

Machine Vision Optimization using Nature-Inspired Algorithms to Model Sunagoke Moss Water Status

Yusuf Hendrawan[#], Dimas Firmanda Al Riza

[#] Department of Agricultural Engineering, University of Brawijaya, Malang 65145, Indonesia
E-mail: yusufhendrawan@gmail.com

Abstract— Machine vision has been widely implemented to monitor water status of plants. The performance of machine vision affects the prediction process of plant water status. Therefore optimization is needed to improve the performance of machine vision. The objective of this study is to optimize the performance of machine vision to model *Sunagoke* moss water status. Back Propagation Neural Network was used to model the relationship of image features and *Sunagoke* moss water status. Multi Objective Optimization (MOO) was used to select 212 image features to get maximum prediction accuracy and minimum number of features subset. Nine nature-inspired algorithms for optimization *i.e.* Genetic Algorithms (GAs), Discrete Particle Swarm Optimization (DPSO), Honey Bees Mating Optimization (HBMO), Simulated Annealing (SA), Ant Colony Optimization (ACO), Intelligent Water Drops (IWD), Discrete Firefly Algorithm (DFA), Discrete Hungry Roach Infestation Optimization (DHRIO), and Fish Swarm Intelligent (FSI) were compared. The result shows generally that the prediction model using feature selection techniques achieved significant prediction accuracy, and the number of feature-subset, and was better than the model without feature selection to predict *Sunagoke* moss water status.

Keywords— Feature selection, machine vision, multi objective optimization, nature-inspired algorithms.

I. INTRODUCTION

Use of biological materials for roof greening is an effective strategy for urban heat island mitigation. *Sunagoke* moss, scientifically known as *Rhacomitrium japonicum* is resilient requiring minimal maintenance, is environmentally sound with high resistance to pest and diseases, retains its dark green colour in extreme cold and high temperatures and is drought resistant with lack of rainfall leaving no permanent ill-effects [1]. Thus, water stress is the main factor limiting moss mat production [2][3]. Fig. 1 shows *Sunagoke* moss mat production in a semi-closed bioproduction system. Mosses exhibit a high level of desiccation tolerance [4][5] making them ideal plant for studying the robustness of water stress detection techniques in plants. Plant water stress is caused by water deficit or flooding. Water stress influences stomata resistance, induces changes in internal and surface leaf structure and leads to breakdown of photosynthesis pigments. These changes can be detected by imaging. Imaging techniques make pre-symptomatic detection of physiological changes in plants possible and in real-time. Combinations of colour and textural features have been used in many studies to detect water stress in moss using imaging techniques [6][7][8][9].

In horticulture and agriculture the commonly used colour spaces such as gray, RGB, HSL, HSV and $L^*a^*b^*$, have often been used for plant detection in image analysis [10]. However, no study has reported conclusively about the combination of some other colour spaces such as XYZ, LCH and Luv. There is also no research has been conducted in image analysis which is implementing the combination of colour features and textural features in various colour spaces such as gray, RGB, HSL, HSV, $L^*a^*b^*$, XYZ, LCH and Luv.



Fig.1. *Sunagoke* moss mat production.

Feature selection techniques have become an apparent need in many bioinformatics applications [11]. Two categories of feature selection techniques that are recently used *i.e.* filter methods and wrapper methods. Filter methods are fast but lack robustness against interactions among features and feature redundancy. Wrapper methods are more effective than filter methods because they evaluate the candidate feature subsets using learning algorithm [3]. Wrapper methods can broadly be classified into two categories based on search strategy: greedy and stochastic. Greedy wrapper methods can easily be trapped into local minima [12]. Stochastic wrapper methods such as Simulated Annealing (SA) [13], Genetic Algorithms (GAs) [14], Particle Swarm Optimization (PSO) [15] are at the forefront of research in feature selection [16].

The natural systems have been one of the rich sources of inspiration for developing new intelligent systems. In this study, artificial intelligence approaches using nature-inspired algorithms for optimizing image feature-subset to predict water status of *Sunagoke* moss were proposed. Artificial Neural Network (ANN) is simplified models of the human central nervous system. ANN has been shown to be effective as computational processors for various task including pattern recognition, classification, modelling, forecasting, combinatorial problem solving and noise filtering [17]. GAs is search algorithms based on the mechanics of natural selection and natural genetics [18]. PSO is an evolutionary computation technique inspired in the behaviour of bird flocks which was first introduced by Kennedy and Eberhart [19]. The Honey Bees Mating Optimization (HBMO) algorithm simulates the marriage behaviour of bees [20]. SA takes inspiration from the process of shaping hot metals into stable forms through a gradual cooling process whereby the material transits from a disordered, unstable, high-energy state to an order, stable, low-energy state [21]. Ant Colony Optimization (ACO) is inspired by the foraging behaviour of ant colonies, and targets discrete optimization problems [22]. Intelligent Water Drops (IWD) algorithm was invented by Shah-Hosseini [23] which is based on the dynamic river systems, actions and reactions that happen among the water drops in rivers. Firefly Algorithm (FA) was developed by Xin-She Yang in 2007 based on the flashing characteristics of fireflies [24]. Hungry Roach Infestation Optimization (HRIO) is inspired by recent discoveries in the social behaviour of cockroaches [25]. Fish Swarm Intelligent (FSI) was developed by Fernandes *et al.* [26] which is inspired by the fish swarm behaviour inside water.

The objectives of this study is (1) to compare the water status prediction ability of colour and texture analysis in *Sunagoke* moss; and (2) to compare the performance of nature-inspired algorithms to find the most significant set of image features suitable for predicting water content of cultured *Sunagoke* moss. Multi-Objective Optimization (MOO) is an optimization problem that involves multiple objectives or goals. It is necessary to be aware that feature selection problem in this study is a MOO problem in the sense of prediction accuracy maximization and feature subset size minimization.

A. Materials and equipments

Ten samples of high density and immature-based type of cultured *Sunagoke* moss *R. japonicum* (VARORE Co., Japan) were grown in polyvinyl netting and anchored in glass wool media in growth chamber (Biotron NK 350, Japan) with the optimum environment parameters air temperature = 15 °C, RH = 80%, the CO₂ gas = 400 ppm, light intensity = 86.5 μmol m⁻² s⁻¹, light duration = 12 h. Those samples were placed in a 110 x 80 x 25 mm glass vessel. The average initial dry weight of the samples was 12.5 g. As a mean of manipulating their physiological status, the samples were subjected to different water states. Water status was defined as the average amount of water available for each sample in each day of data acquisition in grams per gram of its initial dry weight [8].

B. Model of Study

First process is image acquisition in a dark chamber, in which the moss images were captured using digital camera (Nikon Coolpix SQ, Japan) placed at 330 mm perpendicular to the sample surface. The image size was 1024 x 768 pixels. Imaging was done under controlled and well distributed light conditions. Light was provided by two 22W lamps (EFD25N/22, National Corporation, Japan). Light intensity over the moss surface was uniform at 100 μmol m⁻² s⁻¹ PPF (Photometer, Li6400, USA) during image acquisition. A total of 649 image data which varies at different water status (dry, semi-dry, wet and soak condition) were acquired. Image features which consist of Colour Features (CFs) and Textural Features (TFs) were extracted from each image data.

Selection process for selecting relevant image features is done using nine alternative nature-inspired approaches *i.e.* GAs, Discrete Particle Swarm Optimization (DPSO), HBMO, SA, ACO, IWD, Discrete Firefly Algorithm (DFA), Discrete Hungry Roach Infestation Optimization (DHRIO) and FSI. MOO concerns optimization problems with multiple objectives [27]. The fitness is calculated as follows:

$$function_1 = weight_1 \times RMSE_{(x)} \quad (1)$$

$$function_2 = weight_2 \times \frac{IF_{(x)}}{f_t} \quad (2)$$

$$fitness(x) = function_1 + function_2 \quad (3)$$

where $RMSE_{(x)}$ is the Root Mean Square Error of validation-set data of Back-Propagation Neural Network (BPNN) using only the expression values of the selected image features in a subset x , where $IF_{(x)}$ is the number of selected image features in x , f_t is the total number of image features, $weight_1$ and $weight_2$ are two priority weights corresponding to the importance of the accuracy and the number of selected image features, respectively, where $weight_1 \in [0.1, 0.9]$ and $weight_2 = 1 - weight_1$. In this study, the accuracy is more important than the number of selected image features in a feature-subset.

C. Colour Features (CFs)

CFs include colour mean value and excess RGB index. Colour mean value can be described as follows [3]:

$$\text{colour mean value} = \frac{1}{M} \sum_{i=1}^M \text{colour value} \quad (4)$$

where: colour value can be defined as the range of each colour space in the pixel *i.e.* red, green, blue, grey, hue, saturation_(HSL), saturation_(HSV), lightness_(HSL), value_(HSV), X_(XYZ), Y_(XYZ), Z_(XYZ), L*, a*, b*, C_(LCH), H_(LCH), u_(Luv) and v_(Luv). *M* is the total number of pixels in the image.

Excess RGB index was calculated by [8]:

$$ER_n = \frac{2R-G-B}{R+G+B}; EG_n = \frac{2G-R-B}{R+G+B}; EB_n = \frac{2B-R-G}{R+G+B} \quad (5)$$

where ER_n , EG_n and EB_n are the normalized excess Red (*R*) index, excess Green (*G*) index and excess Blue (*B*) index, respectively. The total number of CFs are 22 features.

D. Textural Features (TFs)

The textural analysis can be considered as one of applicable techniques for extracting image features [28]. The Colour Co-occurrence Matrix (CCM) procedure consists of three primary mathematical processes: (1) the image is transformed from RGB colour representation to other colour representation such as gray [29], HSL and HSV [30], L*a*b* and XYZ [31], LCH [32] and Luv [33]; (2) generation of Spatial Gray-Level Dependence Matrices (SGDMs) [34], resulting in one CCM for each colour space, the CCM was calculated based on normalization value; and (3) determination of ten Haralick Textural Features [35].

Ten Haralick's TFs are as follows:

$$\text{Energy} = \sum_i^M \sum_j^N P^2[i, j] \quad (6)$$

$$\text{Entropy} = -\sum_i^M \sum_j^N P[i, j] \log P[i, j] \quad (7)$$

$$\text{Contrast} = \sum_i^M \sum_j^N (i-j)^2 P[i, j] \quad (8)$$

$$\text{Homogeneity} = \sum_i^M \sum_j^N \frac{P[i, j]}{1+|i-j|} \quad (9)$$

$$\text{Inverse Difference Moment} = \sum_i^M \sum_j^N \frac{P[i, j]}{|i-j|^k} \quad i \neq j \quad (10)$$

$$\text{Correlation} = \sum_i^M \sum_j^N \frac{(i-\mu)(j-\mu)P[i, j]}{\sigma^2} \quad (11)$$

$$\text{Sum Mean} = \frac{1}{2} \sum_i^M \sum_j^N (iP[i, j] + jP[i, j]) \quad (12)$$

$$\text{Variance} = \frac{1}{2} \sum_i^M \sum_j^N ((i-\mu)^2 P[i, j] + (j-\mu)^2 P[i, j]) \quad (13)$$

$$\text{Cluster Tendency} = \sum_i^M \sum_j^N (i+j-2\mu)^k P[i, j] \quad (14)$$

$$\text{Maximum Probability} = \text{Max}_{i,j} P[i, j] \quad (15)$$

where: $P(i,j)$ is the $(i,j)^{\text{th}}$ element of a normalized co-occurrence matrix, and μ and σ are the mean and standard deviation of the pixel element given by the following relationships:

$$P[i, j] = \frac{N(i, j)}{M} \quad (16)$$

$$\mu = \sum_i^M i \sum_j^N P[i, j] \quad (17)$$

$$\sigma = \sum_i^M (i-\mu)^2 \sum_j^N P[i, j] \quad (18)$$

where: $N(i,j)$ is the number counts in the image with pixel intensity *i* followed by pixel intensity *j* at one pixel displacement to the left, and *M* is the total number of pixels.

Based on the results of preliminary observation in various combination of angle ($\theta = 0, \theta = 45, \theta = 90, \theta = 135$) and distance ($d = 1, d = 2, d = 3$), it was showed that combination of angle ($\theta = 0$) and distance ($d = 2$) performed better than the other combination of θ and d to identify water content. Therefore, in this study, TFs were extracted at those values of θ and d . A total of 190 TFs were extracted *i.e.* 10 TFs each for R, G, B, gray, hue, saturation_(HSL), saturation_(HSV), lightness_(HSL), value_(HSV), X_(XYZ), Y_(XYZ), Z_(XYZ), L*, a*, b*, C_(LCH), H_(LCH), u_(Luv) and v_(Luv). Therefore, the total image features (f_i) which were extracted from both CFs and TFs are 212 features.

E. Back-Propagation Neural Network (BPNN)

A three layers BPNN structure which consists of input layer, one hidden layer and output layer has been developed for predicting *Sunagoke* moss water status. Learning rate and momentum value were chosen at 0.6 and 0.8, respectively based on the results of preliminary runs. Five models of hidden nodes architecture in the hidden layer were developed *i.e.* 10, 15, 20, 25 and 30. The output was water content corresponding to the input features. The training, validation and testing performance criterion for the prediction was RMSE as follows:

$$\text{RMSE} = \sqrt{\frac{1}{N_n} \sum_{i=1}^{N_n} (S_i - St_i)^2} \quad (19)$$

where N_n is number of input feature vectors, S_i is the water content predicted by BPNN model, and St_i is the target water content. The 649 samples data were randomized and divided into three parts which were 325 data as training-set, 162 data as validation-set and 162 data as testing-set. The best number of hidden nodes was determined by validation-set RMSE. The best hidden nodes will be used to calculate testing-set RMSE. The procedure of training, validation and testing included: (1) divide data into training, validation and testing-set; (2) select certain hidden nodes model; (3) train selected hidden nodes model using the training-set; (4) evaluate selected hidden nodes model using the validation-set; (5) repeat steps 2 through 4 using different hidden nodes model; (6) select the best hidden nodes model; and (7) assess this final model using the testing-set. The iteration in the training process will stop when the error will reach to a threshold value.

F. Genetic Algorithms (GAs)

The steps of N-GAs are as follows [36][37]:

1. Generate population randomly in which individuals (number of individuals $ni = 70$) characterized by chromosomes represent a set of possible solutions (e.g. $ga_j: 0,1,1,0,0,0,1,0,0,1,0,0,1,0, \dots, f_i$), where f_i is the number of

total image features which equals to 212 features. A value of 0 indicates that the corresponding feature is not selected and will not be added as the input of BPNN, while a value of 1 means that the feature is selected and will be added as the input of BPNN.

2. Compute the fitness of each individual in the population using Eqs. (3).
3. Select the fittest individuals to be parents for reproducing offspring using roulette wheel selection strategy.
4. Create offspring with two point crossover (crossover rate = 0.5) and mutation operators (mutation rate = 0.1) by changing the selected individuals during the mating periods. Crossover rate and mutation rate were determined based on the results of preliminary runs. Two points crossover (point1 and point2) are selected randomly, where $\text{point1} < \text{point2}$, $\text{point1} > 1$ and $\text{point2} < f_i$.
5. Displace the parents with good offspring to compose the subsequent generation according to probability best chromosome which is set to 0.2.
6. The search will terminate if the iteration has reached 500 iterations.

G. Discrete Particle Swarm Optimization (DPSO)

Pan *et al.* [38] have presented a DPSO optimization algorithm to tackle the discrete spaces which can not be solved by PSO, the steps of DPSO are as follows [39]:

1. Generate a population of particles, $pso^n = [pso_1^n, pso_2^n, \dots, pso_{np}^n]$ where np is the number of particles ($np = 70$) and n is global iteration ($n = 500$). Each particle in the swarm population has the following attributes: a current position represented as $pso_i^n = [pso_{i1}^n, pso_{i2}^n, \dots, pso_{im}^n]$ e.g. $pso_1^n = 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, \dots, f_i$; a current personal best position represented as $p_i^n = [p_{i1}^n, p_{i2}^n, \dots, p_{im}^n]$; and a current global best position represented as $g_i^n = [g_{i1}^n, g_{i2}^n, \dots, g_{im}^n]$.
2. Evaluate the fitness of each particle in the population using Eqs. (3).
3. Find personal best position. The personal best position of each particle is updated using:

$$p_i^n = \begin{cases} p_i^{n-1} & \text{if } F(pso_i^n) \geq F(p_i^{n-1}) \\ pso_i^n & \text{if } F(pso_i^n) < F(p_i^{n-1}) \end{cases} \quad (20)$$

4. Find global best position.

$$g^n = \begin{cases} \arg \min_{p_i^n} f(p_i^n) & \text{if } \min F(p_i^n) < F(g^{n-1}) \\ g^{n-1} & \text{else} \end{cases} \quad (21)$$

5. Update particles of population.

$$pso_i^n = c_2 \oplus CR(c_1 \oplus CR(w \oplus F_\rho(pso_i^{n-1}), p_i^{n-1}), g^{n-1}) \quad (22)$$

The update particles of population consists of three components: The first component is $a_i^n = w \oplus F_\rho(pso_i^{n-1})$, which represents the velocity of the particle. F_ρ represents the mutation operator with the mutation strength of ρ and the mutation probability of w ($w = 0.5$). The second component is $b_i^n = c_1 \oplus CR(a_i^n, p_i^{n-1})$, which is the cognition part of the particle representing the private thinking of the particle itself. CR represents the crossover operator between a_i^n and p_i^{n-1} with the probability of c_1 [0, 1]. The third component is

$pso_i^n = c_2 \oplus CR(b_i^n, g^n)$, which is the social part of the particle representing the collaboration among particles. CR represents the crossover operator between b_i^n and g^n with the probability of c_2 [0, 1]. Probability of c_1 and c_2 were chosen at 0.8 based on the results of preliminary runs. Here crossover is performed by two points crossover.

6. The search will terminate if the global iteration n has been reached.

H. Honey Bees Mating Optimization (HBMO)

The steps of proposed HBMO are given as follows:

1. Initialisation of HBMO parameters. The maximum iteration is 500. The number of bee's population is 70, the capacity of spermatheca is 50 and the number of worker is 40. The initial *speed* and *energy* of the queen are 0.9 and 1, respectively based on the results of preliminary runs.
2. Generate the initial value of worker randomly [0, 1].
3. Generate the initial population of bees randomly (e.g. $bee_i: 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, \dots, f_i$).
4. Evaluate the fitness of bee_i using Eqs. (3).
5. Compute the individual solution $F(bee_i)$.
6. Based on the individual solution $F(bee_i)$, only in the first iteration, the best member of the initial population of bees is selected as the queen of the hive. The number of queen is one. All, the other members of the population are the drones.
7. Selection of *drone*. A drone mates with a queen using annealing function [40]:

$$Prob(D) = e^{\left[\frac{-1 \times |F(bee_i) - F(queen)|}{speed(t)} \right]} \quad (23)$$

where $Prob(D)$ is the probability of adding the sperm of drone D to the spermatheca of the queen, $|F(bee_i) - F(queen)|$ is the absolute difference between the fitness of D and the fitness of the queen and $speed(t)$ is the speed of the queen at time t .

8. Add sperm of the drone to queen's spermatheca.
9. Update *speed* and *energy* of the queen.

$$speed(t+1) = \alpha \times speed(t) \quad (24)$$

$$energy(t+1) = \alpha \times energy(t) \quad (25)$$

where α is a factor [0, 1] that determines the amount that the speed and the energy will be reduced after each transition and each step.

10. Selection of the worker. *worker* is randomly selected from the list. Random number: $rnd[0, 1]$ is generated to determine the probability of worker to do crossover function (between the queen genotype and the selected sperm) or mutation function (selected sperm) to the brood. Broods are generated from the cross-over and mutation process.
11. Evaluate the fitness of $brood_i$ using Eqs. (3).
12. Compute the individual solution $F(brood_i)$.
13. Replace the queen and update the fitness of the queen if the solution of the brood is better than the solution of the current queen.

$$F(queen) \begin{cases} F(brood) & \text{if } q(F_{(brood)}) > q(F_{(queen)}) \\ F(queen) & \text{otherwise} \end{cases} \quad (26)$$

where function $q(\cdot)$ gives the quality of the solution.

14. Update the total best solution T^{TB} :

$$T^{TB} \begin{cases} T^{TB} & \text{if } q(T^{TB}) \geq q(T^{IB}) \\ T^{IB} & \text{otherwise} \end{cases} \quad (27)$$

15. The search will terminate if the maximum iteration has been reached.

I. Simulated Annealing (SA)

The N-SA steps are as follows [39]:

1. Set the maximum iteration ($n = 70$), set the initial value of temperature (T) and randomly generate an initial solution of feature-subset (sa). Set this solution as the current solution as well as the best solution. The independent variable sa in the SA procedure is set to subset of features which is determined by random value $[0, 1]$, e.g. $sa: 0,1,1,0,0,0,0,1,0,0,1,0, \dots, f_i$.
2. Evaluate the fitness $F(sa)$ using fitness function in Eqs. (3).
3. Generate another feature-subset (sa').
4. Evaluate fitness $F(sa')$. If sa' improves on sa , it is accepted; if sa' is worse than sa , then sa' is accepted with a probability which depends on the difference in objective function value $F(sa) - F(sa')$, and on a parameter T . T is lowered during the run of the algorithm, reducing in this way the probability of accepting solutions worse than the current one. The probability p_{accept} to accept a solution sa' is often defined according to the Metropolis distribution [41]:
$$p_{accept}(sa, sa', T) = \begin{cases} 1, & \text{if } F(sa') < F(sa) \\ \exp\left(-\frac{F(sa) - F(sa')}{T}\right), & \text{otherwise} \end{cases} \quad (28)$$
5. Update Fsa' , global minimum and memorize sa' .
6. Update temperature. The decrement function for decreasing the value of T is given by $T = T - (\alpha_n * T)$, where α_n is an anneal factor which can be defined as a constant smaller than but close to 1. Typical values lie between 0.8 and 0.99. Based on the results of preliminary runs using some variations of T value, we determine the best α_n is 0.9.
7. The search will terminate if the iteration has reached 500 iterations or the current solution does not change for more than 300 iterations.

J. Ant Colony Optimization (ACO)

The steps of N-ACO [35]:

1. Set the initial parameters *i.e.* the number of ant population ($ant_1, ant_2, ant_3, \dots, ant_{na}$) is $na = 70$; maximum of allowed iterations = 500; heuristic ($\eta_{i\psi}$) which is defined as the inverse of the validation-set RMSE between two features (i, ψ) as the input of BPNN; intensity of pheromone trail level ($\tau = 100$); the best ants selected ($k = 8$); pheromone constant ($pc = 1$); heuristic constant ($hc = 1$); and evaporation rate of pheromone $er_\rho [0, 1]$. er_ρ was determined to be 0.2 through preliminary runs.
2. Generating ants for solution generation. Each ant movement for finding the trail path is based on the pheromone and heuristic probability.

$$p_{i\psi}^{ant} = \frac{\tau_{i\psi}^{pc} \times \eta_{i\psi}^{hc}}{\sum_{ant \in allowed_{ant}} (\tau_{i\psi}^{pc} \times \eta_{i\psi}^{hc})} \quad (29)$$

If an ant is not able to optimize fitness function in ten successive steps, it will finish its work and exit. Each ant consist of feature-subset with selected features as ant paths (e.g. $ant_i: 0,1,1,0,0,0,0,1,0,0,1,0, \dots, f_i$).

3. Evaluation of ants (ant) using Eqs. (3).
4. Update the global best solution (T^{best}) by the current ant solution (T^{ant}).

$$T^{best} \begin{cases} T^{best} & \text{if } q(T^{best}) \geq q(T^{ant}) \\ T^{ant} & \text{otherwise} \end{cases} \quad (30)$$

5. Pheromone updating.

$$\tau_{i\psi}(t+n) = (1-\rho)\tau_{i\psi}(t) + \sum_{ant=1}^k \Delta\tau_{i\psi} \quad (31)$$

where $\Delta\tau_{i\psi}$ represents the sum of the contributions of all ants that used move (i, ψ) to construct their solution between time t and $t+1$. Using the feature subsets of the best k ants, the pheromone trails intensity are updated using the following equation:

For $j = 1$ to k

$$\Delta\tau_{i\psi} = \frac{\max_{g=1:k}(\text{fitness value}_g) - \text{fitness value}_j}{\max_{h=1:k} \left(\max_{g=1:k}(\text{fitness value}_g) - \text{fitness value}_h \right)} \quad (32)$$

In the first iteration, each ant will randomly choose a feature subset of f_i features. Only the best k subsets, $k < na$, will be used to update the pheromone trail and influence the feature subsets of the next iteration.

6. Generation of new ants.
7. The search will terminate if the maximum iteration has been reached.

K. Intelligent Water Drops (IWD)

The steps of the proposed N-IWD are as follows [42]:

1. Initialisation of static parameters. Set the maximum iteration (global iteration = 500). The number of IWD population ($IWD_1, IWD_2, IWD_3, \dots, IWD_n$) is = 70. Each IWD consist of feature-subset with selected features as the river paths (e.g. $IWD_1: 0,1,1,0,0,0,0,1,0,0,1,0, \dots, f_i$). For velocity updating, the parameters are $a_v = 1$, $b_v = 0.01$ and $c_v = 1$. For soil updating, $a_s = 1$, $b_s = 0.01$ and $c_s = 1$. The local soil updating parameter ρ_n and the global soil updating parameter ρ_{IWD} were chosen at 0.3 based on the results of the preliminary runs. The initial soil on each path is denoted by the constant $InitSoil$ such that the soil of the path between every two nodes i and j is set by $soil(i, j) = InitSoil$. $InitSoil = 10000$ and $InitVel = 4$.
2. Initialisation of dynamic parameters. Every IWD has a visited node list V_c (IWD). Each IWD's velocity is set to $InitVel$. All IWD are set to have zero amount of soil.
3. Spread the IWDs randomly on the nodes as their first visited nodes.
4. Update the visited node list of each IWD to include the nodes just visited.

5. Repeat steps 5.1 to 5.4 for those IWDs with partial solutions.

5.1 For the IWD residing in node i , choose the next node j , which is not in the visited node list $V_c(IWD)$ if the IWD, using the following probability $p_i^{IWD}(j)$:

$$p_i^{IWD}(j) = \frac{f(soil(i, j))}{\sum_{k \notin V_c(IWD)} f(soil(i, k))} \quad (33)$$

such that

$$f(soil(i, j)) = \frac{1}{\epsilon_s + g(soil(i, j))} \quad (34)$$

$$g(soil(i, j)) = \begin{cases} soil(i, j) & \text{if } \min_{l \in V_c(IWD)} (soil(i, l)) \geq 0 \\ soil(i, j) - \min_{l \in V_c(IWD)} (soil(i, l)) & \text{else} \end{cases} \quad (35)$$

Then, add the newly visited node j to the list $V_c(IWD)$.

5.2. For each IWD moving from node i to node j , update its velocity $vel^{IWD}(t)$ by

$$vel^{IWD}(t+1) = vel^{IWD}(t) + \frac{a_v}{b_v + c_v \times soil^2(i, j)} \quad (36)$$

where $vel^{IWD}(t+1)$ is the updated velocity of the IWD.

5.3. For the IWD moving on the path from node i to j , compute the soil $\Delta soil(i, j)$ that the IWD loads from the path by

$$\Delta soil(i, j) = \frac{a_s}{b_s + c_s \times time^2(i, j; vel^{IWD}(t+1))} \quad (37)$$

such that

$$time(i, j; vel^{IWD}(t+1)) = \frac{HUD(i, j)}{vel^{IWD}(t+1)} \quad (38)$$

where the heuristic undesirability $HUD(i, j)$ is defined as the validation-set RMSE of two nodes (i, j) . The lower validation-set RMSE value between two nodes (i, j) , the lower $HUD(i, j)$ value.

5.4. Update the soil $soil(i, j)$ of the path from node i to j traversed by that IWD and also update the soil that the IWD carries $soil^{IWD}$ by

$$soil(i, j) = (1 - \rho_n) \times soil(i, j) - \rho_n \times \Delta soil(i, j) \quad (39)$$

$$soil^{IWD} = soil^{IWD} + \Delta soil(i, j) \quad (40)$$

6. Evaluate each IWD's solution (T^{IWD}) using the fitness function in Eqs. (3).

7. Find the iteration-best solution T^{IB} from all the solutions T^{IWD} found by the IWDs using

$$T^{IB} = \arg \max_{\forall T^{IWD}} q(T^{IWD}) \quad (41)$$

8. Update the soils on the paths that form the current iteration-best solution T^{IB} by

$$soil(i, j) = (1 + \rho_{IWD}) \times soil(i, j) - \rho_{IWD} \times \frac{1}{(N_{IB} - 1)} \times soil^{IWD} \quad \forall (i, j) \in T^{IB} \quad (42)$$

where N_{IB} is the number of nodes (selected features) in the solution T^{IB} .

9. Update the total best solution T^{TB} by the current iteration-best solution T^{IB} using

$$T^{TB} \begin{cases} T^{IB} & \text{if } q(T^{IB}) \geq q(T^{TB}) \\ T^{TB} & \text{otherwise} \end{cases} \quad (43)$$

10. The search will terminate if the global iteration has been reached.

L. Discrete Firefly Algorithm (DFA)

The steps of the proposed DFA are given as follows:

1. Set the initial parameters of DFA *i.e.* the number of fireflies population ($fa_1, fa_2, fa_3, \dots, fa_{nf}$) is $nf = 70$ and the maximum iteration (global iteration = 500).
2. Generate the location of each firefly randomly $[0, 1]$ (e.g. $fa_i: 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, \dots, f_i$).
3. Determine light intensity I_i at fa_i as $F(fa_i)$ based on the fitness function in Eqs. (3).
4. Rank the fireflies from the worst to the best and find the current best solution (fa_{best}).
5. Repeat steps 5.1 to 5.4 for those fa_i with partial solutions.
 - 5.1 Search over all the dimensions to find other firefly with brighter light intensity (fa_j).
 - 5.2 If I_j is brighter than I_i or we can say if the solution of $F(fa_j)$ is better than the solution of $F(fa_i)$ then there will be two point crossover between firefly fa_j and firefly fa_i . The output of this crossover process is a new location ($fa_{(i,j)}$).
 - 5.3 If the new solution $F(fa_{(i,j)})$ is better than the previous solution $F(fa_i)$ then replace the current location of fa_i with $fa_{(i,j)}$ and update the current individual solution.
 - 5.4 In case of fa_i can not find any brighter firefly then it will move randomly. The output of this process is fa_{rnd} . If $F(fa_{rnd})$ is better than the previous solution $F(fa_i)$ then replace the current location of fa_i with fa_{rnd} and update the current individual solution.
6. Update global best solution as the best feature-subset from the fireflies' population.
7. The search will terminate if the global iteration has been reached.

M. Discrete Hungry Roach Infestation Optimization (DHRIO)

The steps of N-DHRIO are as follows [43]:

1. Initialisation of DHRIO parameters. The maximum iteration ($t_{max}=500$), the number of roach population (N_a) is = 70. For neighbours updating, the parameters are $A_1 = 0.49$, $A_2 = 0.63$ and $A_3 = 0.65$. For hunger updating, $t_{hunger} = 100$. The probability of mutation is set ($w = 0.5$) and the probability of crossover is set ($C_o = 0.5$) based on the results of preliminary runs.
2. Generate roach location (rio_i) randomly and $hunger_i = rand\{0, t_{hunger}-1\}$. Each roach consist of feature-subset (e.g. $rio_i: 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, \dots, f_i$).
3. Evaluate the fitness of each roach (rio_i) using Eqs. (3).
4. Update the individual solution $F(rio_i)$.
5. Calculate neighbours threshold value (d_g):

$$M = [M_{jk}] = \frac{\|F(rio_j) - F(rio_k)\|}{2} \quad (44)$$

$$d_g = median\{M_{jk} \in M : 1 \leq j < k \leq N_a\} \quad (45)$$

6. Repeat steps 6.1 to 6.4 for those rio_i with partial solutions.

6.1 Updating personal best solution (p_i) for the individual cockroach agent:

$$p_i \begin{cases} p_i = rio_i & \text{if } F(rio_i) < F(p_i) \\ p_i & \text{otherwise} \end{cases} \quad (46)$$

6.2 Compute the neighbours (N_i) of roach i .

For $h = 1$ to N_a

$$N_i \begin{cases} N_i = N_i + 1 & \text{if } h : 1 \leq h \leq N_a, h \neq i \text{ AND } M_{ih} < d_g \\ N_i & \text{otherwise} \end{cases} \quad (47)$$

6.3 Update the darkest local location or group best solution (l_i) according to:

For $r = 1$ to N_i

$$l_i \begin{cases} l_i = l_j, = \arg \min_h \{F(p_h)\}, h = \{i, j, r\} & \text{if } \text{rand}[0,1] < A_{\min\{N_i,3\}} \\ l_i & \text{otherwise} \end{cases} \quad (48)$$

where $\{i, j\}$ are the indices of the two socializing cockroaches and p_h is the darkest known location for the individual cockroach agent personal best.

6.4 Update roach location (rio_i):

$$rio_i \begin{cases} rio_i = C_o \oplus CR(C_o \oplus CR(w \oplus F_p(rio_i), p_i), l_i) & \text{if } \text{hunger}_i < t_{\text{hunger}} \\ rio_i = \text{random} & \text{otherwise} \end{cases} \quad (49)$$

6.5 Evaluate each roach (rio_i) using Eqs. (3).

6.6 Update the individual solution $F(rio_i)$.

6.7 Update hunger_i :

$$\text{hunger}_i = \text{hunger}_i + \text{rand}[0,1] \times t_{\text{hunger}} \quad (50)$$

6.8 Update iteration-best solution T^{IB} .

$$T^{IB} = \arg \max q(F(rio_i)) \quad (51)$$

7. Update the total best solution T^{TB} by the current iteration-best solution T^{IB} using:

$$T^{TB} \begin{cases} T^{TB} & \text{if } q(T^{TB}) \geq q(T^{IB}) \\ T^{IB} & \text{otherwise} \end{cases} \quad (52)$$

8. Update the best feature-subset.

9. The search will terminate if the maximum iteration has been reached.

N. Fish Swarm Intelligent (FSI)

The steps of the proposed FSI are as follows:

1. Initialisation of FSI parameters. The maximum iteration is 500. The number of fish population (N_{fish}) is 70, the crowded parameter (C_p) is 0.3 and the leaping value (leap) is 10 based on the results of preliminary runs.
2. Generate the location of each fish randomly $[0, 1]$ (e.g. fish_i : 0,1,1,0,0,0,0,1,0,0,1,0,.... f_i).
3. Evaluate the fitness of fish_i using Eqs. (3).
4. Update the individual solution $F(\text{fish}_i)$.
5. Find the best solution ($\text{fish}_{\text{best}}$).
6. Calculate visual scope ($\text{vis}_{\text{scope}}$):

$$M = [M_{jk}] = \frac{\|F(\text{fish}_j) - F(\text{fish}_k)\|}{2} \quad (53)$$

$$\text{vis}_{\text{scope}} = \text{median}\{M_{jk} \in M : 1 \leq j < k \leq N_{\text{fish}}\} \quad (54)$$

7. Calculate central point of the population ($\text{cent}_{\text{point}}$):

$$\text{features}_{ft} = \frac{\sum_{i=1}^{N_{\text{fish}}} \text{features}_{(i)}}{N_{\text{fish}}} \quad (55)$$

$$\text{features}_{ft} \begin{cases} 1 & \text{if } \text{features}_{ft} \geq 0.5 \\ 0 & \text{otherwise} \end{cases} \quad (56)$$

e.g. $\text{cent}_{\text{point}}$: 0,0,0,1,1,0,0,1,0,0,.... f_i

8. Repeat steps 8.1 to 8.4 for those fish_i with partial solutions.

8.1 If the visual scope of fish_i is empty ($\text{vis}_{\text{scope}} = 0$) then fish_{i+1} will generate its location randomly. Otherwise it goes to *condition1*.

8.2 *Condition1*: If the visual scope of fish_i is crowded ($\text{vis}_{\text{scope}} > C_p$) then *searching* which means fish_i will generate random location $\text{fish}_{\text{rand}}$. If $F(\text{fish}_{\text{rand}})$ is better than $F(\text{fish}_i)$ then there will be a two points crossover process between $\text{fish}_{\text{rand}}$ and fish_i . Otherwise it goes to *condition2*.

8.3 *Condition2*:

1st *process*: If the $F(\text{cent}_{\text{point}})$ is better than $F(\text{fish}_i)$ then *swarming* which means there will be a two points crossover process between $\text{cent}_{\text{point}}$ and fish_i , but if the $F(\text{cent}_{\text{point}})$ is not better than $F(\text{fish}_i)$ then *searching*. The output of the 1st *process* is fish_{y1} .

2nd *process*: If the $F(\text{fish}_{\text{best}})$ is better than $F(\text{fish}_i)$ then *chasing* which means there will be a two points crossover process between $\text{fish}_{\text{best}}$ and fish_i . But if the $F(\text{fish}_{\text{best}})$ is not better than $F(\text{fish}_i)$ then *searching*. The output of the 2nd *process* is fish_{y2} .

$$\text{fish}_{i+1} \begin{cases} \text{fish}_{y1} & \text{if } q(F(\text{fish}_{y1})) > q(F(\text{fish}_{y2})) \\ \text{fish}_{y2} & \text{otherwise} \end{cases} \quad (57)$$

8.4 Update $\text{vis}_{\text{scope}}$, $\text{cent}_{\text{point}}$ and $\text{fish}_{\text{best}}$.

9. If the $F(\text{fish}_i)$ equal to $F(\text{fish}_{i+1})$ then update the *leap* factor. If the *leap* factor reaches the threshold point ($\text{leap} = 10$), then fish_{i+1} will move randomly.

10. Update the best feature-subset.

11. The search will terminate if the maximum iteration has been reached.

III. RESULTS AND DISCUSSION

The main assumption underlying this study was that changes in the external appearances and surface structure of a plant which is caused by water stress can be detected by visible light imaging techniques. In Fig. 2 typical sample images of *R. japonicum* used in this study are shown. Image Fig. 2a shows dry moss, Fig. 2b semi-dry moss, Fig. 2c wet moss or well watered moss and Fig. 2d soak moss or too much water status inside moss. Images Fig. 2c and Fig. 2d are not distinguishable. Therefore, the application of intelligent approaches is required to solve this problem. BPNN model performance was tested successfully to describe the relationship between *Sunagoke* moss water status and image features. It indicates that colour and texture can be good indicators to predict water content in moss.

Fig. 3 shows the average testing-set RMSE of BPNN model trained using various colour spaces. Based on the average testing-set RMSE, R TFs (RMSE = 1.51×10^{-2}) had the lowest prediction error followed by gray TFs, L* TFs, lightness_(HSL) TFs, value_(HSV) TFs, G TFs, L*a*b* mean value, X_(XYZ) TFs, Luv mean value, hue TFs, v_(Luv) TFs, Y_(XYZ) TFs, b* TFs, Z_(XYZ) TFs, H_(LCH) TFs, LCH mean value, saturation_(HSV) TFs, B TFs, C_(LCH) TFs, u_(Luv) TFs, a* TFs, HSV mean value, Excess RGB index, saturation_(HSL) TFs, HSL mean value, RGB mean value and XYZ mean value in that order, respectively. The v_(Luv) TFs shows the least absolute deviation, which means that v_(Luv) TFs shows the highest consistency and the highest reliability in predicting water content of *Sunagoke* moss. ANOVA analysis shows that the significant level is less than 0.01, which means there

is significant difference between the groups showed in Fig. 3 with a confident level of 99%. The TFs feature-subset in each colour space, for example R TFs, consist of ten Haralick's TFs *i.e.* R energy, R entropy, R contrast, R homogeneity, R inverse difference moment, R correlation, R sum mean, R variance, R cluster tendency and R maximum probability. HSV mean value means the combination of hue mean value, saturation_(HSV) mean value and value_(HSV) mean value. Excess RGB index consists of excess R index, excess G index and excess B index.

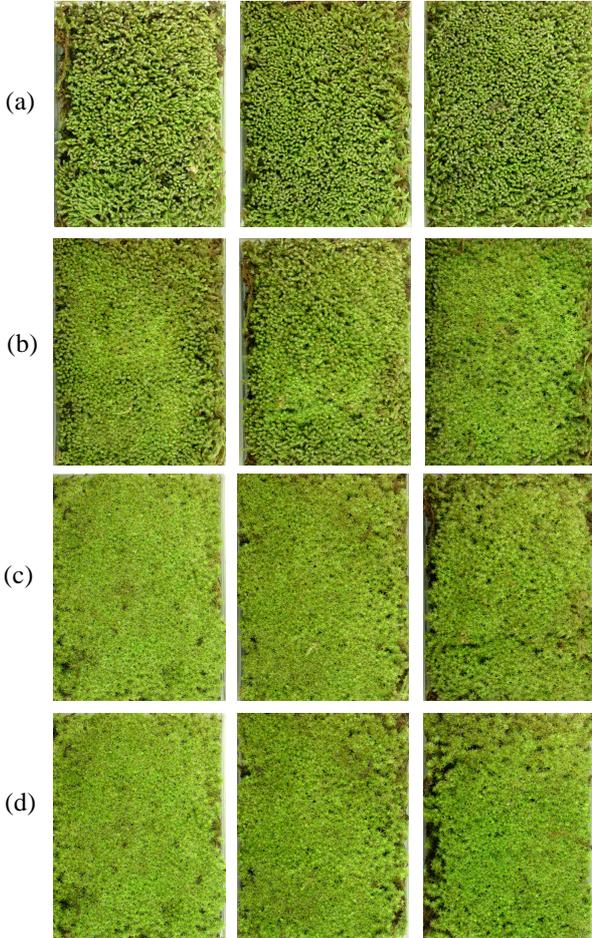


Fig. 2. Cultured *Sunagoke* moss in various water content: (a) dry; (b) semi-dry; (c) wet; (d) soak.

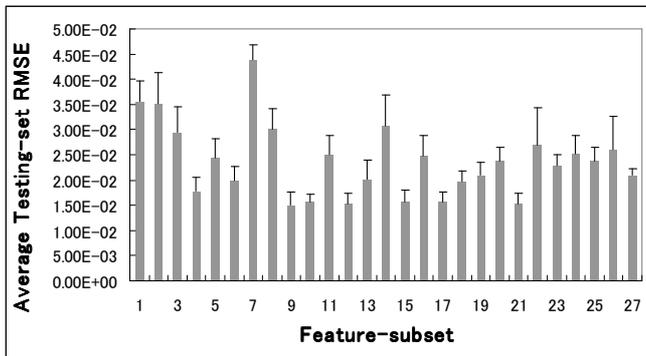


Fig. 3. Performance of BPNN based on: (1) RGB mean value; (2) HSL mean value; (3) HSV mean value; (4) L*a*b* mean value; (5) LCH mean value; (6) Luv mean value; (7) XYZ mean value; (8) Excess RGB index; (9) R TFs; (10) G TFs; (11) B TFs; (12) gray TFs; (13) hue TFs; (14) saturation_(HSL) TFs; (15) lightness_(HSL) TFs; (16) saturation_(HSV) TFs; (17) value_(HSV) TFs; (18) X_(XYZ) TFs; (19) Y_(XYZ) TFs; (20) Z_(XYZ) TFs; (21) L* TFs; (22) a* TFs; (23) b* TFs; (24) C_(LCH) TFs; (25) H_(LCH) TFs; (26) u_(Luv) TFs; (27) v_(Luv) TFs.

The performance of BPNN using individual feature-subset (R TFs) is satisfactory. The performance of BPNN is getting better through all the iterations as shown in Fig. 4. This indicates that the BPNN used for learning algorithm is effective. It means that iteration size of 10000 is appropriate for BPNN to predict water content of *Sunagoke* Moss. The smallest training-set RMSE of BPNN using R TFS is 6.62×10^{-3} . In general, training of BPNN may be terminated at the convergence with the total training-set RMSE value less than 10% [17].

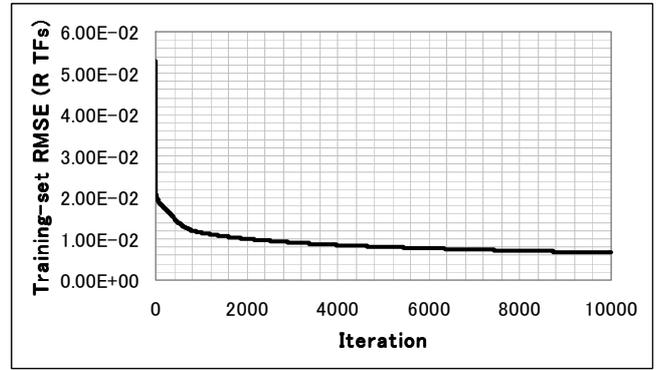


Fig. 4. Performance of BPNN model trained using R TFs.

Table 1 and table 2 show the performance of nine nature-inspired features selection algorithms using different values of $weight_1$ and $weight_2$. A value of the form $x \pm y$ represents an average value x with a standard deviation y . Overall, the prediction accuracy (based on the testing-set data) and the number of selected image features fluctuated because of the diversity of the solutions based on adjusted weights. Moreover, MOO searches simultaneously the solution which is superior in one objective, but poor at others. Based on the objective of the MOO used in this study, where the prediction accuracy is more important than the number of selected image features, the results show that the best performance of each feature selection method are GAs with the average testing-set RMSE of 8.98×10^{-3} and the average feature subset of 46.0 using $weight_1 = 0.9$ and $weight_2 = 0.1$; DPSO with the average testing-set RMSE of 1.32×10^{-2} and the average feature subset of 31.2 using $weight_1 = 0.9$ and $weight_2 = 0.1$; HBMO with the average testing-set RMSE of 9.96×10^{-3} and the average feature subset of 23.2 using $weight_1 = 0.9$ and $weight_2 = 0.1$; SA with the average testing-set RMSE of 1.28×10^{-2} and the average feature subset of 19.0 using $weight_1 = 0.9$ and $weight_2 = 0.1$; ACO with the average testing-set RMSE of 1.24×10^{-2} and the average feature subset of 9.6 using $weight_1 = 0.9$ and $weight_2 = 0.1$; IWD with the average testing-set RMSE of 1.36×10^{-2} and the average feature subset of 9.6 using $weight_1 = 0.9$ and $weight_2 = 0.1$; DFA with the average testing-set RMSE of 9.78×10^{-3} and the average feature subset of 18.2 using $weight_1 = 0.9$ and $weight_2 = 0.1$; DHRIO with the average testing-set RMSE of 1.48×10^{-2} and the average feature subset of 19.6 using $weight_1 = 0.9$ and $weight_2 = 0.1$; and FSI with the average testing-set RMSE of 1.19×10^{-2} and the average feature subset of 18.0 using $weight_1 = 0.9$ and $weight_2 = 0.1$.

TABLE I
AVERAGE TESTING-SET ROOT MEAN SQUARE ERROR (RMSE) AND
AVERAGE FEATURE-SUBSET IN DIFFERENT VALUES OF WEIGHT (w_1 AND w_2)
(5 RUNS ON AVERAGE)

Weight		GAs		DPSO		HBMO	
w_1	w_2	RMSE	features	RMSE	features	RMSE	features
0.1	0.9	2.00E-		2.08E-		2.35E-	
		2±	12.0±	02±	9.8±	02±	3.4±
		5.28E-	0.71	4.22E-	3.03	3.83E-	0.55
		04		04		04	
0.2	0.8	2.12E-		2.10E-		1.83E-	
		02±	11.4±	02±	10.2±	02±	3.6±
		1.68E-	0.89	6.93E-	2.68	7.83E-	0.89
		03		04		04	
0.3	0.7	1.68E-		2.05E-		1.57E-	
		02±	12.8±	02±	10.4±	02±	6.0±
		1.68E-	0.45	1.03E-	2.51	9.59E-	1.41
		03		04		04	
0.4	0.6	1.72E-		2.05E-		1.55E-	
		02±	11.6±	02±	10.8±	02±	6.4±
		7.49E-	0.55	1.58E-	2.68	7.39E-	0.89
		04		04		04	
0.5	0.5	1.57E-		2.05E-		1.40E-	
		02±	14.8±	02±	11.0±	02±	9.0±
		1.17E-	1.79	1.80E-	2.83	1.57E-	1.41
		03		04		03	
0.6	0.4	1.34E-		1.55E-		1.28E-	
		02±	17.0±	02±	20.6±	02±	8.0±
		5.76E-	0.71	2.55E-	3.21	2.41E-	0.71
		04		04		04	
0.7	0.3	1.14E-		1.55E-		1.18E-	
		02±	23.2±	02±	20.8±	02±	11.8±
		8.01E-	0.45	2.89E-	3.90	1.06E-	1.30
		04		04		03	
0.8	0.2	1.07E-		1.47E-		1.17E-	
		02±	28.8±	02±	25.0±	02±	14.6±
		2.12E-	1.10	1.37E-	3.32	1.79E-	0.89
		04		03		04	
0.9	0.1	8.98E-		1.32E-		9.96E-	
		03±	46.0±	02±	31.2±	03±	23.2±
		7.55E-	1.41	4.31E-	3.49	7.69E-	1.30
		04		04		04	

TABLE II
AVERAGE TESTING-SET ROOT MEAN SQUARE ERROR (RMSE) AND
AVERAGE FEATURE-SUBSET IN DIFFERENT VALUES OF WEIGHT (w_1 AND w_2)
(5 RUNS ON AVERAGE)

Weight		SA		ACO		IWD	
w_1	w_2	RMSE	features	RMSE	features	RMSE	features
0.1	0.9	2.72E-		1.53E-		1.59E-	
		02±	0.45	02±	0.89	02±	0.89
		2.96E-		5.71E-		8.24E-	
		03		04		04	
0.2	0.8	2.41E-		1.54E-		1.59E-	
		02±	7.6±	02±	4.2±	02±	5.6±
		3.88E-	1.14	4.53E-	0.45	8.09E-	0.89
		03		04		04	
0.3	0.7	1.88E-		1.54E-		1.62E-	
		02±	7.8±	02±	4.4±	02±	6.2±
		1.60E-	1.30	4.34E-	0.89	1.07E-	1.30
		03		04		03	
0.4	0.6	1.67E-		1.55E-		1.64E-	
		02±	9.6±	02±	5.0±	02±	6.2±
		2.17E-	0.89	6.38E-	0.71	5.56E-	1.30
		04		04		04	
0.5	0.5	1.80E-		1.60E-		1.61E-	
		02±	9.4±	02±	5.8±	02±	7.0±
		1.27E-	1.95	3.79E-	1.09	9.87E-	1.73
		03		04		04	
0.6	0.4	1.59E-	12.2±	1.47E-	5.6±	1.65E-	7.4±

		02±	0.45	02±	0.89	02±	2.30
		2.68E-		1.42E-		5.37E-	
		04		03		04	
0.7	0.3	1.39E-		1.46E-		1.62E-	
		02±	16.8±	02±	6.4±	02±	7.6±
		5.21E-	1.79	1.34E-	1.34	1.39E-	2.19
		04		03		04	
0.8	0.2	1.39E-		1.38E-		1.41E-	
		02±	17.0±	02±	6.6±	02±	8.2±
		5.04E-	2.12	7.61E-	0.89	1.07E-	1.79
		04		04		03	
0.9	0.1	1.28E-		1.24E-		1.36E-	
		02±	19.0±	02±	9.6±	02±	9.6±
		2.71E-	1.41	1.92E-	0.89	4.46E-	1.52
		04		04		04	

Comparative studies was conducted to see how good feature selection effect on the prediction performance of water status using image features. The results show that feature selection models using GAs, DPSO, HBMO, SA, ACO, IWD, DFA, DHRIO and FSI have better performance for minimizing prediction error than model using individual feature-subset or without feature selection method (R TFs). Based on *t-test* statistical analysis, there is significant difference between feature-subset using GAs and feature-subset without feature selection (R TFs) at $\alpha = 0.01$ significant level, and there is significant different between feature-subsets obtained from other feature selection methods (DPSO, HBMO, SA, ACO, DFA, FSI) and R TFs at $\alpha = 0.05$ significant level. However, though the average testing-set RMSE of IWD or DHRIO is better than R TFs, but there is no significant different between feature-subsets obtained from IWD or DHRIO and R TFs. Based on this result, we can conclude that feature selection method improves the performance of prediction using BPNN.

The plots of best fitness values of MOO using all feature selection methods are displayed in Fig. 5 to highlight the search process in each feature selection method. At the beginning of the iteration, all feature selection methods (GAs, DPSO, HBMO, SA, ACO, IWD, DFA, DHRIO and FSI) were given the same feature-subset which is defined as the initial feature-subset. The fitness value obtained from the initial feature-subset and then normalized by the value of 1.00. During the optimization process (minimizing the RMSE of validation-set data and minimizing the number of selected features) the fitness value continues to decrease, searching for the most minimum fitness value. Using the same *weight* parameter ($weight_1 = 0.9$ and $weight_2 = 0.1$), it shows that FSI has the best performance to minimize the fitness value (normalized fitness value = 0.45), followed by DFA (normalized fitness value = 0.65), DHRIO (normalized fitness value = 0.67), GAs (normalized fitness value = 0.67), HBMO (normalized fitness value = 0.74), ACO (normalized fitness value = 0.78), SA (normalized fitness value = 0.81), DPSO (normalized fitness value = 0.84) and IWD (normalized fitness value = 0.89) in that order, respectively. Most of all feature selection methods can quickly minimize the fitness value at the beginning of 50 iterations, but based on the comparison analysis on the performance of all feature selection methods, it shows the superiority of FSI to minimize the fitness value in early iterations, followed by DFA, DHRIO, HBMO, ACO, SA, GAs, IWD and DPSO, respectively. However, the performance of feature selection method to predict the water status of *Sunagoke* moss is

determined by the performance of its testing-set data. Based on the testing-set data and the objective of this study to minimize the prediction error, it shows that GAs has the most minimum testing-set RMSE. The best GAs' fitness function converged with the lowest testing-set RMSE of 8.26×10^{-3} when using 45 features. The best DPSO's fitness function converged with the lowest testing-set RMSE of 1.27×10^{-2} when using 25 features. The best HBMO's fitness function converged with the lowest testing-set RMSE of 9.75×10^{-3} when using 24 features. The best SA's fitness function converged with the lowest testing-set RMSE of 1.26×10^{-2} when using 19 features. The best ACO's fitness function converged with the lowest testing-set RMSE of 1.22×10^{-2} when using 10 features. The best IWD's fitness function converged with the lowest testing-set RMSE of 1.33×10^{-2} when using 10 features. The best DFA's fitness function converged with the lowest testing-set RMSE of 9.64×10^{-3} when using 18 features. The best DHRIO's fitness function converged with the lowest testing-set RMSE of 1.46×10^{-2} when using 24 features. The best FSI's fitness function converged with the lowest testing-set RMSE of 1.17×10^{-2} when using 19 features. From Fig. 5, we can see that the fitness value changed and it is getting better through all the iterations. It indicates that GAs, DPSO, HBMO, SA, ACO, IWD, DFA, DHRIO and FSI are effective. In all of the iterations, the validation-set RMSE of all feature selection methods changed most at the beginning of iterations. It means that iteration size of 500 is appropriate for all feature selection methods.

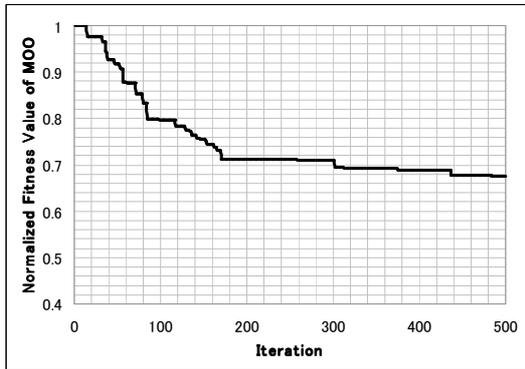


Fig. 5(a)

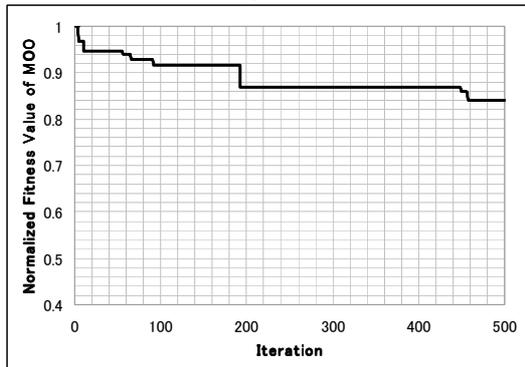


Fig. 5 (b)



Fig. 5 (c)

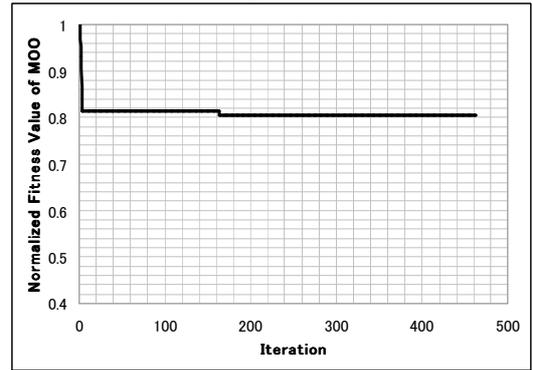


Fig. 5(d)

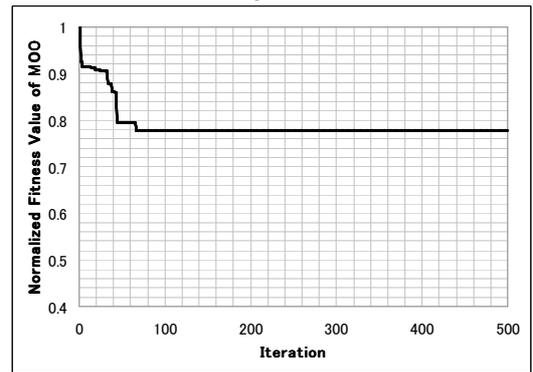


Fig. 5 (e)

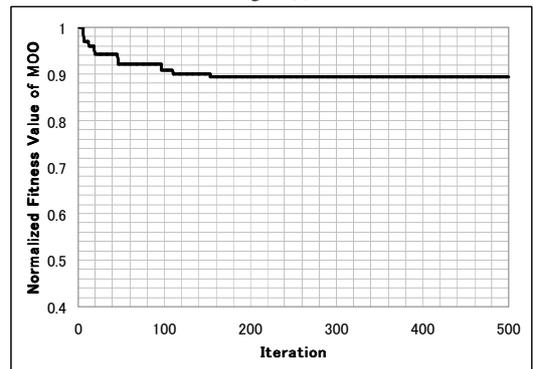


Fig. 5 (f)

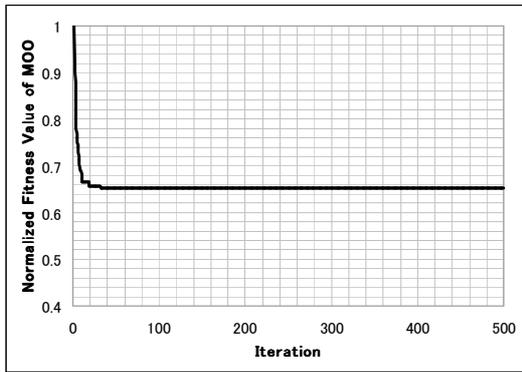


Fig. 5(g)



Fig. 5 (h)

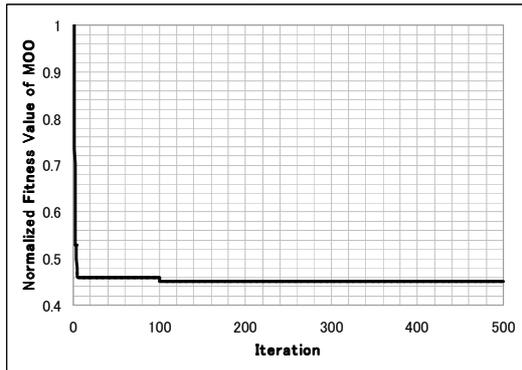


Fig. 5(i)

Fig. 5. Plot of best normalized fitness values of MOO using: (a) GAs; (b) DPSO; (c) HBMO; (d) SA; (e) ACO; (f) IWD; (g) DFA; (h) DHRIO; (i) FSI.

Fig. 6 indicates that the performance of BPNN models using (a) GAs; (b) DPSO; (c) HBMO; (d) SA; (e) ACO; (f) IWD; (g) DFA; (h) DHRIO; (i) FSI are satisfactory. The smallest value for training-set RMSE of all BPNN models (GAs = 6.08×10^{-4} ; DPSO = 1.72×10^{-3} ; HBMO = 1.28×10^{-3} ; SA = 2.13×10^{-3} ; ACO = 6.32×10^{-3} ; IWD = 2.22×10^{-3} ; DFA = 2.03×10^{-3} ; DHRIO = 1.48×10^{-3} ; FSI = 2.38×10^{-3}) are less than 10%. The BPNN performances also indicate that the BPNN used for learning algorithms is effective. Finally, the weights value obtained from the BPNN model using relevant feature-subset can be used to detect water stress in moss. For the future works, the results from this study can be used to develop machine vision-based irrigation system for moss production in closed bio-production system.

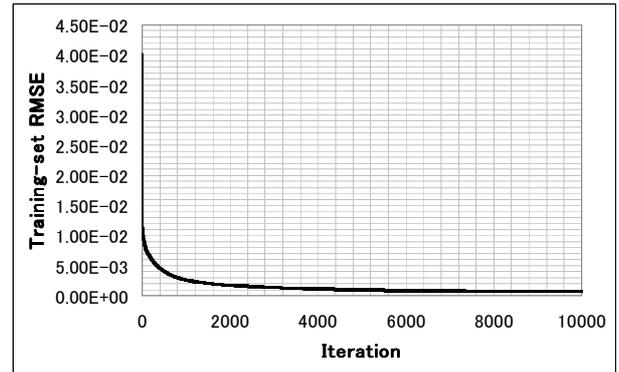


Fig. 6(a)

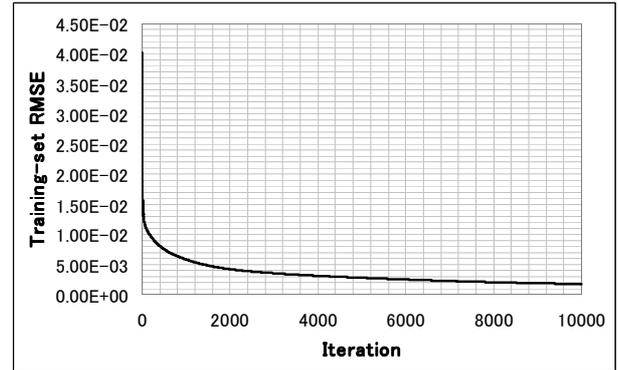


Fig. 6(b)

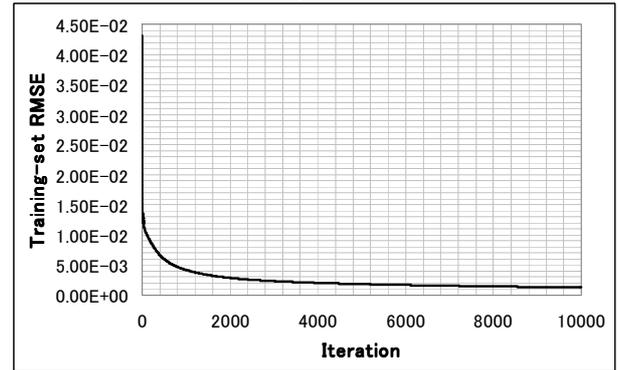


Fig. 6(c)

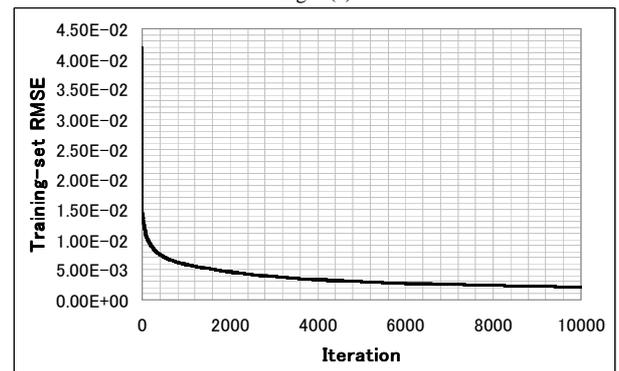


Fig. 6(d)

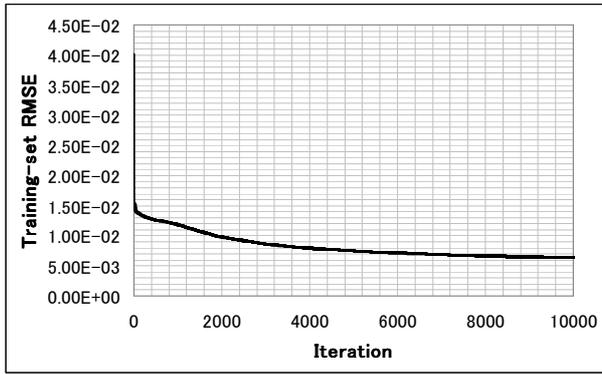


Fig. 6(e)

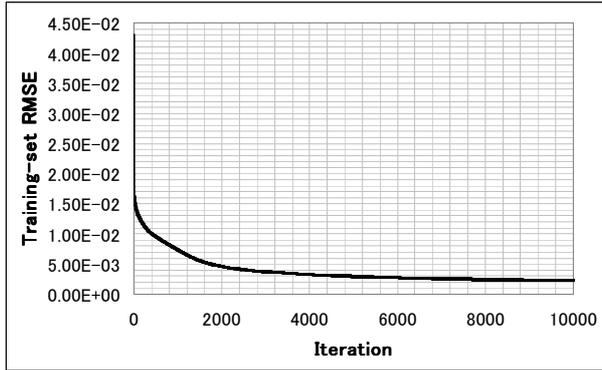


Fig. 6(f)

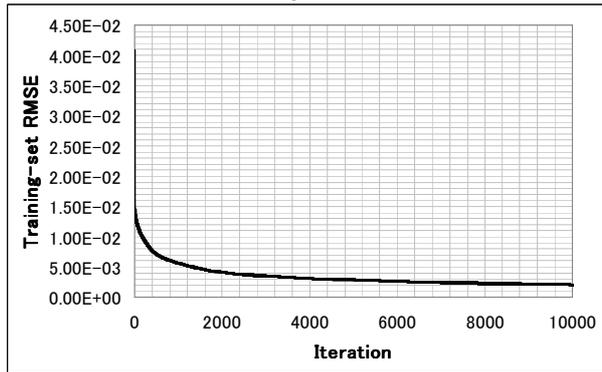


Fig. 6(g)

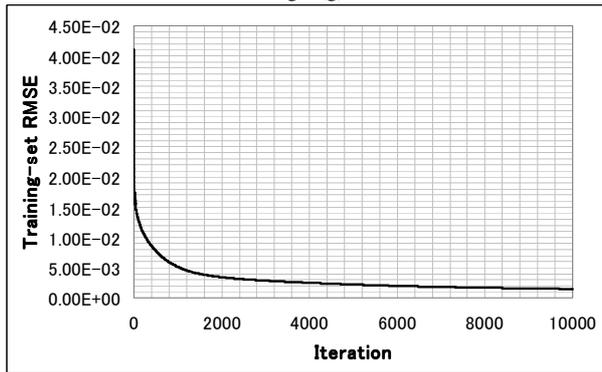


Fig. 6(h)

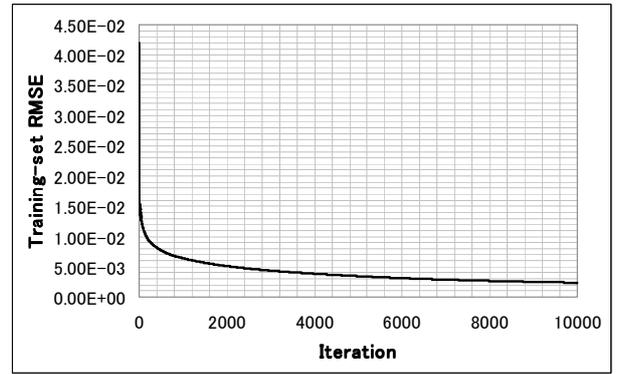


Fig. 6(i)

Fig. 6. Training performance of BPNN: (a) GAs; (b) DPSO; (c) HBMO; (d) SA; (e) ACO; (f) IWD; (g) DFA; (h) DHRIO; (i) FSI.

In this study, we found that the prediction accuracy and the number of selected features using nine feature selection methods were not equal when using different values of $weight_1$ and $weight_2$. This result shows that there are many irrelevant image features, and some of them act negatively on the accuracy acquired by the relevant image features. The weights obtained from the ANN model using relevant image features which has been developed in this study can be applied for water stress detection in cultured *Sunagoke* moss mat production in closed bio-production system. Though, *Sunagoke* moss mat was used in this study, the methods can be extended to other plants or to other purposes for solving feature selection problems.

IV. CONCLUSIONS

In this study, colour and textural features from various colour spaces such as gray, RGB, HSV, HSL, $L^*a^*b^*$, XYZ, LCH and Luv were used to predict water status in *Sunagoke* moss. Back-propagation Neural Network (BPNN) has been tested successfully to describe relationship between image features and water status of cultured *Sunagoke* moss. Based on the testing results, red textural feature-subset had the best performance in prediction using BPNN as individual feature-subset than those extracted from other colour spaces. Feature Selection methods improved the BPNN performance for prediction. Overall, there is a significant difference between methods using feature selection and methods without feature selection. Based on the optimization performance, Fish Swarm Intelligent (FSI) has the best performance for optimizing the fitness function of Multi-Objective Optimization (MOO) problem, followed by Discrete Firefly Algorithm (DFA), Discrete Hungry Roach Infestation Optimization (DHRIO), Genetic Algorithms (GAs), Honey Bees Mating Optimization (HBMO), Ant Colony Optimization (CO), Simulated Annealing (SA), Discrete Particle Swarm Optimization (DPSO) and Intelligent Water Drops (IWD) in that order, respectively. However, in the testing process, BPNN model using feature-subset obtained from GAs has the best prediction accuracy which has a higher ability and reliability to predict water status in *Sunagoke* moss. The best prediction performance using feature-subset obtained from GAs (45 features) has the lowest testing-set Root Mean Square Error (RMSE) of 8.26×10^{-3} .

REFERENCES

- [1] Ondimu, S. S. and Murase, H., 2007. Combining galerkin methods and neural network analysis to inversely determine thermal conductivity of living green roof materials. *Biosyst. Eng.* 96, 541-550.
- [2] Skre, O. and Oechel, W. C., 1981. Moss functioning in different taiga ecosystems in interior Alaska I. Seasonal, phenotypic and drought effects on photosynthesis and response patterns. *Ecologia.* 48, 50-59.
- [3] Hendrawan, Y. and Murase, H., 2009. Precision irrigation for Sunagoke moss production using intelligent image analysis. *Environ. Control Biol.* 47(1), 21-36.
- [4] Aro, E. M., Niemi, H., Valanne, N., 1981. Structural and functional studies on bryophyte photosynthesis. In "Photosynthesis" (ed. By Akoyunoglou, G.). Vol. 3. International Science Services, Balaban Philadelphia, p 327-335.
- [5] Valanne, N., 1984. Photosynthesis and photosynthetic products in mosses. In "The Experimental Biology of Bryophytes" (ed. By Dyer, A.J., Duckett, J.G.), Academic press, London, p 257-273.
- [6] Ushada, M., Murase, H., Fukuda, H., 2007. Non-destructive sensing and its inverse model for canopy parameters using texture analysis and artificial neural network. *Comput. Electron. Agr.* 57(2), 149-165.
- [7] Ondimu, S. S. and Murase, H., 2008a. Water stress detection in Sunagoke moss (*Rhacomitrium canescens*) using combined thermal infrared and visible light imaging techniques. *Biosyst. Eng.* 100(1), 4-13.
- [8] Ondimu, S.S. and Murase, H., 2008b. Comparison of plant water stress detection ability of color and gray-level texture in Sunagoke moss. *T. ASABE.* 51(3), 1111-1120.
- [9] Hendrawan, Y. and Murase, H., 2008. Intelligent irrigation control using color, morphological and textural features in Sunagoke moss. An ASABE Annual International Meeting, Providence, USA, June 29-July 2, paper number: 083858.
- [10] Philipp, I., and Rath, T., 2002. Improving plant discrimination in image processing by use of different colour space transformations. *Comput. Electron. Agr.* 35, 1-5.
- [11] Saeys, Y., Inza, I., Larranaga, P., 2007. A review of feature selection techniques in bioinformatics. *Bioinformatics.* 23(19), 2507-2517.
- [12] Somol, P., Pudil, P., Novovicova, J., Paclik, P., 1999. Adaptive floating search methods in feature selection. *Pattern Recogn. Lett.* 20(11-13), 1157-1163.
- [13] Lin, S. W., Tseng, T. Y., Chou, S. Y., Chen, S. C., 2008. A simulated-annealing-based approach for simultaneous parameter optimization and feature selection of back-propagation networks. *Expert Syst. Appl.* 34, 1491-1499.
- [14] Siedlecki, W. and Sklansky, J., 1989. A note on genetic algorithms for large scale feature selection. *Pattern Recogn. Lett.* 10(5), 335-347.
- [15] Wang, X., Yang, J., Teng, X., Xia, W., Jensen, R., 2007. Feature selection based on rough sets and particle swarm optimization. *Pattern Recogn. Lett.* 28(4), 459-471.
- [16] Gheyas, I. A. and Smith, L. S., 2010. Feature subset selection in large dimensionality domains. *Pattern Recogn.* 43, 5-13.
- [17] Patterson, D. W., 1996. Artificial neural networks: theory and applications. Prentice Hall. Singapore.
- [18] Goldberg, D. E., 1989. Genetic algorithms in search, optimization and machine learning. Addison Wesley Longman, Inc, United States of America.
- [19] Kennedy, J. and Eberhart, R. C., 1995. Particle swarm optimization. Proceedings of the IEEE International Conference on Neural Networks. Vol 4, pp: 1942-1948, ISBN: 0780327683, Perth, Western Australia November 1995, IEEE.
- [20] Marinaki, M., Marinakis, Y., Zopounidis, C., 2010. Honey bees mating optimization algorithm for financial classification problems. *Appl. Soft. Comput.* 10(3), 806-812.
- [21] Floreano, D. and Mattiussi, C., 2008. Bio-Inspired Artificial Intelligence. The MIT Press, Cambridge, Massachusetts, London, England.
- [22] Dorigo, M. and Stutzle, T., 2004. Ant Colony Optimization. A Bradford book, The MIT Press, The United States of America.
- [23] Shah-Hosseini, H., 2009. The intelligent water drops algorithm: a nature-inspired swarm-based optimization algorithm. *International Journal of Bio-Inspired Computation.* 1, Nos. 1/2, 71-79.
- [24] Yang, X.S., 2008. Nature-inspired metaheuristic algorithms. Luniver Press, University of Cambridge, United Kingdom.
- [25] Havens, T. C., Spain, C. J., Salmon, N. G., Keller, M., 2008. Roach Infestation Optimization. IEEE Swarm Intelligence Symposium. St. Louis MO USA, September 21-23.
- [26] Fernandes, E.M.G.P., Martins, T.F.M.C., Rocha, A.M.A.C., 2009. Fish swarm intelligent algorithm for bound constrained global optimization. In the proceedings of the International Conference on Computational and Mathematical Methods in Science and Engineering, CMMSE, 30 June-3 July.
- [27] Handl, J., Douglas, B. K., Knowles, J., 2007. Multiobjective optimization in bioinformatics and computational biology. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 4(2): April-June 2007.
- [28] Haralick, R. M., Shanmugam, K., Its'hak Dinstein., 1973. Textural features for image classification. *IEEE T. Syst. Man Cyb.* 3(6), 610-621.
- [29] Rotterman, Y. and Porat, M., 2006. Colour image coding using regional correlation of primary colours. *Image Vision Comput.* 25, 637-651.
- [30] Angulo, J. and Serra, J., 2007. Modelling and segmentation of colour images in polar representations. *Image Vision Comput.* 25, 475-495.
- [31] Leon, K., Mery, D., Pedreschi, F., Leon, F., 2006. Color measurement in $L^*a^*b^*$ units from RGB digital images. *Food Res. Int.* 39, 1084-1091.
- [32] Kim, M.C., 2008. Comparative color gamut analysis of xvYCC standard. *Displays.* 29, 376-385.
- [33] Palm, C., 2004. Color texture classification by integrative Co-occurrence matrices. *Pattern Recogn.* 37, 965-976.
- [34] Pydipati, R., Burks, T. F., Lee, W. S., 2006. Identification of citrus disease using colour texture features and discriminant analysis. *Comput. Electron. Agr.* 52, 49-59.
- [35] Hendrawan, Y. and Murase, H., 2011a. Non-destructive sensing for determining Sunagoke moss water content: Bio-inspired approaches. *Agricultural Engineering International: CIGR Journal. Manuscript No. 1564. Vol. 13, No.1.*
- [36] Zhang, P., Verma, B., Kumar, K., 2005. Neural vs. statistical classifier in conjunction with genetic algorithm based feature selection. *Pattern Recogn. Lett.* 26, 909-919.
- [37] Hendrawan, Y. and Murase, H., 2010. Neural-Genetic Algorithm as feature selection technique for determining sunagoke moss water content. *Engineering in Agriculture, Environment and Food (EAEF); Asia Agricultural and Biological Engineering Association (AABEA) Journal.* 3(1), 25-31.
- [38] Pan, Q. K., Tasgetiren, M. F., Liang, Y. C., 2008. A discrete particle swarm optimization algorithm for the no-wait flowshop scheduling problem. *Comput. Oper. Res.* 35(9), 2807-2839.
- [39] Hendrawan, Y. and Murase, H., 2011d. Bio-inspired feature selection to select informative image features for determining water content of cultured Sunagoke moss. *Expert Syst. Appl.* 38(11), 14321-14335.
- [40] Abbas, H.A., 2001. A monogenous MBO approach to satisfiability. In: Proceeding of the international conference on computational intelligence for modelling, control and automation, CIMCA' 2001, Las Vegas, NV, USA.
- [41] Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A., Teller, E., 1953. Equation of state calculations by fast computing machines. *J. Chem. Phys.* 21, 1087-1092.
- [42] Hendrawan, Y. and Murase, H., 2011c. Neural-intelligent water drops algorithm to select relevant textural features for developing precision irrigation system using machine vision. *Comput. Electron. Agr.* 77(2), 214-228.
- [43] Hendrawan, Y. and Murase, H., 2011b. Neural-discrete hungry roach infestation optimization to select informative textural features for determining water content of cultured Sunagoke moss. *Environ. Control Biol.* 49(1), 1-21.