Penalty Strategy in The Fitness Function of Grey Wolf Optimizer for Minimum Spanning Tree Problem

Z Zukhri¹, I V Paputungan², N S Handayani³ and V Satriadi⁴

¹,²Department of Informatics, Universitas Islam Indonesia
³Master Program in Informatics, Universitas Islam Indonesia
⁴Undergraduate Program in Informatics, Universitas Islam Indonesia

E-mail: ¹zainudin@uui.ac.id, ²irving@uui.ac.id, ³nadya.satya007@gmail.com, ⁴vebri.satriadi@uui.ac.id

Abstract. The Grey Wolf Optimizer (GWO) is a relatively new population-based optimizer. Various optimization problems have been solved using GWO. This paper presents an experiment on how to apply GWO to solve the minimum spanning tree (MST) problem. MST is normally solved using revision strategy when formulating the fitness in other population-based algorithms. Another strategy, called Penalty strategy, is used in the experiment. Existing dataset for MST problem is tested using GWO. The experiment showed that implementation of penalty strategy in the fitness function of GWO can find a solution with almost 96% accuracy.

1. Introduction

Many real-world problems in the public services area such as minimizing cables of the power electric network, minimizing optical fibre of the communication networks, and minimizing pipes of the water supply networks, can be modelled as minimum spanning tree (MST) problems. In MST, the solution is modelled as a tree that consists of feasible connected edges and vertices where the total length of edges is minimum [1]. The feasible edge is the connector between two edges that are available in the graph. Meanwhile, an infeasible edge is a virtual connector between two edges and not available in the graph. MST is not only able to minimize the use of material to build the network, but also reduce the material waste. Minimum material will consequently save the budget for material procurement. For example, when developing an electrical network, longer cable will definitely need more money. However, at the same time, it makes more power loss [1]. In 2015, there was a power loss of almost 300 MVA within the Java-Bali power transmission line captured by Indonesian State Electricity Company [2]. The alternative solution to reduce this situation is by minimizing the energy transmission line. It can be done by minimizing the length of installation materials.

This paper describes how GWO finds the solution for MST problem. Similar to other population-based optimizer, the solution is normally generated in two different types of process in GWO. First is a random process which usually is done at the initial step, and a process that involves certain operators which usually is performed after initial step. However, both may produce infeasible solution. To handle such issue, two strategies, namely the revision and penalty strategy, can be applied [3]. Revision strategy means the optimizer is designed to avoid infeasible solutions. It works by repeating the process to generate feasible solution or transforming the infeasible solution into feasible one. Whereas in penalty strategy, infeasible solution is managed and carefully handled. When optimizer produces infeasible solutions during the process, the fitness value of these solutions will be reduced in order to lower the probabilities to be selected as the final solution.
GWO is a bio-inspired algorithm that adopts how grey wolf pack hunting process [5]. How the herd finds its prey is modelled as computation to find a solution. It starts from the gathering of wolves at a certain point and will eventually move closer to the target using certain rules until the prey animal can be finally caught. Since the computational process is aimed to find a solution, the positions of the wolf pack must be decoded for each iteration as a set of solution candidates. The quality of each solution in this set is represented by a fitness function. By comparing the fitness value from all solutions, the best one is then produced.

Prior to decoding the position of the wolf pack, it should be noted that candidate solutions of MST are not always obtained merely based on the generated position. It is because a position either randomly generated or another movement of wolf pack. In other words, wolf pack position is occasionally not representing a tree or it is a tree but contains infeasible edge. Penalty strategy is proposed to handle this condition. This strategy is much simpler than revision strategy to find MST solution. The revision strategy, though, will normally add significant computational steps. The proposed fitness function is formulated by reducing the value when the decoding result is not a tree. Since minimization is the ultimate goal of MST, if such goal is used as the objective function, we then need to reverse this assumption in penalty strategy, i.e., penalty value must be escalated. When the decoding result is not a tree, the fitness value must be added by a certain term or it must be multiplied by a certain factor. How the term formulation must be added by the fitness value or the factor formulation must be multiplied by the fitness value will be presented in this paper.

This paper is presented in the following structure. Literature review on related works and infeasible solution in using GWO to solve MST problem is presented in section 2. Section 3 presents the methods of the research. The results and discussions are presented in section 4. Lastly, the conclusions and subsequent work of our research are presented.

2. Literature Review
This section presents several works on MST problem and also discusses what kind of infeasible MST solution that will be handled.

2.1. Related Works
MST problem is an interesting classical model that is able to solve real world problems while evolving the model. Currently, it is not only about the power network, the optical fibre network, or the water supply network, but also the design of logistics networks in inter Island Ocean [6] or even MST has been implemented as the standard protocol for wireless Internet access points [7], etc.

Several studies have been performed to solve MST problem. The revision strategy is applied in the placement of phasor measurement units [8]. Genetic Algorithm (GA) was used to solve the MST. The researcher called the chromosome that generates an infeasible solution as an unobservable chromosome. When an unobservable chromosome is generated, they repeat the chromosome generation procedure until a feasible solution is obtained. The revision strategy is also applied in solving MST using evolutionary algorithm [9][10]. Such work shows that only the feasible solutions are inspected. The evolutionary algorithm can only produce less than 70% feasible solutions or almost a third of the total chromosomes would be ignored. Zuhri used a simplified Genetic Algorithm to solve the MST problem [10], however the study implemented a revision strategy. All in all, it is difficult to find studies that proposed penalty strategy to solve MST.

2.2. Infeasible solution in using GWO to solve MST problem
As mentioned in the introduction, applying GWO to solve MST problem might produce infeasible solution. It can be caused by either the position of the wolf pack which does not represent the tree or the presence of an infeasible edge in the tree. Pseudocode of GWO in Figure 1 shows how the infeasible solution is generated from the position of a wolf. Such pseudocode is adopted from how the grey wolf pack hunts, see Figure 2 [5].

At the initial point during hunting, the wolves from the gathering point will form a formation to randomly chase (Figure 2.A) and surround the target animal (Figure 2.B–2.E). Likewise, when moving to intimidate prey, wolves will make certain adjustments to each other in response to the movement of
the prey. Changing the wolf pack’s position will transform the MST solution. Since the updated wolf pack positions can be anywhere in the area around the prey and some conditions must be met to become an MST solution, it is possible that the transformed MST solution will not meet the requirements. It is called an infeasible solution.

![Figure 1. The pseudocode of GWO [5]](image)

In Figure 1, statements that can generate an infeasible solution are considered as the initialization statement on the first line. The agent position then updates statement on a nested loop (the starred lines). Both starred statements are potential to generate random wolf positions. Although the number of edges is five, such edges will not surely form the MST solution since it is not a tree or a tree that contains an infeasible edge.

3. Methods

This section shows the customization part of GWO to solve MST problem with two alternative fitness functions containing 1) modelling MST problem, 2) decoding the position of the wolf pack, 3) fitness function, 4) dataset, 5) measurement of the solution quality, and 6) hypothesis testing.
3.1. Modelling MST problem to be solved by GWO

MST problem is shown in a graph that consists of m edges and n vertices where (n-1) edges must be selected from m to form a tree connection of n vertices. Virtual edges that are not available in the graph are ignored in the proposed method. The selected edges in this method are always available in the graph. Hence, an infeasible solution is only generated by circular edges.

3.2. Decoding the position of the wolf pack as a solution candidate

According to the proposed model, each wolf position in GWO must be decoded into a solution candidate, both feasible and infeasible solution. Wolf position is limited to the interval 0 to 100 in the coordinate system. This value will be used to select one of the available edges in the sorted edges, from 1 to m. This selection will be made (n-1) times expecting that n vertices in the graph will be connected. Consequently, the edges that have already been selected will not be on the set of sorted edges for the next selection. Thus, for each iteration, the number of edges in the set of sorted edges will be decreased by one. Illustration of the decoding wolf position to the solution candidate is shown in Figure 3.

The sequence of available edges in the graph (E)  

\[ e_1, e_2, e_3, \ldots, e_{m-2}, e_{m-1}, e_m \]

Wolf Coordinate  

\[ x_1, x_2, x_3, \ldots, x_{n-1} \]

Selected edges (E')  

\[ e'_1, e'_2, e'_3, \ldots, e'_{n-1} \]

where  
\( e_i \) is the ith edge in E, \( 1 < i < m \)  
\( x_j \) is the jth coordinate of the pack  
\( e'_j \) is the jth edge in the solution candidate, \( 1 < j < n - 1 \)  
At jth iteration of the edge selection, the number of available edge will decrease as mj

**Figure 3.** Illustration of selecting (n-1) edges in the solution candidate of m edges in the graph of the proposed model

The index of the selected edge is selected in the range 1 to m, hence the index of selected edge, j, can be obtained based on equation (1) and (2).

\[ e'_j = e_{\text{index}} \]  

\[ \text{index} = \text{floor} \left( \frac{x_j}{100} \right) + 1 \]  

In order to make the index less than or equal to mj, the maximum value of xj must be 100. Therefore, if the index is greater than mj since xj equal to 100, the index must be equal to mj as shown in equation (3).

\[ \text{index} = mj, \text{for } x_j = 100 \]

3.3. Fitness function

How to formulate the fitness function is very important in this work. The objective function is set as fitness function. Since the proposed method is penalty strategy, there is a part of this function that penalizes the infeasible solution. Thus, the proposed fitness function will mainly consist of the objective and the penalty function part. As MST is minimizing the total distance, penalty function must be added to the fitness function value. Additional or multiplication form which is equivalent to the infeasibility level of the solution candidates is proposed. A higher level makes a bigger penalty value and vice versa. The lowest level makes the penalty value = 0. It is easy to measure the infeasibility level of the solution candidates. It is indicated by the circular connector of several selected edges. The circular connector will separate one or more vertices. Therefore, the penalty value must be proportional to the number of unconnected vertices.

To formulate the penalty value that is proportional to the infeasibility of the solution, we do not make an experiment with penalty functions mentioned by Zukhri for generic constrained optimization
problems [4]. We propose the penalty function as simple as possible. The first formula is equal to the multiplication of unconnected vertices number \((n_c)\) and the maximum weight of the edge in the graph \((w_i)\), where \(i\) in range between 1 and number of edge \((n)\) and \(w_i\) is the weight of \(e_i\) in Figure 3. The second one is equal to the multiplication of unconnected vertices number and the square of the maximum weight of the edge. We also do not use the MST objective function that is a function defined on the matrix that is usually applied in operational research. We prefer to define the MST objective function based on the MST definition as mention in the introduction of this paper [1] since it is a function defined on the array as equation (4) that more simple, less complicated. Finally, we get two alternatives of fitness function we proposed as the sum of the objective function & penalty function as equation (5) and equation (6). We needed to simulate both alternatives in the experiments to get one of them whose better performance.

\[
\begin{align*}
    f_{obj} & = \min \sum_{j=1}^{n} w_j \\
    f_{fitness_1} & = f_{obj} + n_c \cdot \max(w_i) \\
    f_{fitness_2} & = f_{obj} + n_c \cdot (\max(w_i))^2
\end{align*}
\]

where \(i\) is the index of edge in the graph and \(j\) is the index of edge in the solution.

3.4. Dataset
The experiments to test the proposed method are dataset from previous research as shown in Figure 4 [11], Figure 5 [11], and Figure 6 [12]. In order to make the graph model read easily, the model is not-to-scale. The minimum solutions for the three datasets are 16, 13, and 202 [11, 12], respectively.

3.5. Measurement of the solution quality
We have discussed that a fitness function is the measurement of the solution quality in the computation. But it is only used to compare the solution with each other & it cannot be used to measure how good the solution is compared by the known solution. For this matter, we use the concept of accuracy to measure the quality of the solution. The concept should be applied for all dataset and support to the determined hypotheses testing. The best solution is equal to the known solution from the previous researches. We will call it as the minimum solution. Oppositely, there is the worst solution. It is the maximum value of feasible solution found in the computation. This value is called the maximum solution. Hence, the accuracy is defined as the proportion of the difference between the maximum solution and the solution found by the optimizer compared to the difference between the maximum solution and the minimum solution as shown in equation (7).

\[
\text{accuracy} = \frac{\text{max solution} - \text{solution}}{\text{max solution} - \text{min solution}} \cdot 100\% \tag{7}
\]
3.6. Hypothesis testing
The hypothesis testing in this study is carried out based on the statistical testing data, i.e. the solution quality found by GWO with fitness₁ is assumed to equal the quality found by GWO with fitness₂ as in equation (8) and equation (9).

\[ H_0: \mu_1 = \mu_2 \] (8)
\[ H_1: \mu_1 \neq \mu_2 \] (9)

We use the two-sample t-test. The null hypothesis \( H_0 \) fails to be rejected if the test value calculated based on statistical data is at the critical t interval on both sides and vice versa if the statistical value is outside the critical t interval on both sides, the alternative hypothesis \( H_1 \) will be accepted.

The experiment is performed using an open source program at https://github.com/7ossam81/EvoloPy with inserting the proposed MST fitness functions into benchmarks.py file and adding dataset as input text.

4. Results and Discussions
Different fitness function does not make a significant difference to the GWO algorithm. The penalty function shows that the second fitness function will require a longer computation time, but it is not that significant. Therefore, we can skip the computational time. To shorten the experiment time, we ran the simulation software on 2 different machines. Each data will be subjected to 349 trials for performance testing. The number of trials is based on the Isaac and Michael Tables [13] and the optimizer parameter adjusts to the number of vertices in the graph. The parameters of each testing, the minimum known solution, and the maximum solution of each data were found in the test as shown in Table 1. In testing, the optimizer with both penalty functions can find the minimum solution. The results of two-sample t-test at the level of significance = 1% are shown in Table 2.

Table 1. Dataset and parameter testing

|          | Data 1 | Data 2 | Data 3 |
|----------|--------|--------|--------|
| Population Size | 6      | 10     | 45     |
| Iteration  | 50     | 200    | 3000   |
| Max Solution| 37     | 37     | 287    |
| Min Solution| 16     | 13     | 202    |

The experiments showed that GWO with two alternatives penalty functions can find solutions to the MST problem in all datasets. Based on the accuracy means in the t-sample t-test table, GWO with fitness₁ showed slightly better results than another. But the results of the two-sample t-test indicated that
the difference in mean value is not significant enough. It is indicated by the t-critical value in the interval of ± t critical two-tail value interval or $H_0$ is accepted. However, we have to consider the computation time that we don't need to test. It is clear that the quadratic form in the penalty function will slightly increase the time complexity of the algorithm. Of course, the multiplication operator will make the fitness function simpler than the squares operator. Finally, it can be concluded that penalty strategy in GWO for solving the MST problem can be formulated by increasing the value of objective function slightly more than the possible maximum solution. We do not much more value than the maximum solution.

| Table 2. Two-sample t-test results. |
|-----------------------------------|
| Accuracy | Data 1 | Data 2 | Data 3 |
|----------|--------|--------|--------|
|          | Fitness 1 | Fitness 2 | Fitness 1 | Fitness 2 | Fitness 1 | Fitness 2 |
| Mean     | 98.376  | 98.717  | 96.538  | 97.576  | 96.572  | 96.760   |
| Variance | 18.142  | 13.598  | 38.565  | 29.380  | 7.379   | 8.658    |
| Observations | 349    | 349    | 349    | 349    | 349    | 349      |
| Pooled Variance | 15.870  | 33.973  | 8.019    |
| Hypothesized Mean Difference | 0     | 0     | 0     |
| df       | 696    | 696    | 696    |
| t Stat   | -1.131 | -2.354 | -0.881 |
| P(T<=t) one-tail | 0.129  | 0.009  | 0.189  |
| t Critical one-tail | 2.332  | 2.332  | 2.332  |
| P(T<=t) two-tail | 0.258  | 0.019  | 0.379  |
| t Critical two-tail | 2.583  | 2.583  | 2.583  |

Meanwhile, the value of p on the second test shows that the accuracy value of the experiment is not homogeneous ($p \leq 0.05$). Although the accuracy value in the second test is homogeneous ($p > 0.05$) we can say that GWO produces a non-homogeneous solution. It is possible since we cannot give specific parameter values to simulate wolf pack behaviour. We can only set the common parameter of the algorithm, i.e., population size and number of iterations. It is possible that a more homogeneous solution value would be generated if we could set the specific algorithm parameter. GWO needs to be developed by observing other behaviour of wolf pack to insert a certain specific parameter for this algorithm in order to explore more searching space.

5. Conclusions
The penalty strategy can be applied at GWO to solve MST problems with almost 96% accuracy regardless of the computation time. We do not need to penalize all the infeasible solutions to this problem. The coding of the problem solutions in GWO for solving this problem as agent positions can be designed to prevent the formation of unavailable edges in the graph. Penalties are only applied to solution candidates generated by circular edges. It will reduce the number of infeasible solutions. Based on the tested penalty functions, we only need a penalty formula that gives slightly greater than the maximum possible solution. It is simpler than penalizing the infeasible solution with much more value than the maximum solution. In the future, GWO can be developed by observing wolf pack behaviour in more depth so that there are certain parameters besides population size and number of iterations. That way, the behaviour of the wolf pack can be changed and provide the possibility for exploration of search space.

Acknowledgments
This work is part of the project sponsored by the Department of Informatics, Universitas Islam Indonesia, Yogyakarta, Indonesia, 2020.
References

[1] Sedgewick R and Wayne K 2018 Minimum Spanning Tree Retrieved on October 2020 from https://algs4.cs.princeton.edu/43mst/

[2] Pamungkas M A P, Priharto D and Putranto H 2019 Study of technical and non-technical factors in energy consumption on 20 kV distribution networks Frontier Energy System and Power Eng. vol 1 no 2 (Malang: Electrical Engineering, Universitas Negeri Malang) pp 1–6

[3] Sulistiawati I B and Ashari M I 2016 Artificial bee colony algorithm for optimal power flow on transient stability of Java-Bali 500 KV Proc. of Second Int. Conf. on Electrical Systems, Technol. and Information 2015 (Singapore: Springer Science+Business Media) p 247

[4] Zukhri Z 2014 Algoritma Genetika, Metode Komputasi Evolusioner untuk Menyelesaikan Masalah Optimasi (Yogyakarta: Penerbit Andi)

[5] Mirjalili S, Mirjalili S M and Lewis A 2014 Grey wolf optimizer Advances in Eng. Software vol 69 (Amsterdam: Elsevier) pp 46–61

[6] Kismanti T S 2016 Penentuan Jaringan Logistik pada Transportasi Laut Menggunakan Fuzzy C-Means dan Minimum Spanning Tree Berbasis Hybrid Genetic Algorithm (Surabaya: Institut Teknologi Sepuluh Nopember Surabaya)

[7] Cisco Systems Inc 2017 Cisco IOS Configuration Guide for Autonomous Cisco Aironet Access Points - Releases 15.3(3) JE and Later (Cisco Systems Inc)

[8] Devi M M and Geethanjali M 2020 Hybrid of genetic algorithm and minimum spanning tree method for optimal PMU placements Measurement vol 154 (Amsterdam: Elsevier) pp 1–12

[9] Bolton C C, Rey C, Cossio S R, Rodriguez C, Gatica F and Parada V 2016 Automatically produced algorithms for the generalized minimum spanning tree problem Scientific Programming vol 2016 (London: Hindawi) pp 1–11

[10] Zukhri Z 2017 Penyelesaian masalah pohon rentang minimum dengan algoritme genetika yang disederhanakan suatu adopsi operator dari sistem kekebalan buatan Seminar Nasional Inovasi dan Aplikasi Teknologi di Industri (Malang: Institut Teknologi Nasional) p A.39.1

[11] Hidayat T 2000 Algoritma genetik untuk pemecahan persoalan minimum spanning tree Seminar Nasional Aplikasi Sistem Cerdas dalam Rekayasa dan Bisnis (Yogyakarta: Universitas Islam Indonesia) p 53

[12] Setiawan S and Wibowo A M 1999 Struktur Data Graf Retrieved on October 2020 from http://aren.cs.ui.ac.id/sda/archive/1998/handout/handout20.html

[13] Sugiyono 2018 Statistik Nonparametris untuk Penelitian 2nd edition (Bandung: Alfabeta)