taylor expansion at the first order is:

\[ \Delta \rho_i = \rho_i - \rho_i = a_{xi} \Delta x_i + a_{yi} \Delta y_i + a_{zi} \Delta z_i \]

(4)

where:

\[ a_{xi} = \frac{X_i - X_u}{r_i} ; a_{yi} = \frac{Y_i - Y_u}{r_i} ; a_{zi} = \frac{Z_i - Z_u}{r_i} \]

\[ r_i = \sqrt{(X_i - X_u)^2 + (Y_i - Y_u)^2 + (Z_i - Z_u)^2} \]

(5)

Let assume \( N_{\text{sat}} \) be the number of visible satellites. The matrix \( H \) is as follows:

\[
H = \begin{bmatrix}
  a_{x1} & a_{y1} & a_{z1} & 1 \\
  a_{x2} & a_{y2} & a_{z2} & 1 \\
  \vdots & \vdots & \vdots & \vdots \\
  a_{xN_{\text{sat}}} & a_{yN_{\text{sat}}} & a_{zN_{\text{sat}}} & 1 
\end{bmatrix}
\]

(6)

Let define the \( G \) matrix:

\[ G = (HTH)^{-1} \]

(7)

The GDOP is:

\[ GDOP = \sqrt{\text{trace}(G)} \]

(8)

3. EVOLUTIONARY ALGORITHMS (EAs)

In this section we explain four mentioned EAs briefly.

3.1 Particle Swarm Optimization and New Particle Swarm Optimization

The idea of PSO was first proposed by J. Kennedy and R. Eberhart in 1995 [10]. PSO is an evolutionary computing algorithm inspired by nature and is based on repetition. The
social behavioral of animals such as birds and fish when they are together has been the inspiration source for this algorithm [10]. PSO, the same as other evolutionary algorithms, starts with a random matrix as a primary population. Unlike genetic algorithms (GA), standard PSO doesn’t have evolutionary operators like mutation and breeding. Each member of the population is named a particle. As a matter of fact, in the PSO algorithm a certain number of particles that are formed randomly make the primary values. There are two parameters for each particle including position and velocity of the particle, which are defined by a space vector and a velocity vector, respectively. These particles form a pattern in an n-dimensional space and move to the desired value. The best position of each particle in the past and the best position among all particles are stored separately. According to the experience from the preceding moves, the particles make a decision about how to decide the next move. In each iteration, all particles in the n-dimensional problem space move to an optimum point. In each iteration, the position and velocity of each particle can be modified according to the following equations:

\[
v_i(t+1) = w v_i(t) + C_1 r_1 (p_{best}(t) - x_i(t)) + C_2 r_2 (g_{best}(t) - x_i(t))
\]

\[
x_i(t+1) = x_i(t) + v_i(t+1)
\]

where \(n\) shows the dimension (1 ≤ n ≤ N), \(C_1\) and \(C_2\) are positive constants, generally considered 2.0. \(r_1\) and \(r_2\) are random numbers uniformly between 0 and 1; \(w\) is inertia weight that can be constant [11].

Equation (9) states that the velocity vector of each particle is updated \((v_i(t+1))\) and the new and previous values of the vector position \((x_i(t))\) create the new position vector \((x_i(t+1))\). As a matter of fact, the updated velocity vector influences both local and global values. The best response of the local positions is the best solution of the particle until current execution time \((p_{best})\) and the best global solution is the best solution of the entire particles until current execution time \((g_{best})\). Flowchart of PSO is presented in Figure 2.

Figure 2. Flowchart of PSO

Because PSO stays in local minima of fitness function, we use NPSO. In each iteration, as it was mentioned in PSO, global best particle and local best particle are calculated. NPSO strategy uses the global best particle and local “worst” particle, the particle with the worst fitness value of the particle until current execution time [12]. It can be defined as:

\[
v_i(t+1) = w v_i(t) + C_1 r_1 (p_{worst}(t) - x_i(t)) + C_2 r_2 (g_{best}(t) - x_i(t))
\]

3.2. Genetic Algorithm

A GA has three major steps. In the first step we create an initial population of \(m\) randomly selected individuals. The first generation is created by the initial population. The second step inputs \(m\) individuals and gives as output an assessment for each of them based on an objective function known as fitness function. This evaluation explains proximity value of our demands respect to each one of these \(m\) individuals. The third step is responsible for the formulation of the subsequently generation [8,13].

The fittest individuals of the previous one create a new generation. This evaluation procedure of generation \(N\) and production of generation \(N+1\) is iterated until a performance criterion is joined. The creation of offspring based on the fittest individuals of the previous generation is named as breeding. The breeding process has three basic genetic
operations, namely, reproduction, crossover and mutation [8,13].

In each iteration of GA, some of the individuals with highest fitness value are eliminated and replaced by new members. New members are produced by so called parents. Crossover selects probabilistically two of fittest individuals of generation N; then in a random way selects a number of their characteristics and exchanges them in a way that the selected characteristics of the first individual would be obtained by the second and vice versa. Following this procedure creates two new children that both belong to the new generation. Mutation is a physiologically-inspired disturbance to the system and is often used to avoid possible local minima. To model this for real-world optimization, initially, a random number for each bit of chromosome, then, if the random number is greater than a pre-defined “mutation threshold”, that bit is flipped. Finally, the flowchart of GA is presented in Figure 3 [8,13].

3.3. Imperialist Competitive Algorithm

ICA is a new population based optimization algorithm proposed in 2007 by Atashpaz-Gargari and Lucas [14]. Nowadays ICA has numerous applications like designing controller for industrial systems, solving optimization problems in PID controller, communication systems, and training and analysis of artificial NNs [15-18].

Similar to every evolutionary algorithm, this algorithm starts with the preliminary population with random numbers that each of them is called a “Country”. A number of the members of the population that have best fitness values are chosen as imperialists. Each member of the remaining population is named a colony. Total fitness values of an empire depends on both the power of the imperialist country and the power of its colonies [13,15].

In each step, the countries go toward their related imperialist. If the fitness value of a colony achieves more than its related fitness value of imperialist then, this colony and its related imperialist transform to imperialist and colony, respectively. In each step, the weakest colony of the weakest empire moves toward the closest empire, and the empire without any colony is eliminated. After a while, all empires fall down except for the most powerful one and all the colonies go under the control of this unique empire. Finally the flowchart of the ICA is illustrated in Figure 4 [13-16].

Figure 3. The flowchart of GA-based optimization cycle

Figure 4. Flowchart of the ICA
4. THE GPS GDOP CLASSIFICATION METHODS

First, in order to reduce the instruction time, all of the input and output variants become between 0 and 1 by normalizing. Since $H^TH$ is a $4 \times 4$ matrix, it has four eigenvalues, $\lambda_i$ $(i = 1, 2, 3, 4)$, and it is clear that the four eigenvalues for the matrix $(H^TH)^{-1}$ will be $\lambda_i^{-1}$. Based on the fact that the trace of a matrix is the sum of its eigenvalues, Equation (8) would be as below [4,5]:

$$GDOP = \sqrt{\lambda_1^{-1} + \lambda_2^{-1} + \lambda_3^{-1} + \lambda_4^{-1}}$$

(12)

Mapping with the definition of four variants would be done as below:

$$x_1(\vec{\lambda}) = \lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 = \text{trace}(H^TH)$$

(13)

$$x_2(\vec{\lambda}) = \lambda_1^2 + \lambda_2^2 + \lambda_3^2 + \lambda_4^2 = \text{trace}((H^TH)^2)$$

(14)

$$x_3(\vec{\lambda}) = \lambda_1^3 + \lambda_2^3 + \lambda_3^3 + \lambda_4^3 = \text{trace}((H^TH)^3)$$

(15)

$$x_4(\vec{\lambda}) = \lambda_1\lambda_2\lambda_3\lambda_4 = \det(H^TH)$$

(16)

The GDOP can be considered as a mapping directly from $\vec{X}$ to GDOP classes. Mapping from $\vec{X}$ to the GDOP classes is very non-linear and cannot be determined analytically, but it can be determined exactly by the NN. In this paper, NN is planned to do the mapping from $\vec{X}$ to the GPS GDOP classes. Figure 5 shows the whole classification block diagram of GPS GDOP using NN. As mentioned before in this paper we propose four EAs including ICA, GA, PSO, and NPSO.

5. SIMULATIONS AND DISCUSSIONS

Six classes or outputs for the proposed NNs are considered for GPS GDOP classification. When the GDOP is higher than the first threshold, NN outputs are [1 0 0] (for the large state) and when the GDOP is placed between the first and second threshold, NN outputs are [0 1 0] (for the average case) and when the GDOP is smaller than the second threshold, NN outputs are [0 0 1] (for the low state). Table 2 presents the accuracy numbers of GPS GDOP classification. Table 5 shows comparing accuracy between these three methods for GPS GDOP. The simulation results demonstrate that GPS GDOP classification using NN and ICA has better accuracy than other NN-bases approaches including GA, PSO, and NPSO.

![Figure 5: Classification block diagram of GPS GDOP using NN](image)

Table 2: Comparison of classification rates for proposed methods

<table>
<thead>
<tr>
<th>Classification Methods</th>
<th>NN trained by ICA</th>
<th>NN trained by GA</th>
<th>NN trained by NPSO</th>
<th>NN trained by PSO</th>
<th>BPNN [1]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Classification rates</td>
<td>95.15%</td>
<td>94.31%</td>
<td>94.84%</td>
<td>94.17%</td>
<td>93.16%</td>
</tr>
</tbody>
</table>

6. CONCLUSIONS

GPS GDOP shows the geometrical relationship between positioning accuracy and satellite configuration. It can be calculated by using the positions of four visible satellites and the position of the receiver. The ideal GPS GDOP value is equal or less than one. Values of two to three are considered acceptable, whereas values greater than nine are considered high and should be discarded. Existing approaches for GPS GDOP approximation and classification are time-consuming and unreliable. In this paper, we propose four EAs, namely PSO, NPSO, GA, and ICA, to train an NN. The simulation results demonstrate that GPS GDOP classification using NN and ICA has better accuracy than simple classification using NN and other NN-based approaches based on the EAs.

7. REFERENCES


