A Dynamical Simplified Swarm Optimization Algorithm for the Multiobjective Annual Crop Planning Problem Conserving Groundwater for Sustainability

Chun-Cheng Lin, Der-Jiunn Deng*, Jia-Rong Kang, and Wan-Yu Liu

***Abstract*—In large-scale agriculture, insufficient irrigation water may lead to over-pumping of groundwater, increasing the risk of land subsidence. Growing dryland crops can effectively decrease the demand for irrigation water. However, the previous works on annual crop planting (ACP) focused on maximizing the profit through growing wetland crops, consuming much water. For sustainability, this work proposes a mathematical programming model for an ACP that allocates a land area for growing dryland and wetland crops to maximize the total profit and minimize the total irrigation water used for multiple cropping, under practical constraints. Simplified swarm optimization (SSO) improves the PSO with four probabilities to determine the operations of updating solutions. We further propose dynamic SSO to solve the concerned ACP, in which the four probabilities are adjusted dynamically according to performance of the operations executed. Through simulation on a case study, the proposed DSSO demonstrates high performance over some classical approaches.**

***Index Terms*—Crop planning, sustainability, land subsidence, irrigation water, simplified swarm optimization, swarm intelligence**

I. Introduction

Recently, land subsidence has led to destruction of buildings, structures, railways, and highways [1]. Based on the monitoring data in [2], the land subsidence rate of some cities in the world has been over 50 cm/yr. Land subsidence is generally caused by human activities. For rapid economic development, a lot of countries have exploited a large amount of flat areas or coastal areas for rice cultivation, aquaculture, and urbanization. Over-pumping groundwater for fish ponds and irrigation for paddy fields has made the groundwater throughout the agricultural region rapidly drop and further caused land subsidence [3].

For sustainability of agriculture, one of the most effective ways to reduce the chance of land subsidence in vulnerable farming fields is to plant dryland crops or water-saving crops (e.g., green corn, field corn, peanuts, soybeans, sweet potatoes, sorghum, adzuki beans, and potatoes) rather than wetland crops (e.g., wet-paddy rice, which is popular in East Asia, Southeast Asia, and South Asia) [4]. The amount of water requirement for planting dryland crops is generally less than a half of that for planting paddy rice. Hence, planting dryland crops can effectively reduce the demand for groundwater, further avoiding occurrence of land subsidence. However, the operating cost for planting some dryland crops is higher than that for planting paddy rice. For instance, the seedling cost of dryland crops is two times more than that for paddy rice. In addition, a low degree of mechanization for dryland crops causes high labor cost, which is two to five times higher than that for paddy rice.

The annual crop planning (ACP) is concerned about making an annual plan of allocating a land area for planting crops. Previous works have solved various ACP problems with mathematical programming, e.g., considering new policies [5], deficit irrigation [6], uncertainty [7], and possibility measure [8]. However, most of the previous works only focused on the objective of maximizing the profit earned by harvesting crops, but neglected the objective of minimizing the amount of water requirement for irrigation, which is crucial in conserving groundwater [9] and avoiding occurrence of land subsidence.

In light of the above, this work proposes a two-objective ACP problem that allocates a land area for multiple cropping of dryland and wetland crops so that the total profit earned by harvesting crops is maximized, and the total irrigation water amount is minimized simultaneously. Note that this problem is different from the previous work in [10] that considered only the former objective. The constraints of the proposed ACP model include the constraint of the maximal land area, the constraint of balancing supply and demand in markets, and the constraint of the maximal irritation water amount. In addition, different from planting the same crop in the same area, crop

C.-C. Lin is with Department of Industrial Engineering and Management, National Chiao Tung University, Hsinchu 300, Taiwan; Department of Business Administration, Asia University, Taichung 413, Taiwan; and also Department of Medical Research, China Medical University Hospital, China Medical University, Taichung 404, Taiwan (e-mail: cclin321@nctu.edu.tw).

D.-J. Deng is with Department of Computer Science and Information Engineering, National Changhua University of Education, Changhua 500, Taiwan (e-mail: djdeng@cc.ncue.edu.tw).

J.-R. Kang is with Department of Information Management, Tatung University, Taipei 104, Taiwan (e-mail: klkang8987@gmail.com).

W.-Y. Liu is with Department of Forestry, National Chung Hsing University, Taichung 402, Taiwan; and Innovation and Development Center of Sustainable Agriculture, National Chung Hsing University, Taichung 402, Taiwan (e-mail: wyliu@nchu.edu.tw).

* D.-J. Deng is the corresponding author of this paper.

rotation (which sequentially plants various types of crops in the same area) is not only friendly to the environment, but also drastically increases the profit from harvesting crops [11]. Therefore, different from previous problems, the concerned problem additionally considers the constraint of crop rotation.

This work establishes a mathematical programming model for the concerned problem. Furthermore, as a branch of artificial intelligence (AI), swarm intelligence (SI) [12] simulates a swarm of agents that search for solutions for the concerned problem. Particle swarm optimization (PSO) is a popular SI approach which simulates a swarm of particles (each of which is associated as a position representing a candidate solution), and updates each particle's position based on a formula referring to the best position found by the particle so far (pBest) as well as the global best position found by all particles so far (gBest). Instead of adopting the position updating formula in the PSO, simplified swarm optimization (SSO) [13] updated each particle's position with the following four operations based on four given probabilities: no change, setting as the pBest, setting as the gBest, and setting as a random position. To improve the SSO, this work further proposes a novel dynamic SSO (DSSO) approach to solve the concerned problem. The DSSO includes a novel scheme of dynamically adjusting the four probabilities to search for better solutions. Through simulation, the proposed DSSO demonstrates outperformance over the other SI approaches, including SSO, PSO, genetic algorithm (GA), and cuckoo search (CS).

The main contributions of this work are as follows:

- Inspired by reducing irritation water used for plating crops for sustainable agriculture and avoiding land subsidence, this work proposes a mathematical programming model for a novel ACP problem which allocates a land area for multiple cropping of dryland and wetland crops to maximize the total profit and minimize the total irritation water used for cropping. Its differences from previous problems include adding the objective of minimizing the total irritation water, constraint of crop rotation, and constraint of two cultivations of paddy rice.
- This work solves the concerned problem by a novel DSSO algorithm, which improves the SSO with a scheme of dynamically adjusting the four probabilities that decides the operations of updating solutions. Through simulation on a case study, the proposed DSSO performs better than the SSO, PSO, GA, and CS in terms of convergence rate and stability, and it can obtain high quality solutions.

II. Literature review

A. ACP

The work in [14] considered the multiobjective crop planning problem, which is NP-hard; and reviewed a variety of algorithms for addressing this problem. The work in [5] developed a decision-support tool based on a multi-stage linear programming model that maximizes the rent return of cropping planning under the Common Agricultural Policy. The work in [6] created a liner model for multi-crop planning, in which the objective is to maximize the net profit of the yield produced over a planning horizon, under the constraints of the total land area, crop production quota, and water amount. The work in [7] developed a system based on linear programming to solve a complicated ACP problem with uncertain parameters such as randomness of irrigation patterns, market price, and policies. The work in [8] utilized linear programming to solve an ACP problem with fuzzy numbers of profit coefficients to maximize the income of a farm.

The work in [15] applied PSO to optimize an agricultural plain with crops areas and water depletion of wells. The work in [16] developed an Internet-of-Things (IoT) system based on GA to predict rainfall so as to decide the used volume of manual water supply. The work in [17] utilized a CS algorithm to optimize the utilization of water resources in hydropower station operation. The work in [10] created a model that maximizes the gross margin, subject to the constraints of land area and water allocation. Then, they solved it with some SI approaches, including CS, firefly algorithm (FA), glowworm swarm optimization (GSO), and GA; and they found that the CS is the best approach for the experimental instance.

B. PSO and SSO

Swarm intelligence (SI) is an integral part of the field of AI, and its idea is to adopt the collective behavior of an organized group of animals to address various problems. A variety of SI algorithms have been proposed, e.g., PSO, SSO, FA, and artificial bee colony (ABC) algorithm [18]. This work focuses on PSO and SSO, which are introduced as follows.

The PSO is an SI approach that simulates a swarm of particles (in which the position of each particle represents a candidate solution) flying to cooperatively search for a food source position (representing the optimal solution). Its main idea is explained as follows. Consider an n-dimensional solution space, in which each coordinate represents a solution. The PSO considers a swarm of η particles, in which each particle κ is associated with a n-dimensional position $\chi_\kappa = (\chi_{\kappa 1}, \chi_{\kappa 2}, \ldots, \chi_{\kappa n})$ and a velocity $V_\kappa = (V_{\kappa 1}, V_{\kappa 2}, \ldots, V_{\kappa n})$, which are initialized randomly. Then, the PSO enters a main loop, in which the position and velocity vectors of each particle κ are iteratively updated as follows:

$$V_\kappa^{\text{new}} = \omega \cdot V_\kappa + c_1 \cdot r_1 \cdot (P_\kappa - \chi_\kappa) + c_2 \cdot r_2 \cdot (P^* - \chi_\kappa) \tag{1}$$

$$\chi_\kappa^{\text{new}} = \chi_\kappa + V_\kappa^{\text{new}} \tag{2}$$

where ω is an inertia factor; c_1 and c_2 are parameters for learning the previous best position and the global best position found so far, respectively; r_1 and r_2 are two numbers randomly generated within the range [0, 1]; P_κ is the best position found by particle κ so far; and P^* is the best position found by all particles so far. The main loop is repeated until the stop criteria (e.g., a maximal number of iterations) are achieved, and finally outputs the best solution found so far.

The main idea of the classical SSO [15] is to simplify the formulas of updating the positions of particles in the PSO in (1) and (2). The SSO considers three predetermined parameters C_w, C_p, and C_g to divide the range [0, 1] into four subranges. Consider updating the ith element $\chi_{\kappa i}$ of the position $\chi_\kappa = (\chi_{\kappa 1}, \chi_{\kappa 2}, \ldots, \chi_{\kappa n})$ of particle κ. Based on the four subranges, the

position element $\chi_{\kappa i}$ is updated by the following formula:

$$\chi_{\kappa i}^{\text{new}} = \begin{cases} \chi_{\kappa i}, & \text{if } r(0,1) \in [0, C_{\text{w}}); \\ P_{\kappa i}, & \text{if } r(0,1) \in [C_{\text{w}}, C_{\text{p}}); \\ P_i^*, & \text{if } r(0,1) \in [C_{\text{p}}, C_{\text{g}}); \\ r(LB_i, UB_i), & \text{if } r(0,1) \in [C_{\text{g}}, 1]. \end{cases} \quad (3)$$

where $r(0, 1)$ is a value generated randomly within the range [0, 1]; and LB_i and UB_i are the lower and upper bounds of the ith element of the position. That is, in Case 1, the position element is not modified; in Case 2, it is replaced by the ith element of the previous best position of particle κ (P_κ); in Case 3, it is replaced by the ith element of the best position (P_i^*); and in Case 4, it is assigned to a random value within its feasible range.

Recent works of PSO and SSO are reviewed as follows. The PSO has been applied in various fields, e.g., antenna design [19], electromagnetics [20], and cyber-physical energy-saving control [21]. The works much related to this work include the work in [22], which proposed a PSO approach to find the optimal reservoir operation for irrigation of multiple crops. The work in [23] developed a simulation optimization tool based on the PSO to discover the potential optimal cultivation area of crops with the maximal benefits in some irrigated plains. The work in [24] integrated a neuro-fuzzy inference system with ACO, GA, and PSO prediction of agricultural drought according to various indexes. On improving the PSO, the work in [25] integrated the PSO with stochastic approximation methods to increase the convergence rates of PSO. The work in [26] introduced a lot practical industrial applications based on the PSO and other SI approaches. On the other hand, the SSO has been successful in various applications, e.g., facility location problem [27], vehicle routing problem [28], and training neural networks [29]. To improve the SSO, the work in [30] integrated the classical SSO with an updating mechanism to address non-discrete data. The work in [31] combined the Nelder-Mead simplex with the SSO to identify parameters for a solar cell system.

III. Mathematical Programming Model for the Concerned ACP

This section constructs a mathematical programming model for the concerned ACP problem. In this problem, we play a role of the government authorities that are making an annual crop plan for a state/county-scale land, because the government generally can provide incentives to farmers to achieve this crop plan. Note that this problem is also suitable for private landowners that makes a crop plan for their lands. Consider that through planting dryland crops, the used volume of irrigation water may be reduced. To encourage farmers to plant dryland crops in some area with a high probability of land subsidence, this problem considers an annual mixed plan of planting dryland and wetland crops. Aside from maximizing the total profit earned by harvesting crops, this problem is also concerned about minimizing the irrigation water amount, to reduce the risk of over-pumping an enormous underwater volume to cause land subsidence in this area when the irrigation water is not sufficient to support cropping.

Specifically, given a land area, the concerned ACP problem is to allocate multiple small pieces (say, plots) in this area for planting dryland crops (e.g., sugarcanes, tobacco, peanuts, corn, adzuki beans, soybeans, sorghum, sesames, and sweet potatoes) and wetland crops (e.g., tea and paddy rice), so that the total gross margin earned by harvesting crops is maximized, and the total irrigation water amount is minimized, under the constraints for the maximal land area, balancing the supply and demand in markets, the maximal irritation water amount, and crop rotation. To increase productivity in cropping, this problem becomes complex because the land adopts multiple cropping, i.e., two or more crops can be planted in the same plot. For instance, South China allows triple cropping, although it may adopt double cropping in practice for soil conservation.

Based on the previous model in [10], the mathematical programming model for the concerned ACP problem is given with the notation in Table I.

TABLE I
NOTATIONS USED IN THE PROPOSED MATHEMATICAL PROGRAMMING MODEL.

Index	Definition
k	Index of a multiple-cropping type, in which k = 1, 2, 3 if single-cropping, double-cropping, and triple-cropping plots are concerned, respectively. That is, type k allows different crops planted at k stages on the plots of type k.
i	Index of a stage at which a crop is planted on plots of type k.
j	Index of a crop planted at stage i on plots of type k.
ω	A crop index representing that this crop is the wetland rice.
Parameter	**Definition**
K	Number of multiple-cropping plot types.
M_{ki}	Number of crops planted at stage i on plots of type k.
CP_{kij}	Expected producer price (\$/t) of crop j planted at stage i on plots of type k.
CY_{kij}	Expected yield amount (t/ha) of crop j planted at stage i on plots of type k.
OC_{kij}	Operational cost (\$/ha) of crop j planted at stage i on plots of type k.
WC	Cost of the irrigated water (\$/m^3).
RW_{kij}	The water (mm) required by crop j planted at stage i on plots of type k.
R_{kij}	Estimated rainfall (mm) during planting crop j at stage i on plots of type k.
F_{kij}	Average percentage of a land area requiring irrigation when planting crop j at stage i on plots of type k.
L_{ki}	Total area of the land (ha) allocated for planting crops at stage i on plots of type k.
LB_{kij}	Lower bound of the area (ha) of the land allocated for planting crop j at stage i on plots of type k.
UB_{kij}	Upper bound of the area (ha) of the land allocated for planting crop j at stage i on plots of type k.
T	Total land area (ha) planned for cropping production.
A	The water volume (m^3/ha) per hectare that can support irrigation of the concerned land area.
Response variable	**Definition**
B_{kij}	Gross margin earned for harvesting each hectare (\$/ha) of crop j at stage i on plots of type k.
IW_{kij}	Volume of the irrigated water (m^3/ha) that is applied to crop j planted at stage i on plots of type k.
Decision variable	**Definition**
X_{kij}	Area of the land (ha) allocated to plant crop j at stage i on plots of type k.
Y_{kij}	A binary decision variable deciding whether X_{kij} is zero, i.e., Y_{kij} = 1 if X_{kij} > 0; otherwise, Y_{kij} = 0.

This problem considers the following two objectives:

$$\text{Maximize} \quad \sum_{k=1}^{K}\sum_{i=1}^{k}\sum_{j=1}^{M_{ki}} X_{kij} \cdot B_{kij} \tag{4}$$

$$\text{Minimize} \quad \sum_{k=1}^{K}\sum_{i=1}^{k}\sum_{j=1}^{M_{ki}} X_{kij} \cdot IW_{kij} \tag{5}$$

where

$$B_{kij} = CP_{kij} \cdot CY_{kij} - OC_{kij} - IW_{kij} \cdot WC; \tag{6}$$

$$IW_{kij} = (RW_{kij} - R_{kij}) / 1000 \cdot 10000 \cdot F_{kij}. \tag{7}$$

Objective (4) is to maximize the total gross margin earned by harvesting crops, in which the gross margin B_{kij} for a crop is calculated in (6); and Objective (5) is to minimize the total volume of irrigation water used for planting crops, in which the irrigated water amount for a crop j planted at stage i on plots of type k is calculated in (7), depending on the weather factors of the land location such as rainfall.

Note that although the previous works have considered a lot of factors on sustainable agriculture in their objective functions (e.g., optimization for soil, labors, and fertilizer), these factors may not effectively save water to further avoid land subsidence. Hence, this work combines objectives of the total gross margin earned and the total used volume of irrigation water, i.e., simultaneously considering the economic aspect and saving water to effectively solve the problem of land subsidence.

This problem considers the following constraints:

$$\sum_{j=1}^{M_{ki}} X_{kij} \le L_{ki}, \quad \forall k, i \tag{8}$$

$$LB_{kij} \le X_{kij} \le UB_{kij}, \quad \forall k, i, j \tag{9}$$

$$\sum_{k=1}^{K}\sum_{i=1}^{k}\sum_{j=1}^{M_{ki}} X_{kij} \cdot IW_{kij} \le T \cdot A \tag{10}$$

$$X_{kij} \cdot B_{kij} \ge 0, \quad \forall k, i, j \tag{11}$$

$$Y_{kij} = \left\lceil X_{kij} - \left\lfloor X_{kij} \right\rfloor \right\rceil, \quad \forall k, i, j \tag{12}$$

$$(Y_{kij} + Y_{k(i+1)j}) \cdot (Y_{k(i+1)j} + Y_{k(i+2)j}) \le 2, \quad \forall k \ge 3, i, j \tag{13}$$

$$Y_{k1\omega} \ge Y_{k2\omega}, \quad \forall k \tag{14}$$

$$X_{kj\omega} = 0, \quad \forall k, j \ge 3 \tag{15}$$

Constraint (8) enforces that the total area of the land allocated for planning crops at stage j on plots of type k must not exceed the maximal area L_{ki}. Constraint (9) enforces the decision variable X_{kij} between two bounds. Constraint (10) enforces that the total irrigated water volume must not exceed the maximal irrigated water volume that can be applied ($T \cdot A$). Note that this constraint aims to avoid using too much groundwater when the surface water is not enough for irrigation, and further to avoid land subsidence. Constraint (11) enforces that the gross margin for plating each crop must not be negative. Constraint (12) is used to determine the value of the decision variable Y_{kij}. Considering crop rotation, Constraint (13) enforces that any crop must not be planted at more than two consecutive stages, so as to maintain soil quality.

Paddy rice allows triple cropping, but its production at the second stage is less than that at the first stage and becomes much less if it is planted only at the second stage but not at the first stage. Hence, Constraint (14) enforces that if paddy rice is planted at the first stage, then it can be planted at the second stage; otherwise, it cannot. Furthermore, considering that paddy rice has a small production amount at the third stage (i.e., in fall or winter), Constraint (15) enforces that paddy rice cannot be planted at the third or later stages.

The differences of the proposed model from the previous model in [10] are listed as follows:

- Different from the previous model that only considered a single objective of maximizing the total gross margin earned by harvesting cropps in (4), the proposed model additionally considers the objective of minimizing the total irrigation water volume adopted for planting crops in (5), inspired from sustainability and avoidance from land subsidence.
- The proposed model additionally considers the constraint of crop rotation in (13), which is friendly to the environment.
- The proposed model additionally considers the practice of planting paddy rice in multiple cropping. For maintaining high productivity in practical multiple cropping, paddy rice cannot be planted at the second stage without being planted at the first stage in (14); and it cannot be planted at the third and later stages in (15).

IV. Proposed DSSO for the Concerned ACP

This section proposes a novel DSSO approach for the concerned ACP problem abovementioned. Different from the PSO approach that adopts a formula to update a particle's position to search for solutions, the classical SSO approach [15] is to simply set four fixed probabilities to select the operation of setting a particle's position (i.e., no change, and setting as pBest, gBest, and a random position). Different from the classical SSO approach, the proposed DSSO approach includes a novel scheme of iteratively and dynamically adjusting the four probabilities of selecting the position updating operations based on the performance of the adopted operations. Later experimental results will show outperformance of this scheme.

The proposed DSSO is given in Algorithm 1. The main components designed in the DSSO are detailed as follows.

Algorithm 1 DSSO

For $\kappa \in \{1, 2, \ldots, \eta\}$, each particle κ is associated with three variables $C_{w\kappa}$, $C_{p\kappa}$, and $C_{g\kappa}$, which are initialized so that $0 < C_{w\kappa} < C_{p\kappa} < C_{g\kappa} < 1$
for $k = 1, 2, \ldots, \eta$ **do**
  Initialize the position $\chi_\kappa = (X_{kij}^\kappa)$ of particle κ, and evaluate its fitness $f(\chi_\kappa)$
  If $X_{k1\omega}^\kappa = 0$, then $X_{k2\omega}^\kappa = 0$ to satisfy Constraint (14)
  Update the best position $P_\kappa = (P_{kij}^\kappa)$ of particle κ found so far as the same position of χ_κ, and record the corresponding fitness $f(P_\kappa)$
  Update $P^* = (P_{kij}^*)$ as the best position found by all particles so far, and record the corresponding fitness $f(P^*)$
next for
while the maximal iteration number τ is not achieved **do**
  for $\kappa = 1, 2, \ldots, \eta$ **do**
    Initialize all counter variables $n_{w\kappa} = n_{p\kappa} = n_{g\kappa} = n_{r\kappa} = 0$
    while each X_{kij}^κ in X^κ is not considered **do**
      if $r(0, 1) \ge C_{g\kappa}$ **then**
        $X_{kij}^\kappa = r(LB_j, UB_j)$
        $n_{r\kappa} = n_{r\kappa} + 1$

```
15:             else if r(0, 1) ≥ C_pk then
16:                 X^κ_kij = P*_kij
17:                 n_gκ = n_gκ + 1
18:             else if r(0, 1) ≥ C_wk then
19:                 X^κ_kij = P^κ_kij
20:                 n_pκ = n_pκ + 1
21:             else
22:                 n_wκ = n_wκ + 1
23:             end if
24:         end while
25:         If X^κ_k1ω = 0, then X^κ_k2ω = 0 to satisfy Constraint (14)
26:         Evaluate the fitness f(χ_κ) of particle κ
27:         Update the best position P_κ of particle κ found so far
28:         Update the best position P* found by all particles so far
29:         Adjust C_wκ, C_pκ, and C_gκ by (18), (19), and (20), respectively
30:     next for
31: end while
```

A. Solution representation

In the DSSO, the position of a particle represents a solution for the concerned ACP problem. Specifically, the position χ_κ of particle κ is represented as follows: $\chi_\kappa = \left(X^\kappa_{kij} \mid \forall k,i,j\right)$ excluding the $X_{kj\omega}$'s in Constraint (15), in which X^κ_{kij} is the decision variable of the concerned ACP problem, representing the land area of planting crop j at stage i in plots of type k for $k \in \{1, 2, \ldots, l\}$, $i \in \{1, 2, \ldots, N_k\}$, and $j \in \{1, 2, \ldots, M_{ki}\}$.

B. Fitness evaluation

Given the position χ_κ of a particle κ, the performance of the position is evaluated by the following fitness function:

$$\begin{aligned} f(\chi_k) = {} & \lambda \cdot \Big(\sum_{k=1}^{K}\sum_{i=1}^{k}\sum_{j=1}^{M_{ki}} X_{kij}\cdot B_{kij} \Big/ \sum_{k=1}^{K}\sum_{i=1}^{k}\sum_{j=1}^{M_{ki}} UB_{kij}\cdot B_{kij}\Big) \\ & + (1-\lambda)\cdot\Big(T\cdot A - \sum_{k=1}^{K}\sum_{i=1}^{k}\sum_{j=1}^{M_{ki}} X_{kij}\cdot IW_{kij}\Big) \\ & \Big/ \Big(T\cdot A - \sum_{k=1}^{K}\sum_{i=1}^{k}\sum_{j=1}^{M_{ki}} LB_{kij}\cdot IW_{kij}\Big) \\ & - \delta_1 \cdot \max_{\forall k,i}\Big(\sum_{j=1}^{M_{ki}} X_{kij} - L_{ki}, 0\Big) \\ & - \delta_2 \cdot \max\Big(\sum_{k=1}^{K}\sum_{i=1}^{k}\sum_{j=1}^{M_{ki}} IW_{kij}X_{kij} - T\cdot A, 0\Big) \\ & - \delta_3 \cdot \max_{\forall k,i,j}(-B_{kij}X_{kij}, 0) \\ & - \delta_4 \cdot \max_{\forall k\geq 3,i,j}(2-(Y_{kij}+Y_{k(i+1)j})\cdot(Y_{k(i+1)j}+Y_{k(i+2)j}), 0) \end{aligned} \tag{16}$$

where λ is the weight of the first objective (4), for $\lambda \in [0, 1]$; and δ_1, δ_2, δ_3, and δ_4 are the costs of penalizing Constraints (8), (10), (11), and (13), respectively. That is, the above fitness function is the weighted sum of two normalized terms for Objectives (4) and (5), plus the penalty costs for all constraints.

The first and second terms in the fitness function (16) are detailed as follows. In the fraction of the first term, the numerator is Objective (4) (i.e., the total gross margin); and the denominator is the upper bound of the total gross margin, used for normalization. Hence, this fraction falls in the range [0, 1]. On the other hand, Objective (5) is expected to be minimized, but the fitness function is expected to be maximized. Hence, in the fraction of the second term, the numerator is a to-be-maximized measure equal to the difference of the maximal irrigated water volume $T \cdot A$ from Objective (5) (i.e., the remaining irrigation water amount); and similarly to the first term, the denominator is user for normalization. Then, $\lambda \in [0, 1]$ makes the composed weighted value of the first and second terms of the fitness function within range [0, 1].

C. Position updating

As mentioned in Section II, the classical SSO employs three predetermined parameters C_w, C_p, and C_g to divide the range [0, 1] into four subranges. Different from the classical SSO, the proposed DSSO allows them to be dynamically adjustable; and each particle κ is associated with three of such variables, and hence, we denote them as $C_{w\kappa}$, $C_{p\kappa}$, and $C_{g\kappa}$. That is, the classical SSO adopts 3 fixed parameters; whereas the proposed DSSO adopts $3 \times \eta$ control variables. Note that the $3 \times \eta$ control variables are adjusted dynamically at the end of each iteration of the DSSO main loop, which will be detailed in the next subsection.

Therefore, based on the three control variables $C_{w\kappa}$, $C_{p\kappa}$, and $C_{g\kappa}$ to divide the range [0, 1] into four subranges, the ith element $\chi_{\kappa i}$ of the position $\chi_\kappa = (\chi_{\kappa 1}, \chi_{\kappa 2}, \ldots, \chi_{\kappa \eta})$ of particle κ is updated as follows:

$$\chi^{\text{new}}_{\kappa i} = \begin{cases} \chi_{\kappa i}, & \text{if } r(0,1) \in [0, C_{w\kappa}); \\ P_{\kappa i}, & \text{if } r(0,1) \in [C_{w\kappa}, C_{p\kappa}); \\ P^*_i, & \text{if } r(0,1) \in [C_{p\kappa}, C_{g\kappa}); \\ r(LB_i, UB_i), & \text{if } r(0,1) \in [C_{g\kappa}, 1]. \end{cases} \tag{17}$$

D. Updating dynamical control variables

At the end of each iteration of the main loop of the DSSO, the three control variables $C_{w\kappa}$, $C_{p\kappa}$, and $C_{g\kappa}$ of each particle κ are adjusted accordingly. These control variables divide the range [0, 1] into four subranges. Let $n_{w\kappa}$ denote the total number of times the subrange $[0, C_{w\kappa})$ in (17) was adopted by the n position elements of particle κ. Similarly, $n_{p\kappa}$, $n_{g\kappa}$, and $n_{r\kappa}$ are defined for subranges $[C_{w\kappa}, C_{p\kappa})$, $[C_{p\kappa}, C_{g\kappa})$, and $[C_{g\kappa}, 1]$, respectively. In Algorithm 1, Line 10 initializes the four counter variables; and Lines 14, 17, 20, and 22 iteratively update the four counter variables, respectively. At the end of the main loop, Line 29 adjusts them as follows:

$$C^{\text{new}}_{w\kappa} = (C_{w\kappa} + n_{w\kappa}\cdot\Delta)/(1+n\cdot\Delta) \tag{18}$$

$$C^{\text{new}}_{p\kappa} = (C_{p\kappa} + (n_{w\kappa}+n_{p\kappa})\cdot\Delta)/(1+n\cdot\Delta) \tag{19}$$

$$C^{\text{new}}_{g\kappa} = (C_{g\kappa} + (n_{w\kappa}+n_{p\kappa}+n_{g\kappa})\cdot\Delta)/(1+n\cdot\Delta) \tag{20}$$

where $n = n_{w\kappa} + n_{p\kappa} + n_{g\kappa} + n_{r\kappa}$; Δ is a given step parameter, which is generally set as a very small number. The idea of the above design is explained as follows. See the three axes in Fig 1, which shows the three control variables in iteration t, temporary computation, and iteration $t + 1$.

From iteration t to temporary computation in Fig. 1, each control variable increases by a step depending on the times that the corresponding case is adopted. Note that $n_{w\kappa} + n_{g\kappa} + n_{g\kappa} + n_{r\kappa} = n$. However, the total range in temporary computation is extended to $1 + n\,\Delta$. Hence, the three control variables are divided by $1 + n\,\Delta$ so that they shrink with the same proportion within the range [0, 1].

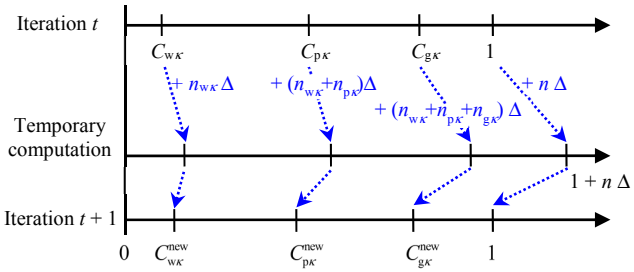

Fig. 1. Illustration of updating dynamical control variables.

V. Experimental Results

A. Experimental case and environment

The experiments in this work are conducted on a problem instance based on the real data in Yunlin (County), Taiwan. Yunlin is one of the most affected countries that suffer from land subsidence [32]. Yunlin belongs to the Chianan Plain, which is the largest plain in Taiwan and is well-known for its agriculture. Yunlin is separated from other counties by the Zhuoshui River (the longest river in Taiwan) and the Beigang River, which have much potential for hydroelectricity. The area of Yunlin is 1,290.83 km^2, and about 68% of the area in Yunlin is used for agriculture (including farming and fishing ponds). Note that although this section shows the experimental analysis on a real data in Yunlin, the proposed DSSO can also be used to solve the ACP problems using other-size data instances.

As a large-scale agricultural county and with the largest agricultural production value, Yunlin requires an enormous water amount to support agriculture. However, since the surface water in Yunlin is not enough to supply all the demand from the agricultural industry, the groundwater is over-pumped, so that the stratum keeps sinking. Whenever it rains heavily, flooding results in enormous loss of agricultural products. In addition, potential land subsidence could destroy crucial infrastructures (e.g., high-speed railways and freeways), further causing the loss of casualties and property.

There are 14 common crops planted in Yunlin, and the statistics of production and irrigation water of the 14 crops in Yunlin [33] are detailed in Tables II and III, respectively. In addition, suppose that F_{kij} = 100% for any k, i, j. Note that NT$ is a stable currency, and US$ 1 has been about NT$ 28–32.

On the crop type in Table II, the crops that are planted for the whole year include 'yearly' and 'perennial' crops, which are denoted by 'Y' and 'P', respectively. Seasonal crops that are planted in a season include 'spring', 'fall', and 'winter' crops. The 'double-cropping' crop that is planted in two seasons (i.e., one is spring, and the other is fall or winter) is denoted by 'D'. The crop that can be planted at 'any season' is denoted by 'A'. Although Yunlin allows triple-cropping, it generally allows only double-cropping. Hence, in Table II, the crops of type 'Y' and 'P' are planted with only one stage; and the crops of the other types can be planted at either one of two stages in our proposed DSSO. Note that for the crop type 'D', the 'planting area' counts the areas planted at two stages, and hence, the lower and upper bounds of a planting area at stage (LB_{kij} and UB_{kij}) are about a half of the 'planting area' in Table II.

Table II.
Statistics of production of 14 crops in Yunlin

Crop	Type	Planting area (ha)	LB_{kij} (ha)	UB_{kij} (ha)	CY (t/ha)	CP (NT$)
Refined sugarcane	Y	2,828.00	1,414	4,242	55.856	964
Fresh sugarcane	Y	64.52	30	100	92.405	10,510
Tea	P	367.52	180	550	0.947	507,370
Paddy rice (1st stage)	S	30,353.16	30,250	30,450	8.635	27,526
Paddy rice (2nd stage)	F	14,298.75	14,200	14,400	6.777	27,526
Tobacco	W	56.42	30	85	2.081	207,410
Peanut	D	15,616.47	7,700	7,900	2.96	53,570
Green corn	D	508.72	125	375	4.67	8,970
Field corn	D	6,117.49	1,500	4,500	8.45	18,880
Adzuki bean	D	0.36	0.09	0.27	0.917	64,370
Soybean	D	164.86	40	120	1.609	41,000
Sorghum	D	0.93	0.2325	0.6975	3.796	26,000
Sesame	D	34.27	9	27	0.502	215,270
Sweat potato	A	4,075.33	1,000	3,000	24.772	13,230

Table III.
Statistics of irrigation water of 14 crops in Yunlin

Crop	$RW_{kij} - R_{kij}$ (mm)	$IW_{kij} \cdot WC_{kij}$ (NT$/ha)	OC_{kij} (NT$/ha)
Refined sugarcane	101.66	1,763.82	42,755
Fresh sugarcane	101.66	1,763.82	637,164
Tea	2,195.84	38,097.75	368,690
Paddy rice (1st stage)	2,019.44	35,037.28	88,136
Paddy rice (2nd stage)	1,663.86	28,868.04	80,392
Tobacco	1,389.41	24,106.32	300,998
Peanut	1,389.41	24,106.32	117,094
Green corn	1,389.41	24,106.32	10,816
Field corn	1,389.41	24,106.32	111,978
Adzuki bean	1,389.41	24,106.32	6,950
Soybean	1,389.41	24,106.32	23,366
Sorghum	1,389.41	24,106.32	41,152
Sesame	1,389.41	24,106.32	31,220
Sweat potato	1,389.41	24,106.32	134,152

The other parameters in the model are set as follows. The total planting area T is 80,042 ha. The maximal irrigation water amount A per hectare per year is 15,381 m^3/ha/y. The water cost WC is NT$ 1.735/m^3. The total land area for planting yearly crops (i.e., those of types 'Y' and 'P') is 3,260 ha. The total land area for crops of type 'S' and the crops of type 'D' planted at the first stage is 43,612 ha. The land area for crops of type 'F' or 'W' and the crops of type 'D' planted at the second stage is 27,614 ha. The DSSO is implemented in C++, and runs on a PC with an Intel i5-3470 CPU @ 3.20 GHz and 8-GB RAM.

B. Parameter analysis

This section conducts experiments under various settings of parameters of the DSSO. First, the DSSO for the concerned two-objective ACP problem adopts the parameter $\lambda \in [0, 1]$ to control the weight of the first objective. The comparison of the experimental results under different λ values is shown in Table IV, which includes the 'fitness' results, the 'gross margin earned' (i.e., the first objective), and the 'irritation water used' (i.e., the second objective) for planting crops with respect to each λ value.

TABLE IV.
COMPARISON OF RESULTS UNDER DIFFERENT λ VALUES

λ	Fitness	Gross margin earned (NT$)	Irritation water used (m^3)
0.0	0.976804	5,282,324,480	34,384
0.1	0.984330	5,282,308,608	36,176
0.2	0.969174	5,305,536,512	107,216
0.3	0.954358	5,305,536,512	107,216
0.4	0.939553	5,305,530,368	105,424
0.5	0.924735	5,305,529,856	105,296
0.6	0.909916	5,305,530,368	105,424
0.7	0.895100	5,305,527,808	104,656
0.8	0.880279	5,305,529,856	105,296
0.9	0.901736	5,997,448,704	57,224,016
1.0	0.976804	6,092,413,952	89,853,264

When λ = 1.0 (i.e., only the gross margin earned is concerned), the result with the maximal gross margin (i.e., NT$ 6,092,413,952) is obtained, but it causes the maximal irritation water used (i.e., 89,853,264 m^3). To the other extreme, when λ = 0.0 (i.e., only the irrigation water used is concerned), the result has the minimal irrigation water (i.e., 34,384 m^3) and the minimal gross margin (NT$ 5,282,324,480). The results when λ is between 0.2 and 0.8 have similar irrigation water used. Hence, we set $\lambda = 0.5$ (i.e., the two objectives are treated equally) for the later experimental results. Although the results between $\lambda = 0$ and $\lambda = 1$ are not linear, the λ value indeed influences the results in terms of the two objectives. The user can flexibly adjust the λ value according to the practical requirement.

To find the best initial values of control variables C_w, C_p, and C_g, we consider all of their combinations in {0.05, 0.10, 0.15, …, 0.95}, i.e., selecting 3 value from the 19 values and resulting in $\binom{19}{3} = 969$ combinations. Among all combinations, the experimental results when C_w = 0.05 and C_g = 0.95 are best. Hence, Table V gives comparison of best, average, worst, and the standard deviation of the experimental results running 100 times of the DSSO under some combinations of the initial value of control variable C_p when C_w = 0.05 and C_g = 0.95; the number of particles η is 20; and number of iterations τ is 10000. From Table V, the best average fitness among all combinations of DSSO is 0.924860 under (C_w, C_p, C_g) = (0.05, 0.85, 0.95).

TABLE V.
COMPARISON OF RESULTS EXECUTING 100 TIMES OF THE DSSO UNDER DIFFERENT INITIAL VALUES OF CONTROL VARIABLES

C_w	C_p	C_g	Best fitnes	Average fitness	Worst fitness	StdDev
0.05	0.70	0.95	0.924903	0.924859	0.924796	0.000474
0.05	0.75	0.95	0.924903	0.924859	0.924796	0.000313
0.05	0.80	0.95	0.924903	0.924859	0.924796	$< 10^{-6}$
0.05	**0.85**	**0.95**	**0.924903**	**0.924860**	**0.924800**	$\mathbf{< 10^{-6}}$
0.05	0.90	0.95	0.924903	0.924859	0.924819	$< 10^{-6}$
0.05	0.95	0.95	0.924326	0.923624	0.923190	0.000710

Similar analyses on the step parameter Δ used in the dynamical probability adjustment scheme and the number of particles η are conducted. It is concluded that, on average, the best fitness is obtained under $\Delta = 10^{-7}$ and $\eta = 80$.

C. Convergence analysis

With the parameter (C_w, C_p, C_g) = (0.05, 0.85, 0.95) based on the results in last subsection, the plots of fitness versus number of iterations under four Δ values (i.e., 10^{-3}, 10^{-5}, 10^{-7}, and 10^{-9}) are shown in Fig. 2. From Fig. 2, all cases tend to converge, and the case under (C_w, C_p, C_g, Δ) = (0.05, 0.85, 0.95, 10^{-7}) obtains the best solution.

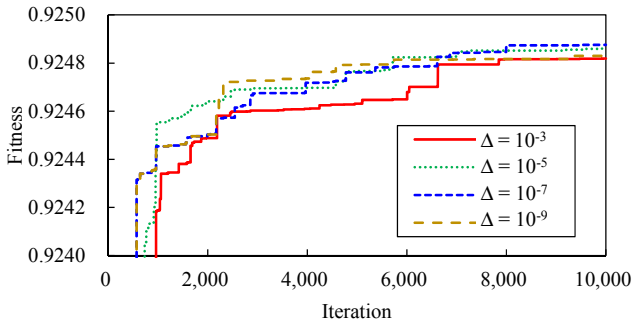

Fig. 2. Convergence analysis of DSSO.

D. Stability analysis

To analyze stability of the proposed DSSO, Fig. 3 shows the results of 100 runs of the DSSO in three cases in which (C_w, C_p, C_g) is set as (0.05, 0.85, 0.95) based on the results in Subsection IV-B, and Δ is set as four values (i.e., 10^{-3}, 10^{-5}, 10^{-7}, and 10^{-9}, respectively). From Fig. 3, the results in all cases are stable. Although the results when $\Delta = 10^{-3}$ look relatively unstable as compared to the other cases, the difference between highest and lowest fitness is still small (< 0.003).

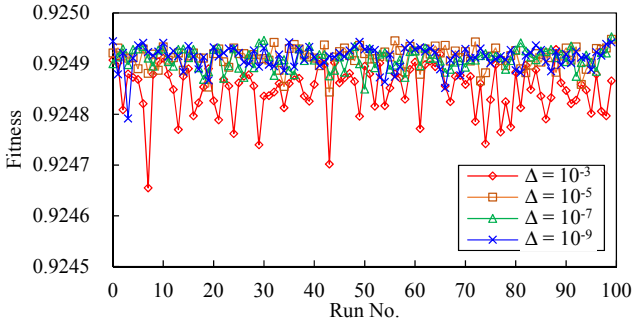

Fig. 3. Comparison of the results of 100 runs of the DSSO under four settings.

E. Comparison with classical algorithms

This subsection compares the experimental results of the proposed DSSO with classical algorithms, including SSO, PSO, GA, and CS. After a lot of experimental trials, the parameters of these algorithms are set as their respective best settings. In addition, the number of iterations in each algorithm is the same (10,000), and the population size (i.e., particles in DSSO, SSO, and PSO; chromosomes in GA; cuckoos in CS) is the same (80). The experimental results using these algorithms are compared in Table VI. From Table VI, in all cases, the DSSO performs better than the others in terms of all measures, including the best fitness, average fitness, worst fitness, and standard deviation of 100 times of experimental results.

TABLE VI.
COMPARISON OF RESULTS USING DSSO, SSO, PSO, GA, AND CS UNDER DIFFERENT NUMBERS OF PARTICLES AND ITERATIONS

Method	Best fitness	Average fitness	Worst fitness	StdDev
DSSO	0.924901	0.924863	0.924806	0.000358
SSO	0.924897	0.924860	0.924791	0.000373
PSO	0.924878	0.917930	0.851715	0.011566
CS	0.913253	0.887307	0.850666	0.014765
GA	0.924219	0.923489	0.923433	0.000190

Fig. 4 is the stability analysis of 100 times of experimental results using DSSO, SSO, PSO, GA, and CS. The boxplots for the 100 times of experimental results are shown in Fig. 5. From Fig. 4 and Fig. 5, DSSO, SSO, and CS are stable (i.e., the difference between highest and lowest fitness is small (about 10^{-4})), but PSO and GA seem relatively unstable (i.e., the difference between highest and lowest fitness is large (> 0.06)).

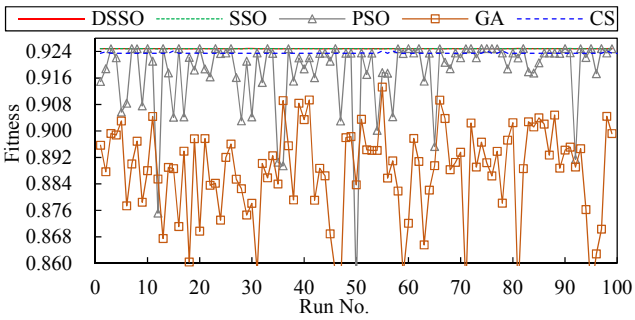

Fig. 4. Comparison of the results of 100 runs of DSSO, SSO, PSO, GA, and CS.

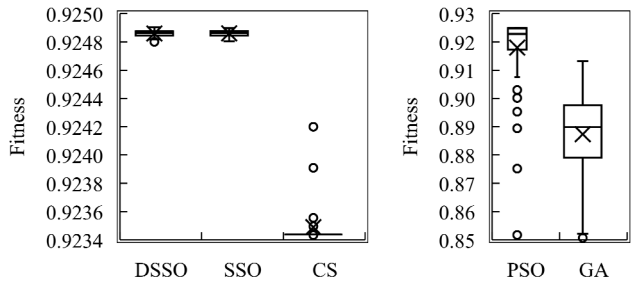

Fig. 5. Boxplot of the 100 results using DSSO, SSO, PSO, CS, and GA.

VI. Conclusion

For sustainability of agriculture, planting dryland crops in vulnerable land areas can effectively reduce the demand for irrigation water and further decrease the risk of land subsidence, which may destroy infrastructures. This work has created a mathematical programming model for the ACP that allocates a land area for planting dryland and wetland crops so that both the total profit earned and the total irrigation water amount are optimized, under some practical constraints. This work further proposes a novel DSSO to solve this problem. The DSSO improves the classical SSO with a scheme of dynamically adjusting the four probabilities that decides the operations of updating solutions. Experimental results on a real case shows that the proposed DSSO performs better than the other approaches of SI (including SSO, PSO, GA, and CS) in all combinations of parameters. This work is helpful for government agencies in agriculture or agricultural businesses in practice. With the required parameters of the proposed model, they can efficiently produce profitable ACP results for planting crops in vulnerable land areas to avoid land subsidence.

Recently, many-objective optimization problems (i.e., those with more than three or four objectives) have received much attention [34]. Therefore, a line of the future work is to consider a variety of additional objectives, e.g., optimizing labor, pollution, carbon emissions, soil, fertilizer, and uncertainty. In addition, it would be of future interest to investigate various cropping methods in the model, e.g., inter-cropping and companion cropping. It would also be interesting to consider mixed cropping methods in the same plots. With advance in development of the IoT and 5G, another line of future work is to conduct more precise analysis of the ACP through using advanced AI algorithms to analyze the data collected.

References

[1] O. Rahmati, et al., "Land subsidence hazard modeling: Machine learning to identify predictors and the role of human activities," *J. Enviro. Manage.*, vol. 236, pp. 466-480, 2019.

[2] S. I. Elmahdy, et al., "Land subsidence and sinkholes susceptibility mapping and analysis using random forest and frequency ratio models in Al Ain, UAE," *Geocarto Int.*, vol. 35, pp. 1-17, 2020.

[3] P, Jeanne, T. G. Farr, J. Rutqvist, and D. W. Vasco, "Role of agricultural activity on land subsidence in the San Joaquin Valley, California," *J. Hydrol.*, vol. 569, pp. 462-469, 2019.

[4] G. Huang, et al., "Water-saving agriculture can deliver deep water cuts for China," *Resour. Conserv. Recycl.*, vol. 154, article no. 104578, 2020.

[5] G. M. Angel, et al., "Multistage linear programming model for optimizing cropping plan decisions under the new common agricultural policy," *Land Use Policy*, vol. 48, pp. 515-524, 2015.

[6] B. F. Bassam, et al., "Optimal multi-crop planning implemented under deficit irrigation," in *Proc. of MELECON 2016*, 2016, pp. 1–6.

[7] C. Zhang, et al., "An inexact robust two-stage mixed-integer linear programming approach for crop area planning under uncertainty," *J. Clean. Prod.*, vol. 204, no. 10, pp. 489–500, 2018.

[8] T. Itoh and H. Ishii, "Fuzzy crop planning problem based on possibility measure," in *Proc. of FUZZ-IEEE 2001*, 2001, pp. 1227–1230.

[9] T. Siegfried, et al., "Multiobjective groundwater management using evolutionary algorithms," *IEEE Trans. Evol. Comput.*, vol. 13, no. 2, pp. 229–242, 2009.

[10] S. Chetty and A. O. Adewumi, "Comparison study of swarm intelligence techniques for the annual crop planning problem," *IEEE Trans. Evol. Comput.*, vol. 18, no. 14, pp. 258–268, 2014.

[11] O. Boyabatlı, J. Nasiry and H. Zhou, "Crop planning in sustainable agriculture: dynamic farmland allocation in the presence of crop rotation benefits," *Manage. Sci.*, vol. 70, no. 3, pp. 1949-2443, 2019.

[12] A. H. Gandomi and A. R. Kashani, "Construction cost minimization of shallow foundation using recent swarm intelligence techniques," *IEEE Trans. Ind. Informat.*, vol. 14, no. 3, pp. 1099-1106, 2017.

[13] W. Yeh, "Optimization of the disassembly sequencing problem on the basis of self-adaptive simplified swarm optimization," *IEEE Trans. Syst. Man Cybern. A Syst. Humans*, vol. 42, no. 1, pp. 250–261, 2012.

[14] S. Saranya and T. Amudha, "Crop planning optimization research – a detailed investigation," in *Proc. of IEEE ICACA 2016*, 2016, pp. 202–208.

[15] I. Sabzzadeh and M. Shourian, "Maximizing crops yield net benefit in a groundwater-irrigated plain constrained to aquifer stable depletion using a coupled PSO-SWAT-MODFLOW hydro-agronomic model," *Journal of Cleaner Production*, vol. 262, article no. 121349, 2020.

[16] S. K. Roy and D. De, "Genetic Algorithm based Internet of Precision Agricultural Things (IopaT) for Agriculture 4.0," *Internet of Things*, article no. 100201, 2020.

[17] X. Meng, J. Chang, X. Wang, and Y. Wang, "Multi-objective hydropower station operation using an improved cuckoo search algorithm," *Energy*, vol. 168, pp. 425-439, 2019.

[18] W. F. Gao, L. L. Huang, S. Y. Liu and C. Dai, "Artificial bee colony algorithm based on information learning," *IEEE Trans. Cybern.*, vol. 45, no. 12, pp. 2827–2839, 2015.

[19] N. Jin and Y. Rahmat-Samii, "Advances in particle swarm optimization for antenna designs: Real-number, binary, single-objective and multiobjective implementations," *IEEE Trans. Antennas Propag.*, vol. 55, no. 3, pp. 560–562, 2007.

[20] G. Ciuprina, D. Ioan, and I. Munteanu, "Use of intelligent-particle swarm optimization in electromagnetics," *IEEE Trans. Magn.*, vol. 38, no. 2, pp. 1037–1040, 2002.

[21] Q. Zhou, et al., "Cyber-physical energy-saving control for hybrid aircraft-towing tractor based on online swarm intelligent programming," *IEEE Trans. Ind. Informat.*, vol. 14, no. 9, pp. 4149–4158, 2018.

[22] H. Noory, A. Liaghat, M. Parsinejad and O. Haddad, "Optimizing irrigation water allocation and multicrop planning using discrete PSO algorithm," *J. Irrig. Drain. Eng.*, vol. 138, no. 5, pp. 437–444, 2012.
[23] M. Fereidoon and M. Koch, "SWAT-MODSIM-PSO optimization of multi-crop planning in the Karkheh River Basin, Iran, under the impacts of climate change," *Sci. Total Environ.*, vol. 63015, pp. 502-516, 2018.
[24] P. Aghelpour, H. Bahrami-Pichaghchi, and O. Kisi, "Comparison of three different bio-inspired algorithms to improve ability of neuro fuzzy approach in prediction of agricultural drought, based on three different indexes," *Comput. Electron. Agric.*, vol. 170, article no. 105279, 2020.
[25] Q. Yuan and G. Yin, "Analyzing convergence and rates of convergence of particle swarm optimization algorithms using stochastic approximation methods," *IEEE Trans. Autom. Control*, vol. 60, no. 7, pp. 1760–1773, 2015.
[26] A. Slowik and H. Kwasnicka, "Nature inspired methods and their industry applications—Swarm intelligence algorithms," *IEEE Trans. Ind. Informat.*, vol. 14, no. 3, pp., 1004–1015, 2017.
[27] C.-M. Lai, C.-C. Chiu, W.-C. Liu, and W.-C. Yeh, "A novel nondominated sorting simplified swarm optimization for multi-stage capacitated facility location problems with multiple quantitative and qualitative objectives," *Appl. Soft Comput.*, vol. 84, article no. 105684, 2019.
[28] C.-L. Huang, et al., "Simplified swarm optimization for the time dependent competitive vehicle routing problem with heterogeneous fleet," in *Proc. of IEEE CEC 2018*, 2018, pp. 1–8.
[29] W.-C. Yeh, "New parameter-free simplified swarm optimization for artificial neural network training and its application in the prediction of time series," *IEEE Trans. Neural Netw. Learn. Syst.*, vol., 24, no. 4, pp. 661–665, 2013.
[30] W. C. Yeh, "An improved simplified swarm optimization," *Knowl. Based Syst.*, vol. 82, pp. 60–69, 2015.
[31] W. C. Yeh, et al., "Simplex simplified swarm optimisation for the efficient optimisation of parameter identification for solar cell models," *IET Renew. Power Gener.*, vol. 12, no. 1, pp. 45–51, 2018.
[32] J. K. Liu, et al., "Land subsidence monitoring with a network of continuously operating GPS stations in Yunlin County, Middle Taiwan," in *Proc. of IEEE IGARSS 2014*, 2014, pp. 4247–4249.
[33] Council of Agriculture, Taiwan, *The Agricultural Statistics Yearbook 2018*. Taipei, Taiwan: Council of Agriculture, 2018.
[34] Y. Chen, et al., "Many-objective optimal power dispatch strategy incorporating temporal and spatial distribution control of multiple air pollutants," *IEEE Trans. Ind. Informat.*, vol. 15, no. 9, pp., 5309–5319, 2019.