FertilizeSmart: Exploiting IoT and Differential Evolution for Optimizing Crop Fertilization

Xu Tao, Christian Cumini, Alessio Sacco, Simone Silvestri,
Salmeron Cortasa Montserrat, Guido Marchetto

***Abstract*—Efficient crop fertilization (e.g., nitrogen) is crucial for maximizing agricultural productivity, ensuring food security, and promoting sustainable farming practices. Traditional methods, such as fixed-rate fertilizer applications or soil sampling, often result in inefficiencies, over-fertilization, and environmental harm, as they fail to account for dynamic in-season weather conditions and varying nutrient needs at different crop growth stages. In this work, we introduce FertilizeSmart, an innovative framework that optimizes crop fertilization by leveraging Internet of Things (IoT) technologies. The goal is to determine the optimal fertilization *strategy* throughout the season. To this purpose, at the core of FertilizeSmart, is an optimization problem that maximizes crop yield while minimizing the amount of fertilizer used. The crop yield in response to different timings and rates of applied fertilizer is estimated using a process-based crop simulation model, namely the Decision Support System for Agrotechnology Transfer (DSSAT). The optimization problem is then solved periodically, by an improved Differential Evolution (DE) algorithm that trades off exploration and exploitation of available solutions, throughout the crop growth cycle, following a Model Predictive Control (MPC) approach. This adaptive approach allows FertilizeSmart to respond to dynamic weather conditions and adjust fertilizer application to meet varying nutrient demands across growth stages. Moreover, we perform extensive simulation experiments and results show that FertilizeSmart significantly outperforms existing fertilizer recommendations, achieving yields approximately 20% higher while reducing fertilizer usage by up to 32% compared to the fixed application rate.**

***Index Terms*—Crop Fertilization, IoT, Differential Evolution.**

I. Introduction

Efficient crop fertilization is essential for enhancing crop productivity, minimizing farming costs, and fostering sustainable agricultural practices. Traditionally, farmers apply fixed fertilizer rates based on local field trials or historical practices, they may lead to inefficient use of resources and potential over-fertilization, which can harm both soil quality and surrounding ecosystems [1]. Several fertilization management techniques and tools have been developed to determine more precise fertilizer strategies. Common approaches include soil nutrient availability testing and plant tissue testing [2], which are used to develop in-season crop fertilizer strategies. Given the time-consuming on-site and laboratory process, soil and plant testing may result in *low adoption rates*. In addition, they may provide *poor accuracy* in determining an optimal fertilization strategy, since they do not account for the dynamic effects of weather and soil nutrient losses and availability [3].

Several works have shown that in-season weather data, such as rainfall, temperature, and solar radiation, is crucial in improving the prediction accuracy of crop fertilizer demands [4], [5]. By incorporating the in-season weather data, process-based crop simulation models, such as Decision Support System for Agrotechnology Transfer (DSSAT) [6], [7] have been developed, and have become ubiquitous tools in crop growth simulations worldwide. The crop models within DSSAT require inputs such as daily weather data, soil profile information, cultivar information, and specified crop management practices. These inputs combined with model calibration are vital for accurately simulating crop growth dynamics and predicting the outcomes of various management strategies. However, despite the potential and flexibility offered by DSSAT, the simulator is not designed to optimize fertilization strategies. Further, DSSAT requires as input the entire in-season weather data, typically derived from historical records [5]. Although the most recent version of DSSAT [8] is able to adjust the yield prediction with in-season weather, it does not adjust the fertilizer strategy accordingly.

In addition to adapting to dynamic in-season weather conditions, accurately determining the rates of fertilizer application across different crop growth stages presents a significant challenge. Firstly, crop yield response to the quantity of fertilizer application is not monotonically increasing nor plateaus, as over-fertilization can decrease yield and harm the environment [9]. Moreover, fertilizer requirements (i.e., nitrogen) fluctuate in different growth stages [10]. As a result, decisions made at one point may not lead to the optimal long-term fertilization strategy if they fail to account for the cumulative effects of nutrient distribution over the entire growth cycle.

To address the aforementioned challenges, we propose FertilizeSmart, a framework to determine an *adaptive strategy* for crop fertilization, leveraging Internet of Things (IoT) technologies and Differential Evolution (DE) techniques. At the core of FertilizeSmart is an optimization problem, which aims at maximizing crop yields while minimizing the amount of fertilizer used, also known as Agronomically Optimum Nitrogen Rate (AONR). To solve this problem, inspired by Model Predictive Control (MPC) [11], FertilizeSmart adap-

Xu Tao (xu.tao@uky.edu), Simone Silvestri (silvestri@cs.uky.edu) are in Computer Science at the University of Kentucky, USA. Salmeron Cortasa Montserrat (msalmeron@uky.edu) is in Plant and Soil Sciences at the University of Kentucky. Christian Cumini (s305399@studenti.polito.it), Alessio Sacco (alessio_sacco@polito.it), and Guido Marchetto (guido.marchetto@polito.it) are in Automatica e Informatica at Politecnico di Torino, Italy.

ISBN: 978-3-903176-71-3

tively adjusts the fertilizer application rate by solving the optimization problem periodically over the crop growth cycle. Here, the fertilizer application dates represent the finite horizon time steps of MPC. At each application date, in-season weather data acquired from IoT technologies are used to inform the decision-making process. The system leverages an improved DE algorithm to determine an optimal fertilizer strategy by strategically tradeoff the exploration and exploitation of available solutions. It begins by generating an initial population of potential solutions, each representing a fertilization strategy containing the fertilizer application rates over the remaining growth cycle. FertilizeSmart then iteratively improves these solutions through an optimized evolutionary processes—mutation, crossover, and selection—based on fitness values, which reflect the gain calculated by the difference between the crop yield and fertilizer costs. The DSSAT crop simulation tool is integrated with the DE algorithm to estimate yields, by incorporating the current available and estimated in-season weather data. Following the MPC approach, only the first element (fertilizer rate of the current date) of the resulting fertilizer strategy is implemented, while future applications will be determined at the next decision cycle, when more weather data will be available. This iterative process allows FertilizeSmart to continuously adapt to changing weather conditions and unforeseen events, ensuring a dynamic and resilient fertilization strategy. Consequently, Fertimilzer not only improves crop yields but also reduces the environmental footprint of fertilization practices.

The main contributions of this paper are:

1) We formulate a novel optimization problem aimed at determining an optimal fertilization strategy.
2) We propose FertilizeSmart, a framework that integrates IoT technologies with a Differential Evolution (DE) algorithm to solve this optimization problem.
3) We conduct extensive experiments, and the results demonstrate that FertilizeSmart outperforms state-of-the-art approaches, achieving yields approximately 20% higher while reducing fertilizer usage by up to 32%.

II. Problem Formulation

Fertilizers are usually applied on several dates $\mathcal{D}$ across the crop growth cycle. Let x_d represent the amount of fertilizer applied on date d, and B denote the budget constraint for the amount of total fertilization used. Denote the crop yield as $Y = y(X|\mathbf{A})$, where $\mathbf{A}$ represents the weather data and the vector $X = [x_1, x_2, ..., x_{|\mathcal{D}|}]$ is the *fertilizer strategy* across all the application dates. Our objective is to find the optimal sequential fertilizer strategy that maximizes crop yield while minimizing the fertilizer utilization, leveraging the available weather data, and not exceeding the budget of B.

$$X^* = arg \max_{X \in [0,B]^{|D|}} \min_{\hat{B} \leq B} Y : y(X|\mathbf{A}) \quad \text{s.t.} \sum_{d \in [1,..,|\mathcal{D}|]} x_d \leq \hat{B} \tag{1}$$

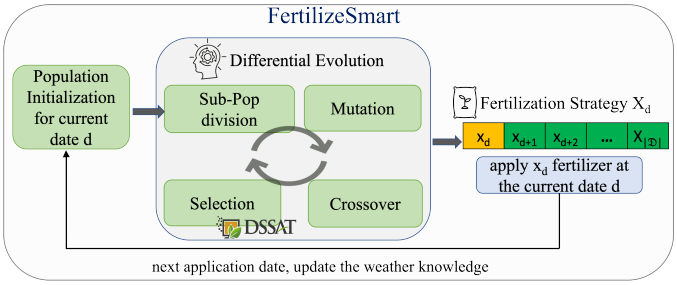


Fig. 1. FertilizeSmart Framework Overview

This problem is hard to solve. First, as noted in [9], crop yields are complex to predict, they depend on a variety of factors, and do not increase linearly with the amount of fertilizer applied. Over-fertilization does not improve yield, and damages the environment. Second, the amount of fertilizer needed vary at different crop growth stages, thus simple strategies such as fertilizing only at a single date or uniformly fertilizing throughout the season, achieve poor performance. Finally, weather conditions play a crucial role in determining fertilizer needs, making it more challenging to optimize the in-season applications, even for the same fields. Consequently, a strategy determined at the beginning of the season, based on weather forecast, would be far from the optimum, and in-season weather needs to be considered.

III. Proposed Solution: FertilizeSmart

A. Overview

We introduce FertilizeSmart, a framework for optimizing the crop fertilizer strategy, outlined in Figure 1. FertilizerSmart draws the concept of MPC to provide an *adaptive* fertilizeration optimization approach. MPC is a technique commonly used in control theory and engineering for controlling dynamic systems [11]. In MPC, the optimization problem is solved over a finite time horizon at subsequent decision steps, by optimizing a control sequence. Specifically, at each decision step, the optimization considers both the current state of the system and a prediction of how the system will evolve over the horizon. Based on this information, MPC selects an optimal action sequence over the entire time horizon. MPC applies the horizon control principle, i.e., only the first control action of the optimized sequence will be implemented. As time progresses, the horizon recedes, and the optimization problem is solved again based on the updated state of the system. This process is repeated at each decision step, allowing for control adjustments as time progresses while considering future system behavior.

Similarly, FertilizerSmart solves the crop fertilization optimization problem over the crop growth cycle (finite time horizon), by optimizing a fertilizer strategy (control sequence). The strategy contains the amount of fertilizer to be provided at each future fertilizer application date, $d \in \mathcal{D}$ (decision steps). At each decision date, FertilizeSmart considers both current available in-field weather data and a prediction of weather conditions of the remaining growing cycle.

ISBN: 978-3-903176-71-3

1) IoT-enabled weather data collection: FertilizeSmart exploits IoT-enabled digital monitoring of various weather parameters in the crop field, including solar radiation, air temperature, precipitation, dew point, wind speed, and humidity to collect weather conditions [12]. These data can be collected from sensors deployed across the field. Multiple approaches have been proposed for this purpose using technologies such as drones [13] and low power wide area network (LPWAN) [14]. The gathered information, along with fertilizer strategies, serves as input for the DSSAT crop simulator to estimate crop yield during the optimization process, as described below.

2) DE-based fertilizer strategy optimization: FertilizeSmart employs a DE-based algorithm to determine a fertilization strategy that maximizes the crop yield while minimizing the total fertilizer usage. DE offers an efficient technique for guiding decision-making in addressing global numerical optimization challenges. The main idea of the DE algorithm is to start from a randomly generated initial population and the individuals within the population are iteratively improved through an evolutionary process that balances exploration and exploitation. Specifically, FertilizeSmart initiates a population with random fertilizer strategies, each representing a potential solution as a vector containing the quantity of fertilizer to be applied across the remaining application dates.

Unlike traditional approaches in DE algorithms that evolve a single population, FertilizeSmart innovatively divides the parent (initialization) population into sub-populations based on their fitness values, calculated from a fitness function. Each sub-population undergoes three key operations: mutation, crossover, and selection. FertilizerSmart applies different *mutation* strategies to different sub-populations, generating mutant vectors based on each individual's proximity to the global optimal solution. *Crossover* operation generates new candidate solutions by combining the elements of different parent solutions. It promotes diversity and allows exploration of the search space. Subsequently, *selection* determines which solutions survive and proceed to the next generation based on their fitness value. This value is calculated through a fitness function that takes into account the yield and amount of fertilizer provided by a solution. FertilizeSmart employs the DSSAT crop simulation tool [15] estimating the yield incurred by each candidate solution during the selection process, incorporating the information of the current and estimated future weather data within the crop growth cycle. The process repeats a certain number of generations. Finally, the individual (candidate solution) with the highest fitness value is selected as the best fertilizer strategy for the current decision date.

We represent the chosen fertilization strategy at the current decision date d as a vector $X_d = [x_d, x_{d+1}..., x_{|\mathcal{D}|}]$. Recall that this is the determined sequence of fertilizer allocation across all the remaining dates starting from the current date d. We then apply the receding horizon control principle and only apply x_d fertilizer to the current date d. As time progresses to the next application date $d+1$, the real weather data of the period between the previous date d and application date $d+1$ becomes available. We update the weather knowledge, as well as the remaining available budget, and repeat the optimization process. This iterative approach, guided by real-time weather data and estimated future weather data, ensures that we dynamically adjust our fertilization strategy to adapt to real-time weather conditions and reduce the uncertainty simulating crop growth and yield in DSSAT. The pseudo-code of FertilizeSmart is illustrated in Algo. 1, as detailed below.

B. Parameter and Population Initialization

The input parameters for FertilizerSmart include the total fertilizer budget B, a set of potential fertilizer application dates $\mathcal{D}$, and parameters for the differential evolution algorithm, such as the population size Np, the maximum number of generations G_{max}, the sub-population division ratio ϵ, two normalization parameters α and β, and two constants γ and δ. The past weather data from the sowing to the current decision date is stored in the matrix $\mathbf{A}_c$, while the estimated future weather data is stored in $\mathbf{A}_e$ [1]. The differential evolution process is performed across all possible application dates (lines 3 onwards). For each date d, the initial population $\mathcal{P}_d^G$ is created as an empty set, with both F_m^G and Cr_m^G parameters set to γ (line 4). F_m^G is a parameter used to generate and update the scaling factor of the mutation operation, while Cr_m^G is another parameter used to generate and update the crossover rate at each generation. Next, we randomly generate the initial population with the size of N_p (lines 5-7). Each i-th individual (potential fertilizer strategy) in the current generation G for the current date d is represented by a vector $X_d^{G,i}$ of size $|\mathcal{D}| - d + 1$, denoted as $X_d^{G,i} = (x_d^{G,i}, x_{d+1}^{G,i}, ..., x_{|\mathcal{D}|}^{G,i})$, where $x_d^{G,i}$ signifies the fertilizer quatity to be applied on decision date d. All elements of the vector are initially generated randomly from (0, 1) (line 6). Subsequently, each element $x_j^{G,i}$ within vector $X_d^{G,i}$ is scaled to fit in the remaining budget B, following Eq. 2 (line 7). The resulting vector $X_d^{G,i}$ is then added to the initial population $\mathcal{P}_d^G$.

$$x_j^{G,i} = \frac{x_j^{G,i}}{\sum_{k=d}^{|\mathcal{D}|} x_k^{G,i}} \cdot B \tag{2}$$

C. Differential Evolution

The differential evolution process entails G_{max} generations (lines 8-27). Each generation involves the computation of fitness values for individuals in the population. The fitness value of individual $X_d^{G,i}$ is denoted as $f(X_d^{G,i})$, which is calculated by the fitness function outlined in Eq. 3.

$$f(X_d^{G,i}) = y(X_d^{G,i}|\mathbf{A}_c, \mathbf{A}_e) \cdot \alpha - \sum_{j=d}^{|\mathcal{D}|} x_j^{G,i} \cdot \beta \tag{3}$$

where, $y(X_d^{G,i}|\mathbf{A}_c, \mathbf{A}_e)$ represents the predicted crop yield given the weather data from the sowing date to the current date $\mathbf{A}_c$ and the estimated future weather data $\mathbf{A}_e$, and the fertilizer strategy $X_d^{G,i}$. The yield prediction is obtained from

[1]Examples of the considered weather data include solar radiation, daily maximum and minimum temperatures, precipitation, dew point, wind speed, and relative humidity.

ISBN: 978-3-903176-71-3

Algorithm 1: FertilizeSmart

Input: $B, \mathcal{D}, N_p, G_{max}, \epsilon, \mathbf{A}_c, \mathbf{A}_e, \alpha, \beta, \gamma, \delta$
$G \leftarrow 1$;
for $d = 1$ *to* $|\mathcal{D}|$ **do**
  $\mathcal{P}_d^G = \emptyset$; $F_m^G \leftarrow \gamma$; $Cr_m^G \leftarrow \gamma$;
  /* Population Initialization */
  for *i =1 to* $|N_p|$ **do**
    Randomly create a vector $X_d^{G,i} = (x_d^{G,i}, x_{d+1}^{G,i}, ..., x_{|\mathcal{D}|}^{G,i})$ st. $\forall x_j^{G,i} = rand(0,1)$;
    Scale the vector $X_d^{G,i}$ with respect to the remaining budget B according to Eq. (2) and add it to the population $\mathcal{P}_d^G$;
  /* Differential Evolution Process */
  while $G < G_{max}$ **do**
    Calculate the fitness value $f(X_d^{G,i})$ of every individual fertilizer strategy $X_d^{G,i}$ in population $\mathcal{P}_d^G$ according to Eq. (3);
    Rank the population in descending order according to the fitness value;
    for *i =1 to* $|N_p|$ **do**
      $F_i = Gaussian(F_m^G, \delta)$;
      $Cr_i = Guassian(Cr_m^G, \delta)$;
      /* Mutation with sub-populations */
      Devise the neighbors and remote relatives for individual $X_d^{G,i}$;
      if $i \leq \epsilon \cdot N_p$ **then**
        Generate mutation vector $V_d^{G,i}$ with Eq. (4);
      if $\epsilon \cdot N_p < i \leq 2\epsilon \cdot N_p$ **then**
        Generate mutation vector $V_d^{G,i}$ with Eq. (5);
      if $2\epsilon \cdot N_p < i \leq 3\epsilon \cdot N_p$ **then**
        Generate mutation vector $V_d^{G,i}$ with Eq. (6);
      /* Crossover operation */
      Create an empty trail vector $U_d^{G,i}$ of $V_d^{G,i}$;
      for $j = d$ *to* $|\mathcal{D}|$ **do**
        $u_j^{G,i} = \begin{cases} v_j^{G,i} & \text{if } rand_{j,i} \leq Cr \text{ or } j = i_{rand} \\ x_j^{G,i} & \text{otherwise} \end{cases}$
      Put trail vector $U_d^{G,i}$ into trail population U_d^G;
    /* Selection Operation */
    for *i=1 to* N_p **do**
      $X_d^{G+1,i} = \begin{cases} U_d^{G,i} & \text{if } f(U_d^{G,i}) \leq f(X_d^{G,i}) \\ X_d^{G,i} & \text{otherwise} \end{cases}$;
    Update F_m and Cr_m according to Eqs. (9), (10), (11) and (12);
    G=G+1;
  Select the best solution $X_d^{G,best}$ in $\mathcal{P}_d^G$ and apply $x_d^{G,best}$ fertilizer to the current application date d;
  Update the current weather knowledge $\mathbf{A}_c$ and estimated weather data $\mathbf{A}_e$, and available budget $B = B - x_d^{G,best}$;
  d=d+1;

the crop simulation tool DSSAT. Here, α and β are used as normalization parameters that can be tuned to prioritize higher yield versus lower budgets.

Subsequently, we rank the population in *descending order* according to the fitness value. The subsequent section illustrates sub-population splitting from the ranked population and the process of mutation and crossover.

1) Sub-population splitting: Most DE algorithms aim for an efficient balance between exploring new solutions and exploiting promising ones within the population to enhance overall performance. To improve the search efficiency, we divide the population into four sub-populations, drawing inspiration from paper [16]. Each sub-population is assigned a distinct search task. The first sub-population comprises the top $\epsilon \cdot N_p$ individuals in terms of fitness, representing solutions closer to the global optimum. Therefore, this sub-population focuses on exploitation, intensively refining already promising solutions. In contrast, the second sub-population encompasses individuals ranked between the $\epsilon \cdot N_p$ and $2\epsilon \cdot N_p$, serving to balance exploration and exploitation throughout the evolutionary process. The third sub-population is composed of individuals between the $2\epsilon \cdot N_p$ and $3\epsilon \cdot N_p$, potentially distant from the global optimum. Here, exploration takes precedence, with individuals undergoing significant perturbations to escape local optima and approach towards more favorable regions. All these three sub-populations will go through a mutation and crossover process. Lastly, the remaining $(1 - 3\epsilon) \cdot N_p$ individuals form the fourth sub-population directly inherited from the parent population, fostering diversity and preventing premature convergence to suboptimal solutions. The four sub-populations then undergo a mutation process.

2) Mutation with different strategies: In the mutation process, we introduce the concepts of *neighbors* and *remote relatives* for each individual within the current generation. These serve as distinct search spaces for different mutation strategies. Specifically, for every individual, we select the top $N_p/4$ of the closest individuals in terms of Euclidean distance within the entire population as its neighbors. Similarly, we also select the top $N_p/4$ of the furthest individuals as its remote relatives. These groups are pivotal for both local neighbor mutation and global remote relative mutation, which aid in generating mutant vectors for the respective individuals.

Different mutation strategies are employed for sub-populations. Specifically, the *DE/current-to-bestNeighbor/1* mutation strategy is applied to the first sub-population with superior individuals, outlined as in Eq 4 (lines 14-15).

$$V_d^{G,i} = X_d^{G,i} + F_i \cdot \left(X_d^{G,best_n} - X_d^{G,i}\right) + F_i \cdot \left(X_d^{G,R1} - X_d^{G,R2}\right) \quad (4)$$

Where $V_d^{G,i}$ represents the mutation vector generated for individual $X_d^{G,i}$, $X_d^{G,best_n}$ denotes the best individual among the *neighbors* of $X_d^{G,i}$. Additionally, $X_d^{G,R1}$ and $X_d^{G,R2}$ represent two distinct individuals randomly chosen from the current population, both of which differ from the target vector $X_d^{G,i}$. The scaling factor F_i governs the magnitude of mutation, dynamically generated at each generation and it is used across all the mutation strategies. Details on its adaptive generation are provided in the following section III-C5.

This mutation strategy aims to guide the target vector $X_d^{G,i}$ towards a superior or similar individual within its neighbors, while introducing a small perturbation to facilitate local exploration. Consequently, the superior individuals within the first sub-population can effectively exploit the current search space, thereby enhancing the likelihood of discovering either local or global optimal solutions.

ISBN: 978-3-903176-71-3

In the case of the second sub-population with medium individuals, we employ the *DE/current-to-best/1* strategy, outlined in Eq. 5 (lines 16 -17).

$$V_d^{G,i} = X_d^{G,i} + F_i \cdot \left(X_d^{G,best_p} - X_d^{G,i}\right) + F_i \cdot \left(X_d^{G,R1} - X_d^{G,R2}\right) \quad (5)$$

Where $X_d^{G,best_p}$ represents the best individual within the entire population. Similar to Eq. (4), X_{R1}^G and X_{R2}^G are two distinct individuals randomly selected from the current population, both of which differ from the target vector X_i^G. This strategy strikes a balance between exploration and exploitation, facilitating effective global search.

Concerning the third sub-population, *DE/current-to-bestRelative/1* strategy is adopted as in Eq. 6 (lines 18-19).

$$V_d^{G,i} = X_d^{G,i} + F_i \cdot \left(X_d^{G,best_r} - X_d^{G,i}\right) + F_i \cdot \left(X_d^{G,R1} - X_d^{G,R2}\right) \quad (6)$$

Where $X_d^{G,best_r}$ denotes the best individual among the *remote relatives* of $X_d^{G,i}$. With this mutation strategy, the target vector $X_d^{G,i}$ undergoes a significant perturbation, aiding its attraction towards a superior individual within the current population. This is facilitated by a larger difference vector, $X_d^{G,best_r} - X_d^{G,i}$, along with an additional random difference vector. Such an approach empowers individuals with lower fitness to explore more extensively within the search space, mitigating the risk of becoming trapped in local regions.

The fourth sub-population stays unchanged without going through the mutation process, it will be directly inherited by the next generation.

3) Crossover operation: After mutation, a crossover operator is performed on the target vector $X_d^{G,i}$ to generate a trail vector (offspring) $U_d^{G,i} = (u_d^{G,i}, u_{d+1}^{G,i}, ..., u_{|\mathcal{D}|}^{G,i})$. Each j-th element in $U_d^{G,i}$ can be set as the corresponding element in the mutant vector or in the target vector $X_d^{G,i}$, as described by Eq. 7 (lines 21 -22).

$$u_j^{G,i} = \begin{cases} v_j^{G,i} & \text{if } rand_j^{G,i} \leq Cr_i \text{ or } j = i_{rand} \\ x_j^{G,i} & \text{otherwise} \end{cases} \quad (7)$$

Where $rand_j^{G,i}$ represents a random number sampled from the interval [0,1]. The parameter $Cr_i \in [0,1]$ denotes the crossover rate, dictating the proportion of individuals inherited from the mutant vector. Cr_i is adaptively generated at each generation, further elaboration on its generation is illustrated in the section III-C5. Additionally, i_{rand} is a randomly selected integer within the range $[d, |\mathcal{D}|]$, ensuring that at least one element of the trial vector is inherited from the mutant vector.

Finally, the trial vector $U_d^{G,i}$ is added to the trail population U_d^G (line 23).

4) Selection operation: The selection operator determines which vectors, either the target or the trail, proceed to the next generation based on their fitness values. Valid vectors that remain within the budget (i.e., with a sum less than or equal to 1) and provide higher fitness values will be selected to advance to the next generation, according to Eq. (8).

$$X_d^{G+1,i} = \begin{cases} U_d^{G,i} & \text{if } f(U_d^{G,i}) \leq f(X_d^{G,i}) \wedge \sum_{j=d}^{|D|} u_j^{G,i} \leq 1 \\ X_i^G & \text{otherwise} \end{cases} \quad (8)$$

Where $f(\cdot)$ is the fitness function defined in Eq. (3).

After $G_{\max}$ generations, the best solution, represented by $X_d^{Gmax,\text{best}}$, is selected from the population $\mathcal{P}_d^{Gmax}$. This solution guides us in determining the optimal fertilizer quantity, denoted as $x_d^{Gmax,\text{best}}$, to be applied on the current application day d. When the time progresses to the next application date, we update the knowledge of the weather conditions including the current weather data and estimated future in-season weather data, and the available budget. The DE progress repeats.

5) Adaptive Parameter Approach: We introduce the adaptive parameter adaptation to both the scaling factor F_i used in the mutation operation and the crossover rate Cr_i.

At every generation G, both the scaling factor F_i and the crossover rate Cr_i for each individual $X_d^{G,i}$ are independently generated from Gaussian distributions (line 12). These distributions have expected values of F_m and Cr_m, respectively, with a standard deviation δ, as outlined below.

$$F_i = Gaussian(F_m, \delta); Cr_i = Gaussian(Cr_m, \delta)$$

If F_i exceeds 1, it is truncated to 1. Conversely, if F_i falls below 0, it is regenerated. Similarly, for Cr_i, if its value exceeds 1, it is set to 1. Otherwise, if Cr_i drops below 0, it is set to 0.

F_m^G and Cr_m^G are initially set to λ and then updated at the end of each generation, as described in Eq. (9) and Eq. (10), respectively.

$$F_m^{G+1} = w_F \cdot F_m^G + (1 - w_F) \cdot F_i \quad (9)$$

$$Cr_m^{G+1} = w_{Cr} \cdot Cr_m^G + (1 - w_{Cr}) \cdot Cr_i \quad (10)$$

Here, w_F and w_{Cr} represent random weight factors. In our implementation we use $0.8 + 0.2 \cdot rand(0,1)$, as recommended in [17].

However, if the parent $X_d^{G,i}$ survives into the next generation, it means that the scaling factor F_i and the crossover rate Cr_i are somewhat invalid. In such instances, F_m^{G+1} is updated according to Eq (11) and Cr_m^{G+1} is updated as in Eq (12).

$$F_m^{G+1} = c_F \cdot F_m^G + (1 - c_F) \cdot rand(0,1) \quad (11)$$

$$Cr_m^{G+1} = w_{Cr} \cdot Cr_m^G + (1 - w_{Cr}) \cdot rand(0,1) \quad (12)$$

Where c_F and w_{Cr} are randomly weight factors. In our implementation use $0.5 \cdot rand(0.1)$, as suggested in [17].

D. Time Complexity of FertilizerSmart

In the following, we evaluate the time complexity of FertilizerSmart. Initially, calculating the fitness value for each fertilizer strategy in the population involves estimating the yield using the DSSAT simulator. Since the inner functionality of the simulator is unknown, we denote this complexity as $\mathcal{O}(C)$, for ease of exposition. Therefore, the total time required for all the individuals within the population is $\mathcal{O}(C \cdot N_p)$. Then, Sorting all individuals within the population based on their fitness value incurs time complexity of $\mathcal{O}(N_p \cdot \log(N_p))$. Following this, pairwise distances between individuals are computed to determine neighbor and remote relatives, incurring a time

ISBN: 978-3-903176-71-3

complexity of $\mathcal{O}(D \cdot N_p \cdot (N_p - 1)/2)$. Subsequently, distances within the first and third sub-populations are sorted, requiring $\mathcal{O}(2\epsilon \cdot N_p \cdot \log(N_p))$ time. These steps repeats over $G_{\max}$ generations, the overall time complexity of FertilizerSmart is $\mathcal{O}(Gmax \cdot (C \cdot N_p + \mathcal{O}(N_p log(N_p)) + \mathcal{O}(D \cdot N_p(N_p - 1)/2) + \mathcal{O}(2\epsilon N_p \cdot log(N_p))))$, which can be simplified as $\mathcal{O}(G_{\max} \cdot (C \cdot N_p + D \cdot N_p^2))$.

IV. Experiments

A. Experimental Setup

1) DSSAT Setup: The Decision Support System for Agrotechnology Transfer (DSSAT) is a software application with dynamic simulation models for over 42 crops. It is complemented by various utilities and applications that handle data related to weather, soil, genetics, crop management, and observational experiments. DSSAT also provides example datasets for all its crop models. These simulation models predict crop growth, development, and yield by analyzing the interactions between soil, plants, and atmospheric conditions. We use the DSSAT v.4.8.2 [6], [8] for the following experiments.

The DSSAT simulation environment provides a set of calibrated experimental profiles. We opted to utilize one such experiment, namely FLSC8101 for the crop of maize. We modified the planting date to May 1 and other soil, crop coefficients, and management input settings were left unmodified. The typical growth cycle for maize spans around 120 days. Our experiments explore nitrogen fertilizer strategies, pivotal for maize growth and yield.

2) Weather Dataset: We use weather data from the NASA Prediction of Worldwide Energy Resources (POWER) project [18]. POWER provides over 200 solar and meteorological near real-time datasets. In our experiments, we utilize the daily weather data spanning from 2020 to 2023 in Florence, South Carolina. The daily weather data includs solar radiation, daily maximum and minimum temperatures, precipitation, dew point, and wind speed. [2].

3) FertilizeSmart Setup: The normalization factors used in the fitness function are set to $\alpha = 0.165$ and $\beta = 1.1$. For generating the mutation scaling factor and crossover rate, the parameters γ and δ are set to 0.5 and 0.1, respectively. The maximum number of generations $G_{\max}$, is set to 400.

B. Benchmark Description

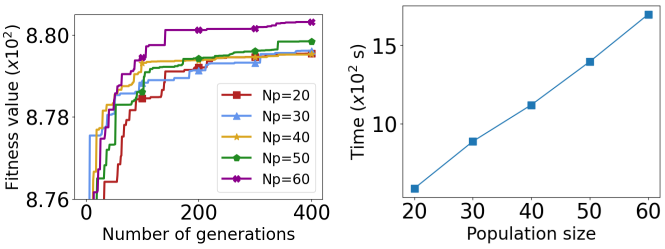
We consider a benchmark strategy based on the fertilization best practices described in [19]. In particular, for corn cultivars, the growth phase that requires the most nitrogen is around the last vegetative stages, or around 30-40 days from sowing. This is when most of the fertilizer should be applied. The grain-filling stage, towards the end of the growth process, is also a phase when fertilizer is needed. Based on this, we model the rates of fertilizer application as a normal distribution centered on the 35th day with a standard deviation of 10 days. This approach allocates 70% of the nitrogen budget early on, with the remaining 30% uniformly distributed after the 75th day to support the grain-filling stage. All are then scaled to fit the budget.

[2]In our experiments, we consider the ground truth data for future weather. Several weather prediction models can be adopted by FertilizeSmart, which is out of the scope of this paper.

C. Experiment Results

Experiment I: Impact of the population size (N_p) on the qualify of the solution. We first consider the evolutionary aspects and convergence behaviors of FertilizerSmart. Specifically, we investigate how the population size (N_p) influence the quality of solutions, measured by the fitness value derived from Eq. (3) and computational efficiency (i.e., execution time). We vary the population size (N_p) from 20 to 75 individuals and iterate through 400 generations. Leveraging weather data from 2023, we distribute 9 application dates evenly throughout the growth period. They are $0, 14, 28, 42, 56, 70, 84, 94, 105$ days after sowing. In addition, we set fertilizer budget of 200 Kg/ha.

Figure 2 (a) shows the fitness value of the best solution during the execution of FertilizerSmart, under different values of the population size. As expected, FertilizerSmart finds better solutions with larger population sizes. Moreover, the approach is able to converge towards good solutions after 300 generations across all considered population size. Figure 2 (b) shows the computational time versus the population size. Note that, although the computational complexity is upperbounded by a quadratic function in N_p, the experimental results show a linear behavior for the considered value of the population size.



(a) (b)

Fig. 2. Experiment I: (a) Fitness value versus population size and number of generations; (b) Execution time versus population size.

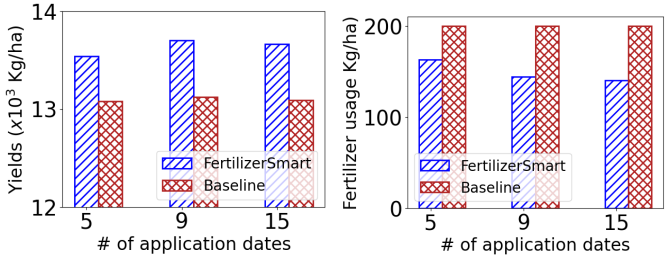
Experiment II: Impact of the number of application dates. This set of experiments shows the impact of the number of application dates on the yields. We study three different options with 5, 9, and 15 application dates, respectively. The distribution of the dates are shown in Table I and we guarantee the crucial period for nitrogen application are covered. However, the number of applications and the dates are input for FertilizeSmart that can be adjustted to the farmers preferences and the budget. We set the budget as 200 Kg/ha and the weather data is from 2023. More application dates means more decision variables the algorithm needs to learn, thus requiring a bigger population size. To be fair comparisons, we set the population size to 20, 40, and 60 for the application dates 5, 9, and 15, respectively.

Figure 3 illustrates that FertilizerSmart achieves superior yields with reduced nitrogen usage across all application dates

ISBN: 978-3-903176-71-3

TABLE I
DISTRIBUTION OF NITROGEN APPLICATION DATES

# of application dates	application dates (offset to sowing dates)
5	[0, 28, 55, 83, 105]
9	[0, 14, 28, 42, 56, 70, 84, 94, 105]
15	[0, 14, 21, 28, 35, 42, 49, 56, 63, 70, 77, 84, 91, 98, 105]



(a) (b)

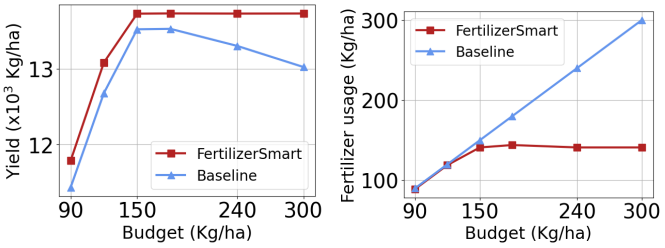
Fig. 3. Experiment II: (a) Yields and (b) Fertilizer usage varying the number of application dates.

compared to the baseline approach. Specifically, Figure 3 (a) shows that increasing the number of nitrogen application dates from 5 to 9 results in higher yields. This improvement is attributed to the more precise fertilizer strategy, as additional application dates are scheduled during the rapid growth period from day 30 to day 50, making the strategy more adaptive to weather conditions. Notably, FertilizerSmart demonstrates a more significant yield increase compared to the baseline approach. However, further increasing the application dates beyond 9, does not lead to additional yield improvements due to the saturation of nitrogen needs.

In contrast, Figure 3 (b) reveals that FertilizerSmart consistently achieves higher yields with all application schedules, using less fertilizer overall. The adaptability and precision fertilization technique offered by FertilizerSmart allows it to save nitrogen as the number of application dates increases from 5 to 9, after which fertilizer usage remains stable even with additional application dates. Based on this observation, we select 9 application dates for the subsequent experiments.

Experiment III: Scalability over budget. In this set of experiments, we assess the impact of fertilizer budgets concerning yields by varying the maximum nitrogen budget from 90 Kg/ha to 300 Kg/ha. We utilize the weather data from 2023, and set the population size of 50, as it provides the best tradeoff between performance and computation time as shown in Experiment I, Figure 2 (b).

Figure 4 demonstrates the superior performance of FertilizerSmart in achieving higher yields with reduced nitrogen usage across all the fertilizer budgets, when compared to the baseline method. Specifically, as depicted in Figure 4 (a), the yield steadily improves with increasing fertilizer budget up to a threshold of 150 Kg/ha for both approaches. Beyond this point, applying excess nitrogen becomes counterproductive, potentially harming yields. FertilizerSmart adeptly manages this balance, reaching yield saturation after the budget hits



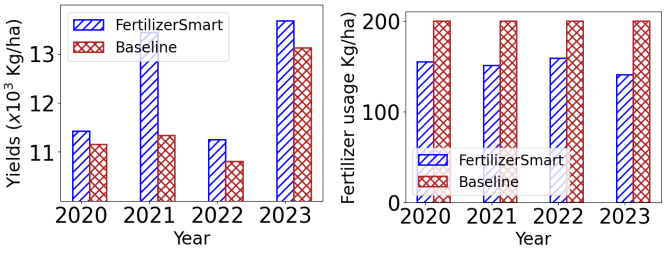
(a) (b)

Fig. 4. Experiment III: (a) Yields and (b) Fertilizer usage varying Budget.

150 Kg/ha, thanks to its incorporation of fertilizer usage optimization within the fitness function (Eq. 3). This optimization empowers the evolutionary process to make more informed decisions, resulting in maximum yields with minimal fertilizer consumption. Conversely, the baseline approach sees yield decreasing beyond 150 Kg/ha, as it adheres to conventional practices that may not necessarily provide the optimal strategy, due to a lack of adaptation to the current weather condition and a lack of learning processing. This is also evident in Figure 4 (b), where FertilizerSmart consistently utilizes significantly less fertilizer, up to 50% less than the available budgets compared to the baseline approach.

Overall, these findings underscore the importance of precision fertilizer techniques with weather adaptation, such as FertilizerSmart, which dynamically adjust fertilization strategies to maximizing crop productivity while minimizing the fertilizer usage.

Experiment IV: Performance over various years. In this set of experiments, we evaluate the performance of FertilizerSmart across the years from 2020 to 2023 with a population size of 50. The variability in weather conditions of different years can influence fertilizer strategy decisions and consequently impact yields. For these experiments, we set the budget at 200 Kg/ha, with a population size of 40 and 400 generations.



(a) (b)

Fig. 5. Experiment IV: (a) Yields and (b) Fertilizer usage across different years.

Figure 5 shows that FertilizerSmart consistently outperforms the baseline approach in terms of both yield and fertilizer usage across all the years. For instance, in 2021, FertilizerSmart achieves yields approximately 20% higher than the baseline, as shown in Figure 5 (a), while reducing fertilizer usage by up to 32%, as depicted in Figure 5 (b). These results demonstrate again the ability of FertilizerSmart to find the optimal fertilizer

ISBN: 978-3-903176-71-3

strategy by adapting the weather condition changes, while the baseline approach employs the fixed recommended strategy.

V. Related Work

Crop fertilization has evolved from conventional methods, like uniform application and soil testing, to modern precision agriculture techniques that improve efficiency and sustainability. Traditional approaches, though simple, often neglect spatial variability within fields, causing nutrient imbalances and resource inefficiencies [20]. Site-Specific Nutrient Management (SSNM) [21] offers a more targeted solution by optimizing nutrient application based on detailed assessments of soil and crop needs, improving yields and reducing waste. However, SSNM can be complex and time-consuming. Process-based simulation tools like DSSAT [8] are more efficient, accounting for environmental conditions and plant-soil-atmosphere interactions, but require calibration with detailed data from local field trails for accurate predictions.

Modern precision agriculture approaches, such as Variable Rate Technology (VRT) [22], leverage IoT technologies, including GPS, sensors, and remote sensing, to collect field data [23]. Geographic Information Systems (GIS) are then used to analyze the data and create detailed field maps that show variations in soil fertility, moisture, and other factors influencing crop growth. These maps enable farmers to apply inputs like fertilizers, pesticides, and seeds more precisely, tailoring the application to the specific needs of different fields. Paper [14] proposed a crop disease detection tool iCrop leveraging the remote imaging sensing and image processing. Recent works have advanced these technologies further, proposing the use of agricultural drones [13] to monitor crop fields through canopy imaging and vegetation indices [24]. By leveraging remote multispectral sensing, these systems can determine the current high-accuracy fertilizer requirements.

Although these approaches provide more targeted solutions by offering a reliable assessment of plant nutrient status, they often make decisions without fully considering evolving environmental and weather conditions later in the growing season. As a result, these solutions, while optimal at the moment of application, may fail to remain globally optimal throughout the entire crop growth cycle. Our work addresses these gaps with FertilizeSmart, a framework that periodically optimizes fertilization strategies using current weather data and forecasts, ensuring the approach adapts to evolving conditions and nutrient demands.

VI. Conclusion

We propose FertilizeSmart, a novel framework that integrates IoT technologies and DE-based learning to optimize crop fertilization. By incorporating in-season weather data collected from IoT sensors, FertilizeSmart dynamically adjusts fertilization strategies throughout the growth cycle, addressing the limitations of traditional methods, such as fixed fertilizer application rates and the inability to respond to weather fluctuations. Experiments demonstrate its effectiveness in maximizing yields while reducing costs and environmental impact.

VII. Acknowledgement

This work is supported by the NSF SCC funded project "Smart Integrated Farm Network for Rural Agricultural Communities" (SIRAC) award Nr.1952045.

References

[1] M. Kazlauskas *et al.*, "Comparative analysis of energy and ghg emissions using fixed and variable fertilization rates," *Agronomy*, vol. 11, no. 1, p. 138, 2021.
[2] N. Ziadi *et al.*, "Soil and plant tests to optimize fertilizer nitrogen management of potatoes," *Sustainable potato production: Global case studies*, pp. 187–207, 2012.
[3] J. Sogbedji, H. Van Es, S. Klausner, D. Bouldin, and W. Cox, "Spatial and temporal processes affecting nitrogen availability at the landscape scale," *Soil and Tillage Research*, vol. 58, no. 3-4, pp. 233–244, 2001.
[4] M. Xie *et al.*, "Weather effects on corn response to in-season nitrogen rates," *Canadian journal of plant science*, vol. 93, pp. 407–417, 2013.
[5] Z. Qin *et al.*, "Application of machine learning methodologies for predicting corn economic optimal nitrogen rate," *Agronomy Journal*, vol. 110, no. 6, pp. 2596–2607, 2018.
[6] J. W. Jones *et al.*, "The dssat cropping system model," *European journal of agronomy*, vol. 18, no. 3-4, pp. 235–265, 2003.
[7] M. Salmerón *et al.*, "Dssat nitrogen cycle simulation of cover crop–maize rotations under irrigated mediterranean conditions," *Agronomy Journal*, vol. 106, no. 4, pp. 1283–1296, 2014.
[8] G. Hoogenboom and others., *Decision Support System for Agrotechnology Transfer (DSSAT) Version 4.8.2*. Gainesville, Florida, USA: DSSAT Foundation, 2023, www.DSSAT.net.
[9] K. Ren *et al.*, "Optimizing nitrogen fertilizer use for more grain and less pollution," *Journal of Cleaner Production*, vol. 360, p. 132180, 2022.
[10] Q.-J. Du *et al.*, "Effects of different fertilization rates on growth, yield, quality and partial factor productivity of tomato under non-pressure gravity irrigation," *PLoS One*, vol. 16, no. 3, p. e0247578, 2021.
[11] M. Schwenzer *et al.*, "Review on model predictive control: An engineering perspective," *The International Journal of Advanced Manufacturing Technology*, vol. 117, pp. 1327–1349, 2021.
[12] M. M. Tiwari *et al.*, "Weather monitoring system using iot and cloud computing," *International journal of advanced science and technology*, vol. 29, pp. 2473–2479, 2020.
[13] X. Tao, E. Damron, and S. Silvestri, "High-precision crop monitoring through uav-aided sensor data collection," in *IEEE International Conference on Communications*. IEEE, 2024.
[14] X. Tao *et al.*, "icrop: Enabling high-precision crop disease detection via lora technology," in *International Conference on Computer Communications and Networks (ICCCN)*. IEEE, 2024.
[15] D. Abayechaw, "Review on decision support system for agrotechnology transfer (dssat) model," *Information Systems*, vol. 10, pp. 126–133, 2021.
[16] L. Cui, G. Li, Q. Lin, J. Chen, and N. Lu, "Adaptive differential evolution algorithm with novel mutation strategies in multiple sub-populations," *Computers & Operations Research*, vol. 67, pp. 155–173, 2016.
[17] S. M. Islam, S. Das, S. Ghosh, S. Roy, and P. N. Suganthan, "An adaptive differential evolution algorithm with novel mutation and crossover strategies for global numerical optimization," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 42, no. 2, pp. 482–500, 2011.
[18] The power project. [Online]. Available: https://power.larc.nasa.gov/
[19] Nitrogen application timing in corn production. [Online]. Available: https://www.pioneer.com/us/agronomy/nitrogen_application_timing.html
[20] S. B. Khose, K. B. Dhokale, and S. Shekhar, "The role of precision farming in sustainable agriculture: advancements and impacts," *Agriculture and Food E-newsletter*, vol. 5, no. 9, pp. 115–119, 2023.
[21] P. Verma, A. Chauhan, and T. Ladon, "Site specific nutrient management: A review," *Journal of Pharmacognosy and Phytochemistry*, vol. 9, no. 5S, pp. 233–236, 2020.
[22] S. R. Saleem, Q. U. Zaman, A. W. Schumann, and S. M. Z. A. Naqvi, "Variable rate technologies: Development, adaptation, and opportunities in agriculture," in *Precision Agriculture*. Elsevier, 2023, pp. 103–122.
[23] A. K. Singh *et al.*, "Smart connected farms and networked farmers to improve crop production, sustainability and profitability," *Frontiers in Agronomy*, vol. 6, p. 1410829, 2024.
[24] S. P. Kumar, A. Subeesh, B. Jyoti, and C. Mehta, "Applications of drones in smart agriculture," in *Smart Agriculture for Developing Nations: Status, Perspectives and Challenges*. Springer, 2023, pp. 33–48.

ISBN: 978-3-903176-71-3