XI TUFUAB 2022
TÜRKİYE ULUSAL FOTOGRAMETRİ VE UZAKTAN ALGILAMA BİRLİĞİ TEKNİK SEMPOZYUMU
MERSİN / 12-14 MAYIS 2022

TUFUAB -MERSIN 2021

https://tufuab2021.mersin.edu.tr/

3B Benzerlik Dönüşümü için Başlangıç Parametrelerinin Evrimsel Arama Algoritmaları ile Kestirilmesi

Mohammed Nasser[*1], **Ümit Haluk Atasever** [1]

[1]*Erciyes University ,Geomatics Engineering,Kayseri,Turkey*

Anahtar Kelimeler
3D Benzerlik Dönüşüm,
Başlatma,
optimizasyon,
Levenberg-Marquardt,

ÖZ

Üç Boyutlu Benzerlik Dönüşümü, Geomatik uygulamalarında yaygın olarak kullanılan bir dönüşümdür. Dönüşüm parametreleri genellikle En Küçük Kareler Tekniği ve Levenberg-Marquardt gibi yöntemler kullanarak hesaplanmasına karşın, bu tekniklerin iyi bir başlangıç değerine ihtiyaçları vardır. Bu problem arama problemi olarak sınıflandırılmış ve araştırmacılar tarafından Monte Carlo araması ve kaba kuvvet araması gibi teknikler kullanılarak çözülmeye çalışılmıştır. Bu makalede, oluşturulan simülasyon verileri ve Sudan Adindan datum ve WGS84 arasındaki dönüşümü gerçekleştirmek için kullanılacak dönüşüm parametrelerinin başlangıç değerleri beş evrimsel arama algoritması (DE, WDE, BSD, Cuckoo Search ve GSA) kullanılarak hesaplanmış ve performansları incelenmiştir.. Ayrıca Bayes optimizasyonu, her bir algoritmanın kontrol parametrelerini ayarlamak için bir araç olarak kullanılmıştır, Sonuç olarak, DE ve BSD algoritmalarının hızlı ve istikrarlı performans nedeniyle başlangıç değerlerini hesaplamak için en uygun algoritmalar olduğu görülmüştür. Bu algoritmalar için ortalama çalışma süresi sırasıyla 0.3225 sn ve 0.3512 sn iken, WDE algoritması ancak ortalama 1.0856 sn çalışma süresi ile istikrarlı bir performansa sahiptir, Guguk kuşu arama algoritması, kontrol noktaları için daha düşük aralıklarda iyi bir performans sergiledi ve daha yüksek aralıklarda ise performansta düşüş gözlemlenmiştir. Ancak diğer taraftan GSA algoritmasının performansının yetersiz olduğu gözlemlenmiştir.

Estimation of Initial Parameters for 3D Similarity Transformation with Evolutionary Search Algorithms

Keywords
3D Similarity Transformation,
initialization,
Optimization,
Levenberg-Marquardt,

ABSTRACT

Three-Dimensional Similarity Transformation is a widely used transformation in geomatics applications. Commonly the estimation of transformation parameters is done using methods such as Least Squares and Levenberg-Marquardt, these algorithms need good initialization. This problem has been classified as a search problem and has been tried to be solved by the researchers using techniques such as Monte Carlo search and Brute-Force search. In this article, both simulation-generated data and real data from Sudan-Adindan and WGS84 datums were used to examine five search algorithms (Differential Evolution, Weighted Differntial Evolution, Bernstain-search differential evolution, cuckoo search and Gravitational Search Algorithms) and its ability to estimate initializations for transformation parameters. Bayesian optimization was used as a tool to adjust the control parameters of each algorithm. As a result, it has been found that DE and BSD algorithms are the most suitable algorithms for calculating initial values due to their fast and stable performance. The average running time for these algorithms is 0.3225 s and 0.3512 s respectively, the WDE algorithm has stable performance but with an average run time of 1.0856 s, the Cuckoo search algorithm performed well at lower ranges for coordinates and decreased in performance at higher ranges. However, on the other hand, it has been observed that the performance of the GSA algorithm is insufficient for initialization.

[*] **Sorumlu Yazar**

[*](momaabna2019@gmail.com) ORCID ID 0000-0002-0990-452X
(uhatasever@erciyes.edu.tr) ORCID ID 0000-0002-3011-9868

Kaynak Göster;

Nasser M & Atasever Ü H (2022). 3B Benzerlik Dönüşümü için Başlangıç Parametrelerinin Evrimsel Arama Algoritmaları ile Kestirilmesi. *11. Türkiye Ulusal Fotogrametri ve Uzaktan Algılama Birliği (TUFUAB) Teknik Sempozyumu*, 10-15, 12-14 Mayıs 2022, Mersin, Türkiye.

1. INTRODUCTION

Three-Dimensional Similarity transformation is widely used in Geomatics application to convert between three dimensional frames. 3D similarity transformation is a special case of 3D coordinates transformation that handle only 3 translations, 3 rotations and 1 global scale factor (Amiri-Simkooei, 2018) .

The determination of these parameters is essential in multiple Geomatics applications such as geodetic datum transformation, close-range photogrammetry, Geographical Information System (GIS), Remote Sensing(RS) and Laser Scanning (Mikhail et al., 2001; Pandey et al., 2010; De Agostino et al., 2012).

Non-linearity of this transformation makes it very necessary to start its iterative solution by initializing its parameters.

The process of initialization is a very simple procedure in some applications due to trivial differences between original and destination geodetic datum such as 3D datum frames conversion in geodesy, in this case initialization start around zero for both translation and rotations and starts from 1 for scale factor.

However, in other applications such as 3D laser scanning and close-range photogrammetry, the rotations and translations are widely differentiating and hard to be initialized, because of that, poor initialization leads to non-convergence in iterative solution.

In this paper multiple search algorithms were tested for the purpose of parameters initialization. The performance of these algorithms was compared to find the best search algorithm performance, so corresponding search algorithm can be used for the purpose of three-dimensional similarity transformation's parameters initialization.

2. Initialization Problem Literature

Least Square Solution for non-linear systems suffer from bad initialization, however bad initialization for the parameters can lead to non-convergence.

Levenberg-Marquardt algorithm as mentioned (Levenberg, 1944), came partially to solve the sensitivity of initialization problem, in spite of that initialization problem still facing engineering solution for non-linear problem. Classification of initialization problem as search problem in Geomatics started early by Cronk et al who classified the initializing problem for relative orientation parameters as a search strategy using a Monte Carlo approach (Cronk et al., 2006).

Another work is done by Seed Ahmed who solved exterior orientation parameters based on coo-linearity model using linear solution for camera position and grid search to initialize rotation angles (Seedahmed, 2008).

Inspired by earlier mentioned strategies that classified initialization as search problem, and due to the huge improvements optimization techniques in the last decades and the absence of universal solution for initialization of 3D parameter estimation problem multiple search techniques are compared in different real time and simulation scenario to find the optimum technique which can be reliable for initialization of 3D similarity transformation.

3. METHODS and MATERIALS

In order to solve the 3D similarity non-linear system some Methods such as Least Squares can reach the solution given good initialization. Nowadays more efficient algorithms such as Levenberg-Marquardt which shows more stability but the problem with this algorithm is that poor initial guess can contribute to non-convergence of this algorithm (Zhang, 2021).

Knowing that three of similarity transformation parameters are bounded between (π and -π), and other parameters also can be bounded in the most of Geomatics application and that leads to bounded search problem so it can be solved using well known search algorithms.

In this article five search algorithms were used for the purpose of finding initialization for Levenberg-Marquardt algorithm to estimate three-dimensional similarity transformation parameters.

3D similarity Transformation is given by Equation 1

$$X_{2i} = s * R * X_{1i} + T \tag{1}$$

In Equation 1, (X_{2i}, X_{1i}) are the coordinates of i^{th} point in first and second coordinate system respectively, s is the global scale factor, T is Translation in $(x, y\ and\ z)$ and R is the rotation matrix in 3D coordinate system and given by equation (2)

$$\begin{pmatrix} c(\kappa)\,c(\phi) & -c(\phi)\,s(\kappa) & s(\phi) \\ c(\omega)\,s(\kappa) + c(\kappa)\,s(\omega)\,s(\phi) & c(\kappa)\,c(\omega) - s(\kappa)\,s(\omega)\,s(\phi) & -c(\phi)\,s(\omega) \\ s(\kappa)\,s(\omega) - c(\kappa)\,c(\omega)\,s(\phi) & c(\kappa)\,s(\omega) + c(\omega)\,s(\kappa)\,s(\phi) & c(\omega)\,c(\phi) \end{pmatrix} \tag{2}$$

In Equation 2, ω is the rotation around *x*-axis, ϕ is the rotation around y-axis, κ is the rotation around **z**-axis, c and s are cosine and sine respectively .

Objective function is defined by equation 3

$$F = \sum_{i=1}^{n} f_i^t * f_i \tag{3}$$

In Equation 3, n is number of known points and f is given by Equation 4

$$f_i = X_{2i} - [s * R * X_{1i} + T] \tag{4}$$

3.1. Parameters Estimation

Parameter estimation is very repetitive procedure in multiple Geomatics applications, in this article Levenberg–Marquardt (LM) algorithm was implemented and used for parameter estimation.

Algorithm 1 shows the implementation of LM Algorithm.

Table 1. Algorithm 1 Levenberg–Marquardt Algorithm

Algorithm 1 Levenberg–Marquardt Algorithm	
λ = 1	
Initialize V0	(Initialization)
Initialize S0 = inf	(Initialization)
while min (Si) > 10−12 do	
F(Vi), ∇F(Vi),[H]	(Calculate Objective function, divergence and Hessian Matrix)
Si = − [H + λI]−1∇F(Vi)	(Calculate corrections direction)
Vi+1 = Vi + Si	(Apply corrections)
if F (V_{i+1}) < F(Vi) then	
λ = λ/2	
else	
λ = 2λ	
end if	
Vi = V_{i+1}	
end while	
display V_{i+1}	

In the stated Algorithm $\nabla F(V_i)$ refer to the divergence of loss function, H is the Hessian Matrix, λ is a scalar and S is correction direction.

3.2. Initialization

In this article five algorithms have been compared in term of running time to be used as an initialization approaches for LM algorithm, the five algorithms were tested with variable values for its hyper parameters, the used algorithms were

- Differential Evolution Algorithm (DE)
- Weighted Differential Evolution Algorithm (WDE)
- Cuckoo Search Algorithm
- Bernstain-search differential evolution algorithm (BSD)
- Gravitational Search Algorithm (GSA)

Pairs of combination between algorithms with LM algorithm was used hyper parameters for these algorithms was optimized using Bayes Optimization technique to find the best hyper-parameters that minimize running time for each combination algorithm.

3.2.1. Differential Evolution Algorithm (DE)

Differential evolution reported for the first time in 1995 by Storn and Rainer as technical report, and performed well in the First International Contest on Evolutionary Optimization in the next year. As mentioned in review article entitled ”” by (Das & Suganthan, 2010), Two academic articles (Storn, 1995; Storn & Price, 1997) followed in fast succession, each presenting the algorithm in sufficient way.

In this article DE/rand/1/bin strategy was used this strategy can be briefly explained in four main steps:

Initialization: In this step, the pattern matrix (X) is randomly generated and the parameters of the DE are established, including the number of population (NP), the crossover probability (P), and the mutation scale (s).

Mutation: for mutation, strategy uses a differential mutation function. Three vectors are randomly selected from the old pattern matrix $(X_i, X_j and X_k)$ and a trial vector V_i is created using equation 5

$$V_i = X_i + s(X_j - X_k), i \neq j \neq k \qquad (5)$$

Crossover: In this step a new Offspring is created from the old generation or trial vector based on equation 6

$$T_i = \begin{cases} V_i, & \text{if } r \leq P \text{ or } r = I \\ X_i, & \text{if } r > P \text{ and } r \neq I \end{cases} \qquad (6)$$

Selection: In this step, a new generation is created from the old generation or offspring by using the objective function and equation 7.

$$New\ X_i = \begin{cases} T_i, & \text{if } F(T_i) < F(X_i) \\ Old\ X_i, & \text{otherwise} \end{cases} \qquad (7)$$

3.2.2. Weighted Differential Evolution Algorithm (WDE)

As mentioned in the original paper of Weighted Differential Evolution Algorithm WDE be Civicioglu et al. (2019), WDE algorithm is designed as Global optimization Algorithm which preform bounded and unbounded search unlike DE algorithm this algorithm is not sensitive to problem type and has no need for parameters tuning (Civcioglu et al., 2020).

This algorithm can be implemented in the following steps

Initialization: in this step the population is randomly generated and objective function is calculated for elements in population. This algorithm is bi-population based so 2N random element is generated.

Mutation: mutation In WDE algorithm is different

- *TempP* is generated with given equation 8

$$TempP = \sum(w \,.\, P_i) \qquad (8)$$

In Equation 8, $w = w^* \cdot \delta, w^* = k^3, w *:= \frac{W^*}{w^*}, \delta = [1]_{[N,1]}$

- *TempP* and randomly selected *SubP* of size *[N,D]* is used to generate trial matrix T using equation 9.

$$T = SubP + F \times M \times (TempP.SubP\ \ ) \qquad (9)$$

In Equation 9, F , M parameters are calculated by the algorithm.

Selection: in this algorithm selection between P and T is done based on fitness of the population elements and mutation results.

3.2.3. Cuckoo Search Algorithm

As mentioned in the original article (Rajabioun, 2011), the main idea of this algorithm was invented by following the life of the Cuckoo birds.

In this algorithm, the result development is provided in two steps.

Lévy Flights: New results are created with Lévy Flights, if these results are better than the old results, these new results will replace the old results.

Equation 10 ,shows that, the best result can be estimates using Lévy Flights.

$$X^{i+1} = X^i + a \oplus Lévy(\lambda) \tag{10}$$

Exploration: In this step, the worst result is going to be compared with new randomly selected solutions. The best solution between the worst and randomly generated is going to be selected for the next generation.

3.2.4. Bernstain-search differential evolution algorithm (BSD)

Bernstain-search differential evolution BSD algorithm is a genetic DE algorithm. In BSD, each pattern vector in the pattern matrix is developed separately. In this algorithm, the structural parameter values of BSD are randomly determined. Since the evolution of each model vector is independent of the others, BSD is inherently a parallel search algorithm. In BSD, crossover is controlled using Bernstain polynomials.

Therefore, BSD has no parameters for the crossover process. This algorithm's procedures can be abstracted in four steps

Initialization: the population matrix is randomly generated; the objective function was created and the search space was defined.

Construction of the Mutation Control Matrix (M): In this part, starting from the zero matrix and with the equation 11 this matrix is arranged.

$$M\big(i, u(1\!:\![\rho.D])\big) = 1 \tag{11}$$

In equation 11 u is expressed by equation 12, β is taken randomly.

$$u = \begin{cases} \rho = (1-\beta)^2 \; if \; k_0 = 1 \\ \rho = 2.\beta.(1-\beta) \; if \; k_0 = 2 \\ \rho = \beta^2 \; if \; k_0 = 3 \end{cases} \tag{12}$$

Evolutionary Step Size Generation (F): In this step, F is generated by the algorithm where $\kappa_1, \kappa_2 and \lambda$ are randomly generated, Q is a ones matrix.

$$F = \begin{cases} \Big(\big(\eta^3_{(1,1:D)} \circ |\lambda^3_{(1,1:D)}|\big) \times Q_{(1,1:N)}\Big), & \kappa_2 < \kappa_3 \\ \lambda^3_{(N,1)} \times Q_{(1,1:N)}, & \kappa_2 \geq \kappa_3 \end{cases} \tag{13}$$

Generation of Trial Vectors: In this step, the trial vector is created using equation 14.

$$T = P + F.M.(w^*).3.E + 1 - (w^*).3.(bestP - P) \tag{14}$$

In Equation 14, F is the step size, $E = w.PL_1 + (1-w) \cdot PL_2$ and w, w^* randomly generated.

Update: If the objective function of the trial solution is better the trial solution in this step will replace the old solution.

3.2.5. Gravitational Search Algorithm (GSA)

As explained in the original paper (Rashedi et al., 2009). Gravitational Search Algorithm is an algorithm that based on law of gravity. This algorithm inspired by how bodies and masses interact in the space and that is clear when using this algorithm to solve 2-dimensional space problem the interaction of solutions looks like the movement of the masses in the space.

The gravitational force between two bodies in the universe is directly proportional to the product of their masses and inversely proportional to the square of the distance between them, equation 15 shows the relationship:

$$F = \frac{M_1 \times M_2}{R^2} \tag{15}$$

In this algorithm optimization of the solution is given using the following steps

Initialization: this algorithm starts with random initialization.

Update: update in this algorithm is done in three steps.

- Fitness evaluation.
- Update for masses, best and worst solution.
- Calculation of the total force, acceleration and velocity in different directions.
- Updating solution position.

Update step is repeatedly done until reach the criteria.

3.3. Implementation

MATLAB environment is used for implementation. cuckoo search and WDE algorithm implementation are downloaded from MATLAB file sharing center and edited for our methodology purposes, DE, GSA and BSD are written manually based on original papers.

The Bayes Optimization is implemented using MATLAB software built-in function.

3.3.1. Bayesian Optimization

Bayesian Optimization (BO) is an approach based on Bayes theory to search the optimum solution in noisy or the hard to be evaluated functions such as neural networks (Mockus, 2012).

in this article BO is used for the purpose of finding the best hyper parameters for each algorithm which yield the best performance.

To test performance, the running time was used as an indicator to indicate algorithm performance.

The best performance for each algorithm is tested under multiple coordinate ranges. The coordinate ranges selected exponentially to test algorithms under varied value for cost function to generalize algorithm performance test in multiple scenarios.

Best five performances for each algorithm optimized by Bayesian Optimization was used to calculate average running time as an indicator for performance. Table 2 showing algorithms and search space for its hyper parameters.

Table 2. Show search space for each algorithm's parameters.

Algorithm	Parameter	Search space	
		Start	End
DE	Population	10	200
	Mutation Scale	0.01	1
	Crossover Probability	0.01	1
WDE	Population	10	200
GSA	Population	10	200
	Bita	0.01	1
	Small constant	10^-6	1
BSD	Population	10	200
Cuckoo	Population	10	200
	Discovery rate	0.01	1

3.4. Experimental Study

3.4.1. Simulation Data Experiments

To generalize experiment, Simulation data has been generated randomly based on the following steps:

- Three points are generated randomly as original coordinates system points (O) in specific scales $[1,10^1.,10^2,\dots,10^7]$.
- Known transformation parameters $X = [90°,90°,90°,100m,100m,100m,1.2]$ is applied on these points to generate temporary target coordinates(T_t) .
- Random Noise Applied on the target temporary coordinates (T_t) to simulate real measurements $T = T_t + s.rand(N,3)$.

Original and Target coordinates($O\ and\ T$) generated used to estimate known transformation parameters. For each used search algorithm 100 function evaluation using Bayes optimization was applied to optimize algorithm hyper parameters.

The average of five top performances is calculated as measurable for performance for each algorithm.

3.4.2. Real Data Experiment

Eight points in Geocentric coordinate system is used to calculate transformation parameters between Adindan-Sudan and WGS84 datums from paper by (Mohammed & Mohammed, 2013) .

Running time was measured for each algorithm, this process was repeated 10 times and the average running time calculated for each algorithm.

4. RESULTS

Bayes optimization was implemented, 100 function evaluation for each algorithm, in 8 different ranges for three dimensional coordinates to estimate the best hyper parameters that yield the best performance for each algorithm in each range. The mean of top five performances for each algorithm is shown in table 3.

Bayesian Optimization is implemented and it is found that, the performance of DE algorithm was the best.

Table 3. Running Time Per Seconds for each algorithm in multiple ranges for data.

Range	DE	BSD	WDE	Cuckoo	GSA
0.1	0.2987	0.4017	0.5309	0.1504	>10
10	0.863	0.3952	1.6497	0.2305	>10
10^2	0.5948	0.2322	0.6414	0.1633	>10
10^3	0.155	0.1793	1.7112	0.2193	>10
10^4	0.1342	0.1849	0.51	0.1318	>10
10^5	0.1142	0.3661	0.4042	0.2035	>10
10^6	0.2589	0.4027	0.7526	>10	>10
10^7	0.161	0.6477	2.4848	>10	>10
Average	0.3225	0.3512	1.0856	0.1831	-

Second Experiment was implemented, Table 4 shows the average running time and running time standard deviation for each algorithm.

Table 4. Average and standard deviation running time for DE, BSD, WDE, Cuckoo and GSA algorithms.

	DE	WDE	BSD	Cuckoo
Mean (s)	0.7949	1.7114	0.5231	2.539
Std (s)	0.3087	0.6026	0.2057	2.269

Figure 1 shows best algorithms performance through iterations

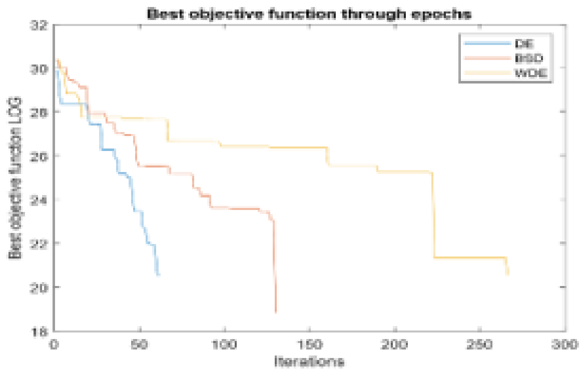

Figure 1. Algorithms performance through iterations.

5. DISCUSSION

From results it is very clear that the performance of DE algorithm was the best comparing with other algorithms. However, Cuckoo search showed better performance in multiple situation but it still has weak performance in wider coordinates system.

It is very important to refer to stability of BSD algorithm in different situations.

From implemented experiments it is very clear that DE algorithm is performing fast while population is small, however small population causing instability of DE algorithm. However, BSD as well as WDE shows a stability in its performance even if population is very small.

Cuckoo search algorithm shows rapidly improvement in its loss function during iterations, loss stay constant for multiple iterations and suddenly improves rapidly.

6. CONCLUSIONS

3D similarity transformation is a transformation widely used in Geomatics application. Estimation of 3D Similarity transformation parameters is a non-linear problem. Solving the non-linear systems usually passing the initialization problem. From the literature it is known that initialization problem can be solved as search problem. In this article five optimization search techniques were compared and it is concluded that DE, BSD and WDE algorithms are stable and recommended for initialization of 3D similarity transformation's nonlinear solution. For the best performance DE algorithm is proposed to be used. However, using GSA algorithm is not sufficient due to low performance. Cuckoo search algorithm is performing well in narrow spaces but that performance cannot be generalized to wider spaces.

Author contributions

The authors contributed equally to the study.

Conflicts of interest

There is no conflict of interest between the authors.

Statement of Research and Publication Ethics

Research and publication ethics were complied with in the study.

REFERENCES

Amiri-Simkooei A (2018). Parameter estimation in 3D affine and similarity transformation:implementation of variance component estimation. *Journal of Geodesy, 92*, 1285--1297.

Civicioglu P & Besdok E (2019). Bernstain-search differential evolution algorithm for numerical function optimization. *Expert Systems with Applications, 138*, 112831.

Civicioglu P, Besdok E, Gunen M A & Atasever U H (2020). Weighted differential evolution algorithm for numerical function optimization: a comparative study with cuckoo search, artificial bee colony, adaptive differential evolution, and backtracking search optimization algorithms. *Neural Computing and Applications, 32*(8), 3923-3937.

Cronk S, Fraser C & Hanley H (2006). Automated metric calibration of colour digital cameras. *The photogrammetric record, 21*(116), 355-372.

Das S & Suganthan P N (2010). Differential evolution: A survey of the state-of-the-art. *IEEE transactions on evolutionary computation, 15*(1), 4-31.

De Agostino M, Lıngua A & Pıras M (2012). SOLDEO: a new solution for 3D GIS data recording. *FIG Working Week.*

Levenberg K (1944). A method for the solution of certain non-linear problems in least squares. *Quarterly of applied mathematics, 2*(2), 164-168.

Mikhail E M, Bethel J S & McGlone J C (2001). Introduction to modern photogrammetry. *New York, 19*.

Mockus J (2012). *Bayesian approach to global optimization: theory and applications* (Vol. 37). Springer Science \& Business Media.

Mohammed A & Mohammed N Z (2013). WGS84 to Adindan-Sudan Datum Transformation Manipulated by ITRF96. *International Journal of Multidisciplinary Sciences and Engineering*, 60-64.

Pandey G, McBride J, Savarese S & Eustice R (2010). Extrinsic calibration of a 3d laser scanner and an omnidirectional camera. *IFAC Proceedings Volumes, 43*(16), 336-341.

Rajabioun R (2011). Cuckoo optimization algorithm. *Applied soft computing, 11*(8), 5508-5518.

Rashedi E, Nezamabadi-Pour H & Saryazdi S (2009). GSA: a gravitational search algorithm. *Information sciences, 179*(13), 2232-2248.

Seedahmed G H (2008). Autonomous initialisation of exterior orientation parameters using a collinearity search-based solution. *The Photogrammetric Record, 23*(121), 90-108.

Storn R & Price K (1997). Differential evolution a simple evolution strategy for fast optimization. *Dr. Dobb's Journal, 22*(4), 18-24.

Zhang, G. a. (2021). Taking the Guess Work Out of the Initial Guess: A Solution Interval Method for Least-Squares Parameter Estimation in Nonlinear Models. *Journal of Computing and Information Science in Engineering*.