

Greedy Algorithm for Fertilization Processes in Agriculture 4.0

Aarón Junior Rocha Rocha
Ing. en Sistemas Computacionales
Tecnológico Nacional de México, campus Zamora
Zamora, Michoacán, México
0000-0002-8788-291X

Ana Celia Segundo Sevilla
Ing. en Sistemas Computacionales
Tecnológico Nacional de México, campus Zamora
Zamora, Michoacán, México
0000-0002-3119-697X

Juan Pablo Guerra Ibarra
Ing. en Sistemas Computacionales
Tecnológico Nacional de México, campus Zamora
Zamora, Michoacán, México
0000-0003-3520-8840

Abstract—Agriculture has been strengthened from its beginnings to the present day in what is known as agriculture 4.0. One of the main objectives of modern agriculture is the optimization of resources such as water and fertilizers, which is specially relevant in hydroponic cultivation systems. Plants, as living organisms, require nutrients for proper growth and development. In agriculture, nutrients are classified into two groups: macronutrients and micronutrients. This classification does not depend on their importance, but rather on the quantity required by plants. One of the main macronutrients is nitrogen, which is important for various physiological and biochemical processes. In agriculture 4.0, the use of algorithms to support farmers' decision-making in nutrient management has become an increasingly valuable tool. One such approach involves greedy algorithms, which construct solutions through a sequence of local decisions, with the expectation that these will lead to an optimal or near-optimal global solution. This paper presents an adaptation of a greedy algorithm designed to formulate nutrient solutions that meet the required parts per million of nitrogen according to the crop's development stage. The results demonstrate an average error of 3.5 parts per million in 10 runs of the algorithm with an average of 600,000 iterations and 15 seconds to converge.

Index Terms—Agriculture 4.0, Greedy Algorithm, Optimization, Vegetable Nutrition

I. INTRODUCTION

The production of food from farming the land is as old as humanity itself, always being fundamental to different civilizations throughout history and developing alongside them with different technological advances. [1], [2]. For example, advances made by Egyptian culture include hydraulic control of the Nile river, advanced irrigation systems, agricultural tools

such as animal-drawn plows, and food preservation for times of scarcity. In America, the Aztec culture made advances with the construction of chinampas, agricultural terraces, and crop rotation. Agricultural progress has been a fundamental basis for the growth of human populations. By year 2050, the human population is projected to grown to about 10 billion people, according to *United Nations* data [3], which implies great challenges for the different food production activities.

In recent years, technological advances applied to farming of the land have generated what is called **Agriculture 4.0**. The application of different types of algorithms and other technological advances allows for the monitoring and control of various crop variables. For example, classical algorithms and **Artificial Intelligence (AI)** algorithms, especially those of **Computer Vision (CV)** applied to the detection and identification of diseases and pests [1], irrigation and fertilization optimization processes often use different **Metaheuristic Algorithms** and **Greedy Algorithms (GA)** [4], [5].

GA are implemented in optimization problems, where the aim is to maximize or minimize an objective function [6]. In each iteration, the algorithm selects the solution considered locally best according to an objective function when considering two candidate responses. The characteristics of GA include simplicity of implementation and speed, but do not guarantee that the solution found will always be the optimal one. This work approaches the problem of determining the quantities of different fertilizers needed to achieve specific parts per million of a nutrient element using a GA.

The organization of the paper is as follows. Section 2

presents a series of works related to Agriculture 4.0. Section 3 describes the materials and methods used in the *GA* adaptation for the formulation of nutrient solutions for crops are described. The objectives of the experiment and the control parameters for the formulation of nutrient solutions using *GA* are presented in Section 4. The results obtained with the adaptation of the algorithm for the formulation of nutrient solutions are presented in Section 5 and the last Section contains the conclusions of the investigation.

II. STATE OF THE ART

In agriculture 4.0, crop fertilization processes are essential to maximize yields through the optimal use of fertilizers and water. Optimization of fertilization processes is addressed using different metaheuristic algorithms. For example, Cropper [7] developed a model to estimate the amount of phosphorus present in the soil and crop requirements in order to use a genetic algorithm to estimate the minimum amount needed to meet crop needs through fertilization processes. [8] developed an expert system to support decision-making regarding nitrogen, phosphorus, and potassium fertilization processes, using data from 2,000 plant nutrition experiments, reporting an increase in rice grain production in fields that used the recommendations made by the expert system. Ahmad [9] estimated the amount of nitrogen needed based on the application's water requirements using a quadratic equation that can be used to reduce costs in different crops. Ahmed [10] implemented a genetic algorithm that uses data from different types of sensors to make adjustments to fertilization processes. The algorithm applied the neighborhood-based strategy for the exploration and exploitation stages. Chen [11] proposed an optimization method based on an algorithm inspired by gray wolf behavior with multiple strategies to make fertilization suggestions for the main macronutrients.

III. MATERIALS AND METHODS

The following sections describe the fundamental concepts of vegetal nutrition and the chemical elements necessary for plant development, in the same way, classical algorithm in combinatorial optimization, the greedy algorithm for the formulation of nutritional solutions, characteristics of the selected for experimentation, and an explanation of the method.

A. Vegetal Nutrition

Through the process of photosynthesis, plants obtain essential elements such as Oxygen (O), Carbon (C), and Hydrogen (H). In addition, the proper development of plants requires other essential chemical elements divided into two categories depending on the quantities required by the crops and not their importance, since they are all necessary [12], which are taken from the soil in traditional agriculture or through fertilization processes in Nutrient Solutions (*NS*) in the case of hydroponic production. Table I shows the names and symbols of the six main macronutrients, while Table II contains data on nine micronutrients.

TABLE I: List of Macronutrients required by crops for their development

No.	Macronutrients 60 kg/ha to 200 kg/ha	
	Chemical element	Symbol
1	Nitrogen	N
2	Phosphorus	P
3	Potassium	K
4	Calcium	Ca
5	Magnesium	Mg
6	Sulfur	S

TABLE II: List of Micronutrients required by crops for their development

No.	Micronutrients Less than 50 kg/ha	
	Chemical element	Symbol
1	Iron	Fe
2	Manganese	Mn
3	Copper	Cu
4	Zinc	Zn
5	Boron	B
6	Molybdenum	Mo
7	Chlorine	Cl
8	Nickel	Ni
9	Sodium	Na

NS are basically a mixture of quantities of fertilizers dissolved in water. Fertilizers are composed of different chemical elements that are necessary for crop development. Table III presents the composition of some commonly used fertilizers.

One way to control the amount of a particular chemical element is by calculating the Parts Per Million (*PPM*) present in a given amount of fertilizer.

To calculate the *PPM* of a chemical element in a quantity of fertilizer, the following is required:

- 1) Quantity of fertilizer (q).
- 2) Atomic weight of the chemical element (a_w).
- 3) Number of atoms of the chemical element in the fertilizer formula (n).
- 4) Total atomic weight of the fertilizer (a_f).

The above parameters are expressed in Equation 1.

$$PPM = \frac{q * a_w * n}{a_f}. \quad (1)$$

Table III contains information regarding the chemical composition of 25 fertilizers commonly used in agriculture 4.0.

B. Greedy Algorithm

The greedy algorithm makes a locally optimal choice in the hope that this will lead to a globally optimal solution. Clearly, greedy algorithms do not always yield de optimal solution. But for a wide range of important problems, is quite powerful.

The algorithm is based on the specifications provided by the domain expert, who determines the nutritional requirements of the plant in terms of a set of essential chemical elements n . For each element i , a target amount is established, expressed

TABLE III: List of fertilizers to be considered for blending.

Fertilizer	Chemical elements											Atomic weight amu
	Ca	C	H	Mg	N	P	k	S	O	Cl	Na	
	Number of atoms in the fertilizer											
1. Calcium Nitrate	1	0	0	0	2	0	0	0	6	0	0	164.12
2. Potassium Nitrate	0	0	0	0	1	0	1	0	3	0	0	101.11
3. Magnesium sulfate	0	0	0	1	0	0	0	1	4	0	0	120.36
4. Diammonium phosphate	0	0	9	0	2	1	0	0	4	0	0	132.11
5. Ammonium nitrate	0	0	4	0	2	0	0	0	3	0	0	80.08
6. Monopotassium Phosphate	0	0	2	0	0	1	1	0	4	0	0	136.09
7. Potassium sulfate	0	0	0	0	0	2	1	4	0	0	0	174.24
8. Ammonium sulfate	0	0	8	0	2	0	0	1	4	0	0	132.18
9. Urea	0	1	4	0	2	0	0	0	1	0	0	60.09
10. Phosphonitrate	0	0	0	0	3	1	0	0	9	0	0	217.04
11. Ammonium Sulfate	0	0	8	0	2	0	0	1	4	0	0	132.18
12. Monoammonium phosphate	0	0	6	0	1	1	0	0	4	0	0	115.06
13. Potassium chloride	0	0	0	0	0	1	0	0	1	0	0	74.54
14. Calcium chloride	1	0	0	0	0	0	0	0	2	0	0	74.54
15. Anodized ammonia	0	0	3	0	1	0	0	0	0	0	0	17.05
16. Ammonium chloride	0	0	4	0	1	0	0	0	1	0	0	53.51
17. Phosphoric Rock	5	0	1	0	0	3	0	0	1	3	0	502.35
18. Superphosphate	1	0	4	0	0	2	0	0	8	0	0	234.08
19. Phosphoric acid	0	0	3	0	0	1	0	0	4	0	0	98.01
20. Phosphorite	0	0	0	0	0	2	0	0	5	0	0	141.96
21. Potassium carbonate	0	1	0	0	0	2	0	3	0	0	0	138.19
22. Sodium nitrate	0	0	0	0	0	1	0	0	3	0	1	141.96
23. Kieserita	0	0	2	1	0	0	0	1	5	0	0	138.38
24. Magnesium nitrate	0	0	0	1	2	0	0	0	6	0	0	148.34
25. Sodium chloride	0	0	0	0	0	0	0	0	1	1	0	58.43

in Parts Per Million (PPM), denoted as qe_i . Thus, the input vector is defined as:

$$Q = \{qe_0, qe_1, \dots, qe_n\}$$

where each qe_i represents the desired concentration of a chemical element in PPM.

Consequently, the objective function of the algorithm is not defined as a single scalar value, but rather as the approximation to the entire target vector Q . Formally, the algorithm seeks to construct a solution $S = \{s_1, s_2, s_3, \dots, s_n\}$ such that:

$$s_i \approx qe_i \quad \forall i \in \{1, 2, \dots, n\}$$

where s_i , denotes the final amount reached for element i after executing the optimization procedure.

For example, if the expert specifies $qe_N = 5$ ppm of nitrogen and $qe_{Mg} = 10$ ppm of magnesium, the objective of the algorithm will be to generate a mixture in which the final concentration of nitrogen and magnesium approximates these target values.

In practice, the contribution of each element is obtained from a set of fertilizers m , where b_{ij} represents the concentration in PPM of element i contributed by fertilizer j . The total concentration reached for element i is:

$$s_i = \sum_{j=1}^m b_{ij}$$

To evaluate the quality of a candidate solution, the algorithm computes a fitness function defined as the sum of the absolute deviations between the target values and the achieved concentrations:

$$\text{Fitness} = \sum_{i=1}^n |qe_i - s_i| = \sum_{i=1}^n \left| qe_i - \sum_{j=1}^m b_{ij} \right|$$

This process is repeated iteratively until either a maximum number of iterations is reached or the fitness function is minimized below a desired threshold (targetFitness).

In the first iteration, the solution S is obtained randomly within the predefined lower and upper bounds.

From that point onward, in each subsequent iteration, new values are generated based on the previous solution. However, the interval used to constrain these variations is not recalculated at every step. Instead, it is initially determined using the fitness value of the first generated solution and remains fixed until the fitness value of a subsequent solution falls below the defined lower bound. At that moment, the threshold is recalculated, and a new interval is established. This ensures that the search process alternates between phases of exploitation within a stable range and adaptive adjustments whenever the current solution crosses the lower limit.

Formally, the threshold $\theta(t)$ is updated according to the following condition:

$$\theta(t) = \begin{cases} \delta \cdot f(S(t)) & \text{if } f(S(t)) < L(t), \\ \theta(t-1) & \text{otherwise.} \end{cases}$$

where δ is the threshold factor ($0 < \delta < 1$),

$$f(S(t))$$

is the fitness of the current solution when the condition is satisfied, $L(t) = f(S(t)) - \theta(t)$ represents the lower bound of the interval.

Thus, the search interval at iteration t is given by:

$$I(t) = [f(S(t)) - \theta(t), f(S(t)) + \theta(t)]$$

which defines the range of within which new candidate solutions are generated.

At predefined intervals, specified by the expCount parameter, the algorithm performs an exploration step. During this step, the new solution is not derived from the previous iteration but is instead sampled uniformly at random within the initial lower and upper bounds. This mechanism introduces diversity into the search process and reduces the risk of premature convergence.

This approach ensures that the search space is gradually refined around the best known solutions, allowing both exploration and exploitation during the optimization process.

Figure 1 shows the flow diagram of the Greedy Nutrient Algorithm (GNA) which describes the operation of the algorithm to find the best nutritional solution required by the plants. The best initial solution is based on a new random element, which is explained in the Algorithm 1, and compared to a fitness value and the number of repetitions. The result of a modular operation between the number of iterations and the exploration data determines whether a new element or a new random element is obtained as a new solution. This is compared to the best solution that already existed and that may be a better solution now or not.

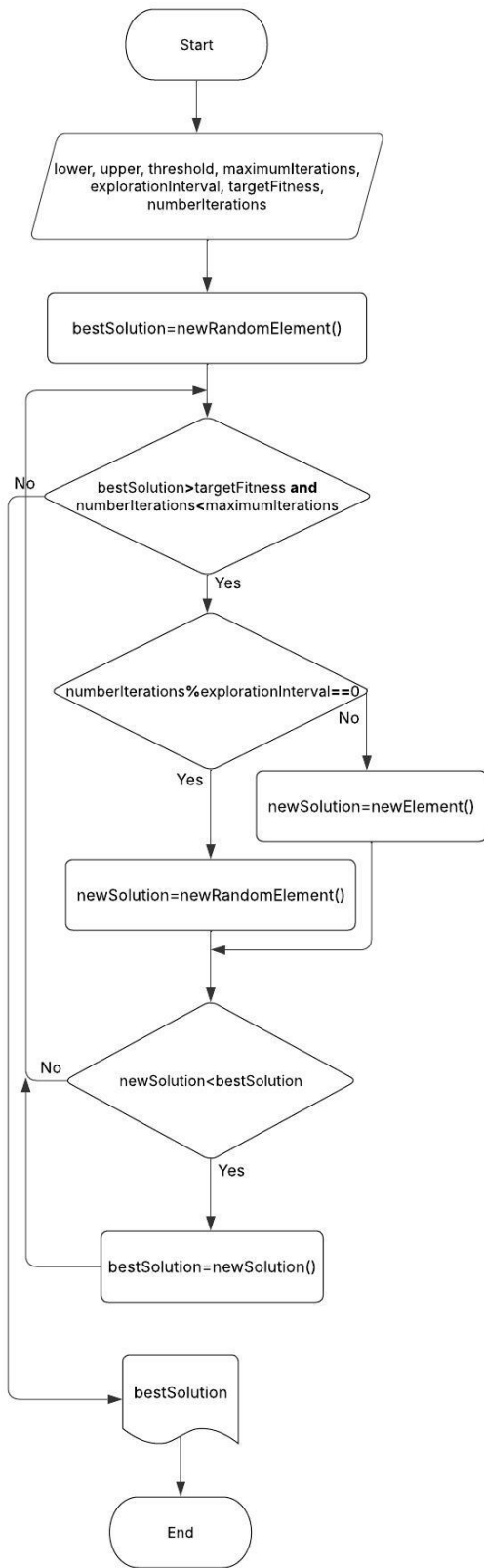


Fig. 1: Greedy Nutrient Algorithm

The functions and parameters involved in the (GNA) algorithm are described below and in case of the relevant Functions an algorithm with Pseudocode of them is shown.

The newRandomElement() function shows which is the better solution founded based on an upper and lower limit, as well as the input data extracted from the fertilizers mentioned in the Table III, to generate random elements that are evaluated based on fitness values. The implementation of the function is shown in the Algorithm 1.

Algorithm 1 Algorithm to create new random element

```

1: function NEWRANDOMELEMENT(update)
2:   new ← answer(lower, upper)
3:   new.generate(data)
4:   new.evaluate(data)
5:   if update is even then
6:     thresholdUpdate(new.targetFitness)
7:     lastThresholdFitness ← new.targetFitness
8:   end if
9:   return new
10: end function

```

The newElement() generates random values that are evaluated based on fitness values and compared with the fitness of the last threshold base. The implementation of the function is shown in the Algorithm 2.

Algorithm 2 Algorithm to create new element

```

1: function NEWELEMENT(baseSolution)
2:   new ← answer(lower, upper)
3:   new.genThreshold(data, baseSolution, threshold)
4:   new.evaluate(data)
5:   if new.targetFitness < lastThresholdFitness – threshold then
6:     thresholdUpdate(new.targetFitness)
7:     lastThresholdFitness ← new.targetFitness
8:   end if
9:   return new
10: end function

```

IV. EXPERIMENTATION

The experimental evaluation was conducted on a case study involving tomato cultivation, where the domain expert determined that nitrogen is the primary nutrient required by the plant. Consequently, the optimization objective was defined as reaching a nitrogen target of 150 parts per million (PPM) in the final nutrient solution.

Based on this requirement, the algorithm was executed under configuration parameters shown in Table IV.

The experiment was repeated five times to assess the effectiveness of the proposed algorithm. The evaluation criteria focused on three aspects: (i) the number of successful runs in which the optimal solution was reached, (ii) the time, measured in the number of iterations, required to reach the solution and the consistency of this value across different runs, and

TABLE IV: Experimentation Parameter-Values

Parameter	Value
lower	1
upper	1,000
threshold	10
maximumIterations	500,000
explorationInterval	10
targetFitness	1

(iii) the structure of the solutions obtained in each repetition, specifically to determine whether the resulting solutions were identical, similar, or distinct.

The algorithm was implemented in the Python programming language, chosen for its versatility and the wide availability of libraries oriented towards numerical computation and optimization. This choice allowed for a clear structure of the objective function, model constraints, and solution search procedures while also facilitating the reproducibility of the experiments.

V. RESULTS

In the experiments conducted, it was observed that the algorithm reached adaptation values that, in most cases, remained close to the target value of 150 PPM of nitrogen. Although in some trials the results plateaued at a mean squared error (MSE) close to zero, this indicates that the solutions obtained were generally quite near the desired target. The experiment results related to the best adaptation values and the number of iterations required to reach them in each trial are shown in Table V.

TABLE V: Final fitness value and difference to objective function. (EN) Experiment Number, (OF) Objective Function, (BFV) Best Fitness Value, (ON) Obtained Nitrogen in PPM, (I) Iteration where BFV obtained

EN	FO	BFV	ON	I
1	150	0.9705	149.03	7,951
2	150	6.4483	143.55	500,000
3	150	0.9763	149.02	262,519
4	150	1.6451	148.35	500,000
5	150	0.9989	149	80,455

As shown in the results Table V, the adaptation values, expressed in terms of nitrogen obtained (ppm), indicate that in three out of five experiments the objective function value was exactly achieved. It can also be observed that the number of iterations required to reach this target varied, which is attributable to the stochastic nature of the implemented algorithm.

In the remaining two experiments, although the exact target was not reached, the difference with respect to the objective value was minimal, yielding concentrations of 143.55 ppm and 148.35 ppm in experiments 2 and 4, respectively. These results can be considered practically acceptable, as the deviations are small.

Moreover, Figures 2 through 6 show that the evolution of the fitness function across iterations follows a very similar pattern in all experiments. In general, the curves exhibit an almost identical trend, reaching stable values close to convergence approximately halfway through the process.

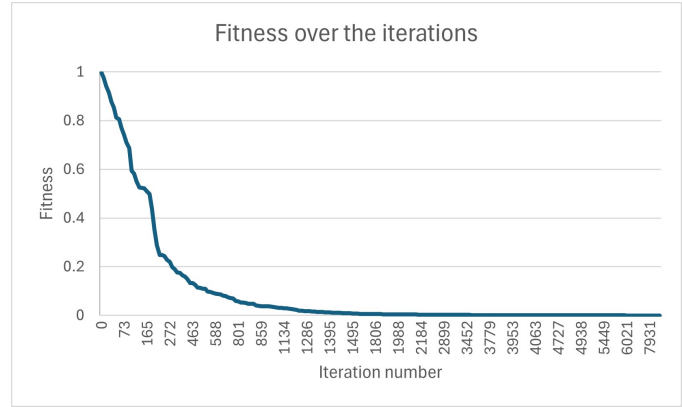


Fig. 2: Fitness curve for experiment 1.

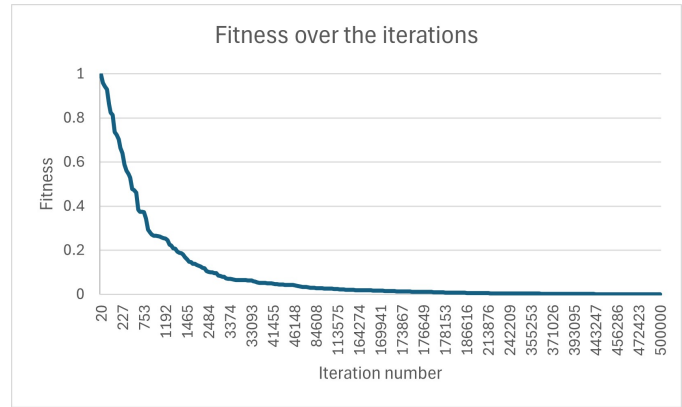


Fig. 3: Fitness curve for experiment 2.

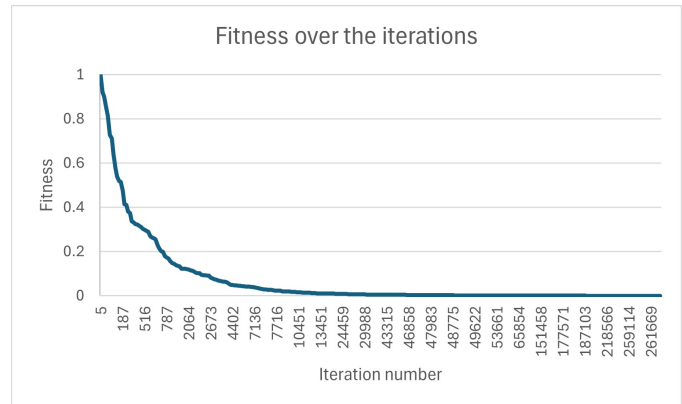


Fig. 4: Fitness curve for experiment 3.

Figure 7 illustrates this overall tendency, highlighting that each run started from different initial values and at varying distances from the objective function. Notably, experiment 1

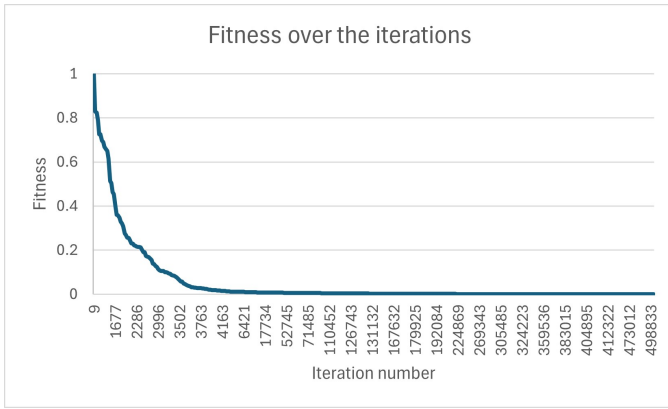


Fig. 5: Fitness curve for experiment 4.

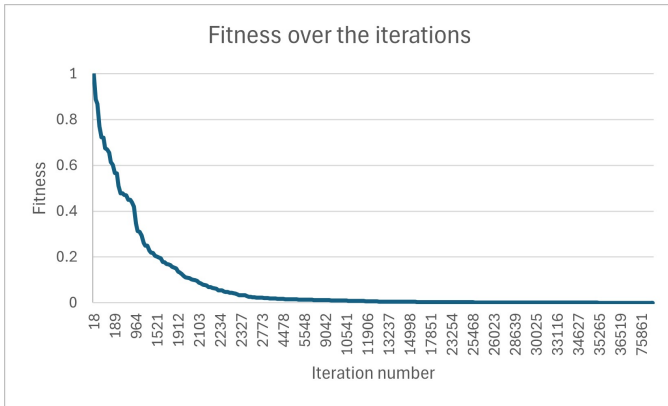


Fig. 6: Fitness curve for experiment 5.

—despite beginning farther from the target— was the one that achieved the goal in the fewest iterations, thus reinforcing the role of randomness in the algorithm’s dynamics.

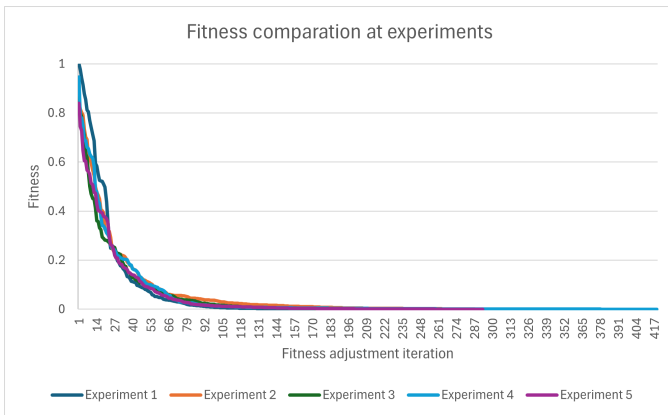


Fig. 7: Fitness comparative at experiments.

Finally, Figure 8 presents the distribution of the quantity of each fertilizer in the final solution obtained from each experiment. It can be observed that each run produced a different solution, which demonstrates, first, that there is not a single feasible solution. Moreover, the results show that the

solutions may vary in their final volume depending on the proportions of fertilizers selected by the algorithm.

It is also noteworthy that in some experiments certain fertilizer quantities tend to approach values close to zero, whereas in other cases their contribution is more significant. This behavior highlights two important aspects: on the one hand, the algorithm’s ability to explore different combinations that satisfy the nutritional objective, and on the other, the possibility of introducing secondary criteria—such as cost reduction or resource efficiency—in order to select the most convenient solution in an actual agricultural context.

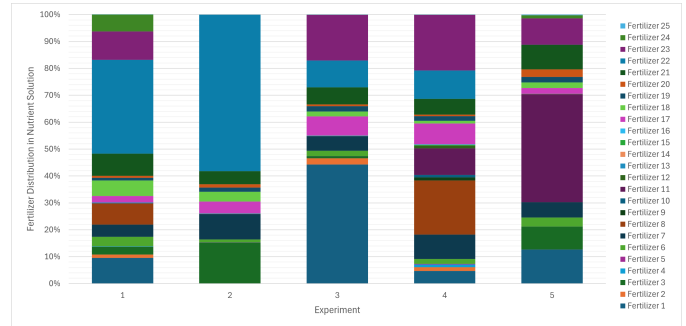


Fig. 8: Fertilizer distribution in five experiments.

VI. CONCLUSIONS

The use of greedy algorithm to determine fertilizer quantities to achieve nitrogen PPMs is efficient in achieving the objective function of the presented method.

The fact that some individual solutions fell slightly below or even slightly exceeded 150 PPM does not represent a significant issue. Rather, it demonstrates that the algorithm successfully explored the region around the global optimum and produced stable results within an acceptable margin of variation. Such fluctuations are expected in this type of method as they depend on search processes that combine both exploration and exploitation of solutions.

With regard to the interpretation of the performance, it can be stated that the algorithm exhibited high effectiveness, as it consistently approached the target nitrogen value. However, efficiency was variable: while in some executions the number of iterations required to achieve convergence was relatively low, in others a considerable computational effort was necessary, reaching up to 500,000 iterations.

In general, the results shown in Table V suggest that the applied strategy allows us to obtain reliable solutions to estimate nitrogen levels, remaining close to the target value. This implies that, although the exact global optimum is not always reached, the solutions achieved are sufficiently accurate for practical purposes, validating the use of this algorithm as an appropriate tool for problems of this nature.

VII. FUTURE WORK

The area of nutrient solution formulation is fundamental to the development of agriculture 4.0, improvement of the greedy algorithm focuses on the following aspects:

- Adapt and test the greedy algorithm to perform a multi-objective search for more than one macronutrient.
- Adapt and test a method for including the price and availability of fertilizers when measuring fitness in a nutrient solution.
- Study and implement alternatives to reduce the randomness factor of the greedy algorithm as the fitness of the nutritional solutions approaches the values of the objective function.

Additionally, the study and adaptation of other metaheuristic algorithms such as genetic algorithms and particle swarm optimization, among others, is explored.

VIII. ACKNOWLEDGEMENTS

Without the support of institutions, it would not be possible to carry out research activities. We would like to thank:

- Secretaria de Ciencias, Humanidades, Tecnología e Innovación.
- Tecnológico Nacional de México.
- Tecnológico Nacional de México, Campus Zamora.
- Centro de Investigaciones en Óptica A.C.

REFERENCES

- [1] Y. Awasthi, "Press "A" for Artificial Intelligence in Agriculture: A Review," *International Journal on Informatics Visualization*, vol. 4, no. 3, pp. 112–116, 2020.
- [2] P. P. Ray, "Internet of things for smart agriculture: Technologies, practices and future direction," *Journal of Ambient Intelligence and Smart Environments*, vol. 9, no. 4, pp. 395–420, 2017.
- [3] FAO, "Our approach — Food Systems — Food and Agriculture Organization of the United Nations Retrieved from <http://www.fao.org/food-systems/our-approach/en/>," 2023.
- [4] Y. Dong, C. Zhao, G. Yang, L. Chen, J. Wang, and H. Feng, "Integrating a very fast simulated annealing optimization algorithm for crop leaf area index variational assimilation," *Mathematical and Computer Modelling*, vol. 58, no. 3–4, pp. 877–885, 2013.
- [5] Y. Bai and J. Gao, "Optimization of the nitrogen fertilizer schedule of maize under drip irrigation in Jilin, China, based on DSSAT and GA," *Agricultural Water Management*, vol. 244, no. July 2020, p. 106555, 2021.
- [6] E.-G. Talbi, *Metaheuristicis. From desing to implementation*. 2009.
- [7] W. P. Cropper and N. B. Comerford, "Optimizing simulated fertilizer additions using a genetic algorithm with a nutrient uptake model," *Ecological Modelling*, vol. 185, pp. 271–281, 2005.
- [8] X. Xu, P. He, F. Yang, J. Ma, M. F. Pampolino, A. M. Johnston, and W. Zhou, "Methodology of fertilizer recommendation based on yield response and agronomic efficiency for rice in China," *Field Crops Research*, vol. 206, pp. 33–42, 2017.
- [9] I. Ahmad, S. A. Wajid, A. Ahmad, M. J. M. Cheema, and J. Judge, "Optimizing irrigation and nitrogen requirements for maize through empirical modeling in semi-arid environment," *Environmental Science and Pollution Research*, vol. 26, no. 2, pp. 1227–1237, 2019.
- [10] U. Ahmed, J. C. W. Lin, G. Srivastava, and Y. Djenouri, "A nutrient recommendation system for soil fertilization based on evolutionary computation," *Computers and Electronics in Agriculture*, vol. 189, no. August, p. 106407, 2021.
- [11] C. Chen, X. Wang, H. Chen, C. Wu, M. Mafarja, and H. Turabieh, "Towards precision fertilization: Multi-strategy grey wolf optimizer based model evaluation and yield estimation," *Electronics (Switzerland)*, vol. 10, no. 18, pp. 1–26, 2021.
- [12] K. MENGEL and E. KIRKBY, *Principles of vegetable nutrition*, vol. 15B,A. 1978.