An Intelligent Approach for Optimizing Energy-Efficient Packets Routing in the Smart Grid Internet of Things

Chih-Kun Ke[1(✉)], Mei-Yu Wu[2], and Chia-Yu Chen[1]

[1] Department of Information Management, National Taichung University of Science and Technology, North Dist, No.129, Sec.3, Sanmin Rd., 40401 Taichung, Taiwan, R.O.C.
ckk@nutc.edu.tw

[2] Department of Business Management, National Taichung University of Science and Technology, 40401 Taichung, Taiwan, R.O.C.

Abstract. This work proposes a multi-criteria artificial bee colony (MABC) algorithm to optimize the energy consumption problem in wireless sensor networks. The approach uses the artificial bee colony (ABC) algorithm to discover sensor nodes in a network as a cluster header combination. Different nodes are dynamically selected according to their current status in the network. The purpose is to cluster sensor nodes in the network in such a way that nodes can transmit packets to their cluster header, and then identify the most energy efficient packet routing from the cluster headers to the Internet of Things (IoT) base station. The routing strategy takes into account nodes' residual energy and energy consumption, routing distance, number of hops, and frequency, in order to assign decision scores to help the algorithm discover a better solution. The use case shows that the MABC algorithm provides energy-efficient packet routing, and thus extends the wireless sensor network lifespan, which is confirmed by the multi-criteria analysis evaluation of the candidate routing. The contribution of this research is its use of swarm intelligence algorithms in wireless sensor network routing, with a multi-criteria artificial bee colony algorithm used in a wireless sensor network to address the problem of fast convergence of the algorithm.

Keywords: Internet of Things · Packet routing · Energy-efficient · Artificial bee colony algorithm · Multi-criteria decision analysis

1 Introduction

The Internet of things (IoT) [1] provides a multitude of convenient services in daily life, and wireless sensor networks (WSN) are a key technology in IoT development. However, WSN sensors have some limitations and challenges, such as limited power, fewer instructions per second (IPS), less storage, and lower network bandwidth. The limited power of sensors, in particular, is a key challenge in IoT-enabled smart grid development. WSN sensors collect data from the sensing environment and communicate

Y.-B. Lin and D.-J. Deng (Eds.): SGIoT 2020, LNICST 354, pp. 260–269, 2021.
https://doi.org/10.1007/978-3-030-69514-9_21

that data to each other through radio signals, with each transmission consuming each node's limited store of energy [2]. In addition, general wireless sensors for environmental monitoring are often installed in areas that people cannot reach. If a wireless sensor's energy is depleted, it is thus difficult to provide power support, meaning the sensors will be abandoned, leaving gaps in the WSN and reducing its ability to warn of environmental threats [3]. A more flexible and faster programming method based on the status of each sensor in the network is thus crucial to extend the lifespans of both wireless sensors and WSNs.

Optimization problems related to engineering scheduling or mathematical sciences are very complex and challenging. The purpose of such problems is to identify the best and most feasible solution from all candidate solutions. However, current methods tend to become trapped in local optima, and require novel solutions to improve the problems [4]. In recent years, researchers have explored the use of optimization algorithms based on Swarm Intelligence (SI), using characteristics of swarms of creatures found in nature. SI algorithms decentralize control between swarms, and emulate self-organized collective behavior. Through these algorithms, each agent follows simple rules to perform operations locally, looking for the best solution for complex problems, and these rules, as the name implies, are inspired by nature. The algorithms tend to be flexible, easy to implement, and sufficiently versatile to handle different types of optimization problems. The artificial bee colony algorithm (ABC) is one such swarm intelligence algorithm [5]. ABC is inspired by the foraging behavior of honey-bee colonies. Various types of bees perform different activities according to their division of labor. By sharing and exchanging foraging information, they can find the optimal solution to a given problem. The main advantage of ABC is that it does not need to consider a specific solution, but rather compares all solutions; even information contained in a bad solution thus becomes useful. Through the local optimization of each type of bee, the optimal value of the problem becomes the solution for the entire bee colony, and the convergence speed is faster.

Multi-criteria decision analysis (MCDA) is decision-making analysis based on multiple conflicting criteria, and MCDA problems can be defined as finding multiple alternatives using multiple criteria [6]. This process can help decision-makers divide those multiple alternatives into a ranking order based on the nature of each alternative, and the characteristics of each criterion. They can then arrange the priorities, evaluate alternatives, and select a plan that is closest to the user's ideal solution. In MCDA, decision-making factors such as the properties, attributes, and criteria of alternatives can be processed using two types of MCDA: multi-objective decision making (MODM), and multi-attribute decision making (MADM). MODM is an alternative plan comprised of a set of restrictive conditions, which seeks the best solution for several objective functions, while MADM provides a set of alternative plans, considering multiple attributes (criteria) to evaluate the alternatives, with the best option determined by the order of the evaluation results [7]. A simple additive weighting method (SAW) was developed by Churchman et al. in 1954 [8]. In the SAW meth-od, each attribute (criteria) is assigned a weight, and then the performance value of each attribute (criteria) is converted into a number by multiplying it by the weight value. The preliminary priority order can then be arranged according to these scores [9–11].

This study proposes an intelligent approach to optimizing the problem of energy consumption in WSNs. The approach uses the ABC algorithm to discover sensor nodes in the network as a cluster head combination. Different nodes are dynamically selected according to their current status in the network. The purpose is to cluster the sensor nodes in the network in such a way that other nodes can transmit packets to their respective cluster headers, and then find the most energy-efficient packet routing from the cluster headers to the IoT base station. The routing method takes into account multiple attributes, including nodes'residual energy, energy consumption, routing distance, number of hops and frequency to derive decision scores, which are then used to discover a more energy-efficient routing solution.

The remainder of this paper is organized as follows. Section 2 presents the proposed intelligent approach for optimizing energy-efficient packet routing in the smart grid IoT. The use case demonstration follows in Sect. 3. Finally, Sect. 4 presents the conclusions.

2 The Proposed Approach

This section describes how the artificial bee colony algorithm (ABC) is combined with the simple additive weighting method (SAW) to solve the random selection problem. The proposed approach is called a multi-criteria artificial bee colony algorithm (MABC). Based on the rich attributes in packet transmission, the approach recommends the most energy-efficient routing for each cluster header. First, the MABC algorithm is described, and a detailed analysis and calculations are presented in Sect. 3. Table 1 presents the parameter definitions.

Table 1. The parameter definitions for the proposed MABC algorithm

Parameter	Definition
Cluster_Header_Set	The cluster header set;
Cluster_Set	The cluster set;
Best_Routing_Set	The best routing set calculated by ABC;
Routing _Set	The initial routing set;
Max_ Iter	The maximum iteration of ABC;
Limit	The maximum mining limitation of a food solution of ABC;
Bee_number	The number of bees in the ABC;
MCDA_Parameter	An MCDA parameter;
MCDA_Candidate	A candidate set calculated by MCDA

The pseudo-code of the proposed MABC algorithm is as follows.

Pseudo-code: MABC routing algorithm.
Input: The set of cluster and header information (`Cluster_Set`).
Output: The best routing set calculated by MABC (`Best_Routing_Set`).

```
MABC_Routing(Cluster_Header_Set, MCDA_Parameter){
 Set the parameter Max_ Iter, Limit and Bee_number
 ABC(Routing_set, Max_liter, Limit, Bee_number){
  Initialize food sources
  Evaluate the food sources' the fitness
  Iter_number ← 1
   Do While (Iter_number < Max_liter)
     For i =1:
      Generate new routing solutions in Clus-
       ter_Header_Set.
      Evaluate the new routing solutions' fitness
      Apply greedy selection process
     End for
     MCDA_Candidate = SAW(MCDA_Parameter);
     Sort(MCDA_Candidate);
     //Onlooker bee process
     For i=1:
      Onlooker bees choose the best one in
       MCDA_Candidate.
      Generate new routing solutions using onlooker
       bees
      Evaluate the new routing solutions' fitness
      Apply greedy selection process
     End for
     // Scout bee process
     IF Limit_number > Limit:
      Generate new CH solutions in
       Candiate_Sensor_Set.
      Limit_number = 0
      Memorize the best solution.
      Iter_number ← Iter_number + 1
   End While
   Return best solution.
 }
 Return Best_Routing_Set.
}
```

The routing plan obtains the best cluster header solution and member node information of each cluster. It then uses the ABC algorithm to evaluate the most energy-efficient routing for the cluster header. The algorithm parameters include the number of bee colonies; the number of employee bees, onlooker bees and scout bees; the maximum number of iterations of the algorithm (Max_Iter); the maximum mining limitation (Limits) for each food solution; and the initial food source of the bees. The food source is the current best routing cluster and problem solution, as well as the best routing cluster reconciliation of each cluster header.

Since it is necessary to evaluate the pros and cons of the initial solutions before colony mining, this study calculates the fitness values of all solutions generated. The fitness value represents the total energy consumption of all routings in the problem solution. A smaller problem solution fitness value means that the routing cluster will consume relatively less energy, while a larger fitness value indicates greater energy consumption. Finally, the total value is added to derive the fitness value of the problem solution, and the evaluation step ends when all problem solutions are calculated.

After evaluating the fitness value of all problem solutions, the employee bees take the lead in mining each problem solution. Because the routing is modified, the energy consumption of the routing cluster must be re-evaluated as the new problem solution's fitness. The fitness value is the energy consumed by each cluster header in the routing in order to transmit packets to each other. The energy consumed by each cluster header in order to receive transmitted packets is included to obtain the overall fitness value. If the fitness value of the new problem solution is greater than that of the original solution, the employee bee adopts the greedy selection method to select a better solution. Therefore, the employee bee keeps the original problem solution and discards the new problem solution. If the solution has not been updated, the number of mining (Limits) will increase by one, until the maximum mining limitation is exceeded, and will be eliminated by the scout bee, and so on until the employee bee update is completed.

When all employee bees have updated the problem solution, they will inform the onlooker bees of the information they have mined so far. In the traditional artificial bee colony algorithm, the onlooker bee uses the roulette wheel selection method to select the employee bee position to follow. The purpose of onlooker bees searching for solutions is to find a better solution based on the current problem solution. Therefore, it is important to choose a better solution among the current candidates; however, the roulette selection method has some shortcomings. First, roulette uses the fitness value of each problem solution to calculate its probability of being selected. If the probability value is very different, the selection will be problematic. The current better problem solution will have a greater chance of being selected by the onlooker bee, and the introduction of bias into the solution is sought, leading to premature convergence and loss of diversity. On the other hand, the calculation of the probability value only considers the fitness value at a single level, which means that the pros and cons of the routing are considered at a single level; this may make it difficult to find a better routing problem solution with single-level evaluation routing.

Therefore, this study modified the roulette selection method to introduce MCDA. The multiple attributes in the routing are evaluated to help the onlooker bee search for better solutions. Based on [12, 13], this work takes into account routing attributes, including the total routing distance, total energy consumption, total number of hops, frequency, and remaining total energy, in order to evaluate the routing. Longer routing distances result in greater energy consumption for packet transmission. Total energy consumption consists of the energy consumed by each node in the transmission routing. The total hop count indicates the number of nodes through which a packet is passed from the cluster head node to the IoT base station in each routing, with more hops resulting in greater energy consumption. The remaining energy represents the remaining energy of each node after a packet is transmitted. The frequency represents the number of packets sent or forwarded by each node, and the higher the frequency, the faster the battery power will be consumed.

This work uses the simple additive weighting method (SAW) to set the weight value of each packet transmission routing according to the characteristic attributes, and to evaluate the recommended score of each routing solution. A higher score indicates a better solution. In this way, the onlooker bee chooses the best solution to the mining problem according to the recommended result evaluation, followed by further mining. The SAW calculation process is as follows. The onlooker bee intends to mine a food source. The goal is to find a better solution for each evaluation attribute among existing food solutions. The decision-making solution is based on the problem-solving content mined by the employee bee. The network administrator sets the weights of the evaluation indicators, where each weight is assigned to each evaluation indicator, and the total is 1. The routing solution and its evaluated attributes are converted into a decision matrix, which is then normalized. The normalized matrix is then used to calculate the direction normalization. This study calculated the direction normalization matrix using the minimization criterion. The weight matrix and the direction normalization matrix are comprehensively evaluated to obtain a comprehensive evaluation value. Finally, the comprehensive evaluation values are ranked from large to small, and the observation bee selects the better-ranked solution for updating, according to the sorted problem solution.

When the employee bee and the onlooker bee have finished their work, the scout bee will replace certain food solutions that exceed the mining times (Limits) by randomly generating new routing solutions to replace them with new food solutions, and increase the number of iterations by one. In this way, the algorithm stops computing until the number of loops exceeds the maximum number of iterations (Max_Iter), and returns the best solution after computing.

3 Use Case Demonstration

This study randomly generated the routing for each cluster header to transmit a packet to the base station. The routing included the cluster header. In the cluster header selection step, this study calculated the best cluster header combination in the network as [4, 7–9, 11] to generate six food sources, as shown in Table 2. The numbers 0 to 4 in the routing were represented as [4, 7–9, 11], and the number 5 was the base station. Each employee bee then visited various food sources for mining, and the food source was called the problem solution.

Taking Solution 1 as an example, the transmission routing of the 0th cluster header was [0, 1, 4, 5, 2, 3]. First, the cluster header needed to transmit five packets to the base station. The energy consumed by each cluster header in order to transmit the packets, as well as the energy consumed by each cluster header in order to receive packets, was

Table 2. Routing planning and solutions of the MABC

Solution	Routing
1	[[0, 1, 4, 5, 2, 3], [1, 2, 4, 0, 5, 3], [2, 3, 0, 4, 5, 1], [3, 2, 1, 4, 5, 0], [4, 5, 3, 1, 2, 0]]
2	[[0, 5, 3, 1, 2, 4], [1, 5, 3, 4, 0, 2], [2, 4, 3, 0, 5, 1], [3, 4, 0, 1, 5, 2], [4, 2, 3, 1, 5, 0]]
3	[[0, 2, 3, 1, 4, 5], [1, 3, 5, 0, 2, 4], [2, 1, 0, 4, 3, 5], [3, 4, 2, 0, 1, 5], [4, 2, 0, 1, 5, 3]]
4	[[0, 1, 4, 3, 5, 2], [1, 3, 4, 2, 5, 0], [2, 0, 1, 5, 4, 3], [3, 2, 0, 4, 1, 5], [4, 2, 0, 1, 5, 3]]
5	[[0, 3, 5, 4, 1, 2], [1, 5, 4, 2, 3, 0], [2, 3, 5, 4, 0, 1], [3, 1, 5, 4, 0, 2], [4, 5, 3, 0, 2, 1]]
6	[[0, 2, 3, 1, 4, 5], [1, 0, 5, 3, 2, 4], [2, 4, 1, 5, 3, 0], [3, 5, 1, 2, 4, 0], [4, 3, 5, 1, 0, 2]]

calculated as shown in Table 3, as 0.00002500026 + 0.00005313742 + 0.00005529074 = 0.00013342842 J, and so on to calculate the energy consumed by other routings. The value was added as the fitness value of the problem solution, and the evaluation step ended when all the problem solutions were calculated.

Table 3. Routing energy consumption calculation of the MABC

Routing	Energy consumption calculation
$0 \rightarrow 1$	$500 \times 5\text{e-}8 + (4.47213595499958)^4 \times 500 \times 1.3\text{e-}15 = 0.00002500026$
$1 \rightarrow 4$	$500 \times 5\text{e-}8 + 500 \times 5\text{e-}8 + (46.87216658103186)^4 \times 500 \times 1.3\text{e-}15 = 0.00005313742$
$4 \rightarrow 5$	$500 \times 5\text{e-}8 + 500 \times 5\text{e-}8 + (53.41348144429457)^4 \times 500 \times 1.3\text{e-}15 = 0.00005529074$

Continuing the Solution 1 example, the routing was [[0, 1, 4, 5, 2, 3], [1, 2, 4, 0, 5, 3], [2, 3, 0, 4, 5, 1], [3, 2, 1, 4, 5, 0], [4, 5, 3, 1, 2, 0]], and the employee bee changed the transmission sequence of all routings to [1, 2] to form a new problem solution, so the new problem solution was [[0, 4, 1, 5, 2, 3], [1, 4, 2, 0, 5, 3], [2, 0, 3, 4, 5, 1], [3, 1, 2, 4, 5, 0], [4, 3, 5, 1, 2, 0]]. Because the routing was modified, the energy consumption of the routing cluster header to be recalculated. The fitness value was the energy consumed by each cluster header of the routing in order to transmit packets to each other, plus the energy consumed by each cluster header in order to receive packets. The total available energy was 0.0023852419407 J (J). The original fitness value was 0.00013342842 J (J), which was less than the available energy. If the fitness value of the new problem solution is greater than the original solution, the employee bee adopts a greedy selection method to select a better solution; it will thus keep the original problem solution and discard the new problem solution. If the solution is not updated, the Limits will be increased by one until the maximum mining limitation is exceeded; it will then be eliminated by the onlooker bee when the employee bee has updated.

In MCDA, the SAW method is used to assign weight values to criteria. The SAW calculation process is as follows. The onlooker bee intends to mine the food source.

Solution S = $\{S_1, S_2, S_3, \ldots, S_n\}$, where S includes total routing length, total energy consumption, total number of hops, total remaining energy of nodes, and frequency in the cluster header transmission routing combination set, as shown in Table 4.

Table 4. Candidate solutions of MCDA

No	Routing length	Energy consumption	Hops	Remaining energy	Frequency
1	926.932114095416	0.004265394907499999	12	9.995734605092501	47.4
2	1078.3434271204517	0.004638783719999999	16	9.99536121628	61.2
3	463.7951328053706	0.0024948649999999998	5	9.997505134999999	21.6
4	1117.8385570224605	0.0048117749407	13	9.9951882250593	55.6
5	1129.3773153600205	0.0047363974101	14	9.995263602589901	53.2
6	427.9026888159207	0.0018840385002000001	7	3, 9.9981159614998	12.6

There are five decision evaluation indicators used in MCDA. The decision evaluation indicators of this study refer to the routing attributes [12, 13], as shown in Table 5. C = $\{C_1, C_2, C_3 \ldots\ldots C_m\}$, where C_1, C_2, C_3 and C_5 are minimization criteria, and C_4 is the maximum criterion.

Table 5. Evaluation criteria

Criteria	Routing length ($\boldsymbol{C}_1$)	Energy consumption ($\boldsymbol{C}_2$)	Hops($\boldsymbol{C}_3$)	Remaining energy ($\boldsymbol{C}_4$)	Frequency($\boldsymbol{C}_5$)
Weight	0.15	0.15	0.25	0.3	0.15

A weight matrix W is constructed, and the network administrator sets the weights of the evaluation indicators, where each weight is assigned to each evaluation indicator, and the total is 1:

$$\mathrm{W} = \begin{bmatrix} 0.15 & 0 & 0 & 0 & 0 \\ 0 & 0.15 & 0 & 0 & 0 \\ 0 & 0 & 0.25 & 0 & 0 \\ 0 & 0 & 0 & 0.3 & 0 \\ 0 & 0 & 0 & 0 & 0.15 \end{bmatrix}$$

A decision matrix D is constructed, and then normalized to be a normalization matrix D':

$$\text{MCDA\_Candidate} = \begin{bmatrix} D_{11} \cdots D_{1j} \cdots D_{1m} \\ \vdots \quad\quad \ddots \quad\quad \vdots \\ D_{n1} \cdots D_{1j} \cdots D_{nm} \end{bmatrix} \times [W_1 W_2 W_3 W_4 W_5]$$

$$D = \begin{bmatrix} 926.932114095416 & 0.004265394907499999 & 12 & 9.995734605092501 & 47.4 \\ 1078.3434271204517 & 0.004638783719999999 & 16 & 9.99536121628 & 61.2 \\ 463.7951328053706 & 0.0024948649999999998 & 5 & 9.997505134999999 & 21.6 \\ 1117.8385570224605 & 0.0048117749407 & 13 & 9.9951882250593 & 55.6 \\ 1129.3773153600205 & 0.0047363974101 & 11 & 9.995263602589901 & 53.2 \\ 427.9026888159207 & 0.0018840385002000001 & 3 & 9.9981159614998 & 12.6 \end{bmatrix}$$

$$\frac{12}{12+16+5+13+11+3} = 0.190476193$$

$$D' = \begin{bmatrix} 0.18019013 & 0.18682263 & 0.190476193 & 0.16665899 & 0.18839428 \\ 0.20962359 & 0.20317691 & 0.25396825 & 0.16665277 & 0.24324324 \\ 0.09015903 & 0.10927411 & 0.07936508 & 0.16668851 & 0.08585056 \\ 0.21730121 & 0.21075386 & 0.20634921 & 0.16664988 & 0.22098569 \\ 0.21954428 & 0.20745235 & 0.22222222 & 0.16665114 & 0.21144674 \\ 0.08318176 & 0.08252015 & 0.04761905 & 0.1666987 & 0.05007949 \end{bmatrix}$$

The direction normalization matrix G is calculated using the minimization criteria. Finally, the weight matrix W and the direction normalization matrix are comprehensively evaluated to obtain a comprehensive evaluation value. The comprehensive evaluation values are then ranked from largest to smallest.

$$1 - D' = G = \begin{bmatrix} 0.81980987 & 0.81317737 & 0.80952381 & 0.83334101 & 0.81160572 \\ 0.79037641 & 0.79682309 & 0.74603175 & 0.83334723 & 0.75675676 \\ 0.90984097 & 0.89072589 & 0.92063492 & 0.83331149 & 0.91414944 \\ 0.78269879 & 0.78924614 & 0.79365079 & 0.83335012 & 0.77901431 \\ 0.78045572 & 0.79254765 & 0.77777778 & 0.83334886 & 0.78855326 \\ 0.91681824 & 0.91747985 & 0.95238095 & 0.8333013 & 0.9499205 \end{bmatrix}$$

$$0.81980987\text{x}0.15 + 0.81317737\text{x}0.15 + 0.80952381\text{x}0.25 + 0.83334101\text{x}0.3 + 0.81160572\text{x}0.15 = 0.8221559891937644$$

$$S_1 = 0.8221559891937644 \quad S_2 == 0.891133617849319 \quad S_3 = 0.8756477072779868$$
$$S_4 = 0.8063207016645405 \quad S_5 = 0.8072234161938577 \quad S_6 = 0.891133617849319$$

$$S_6 > S_3 > S_1 > S_5 > S_4 > S_2$$

From the above, Solution S_6 is a better problem solution. The onlooker bee selects S_6 to update according to the problem solution rankings. The algorithm stops computing when the number of loops exceeds the maximum number of iterations (Max_Iter), and returns the best solution after computing.

4 Conclusion

This study uses an intelligent approach called the multi-criteria artificial bee colony (MABC) algorithm to optimize the energy consumption problem in wireless sensor networks. The use case shows that the MABC algorithm provides energy-efficient packet routing, extending the lifespan of the wireless sensor network; this is confirmed by the evaluation of the candidate routing using multi-criteria analysis. The contribution of this work is its use of swarm intelligence (SI) algorithms in wireless sensor network routing. The artificial bee colony algorithm based on multi-criteria decision analysis is used in a wireless sensor network to address the fast convergence problem. Future work will compare the performances of other swarm intelligence algorithms in wireless sensor networks with that of the proposed approach.

Acknowledgement. This research was supported in part by the Ministry of Science and Technology, R.O.C. with a MOST grant 107–2221-E-025–005-.

References

1. Kusek, M.: Internet of things: today and tomorrow. In: 2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), pp. 0335–0338. IEEE, May 2018
2. Albreem, M.A., El-Saleh, A.A., Isa, M., Salah, W., Jusoh, M., Azizan, M.M., Ali, A.: Green internet of things (IoT): an overview. In: 2017 IEEE 4th International Conference on Smart Instrumentation, Measurement and Application (ICSIMA), pp. 1–6. IEEE, November 2017
3. Maksimovic, M.: Greening the future: green Internet of Things (G-IoT) as a key technological enabler of sustainable development. In: Internet of Things and Big Data Analytics Toward Next-Generation Intelligence, pp. 283–313. Springer, Cham (2018)
4. Yang, X.-S., Deb, S., Zhao, Y.-X., Fong, S., He, X.: Swarm intelligence: past, present and future. Soft. Comput. **22**(18), 5923–5933 (2017). https://doi.org/10.1007/s00500-017-2810-5
5. Karaboga, D.: An idea based on honey bee swarm for numerical optimization, vol. 200, p. 10. Technical report-tr06, Erciyes university, engineering faculty, computer engineering department (2005)
6. Triantaphyllou, E.: Multi-criteria decision making methods. In: Multi-criteria Decision Making Methods: A Comparative Study, pp. 5–21 (2000)
7. Guo, S., Zhao, H.: Fuzzy best-worst multi-criteria decision-making method and its applications. Knowl.-Based Syst. **121**, 23–31 (2017)
8. Abdullah, L., Adawiyah, C.R.: Simple additive weighting methods of multi criteria decision making and applications: a decade review. Int. J. Inf. Process. Manage. **5**(1), 39 (2014)
9. Zanakis, S.H., Solomon, A., Wishart, N., Dublish, S.: Multi-attribute decision making: a simulation comparison of select methods. Eur. J. Oper. Res. **107**(3), 507–529 (1998)
10. Fauzi, N., Noviarti, T., Muslihudin, M., Irviani, R., Maseleno, A., Pringsewu, S.T.M.I.K.: Optimal dengue endemic region prediction using fuzzy simple additive weighting based algorithm. Int. J. Pure Appl. Math **118**(7), 473–478 (2018)
11. Goodridge, W., Bernard, M., Jordan, R., Rampersad, R.: Intelligent diagnosis of diseases in plants using a hybrid Multi-Criteria decision making technique. Comput. Electron. Agric. **133**, 80–87 (2017)
12. Chithaluru, P., Ravi P., Subodh, S.: WSN structure based on SDN. In: Innovations in Software-Defined Networking and Network Functions Virtualization. IGI Global, pp. 240–253 (2018)
13. Djellali, H., Djebbar, A., Zine, N. G., Azizi, N.: Hybrid artificial bees colony and particle swarm on feature selection. In: IFIP International Conference on Computational Intelligence and Its Applications, pp. 93–105, Springer, Cham (2018)