

Firefly Algorithm for Adaptive Emergency Evacuation Center Management

Yuhanis Yusof, Nor Laily Hashim, Noraziah ChePa, Azham Hussain

School of Computing
Universiti Utara Malaysia
Sintok, MALAYSIA

Abstract—Flood disaster is among the most devastating natural disasters in the world, claiming more lives and causing property damage. The pattern of floods across all continents has been changing, becoming more frequent, intense and unpredictable for local communities. Due to unforeseen scenarios, some evacuation centers that host the flood victims may also be drowned. Hence, prime decision making is required to relocate the victims and resources to a safer center. This study proposes a Firefly Algorithm (FA) to be employed in an emergency evacuation center management. Experimental analysis of a minimization problem was performed to compare the solutions produced by FA and the ones generated using Tabu Search. Results show that the proposed FA produced solutions with smaller utility value, hence indicating that it is better than the benchmark method.

Keywords—Firefly Algorithm; Swarm Intelligence; Flood Management; Evacuation Center Management

I. INTRODUCTION

Disaster management is extremely important in today's world and it focuses on the organization and management of resources and responsibilities for dealing with all humanitarian aid. Despite all activities accomplished by governments in the disaster preparation stage, flood occurs and affected people's daily routines and the economic flow since offices, businesses and schools are closed. In the past decade, Malaysia has experienced a number of major floods. Floods are caused by a combination of natural and human factors. Malaysians are historically riverside people as early settlements grew on the banks of the major rivers in the peninsula. Coupled with natural factors such as heavy monsoon rainfall, intense convection rain storms, poor drainage and other local factors, floods have become a common feature in the lives of a significant number of Malaysians. The vast increasing numbers of the lost due to flood enforces the government to take proactive steps such as setting up supervisory bodies, implementing flood mitigation programmes, implementing non-structural steps with the setting up of flood forecasting and warning systems for the flood prone area. The evacuation and relocation of flood victims involves a lot of capital. As informed by the Minister in the Prime Minister Department in March 2011, almost USD 21.12 million was spend in for 89,000 flood victims in five states effected by the disaster and it was estimated that 53 percent of that amount was spent on relocation of the victims,

which also includes food and other daily necessities. At the moment, there are 5,143 evacuation centres (EC) which could accommodate 1.4 million flood victims around the country.

Flood evacuation centres in Malaysia are managed by the Department of Social Welfare (*Jabatan Kebajikan Masyarakat*) and works closely with a number of governmental and non-governmental agencies to provide necessary steps to ensure safety and comfort in every evacuation centre. These individuals are the backbone of the flood evacuation centre and often face difficulty in decision making such as for resource allocation. Various work can be seen in resource allocation pertaining to disaster management such as in flood. In the work by Zhu, Huang, Liu and Han [1], the researchers propose a resource allocation model that is aimed at determining the location of reserve depots and the amount and type of resources to be stored. It is modelled based on discrete scenarios that is divided into two; local government and national. Their optimization focuses on the commodities inventory holding and transportation cost. On the other hand, a more recent work [2] was discussed in that identifies the optimal number, location and inventory level of warehouses around the world in the occurrence of a disaster. The model considers uncertainties on product quality, availability and production capacity in affected areas.

This study proposes the employment of a swarm intelligence algorithm (i.e Firefly Algorithm) in the Adaptive Emergency Evacuation Center Management (AEECM) that monitors and manage evacuation centers. Similar to existing work on disaster management, the AEECM focuses on resource allocation. However, the study reported in this article is limited to the management of victims located in ECs. The proposed AEECM adapts a recent computing approach known as Swarm Intelligence. Swarm Intelligence is defined as an emergent collective intelligence of groups of simple agents [3]. It is used to find optimal solutions in hard problems, such as Travelling Salesman (TSP) [4], scheduling [5] and nurse rostering [6]. Examples of algorithms that are considered as Swarm Intelligence are the Ant Colony Optimization, Artificial Bee Colony, Fish School, Bat Algorithm and Firefly Algorithm (FA). In this study, a variant of Firefly Algorithm that provides optimal solution to the management of victims in an evacuation center is presented. In particular, the proposed FA determines to where (i.e which available centers) victims in an evacuation center should be re-located and how many of them should be moved to each identified center.

II. RELATED WORK

Meta-heuristic algorithms are defined as optimization algorithms that determines the best solution (optimal or near optimal solution) from a set of available solutions [7]. The identification of best solution is achieved by evaluating a predefined objective function that can be addressed as either minimum or maximum function. The design of an objective function is based on the problem in-hand, that is if the goal of the problem is to obtain minimum cost then a minimum objective function is designed [8] and vice versa.

Existing meta-heuristic algorithms can be categorised into two types as shown in Figure 1; single meta-heuristic and population meta-heuristic [9]. For the first category, (i.e single meta-heuristic), the algorithm generates a single solution and iteratively enhance it. An example of such algorithm is the Tabu Search (TS) which was introduced by Glover in 1986 [10]. On the other hand, the population meta-heuristic algorithms generate a set of solutions and select one of it as the best solution. The swarm algorithms such as Firefly Algorithm is an example of meta heuristic algorithm [3, 11].

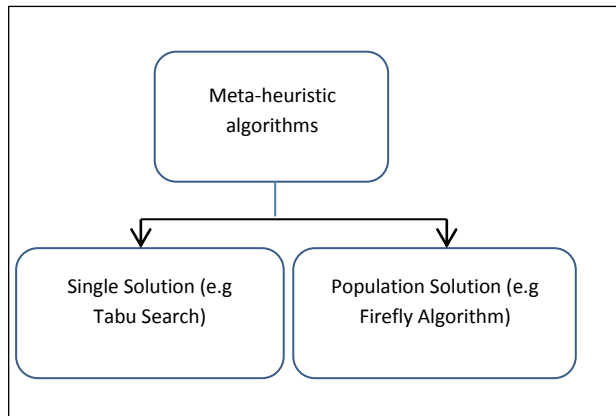


Fig. 1. Categories of Meta-heuristics Algorithm

A. Tabu Search

Tabu Search (TS) generates a single solution by extending the search space beyond local optimality to identify better solution [12, 13]. TS classify a subset of the moves in a neighbourhood as tabu. A neighbourhood constructs to reach adjacent solution from a current solution. The main idea in TS is to avoid recently visited solution space areas and move towards promising area [14]. TS has been adapted into various optimization problems (Glover & Laguna, 2013) such as colour texture histogram [13], scheduling [5], test data software generation [12], cell formation [14], nurse rostering [6], graph colour [15, 16], assignment [16] and max-cut problem [17].

In TS, there are two important factors; tabu moves and tabu condition. The first factor is determined by a function that utilizes information from the search process, while, the second factor is a linear inequality or logical relationships that is used to choose the tabu moves [18]. Figure 2 details the pseudo code of the Tabu Search algorithm [18]. In Step 1, a random solution is selected and assigned as the best solution. A new subset of solutions will be generated based on the

identified best solution. Comparison between solutions is performed in order to identify the best one.

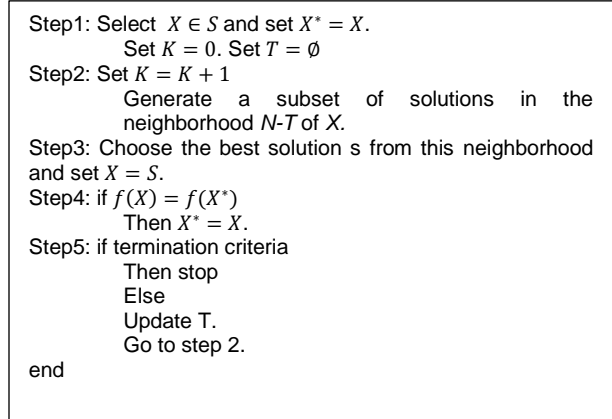


Fig. 2. Pseudo Code of Tabu Search [18]

B. Firefly Algorithm

Firefly algorithm (FA) is an example of algorithm that is based on nature inspired computing. It has the ability to identify global optimal solution [19]. The main concept of Firefly algorithm is realized in two factors; light intensity and attractiveness between fireflies. The light intensity of a firefly is more related with the objective function, $f(x)$, and can be a maximization or minimization function. On the other hand, the attractiveness, β , between fireflies is associated with the distance between two fireflies, where β is based on the change of distance. Figure 3 shows the steps in Firefly Algorithm [11, 20].

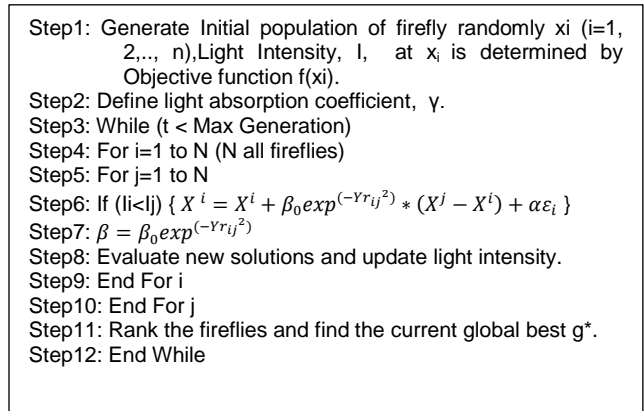


Fig. 3. Pseudo Code of Firefly Algorithm [11, 20]

Firefly Algorithm has been implemented in many optimization problems such as image segmentation [21], traffic forecasting[22], discrete optimization [23], data classification [24], data clustering [25], text clustering [26] and economic dispatch problems [27]. In all of these domains, Firefly Algorithm has proven to be successful in solving the problems and identifying the optimal solution.

III. METHODS

The proposed work is realized by performing 3 phases; data collection, design of algorithms and evaluation.

A. Data Collection

The obtained data on evacuation centers is represented as four independent variables as depicted in Table 1. It covers information on the 106 ECs in one of the district in Malaysia.

TABLE I. INDEPENDENT VARIABLES

Data	Parameter
Size of EC	V1
Distance of EC to the closed EC	V2
Water level of nearby river	V3
Distance of EC to nearby river	V4

B. Design of Algorithms

In this section, elaboration on the adaptation of Tabu Search and Firefly Algorithm in determining number of victims to be evacuated is presented. In the Adaptive Emergency Evacuation Center Management, the two algorithms are triggered when a decision on closing a particular EC is made. Assuming that the chosen EC has *n* number of victims, the proposed Tabu Search and Firefly Algorithm provides suggestion on the location (i.e EC) to where the victims should be relocated. Furthermore, the suggestion also includes information on the number of relevant victims. In the AEECM, the proposed Tabu Search is termed as TS_{Flood} while the variant of FA is known as FA_{Flood}. Both of these algorithms employ an objective function as depicted in Eq. 1

$$\begin{aligned}
 \text{Utility function } F = & \text{Summation of (75\% of V1) of available} \\
 & \text{EC} + \text{Summation of V2 of available EC} + \\
 & \text{Summation of V3 of available EC} - \\
 & \text{Summation of V4 of available EC}
 \end{aligned}
 \tag{Eq. 1}$$

The proposed objective function is of minimum problem as most of the included parameters V1, V2, and V3 prefer small values. For example, an EC with a smaller distance to the closed EC is preferred compared to the EC that has farther distance. On the other hand, the fourth parameter which is V4 is of maximum value as the system needs to avoid EC that is near to a river. In addition, the first variable includes the constraint of 75% usage as we need to ensure that there isn't any EC that is 100% occupied for safety and convenience purposes.

Tabu Search for Optimal Evacuees Management

In Figure 4, pseudo code of the proposed Tabu Search is presented. The TS_{Flood} starts by randomly generating an initial solution, *X*, and denote the solution as the best solution where *X**=*X*. The solution, *X*, is represented in binary form, where the length of the representation is based on the number of evacuation centers. Each bit in this solution represents one EC; if the value is 0 it means that the EC is not chosen and if it is 1 it shows that EC is selected. The best solution will then undergo an evaluation using the objective function.

In Step 6 of the TS_{Flood}, if the termination criteria (a termination criterion is based on number of iteration) is reached then the process is stopped, else continues by

generating a subset of solutions based on the best solution (as shown in step 7). The solutions are produced by adding or deleting item from the best solution. The number of addition is determined by a random number, while the number of for deletion equals the difference between the number of sub solutions and the addition.

The generated sub solutions are later evaluated using objective function (utility function in equation 1) and the best solution is identified. The aim of the proposed TS_{Flood} is to identify a solution that the smallest utility function value.

Input:
Step 1: Input the dataset that includes four variables [v1, v2, v3, v4].
Step 2: Determine the max capacity for EC.

Process:
Step3: Generate random start solution *X*, and set the best solution *X**=*X*.
Step4: Evaluate the best solution using objective function (Equation 1)
Step5: Check the generated solution, it must fulfill the constraint of summation of (75% of v1) of all record in one solution exceed the capacity value. If fail, the generated solution will undergo refinement process (one item will be added randomly to the solution).
Step6: If termination criteria is fulfilled, then stop (termination criterion is based on number of iteration).
Step7: Generate a subset of solutions based on best solution by adding or deleting item from best solutions. The number of adding process is determined by random number, while the number of deleting process equals the difference between the number of sub solutions and the number of adding process.
Step8: Evaluate the sub solutions using objective function as shown in equation 1 and check the generated solution as in Step3.
Step9: Choose the best solution from this sub solutions if *f(x) < f(X*)* then *X**= *X*
Step 10: End

Fig. 4. Pseudo code of proposed Tabu Search algorithm (TS_{Flood})

Firefly Algorithm for Optimal Evacuees Management

The pseudo code of proposed Firefly algorithm (FA_{Flood}) in decision making for AEECM is illustrated in Figure 5. Further, in the input of the proposed FA_{Flood} needs to define some important parameters for operating the algorithm such as the light absorption coefficient γ , where it set to 1 in the algorithm, the value of initial attractiveness β_0 , where it sets to 1, the number of max generation, the value of capacity which equal to value of the number of vectims that need to be in safe places, the number of fireflies which equal to 10% from the number of records in dataset, and finally is the number of initial solution which is equal to the number of fireflies.

The proposed FA_{Flood} starts to operate by generating initial solutions which are represented in binary form [5], where the dimension of one solution equals the number of records in dataset (i.e evacuation centers). Generation of the solutions are performed using two ways: If variable v1 of a record is higher than the capacity, it will take it as one solution by assigning 1 in the solution, else, it randomly generate a solution that has 2 or more records. These solutions then undergo a verification

process that checks if the summation of v1 for all records in one solution exceed the capacity value. After that, the generated initial solution need to be evaluated based on objective function (Utility function) as shown in equation 1. Assign the inverse of utility value (fitness) of each solutions to each firefly as initial light (I). Then, the initial position of each firefly are determined, which is represented the solution.

Fireflies compete between them to determine the best solution that has the highest fitness value. Firefly with brighter light attracts the less bright ones and this is based on the distance between two solutions using Hamman distance as shown in equation 2. Then, the attractiveness between two solutions using equation 3 is calculated. The less bright firefly will move to the brighter one using equation (4).

$$r_{ij} = \frac{\text{the number of dissimilar bit in } i \text{ and } j \text{ solutions}}{\text{the number of records in dataset}} \quad (2)$$

$$\beta = \beta_0 \exp(-\gamma r_{ij}^2) \quad (3)$$

$$X^i = X^i + \beta_0 \exp(-\gamma r_{ij}^2) * (X^j - X^i) + \text{rand} - 0.5 \quad (4)$$

After moving, a new solution is generated, If the summation of v1 in new solution greater or equal to capacity, then, the utility function and fitness for new solution are calculated and compared with old solution, if a new solution is better than old solution then replace it. In the situation where the summation of v1 in a new solution is smaller than the capacity, a mutation process is conducted on the new solution by adding one bit randomly until pass the capacity value. Then the utility function of the new solution is calculated and compared against the old solution in order to identify the best solution. Once the predefined number of iteration is reached, the fireflies are sorted based on their brightness that indicates the utility value.

C. Evaluation

The effectiveness of TS_{Flood} and FA_{Flood} is evaluated based on two criteria; utility value and computational time. The AEECM prefers the method that produces solution with the lowest utility value and computational time. In addition, two scenarios were employed; the first scenario investigates solutions for a to-be closed EC with number of victims that is larger than the capacity of any available EC. Meaning that the solution is expected to consist a combination of ECs. On the other hand, the second scenario represents situations where the to-be closed EC has the same or less number of victims as the EC.

Input:
Step 1: Input the dataset that includes four variables [v1, v2, v3, v4].
Step 2: Define light absorption coefficient γ , where $\gamma=1.0$
Step 3: Define initial attractiveness $\beta_0 = 1$
Step 4: Determine the Max Generation.
Step 5: Determined the Capacity.
Step 6: Determined the number of fireflies which equal 10% from the number of records in dataset.
Step 7: Determine the number of initial solution which is equal the number of fireflies.
Process:
Step 8: The solutions are represented in binary form (0,1), where the dimension of one solution equal the number of records in dataset.
Step 9: Generated the solutions are undertaken in two ways:
Step 9.1: If the variable v1 in one record pass the capacity, it will take it as one solution by assigning 1 in solution.
Step 9.2: If the v1 in one record less than the capacity, then, the solution will generated randomly that take more than one record in one solution.
Step 9.3: Check the generated solution must pass the constrain which is the summation of v1 of all record in one solution pass the capacity value.
Step 10: Calculated the initial utility function for each solution as in Eq 1
Step 11: Assign the utility value to each firefly as initial light (I)...meaning that fitness = 1/utility
Step 12: Determine position of each firefly, which is the initial solution.
Step 13: While (t < Max Generation): Max_Generation = No of Firefly = 0.1 * number of EC
Step 14: For i=1 to N (N all fireflies)
Step 15: For j=1 to N
Step 16: If fitness_i < fitness_j (li<lj) {
Step 17: Calculated the distance between two solutions using Hamman distance the following equation:

$$r_{ij} = \frac{\text{the number of dissimilar bit in } i \text{ and } j \text{ solutions}}{\text{the number of records in dataset}} \quad (2)$$
Step 18: Calculated the attractiveness between two solutions using the following equation:

$$\beta = \beta_0 \exp(-\gamma r_{ij}^2) \quad (3)$$
Step 19: Move less brighter firefly to high brighter firefly

$$X^i = X^i + \beta_0 \exp(-\gamma r_{ij}^2) * (X^j - X^i) + \text{rand} - 0.5 \quad (4)$$
Step 20: If summation of v1 in new solution >= Capacity
Step 20.1: Calculated the utility function and fitness for new solution.
Step 20.2: Compare with old solution.
Step 20.3: Replace old solution with new solution.
Step 21: Else if summation of v1 in new solution < Capacity
Does mutation for one bit random in new solution until pass Capacity.
Step 21.1: Calculated the utility function for new solution.
Step 21.2: Compare with old solution.
Step 21.3: Replace old solution with new solution.
Step 22: End For i
Step 23: End For j
Step 24: End While
Step 25: Rank the fireflies and find the current global best utility function, and best solution.
Output
Step 26: Sort the best solution based on v1.

Fig. 5. Pseudo code of proposed Firefly Algorithm (FAFlood)

IV. RESULTS

In Table 2, result for the first scenario is presented while Table 3 depicts results for the second scenario. The proposed algorithms were executed for three times. For the first scenario, assuming the closed EC is ECID_76 and the number of victims in this EC is (298) greater than the capacity of any EC. As shown in Table 2, the utility values of FA_{Flood} are (5635.15, 5235.45, and 4068.59) which are smaller than TS_{Flood} which are (5754.95, 5545.95, and 4521.59). However, the execution time for TS_{Flood} (305, 368, and 295) is better than FA_{Flood} which took 676, 641, and 641ms. The graphical illustration is provided in Figure 7.

In Table 3, the closed EC is ECID_35 and the number of people in this EC is (221) smaller or equal input EC. As can observe in Table2, the utility values of proposed FA_{Flood} are (260.1, 265 and 276) for three executions better than (minimum is better) TS_{Flood} which are (260.1, 288, and 276), (260.1, 283.1, and 279.05) and (260.1, 265, and 279.05) as shown in Figure 7.c, however, the time execution of proposed TS_{Flood} algorithm is (277, 282 and 266) better (smaller is better) than proposed FA_{Flood} algorithm which are (594) in three executions. The graphical representation is as shown in Figure 7.d.

V. CONCLUSION

Over the past decades, the pattern of floods across all continents has been changing, becoming more frequent, intense and unpredictable for local communities. Even though

various strategies have been implemented in managing flood evacuation centers, unforeseen scenarios may lead to the closure of the center. Hence, in this study, a variant of Firefly Algorithm (FA) that provides optimal solution to the management of victims in an evacuation center is presented. In particular, the proposed FA determines the number of victims to be relocated to a particular center. Evaluation of the proposed FA is undertaken by comparing its result (i.e utility function and computational time) against the one produced by Tabu Search which is also an example of a meta heuristics algorithm. Two types of experiments were performed; number of victims in a to-be closed EC is larger and, smaller or equals the capacity of any available EC. Results of the first experiment show that the average value of utility function produced by FA solutions (i.e 4980) is smaller than the one obtained by Tabu Search (i.e 5274). Similar pattern can also be seen in the utility function values of the second experiment. In this study, as the problem is formulated as a minimization function, FA that produces a smaller utility function is preferred. Nevertheless, in both experiments, it is noted that FA consumes larger computational time compared to Tabu Search. FA requires at least 594ms to produce a solution while Tabu Search only took 266ms.

ACKNOWLEDGMENT

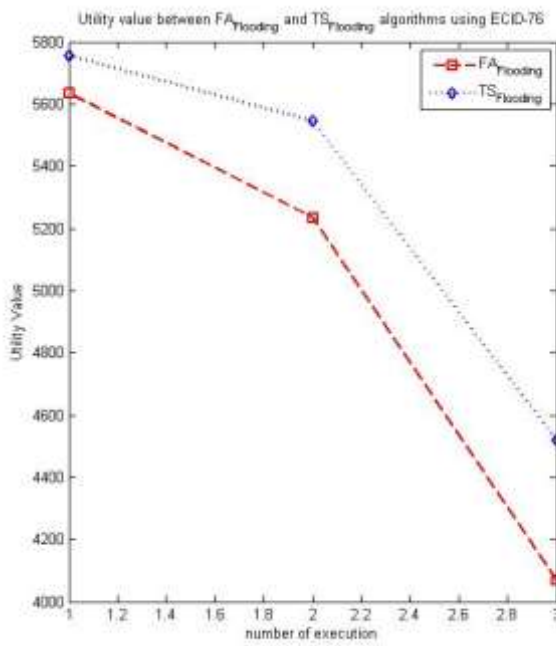
Authors would like to thank the Malaysian Ministry of Higher Education for the financial support given under the Fundamental Research Grant Scheme (S/O Code 13183).

TABLE II. RESULTS OF SCENARIO 1

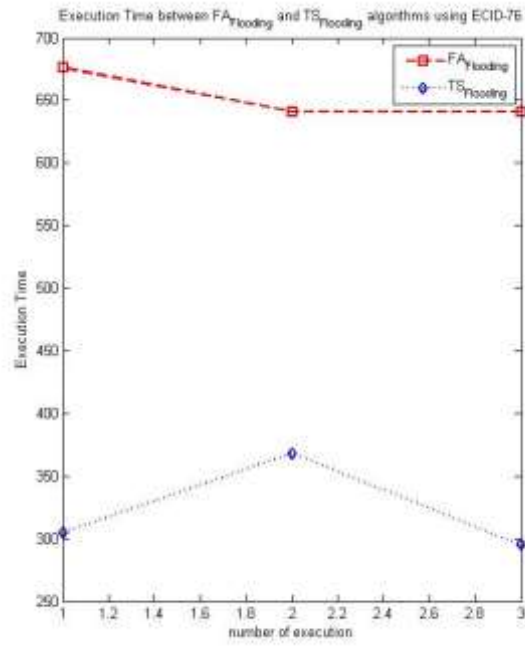
Center to-be closed	Victims	FA_{Flood}			TS_{Flood}		
		Solution	Utility value	Time (ms)	Solution	Utility value	Time (ms)
ECID_76	298	ECID_38 = 255	5635.15	676	ECID_38 = 255	5754.95	305
		ECID_34 = 43			ECID_80 = 43		
		ECID_34 = 245			ECID_80 = 253		
		ECID_64 = 53			ECID_70 = 45		
		ECID_64 = 242			ECID_70 = 249		
		ECID_60 = 56			ECID_90 = 49		
ECID_76	298	ECID_70 = 249	5235.45	641	ECID_70 = 249	5545.95	368
		ECID_34 = 49			ECID_90 = 49		
		ECID_34 = 245			ECID_90 = 203		
		ECID_64 = 53			ECID_125 = 95		
		ECID_64 = 242			ECID_125 = 197		
		ECID_121 = 56			ECID_103 = 101		
ECID_76	298	ECID_88 = 255	4068.59	641	ECID_37 = 189	4521.59	295
		ECID_70 = 43			ECID_46 = 109		
		ECID_70 = 249			ECID_46 = 188		
		ECID_34 = 49			ECID_57 = 110		
		ECID_34 = 245			ECID_57 = 179		
		ECID_64 = 53			ECID_105 = 119		

TABLE III. RESULTS OF SCENARIO 2

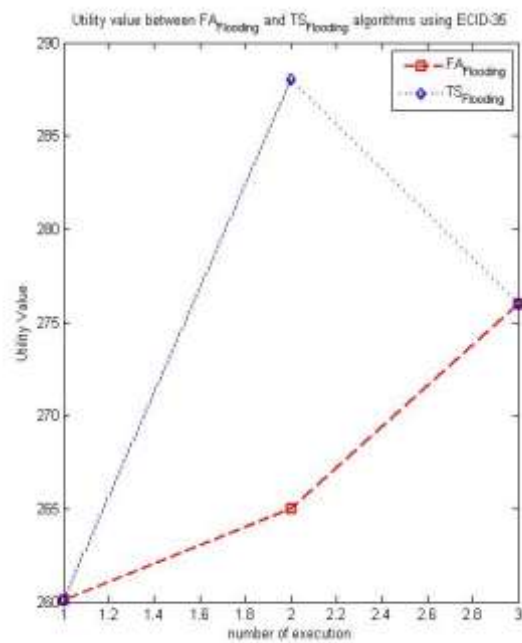
Center to-be closed	Victims	FAFlood			TSFlood		
		Solution	Utility value	Time (ms)	Ssolution	Utility value	Time
ECID_35	221	ECID_34 = 221	260.1	594	ECID_88 = 221	260.1	277
		ECID_38 = 221	265.0		ECID_70 = 221	288.0	
		ECID_64 = 221	276.0		ECID_34 = 221	276.0	
ECID_35	221	ECID_34 = 221	260.1	594	ECID_88 = 221	260.1	282
		ECID_38 = 221	265.0		ECID_70 = 221	283.1	
		ECID_64 = 221	276.0		ECID_80 = 221	279.05	
ECID_35	221	ECID_34 = 221	260.1	594	ECID_88 = 221	260.1	266
		ECID_38 = 221	265.0		ECID_121 = 221	265.0	
		ECID_64 = 221	276.0		ECID_80 = 221	279.05	



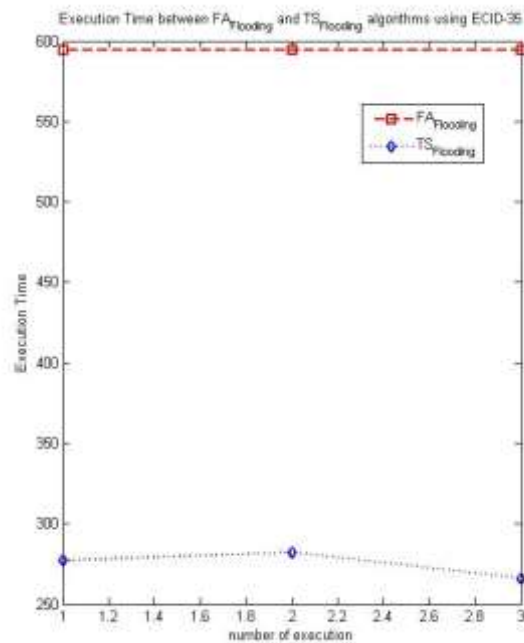
(a)



(b)



(c)



(d)

Fig. 6. A graphical representations of utility function and execution time results: FA_{Flood} vs. TS_{Flood}. (a) Utility for ECID-76 (b) Execution time for ECID-76 (c) Utility for ECID-35 (d) Execution time for ECID-35

REFERENCES

- [1] J. Zhu, J. Huang, D. Liu, and J. Han, "Resources Allocation Problem for Local Reserve Depots in Disaster Management Based on Scenario Analysis," presented at 7th International Symposium on Operations Research and Its Applications (ISORA'08), Lijiang, China, 2008.
- [2] S. Duran, M. A. Gutierrez, and P. Keskinocak, "Pre-Positioning of Emergency Items for CARE," *Interfaces*, vol. 41, pp. 223-237, 2011.
- [3] E. Bonabeau, G. Theraulaz, and M. Dorigo, *Swarm Intelligence: From Natural to Artificial Systems*: Oxford University Press, 1999.
- [4] M. Dorigo and L. M. Gambardella, "Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem," *IEEE Transactions on Evolutionary Computation*, vol. 1, pp. 53-66, 1997.
- [5] A. C. Adamuthe and R. S. Bichkar, "Tabu search for solving personnel scheduling problem," presented at International Conference on Communication, Information & Computing Technology (ICCICT), 2012.
- [6] S. N. Vu, M. H. N. Nguyen, L. M. Duc, C. Baril, V. Gascon, and T. B. Dinh, "Iterated local search in nurse rostering problem," in *Proceedings*

- of the Fourth Symposium on Information and Communication Technology. Danang, Vietnam: ACM, 2013, pp. 71-80.
- [7] S. Das, A. Abraham, and A. Konar, *Metaheuristic Clustering*, vol. 178: Springer-Verlag Berlin Heidelberg, 2009.
- [8] F. Rothlauf, *Design of Modern Heuristics: Principles and Application*: Springer Publishing Company, Incorporated, 2011.
- [9] I. Boussaid, J. Lepagnot, and P. Siarry, "A survey on optimization metaheuristics," *Information Sciences*, vol. 237, pp. 82-117, 2013.
- [10] F. Glover, "Future paths for integer programming and links to artificial intelligence," *Computational Operation Research*, vol. 13, pp. 533-549, 1986.
- [11] X.-S. Yang, *Nature-Inspired Metaheuristic Algorithms: Second Edition*: Luniver Press, 2010.
- [12] A. Rathore, A. Bohara, R. G. Prashil, T. S. L. Prashanth, and P. R. Srivastava, "Application of genetic algorithm and tabu search in software testing," in *Proceedings of the Fourth Annual ACM Bangalore Conference*. Bangalore, India: ACM, 2011, pp. 1-4.
- [13] B. K. Koorra, N. R. Satpute, and A. Adiga, "Tabu search based implementation of object tracking using joint color texture histogram," presented at 2012 IEEE 7th International Conference on Industrial and Information Systems (ICIIS), 2012.
- [14] T. Dinh, T. Dinh, and J. Ferland, "A meta-heuristic approach for cell formation problem," in *Proceedings of the Second Symposium on Information and Communication Technology*. Hanoi, Vietnam: ACM, 2011, pp. 11-18.
- [15] D. Chalupa, "Population-based and learning-based metaheuristic algorithms for the graph coloring problem," in *Proceedings of the 13th annual conference on Genetic and evolutionary computation*. Dublin, Ireland: ACM, 2011, pp. 465-472.
- [16] J. Xie, Y. Mei, and A. Song, "Evolving Self-Adaptive Tabu Search Algorithm for Storage Location Assignment Problems," in *Proceedings of the Companion Publication of the 2015 Annual Conference on Genetic and Evolutionary Computation*. Madrid, Spain: ACM, 2015, pp. 779-780.
- [17] E. Arraiz and O. Olivo, "Competitive simulated annealing and Tabu Search algorithms for the max-cut problem," in *Proceedings of the 11th Annual conference on Genetic and evolutionary computation*. Montreal, Quebec, Canada: ACM, 2009, pp. 1797-1798.
- [18] C. Grosan and A. Abraham, *Intelligent Systems: A Modern Approach*, vol. 17: Springer-Verlag Berlin Heidelberg, 2011.
- [19] R. Tang, S. Fong, X. S. Yang, and S. Deb, "Integrating nature-inspired optimization algorithms to K-means clustering," presented at *International Conference on Digital Information Management (ICDIM)*, 2012.
- [20] X. S. Yang, "Firefly algorithm, stochastic test functions and design optimisation," *Int. J. Bio-Inspired Comput.*, vol. 2, pp. 78-84, 2010.
- [21] M.-H. Horng and T.-W. Jiang, "Multilevel Image Thresholding Selection Based on the Firefly Algorithm," presented at *7th International Conference on Autonomic and Trusted Computing (UIC/ATC)*, 2010.
- [22] Y. Yusof, F. K. Ahmad, S. S. Kamaruddin, M. H. Omar, and A. J. Mohamed, "Short Term Traffic Forecasting Based on Hybrid of Firefly Algorithm and Least Squares Support Vector Machine," presented at *Proceedings of First International Conference on Soft Computing in Data Science*, Putrajaya, 2015.
- [23] M. K. Sayadi, A. Hafezalkotob, and S. G. J. Nainia, "Firefly-inspired algorithm for discrete optimization problems: An application to manufacturing cell formation," *Journal of Manufacturing Systems*, vol. 32, pp. 78-84, 2013.
- [24] N. Sudarshan, S. P. Pratim, and D. Achintya, "Analysis of a Nature Inspired Firefly Algorithm based Back-propagation Neural Network Training," *International Journal of Computer Applications*, vol. 43, pp. 8-16, 2012.
- [25] A. J. Mohammed, Y. Yusof, and H. Husni, "Document Clustering Based on Firefly Algorithm," *Journal of Computer Science*, vol. 11, pp. 453-465, 2015.
- [26] A. J. Mohammed, Y. Yusof, and H. Husni, "GF-CLUST: A Nature Inspired Algorithm for Automatic Text Clustering," *Journal of Information and Communication Technology (IJCT)*, vol. 15, pp. 57-81, 2016.
- [27] X.-S. Yang, S. S. S. Hosseini, and A. H. Gandomi, "Firefly Algorithm for solving non-convex economic dispatch problems with valve loading effect," *Appl. Soft Comput.*, vol. 12, pp. 1180-1186, 2012.