Optimization of Land and Resource Utilization Techniques with Deep Reinforcement Learning

Michael Jon Miller
University of Colorado - Boulder
ASEN 5264
Boulder, CO, United States
michael.miller-5@colorado.edu

***Abstract*—**
In efforts to reduce resource consumption among horticulturists while retaining or increasing crop yields for a growing global population, recent research has been directed towards utilizing Deep Reinforcement Learning (DRL) to manage irrigation and fertilization policies. Growing crops can be thought of as a Sequential Decision Making Problem (SDMP) with sparse rewards only being obtained at the end of a growing season when the crop can be harvested. Additionally, since real-world experiments take a growing season to collect data on the effectiveness of crop management policies, there has been a focus to use Deep Q-Learning Networks (DQN) to solve and iterate upon different crop and resource management practices. This project aims to create a representative Markov Decision Process (MDP) formulation of tomato (cultivar Roma) growth stages and dynamics and utilize the Flux.jl machine learning library to learn an optimized irrigation strategy for a single plant and compare the average output to an expert policy baseline. The DQN agent was able to learn an irrigation policy that performed 7.06 times better than the baseline expert policy. Building on top of those results, this project also aims to optimize for both irrigation and fertilization policies while accounting for environmental factors such as water use and Nitrogen leaching. The DQN agent was unable to learn an optimized combined policy and performed 7.5 times worse than the baseline expert policy. ***Index Terms*—Irrigation, Fertilization, Crop Management, Resource Management, Land Utilization, Deep Reinforcement Learning, Deep Q-Learning**

I. Introduction

Each growing season, horticulturalists around the world strive to cultivate the most productive versions of their crops whether their goals are survival, economic profit, or enjoyment. However, in the recent years there has been an intense focus on how to continually increase crop yield to sustain the growing global population while also balancing scarce land and water resources. With the ever-looming threat of climate change having the potential to change weather patterns, horticulturists are focused on methods to reduce their resource consumption wile maintaining output or to keep their resource consumption constant while increasing their yields. In addition to real-world experiments costing a lot of time and resources to test and collect data on new resource management practices, they are unable to iterate upon different strategies quickly and adjust to rapidly changing conditions from year to year. This all motivates the introduction of Deep Reinforcement Learning (DRL) to solve and learn new, optimized resource management strategies to help boost yield while minimizing environmental impact.

Although expert policies are modified year over year in response to new trends in local climates and more productive or robust cultivars, they usually fail to make use of all the information that would be available to a DRL agent. DRL is a commonly used machine learning strategy in this field as it can overcome the sparse rewards that come with crop cultivation. Where some fields of machine learning would fail to make progress in the face of immediate costs of operation and only distant rewards of harvest, DRL is able to learn a near-optimal strategy through trial and error in a modeled environment. Through verification and validation of the simulated environment in relation to real-world growth patterns and yields, these models can be iterated upon until they properly approximate the target growing region and the crop's response to weather. These models can then be deployed at the start of a growing season and continually update the management policies to offload work from the horticulturalist and increase their overall yields.

While other research has been directed at large-scale agricultural applications and the optimization of different crop management techniques [1] - [4] and crop yield predictions [5], this project aims to learn a management policy for a single plant which can then be scaled accordingly with the goal of making these optimized strategies accessible to backyard and small-scale operations. Additionally, while much of the other literature aims to solve management practices for maize or wheat, this project aims to learn management practices for tomato (cv. Roma) crops as a popular backyard garden choice that often suffers from losses due to diseases, pests, or lack of proper care.

II. Background and Related Work

Recently, [1] developed techniques using both Reinforcement Learning (RL) and Imitation Learning (IL) to train optimized management policies for both irrigation and fertilization of maize crops. This work focused on four different models of reward functions and comparing their yields to real-world data from growing seasons in Florida and Spain. The RL agent was deployed to learn optimal management policies given all available information from the training data and the IL agent was given these trained policies as expert policies

to convert into models that only utilized readily available information to make management decisions. This research showed a more than 45% increase in economic profits from the training experiments while also factoring in environmental impacts.

The research outlined above heavily built upon the results of [2] where similar techniques were used to train a RL agent to learn optimal policies for fertilization of maize crops from experimental data from Florida and Iowa. This work did not optimize for irrigation and did not reduce the state space the models were trained on to only readily available information like in [1]. In both of these research projects, the teams utilized information from a popular crop simulation software called Decision Support System for Agrotechnology Transfer (DSSAT) and an Open-AI gym interface for this simulator called Gym-DSSAT. These simulators, however, do not readily provide an interface for tomato growth or for single plant management optimization and thus were not used for this project.

The results of research that focused solely on optimization of irrigation practices is outlined in [3]. Utilizing maize crop simulations from DSSAT in Temple, Texas, the SARSA-λ algorithm was implemented on outputs of two neural networks to determine an optimal irrigation policy based on the Total Soil Water (TSW). The use of the SARSA-λ algorithm increased the learning rate by propagating rewards backwards with eligibility traces, but rewards were only obtained at the end of the simulation to prevent the agent from reacting negatively to immediate operating costs. With this model, the team was able to increase profits anywhere from 6.4% to 136.2% over the baseline experimental practices.

Reference [6] provides extensive information in order to compare to an expert policy and for information regarding best practices of tomato cultivation. This handbook for tomato production outlines best practices for germination, transplanting, irrigation, fertilization, and pest and disease control. With respect to the best mathematical models that represent tomato growth stages and dynamics, [7] utilizes various sigmoid growth models to simulate tomato growth and development. This decision was made to best represent the growth patterns of tomato plants as a sigmoid acceleration function where the highest acceleration happened in the middle of development with slower acceleration in growth in the beginning and end of development. Finally, [8] investigated an improved tomato plant model using Physiological Development Time (PDT) where their improved model accurately predicted the transition of a tomato plant from one stage of growth to the next with a maximum error of 2 days.

III. Problem Formulation

Farming and horticulture can often be modeled as a Sequential Decision Making Problem (SDMP) wherein the horticulturalist needs to take actions each day in order to cultivate a productive crop. This naturally lead this project to use a MDP formulation of the growth stages and dynamics of a tomato plant in order to show the affects the actions a horticulturalist takes in one day on the next. This problem focuses on the Roma tomato cultivar. This cultivar was selected as it is fairly quick growing and a determinate tomato bush. These characteristics made it perfect for a MDP because the state space wouldn't grow large with larger, slow growing plants and being a determinate variant, meaning that all of its fruits flower and ripen at about the same time, meant that the problem formulation could assume that all fruits were harvested at once.

This project initially had the goal of also addressing the positioning of each plant in a garden bed. This was later dropped in favor of using the best known policies for spacing plants (18-24 inches apart). The change in the problem scope came when it became obvious that attacking irrigation, fertilization, and positioning would create a problem space that would be too large for computation or have to be reduced to a problem that didn't accurately represent the growth dynamics and actions of a real-world horticulturalist. Positioning had the largest impact on the compound state space, taking the amount of states for irrigation times the amount of states for fertilization and raising them to the amount of states for positioning. Since the project's scope was already focused on small operations where irrigation and fertilization practices have a larger impact, the positioning problem was dropped. The project then shifted goals to first learn an optimal irrigation strategy and then to further build on that with fertilization and rewarding environmental consciousness.

The state space for the MDP formulation consisted of the combinations of three states for the irrigation strategy and four states for the combined strategy. The first small state was the growth stage of the plant, S_{plant}, which reflected the maturity of the plant and the size of the fruit it produced. This was split up into four different stages, similar to [8]: $empty$, $germination$, $immature$ with the height being measured in 10 cm increments, and $mature$ with the mass of the fruit being measured in 10 g increments. The average height of a Roma tomato bush at maturity is 70 cm and the average mass of a mature fruit is 40 g [10]. The second small state was the Soil Moisture Capacity at a depth of 1 foot (0.3 meters), S_{smc}. For tomatoes to grow in most soils, the SMC must remain above 1 cm per 0.3 meters with the most optimal growth happening when the SMC is around 2.5 cm per 0.3 meters [6]. For the MDP formulation, S_{smc} was represented with four discrete values ranging from 0-3 cm in 1 cm increments. The third small state represented the days since the start of the growing season, S_{day}. To keep the state space more manageable and in line with [3], each time step in the MDP represented 3 day in real time. This is a reasonable time frame to evaluate upon as many expert policies only tend to the tomatoes once every 3-7 days, depending on the action. With Roma tomatoes having an average growing season of 90 days from planting to harvest, S_{day} consisted of discrete 3-day steps from 1-100 with "day 0" at the end of the season representing the terminal state. The fourth small state space used in the combined strategy was the Nitrogen fertilizer available to the plant in grams, S_N. Expert policies vary widely when it comes to fertilization, especially when some factor in environmental concerns, such

as agricultural run-off and Nitrogen leaching, and others do not. These policies usually prescribe a water soluble Nitrogen and Phosphorus based fertilizer, such as 20-20-20, mixed to a concentration of 0-200 ppm N [6]. For the combined strategy, S_N consisted of two discrete fertilization levels: 0 grams and 0.2374 grams per plant (corresponding with the 0-200 ppm N concentrations). The total state space, S, is then represented by a combination of all four of these small state spaces. In the simulated environment, this looked like: $S = [[empty, SMC = 0cm, N = 0g, Day = 1], [empty, SMC = 1cm, N = 0.2374g, Day = 1], ..., [mature - 40g, SMC = 3cm, N = 0.2374g, Day = 0]]$. The final state space for the irrigation only problem was 2240 states and the final state space for the combined problem was 4480 states.

The action space took a similar approach as the state space, utilizing a combination of smaller action spaces to represent the full action space. The full action space was a combination of the first two smaller spaces for the irrigation only problem and a combination of all three smaller action spaces for the combined problem. The first small action space is related to the growth stage of the plant with A_{plant} consisting of three discrete actions: $wait$, $plant$, and $harvest$. The second small action space represents the amount of water the agent can give to a plant, A_{water}. This smaller space is broken into volumes of water that the agent can give to the plant with four options ranging from 0-3 cm in 1 cm increments. Although these options are given in depths, irrigation management policies usually assume that the specified depth of water is applied over a 1 ft^2 area, or 929.03 cm^3. The third small action space represents the amount of fertilizer that the agent can apply while watering, A_N. This action space consists of two options for total amount of Nitrogen applied to the plant: 0 g Nitrogen and 0.2374g Nitrogen. The total action space, A, is then represented by a combination of all three of these small action spaces. In the simulated environment, this looked like: $A = [[wait, water = 0cm, N = 0g], [wait, water = 1cm, N = 0.2374g], ..., [harvest, water = 3cm, N = 0.2374g]]$. The final action space for the irrigation only problem was 12 actions and the final state space for the combined problem was 24 actions.

The transition matrices utilized a similar method by iterating through transition matrices for smaller state and action spaces and multiplying the combined probabilities together. Transitions for the growth stage of the plant depended on the SMC and amount of Nitrogen available to the plant. The average germination rate of Roma tomatoes is 85% with 15% not making it from seed to first sprout [9]. Additionally, [9] shows that there is an average loss rate of 3% for any stage of growth due to weather, pests, disease, or poor management practices. These two rates remain constant for all plant growth stage transitions. As for both SMC and Nitrogen level transitions, [6] and [8] show how poor management practices can add anywhere from 3 to 7 days per growth stage. Fitting a linear regression fit to this data and interpolating for varying degrees of sub-optimal management practices, three different loss factors, 1.0, 1.67, and 3.33 were used to scale the given average loss rates at every stage of growth. Under optimal conditions, the plants would grow 10 cm every week on average to meet the average maturity age of 76 days after the first true leaf [6].

Transitions for watering and the SMC had to take into account evaporative losses when watering as well as water leaving the plant and ground through evapotranspiration [6]. Evapotranspiration can occur through transpiration of water from the plant due to heat and cellular respiration as well as the natural depletion of SMC due to evaporative effects. On average, a tomato plant can lose 0.5 cm SMC per day through evapotranspiration [6]. Additionally, while watering wind and heat can decrease the amount of water actually absorbed into the ground by 0.25-0.5 cm of the applied irrigation. All accounted for, when watering there can be much less water delivered to the plant than intended, so the agent must learn how to deal with these non-deterministic effects.

Finally, transitions for the amount of Nitrogen available to the tomato plant was modeled after expert policy recommendations in [6]. Here, it is noted that on average, the Nitrogen required to keep a tomato plant growing optimally will be depleted after two weeks with most expert policies aiming to fertilize every week. To create the full transition matrix, T, each action iterated over the transition probabilities of each of the states available and multiplied them together to get the final transition probability. In the simulated environment, T was a dictionary with an entry for each possible action with a $S \times S$ matrix stored as the key.

The reward function, R, for this MDP was represented by the costs of each action and were applied immediately. A negative reward was given for planting a seed with it being proportional to the average cost of a tomato seed from the popular seed website Burpee, or \$0.0158. Similarly for watering, this project is aimed at smaller operations or backyard gardens, most likely using water from a tap or provided from a city. A negative reward was given proportional to the cost of water applied, in cm, using the City of Boulder as an example where the average cost of 1,000 gallons of water over the summer is \$4.47, or \$0.001095 per 1 cm of water applied. A negative reward was given proportional to the amount of Nitrogen applied to a plant and with Nitrogen based fertilizers being relatively cheap, this was modeled with the average price of \$0.0243 per gram of 20-20-20 fertilizer. Finally, a positive reward was given whenever the agent successfully harvested a mature tomato plant. This positive reward was modeled off of the average price of a kilogram of tomatoes in Boulder, Colorado at \$2.78. Roma tomato bushes produce an average of 100 tomatoes per year with the fruit masses in the simulated environment ranging from 0-40 grams. Using a normal distribution with a mean of 100 and a standard deviation of 10, the productivity of a particular tomato plant can be modeled to have some variance. At harvest, the reward function takes this variable amount of tomatoes produced by the plant and multiplies it by the per fruit mass and market price of tomatoes to get the final reward for harvest. In the combined problem, the reward function was modified to give

negative rewards for over-irrigation and over-fertilization. This was represented as giving two times the respective cost of the action if the plant already had sufficient water or fertilizer. This was done to disincentivize the use of more water or fertilizer than is necessary. This MDP formulation also utilizes a discount factor, γ, of 0.99. A high discount factor was chosen to represent the higher value placed on long-term rewards.

IV. Solution Approach

The solution approach for this large problem with sparse rewards was selected to be training an agent with the Deep Q-Learning Network (DQN) algorithm. Utilizing the POMDP.jl and CommonRLInterface.jl libraries, the previously formulated MDP can be converted to a Reinforcement Learning Environment. The implementation of DQN that was used in this project trained a neural network with an input layer of 4 units, 3 hidden layers with 256 units and sigmoid non-linearities, and an output layer with a unit for each possible action [1], [2], [7]. During training, the Flux.jl optimized train was used with the ADAM optimizer. To chose actions, the DQN algorithm used the ϵ-greedy approach with a linear decay on the initial ϵ as the number of epochs trained increased. For each training of both the irrigation only problem and the combined problem, the DQN algorithm used an initial learning rate of 1e-4, trained for 100 epochs and 10,000 episodes per epoch, utilized a batch size of 10,000 data points, and a buffer size of 5,000,000 data points. In order to evaluate the performance of the DQN agents and their learned policies, a heuristic policy was made for the irrigation only problem and the combined problem using the recommendations in [6].

V. Results

Training of the DQN agent occurred under two different optimization problems as outlined previously: optimizing for only irrigation management and optimizing for both irrigation and fertilization management. Initially, the reward function and the average discounted return of an evaluation of the learned policy was supposed to represent the average return in USD for each Roma tomato plant grown. However, this formulation ran into some issues regarding what actions were incentivized to the DQN agent. Figure 1 shows how under these conditions, the DQN agent settled on doing nothing in the environment as the best policy since it had a cost of 0.0.

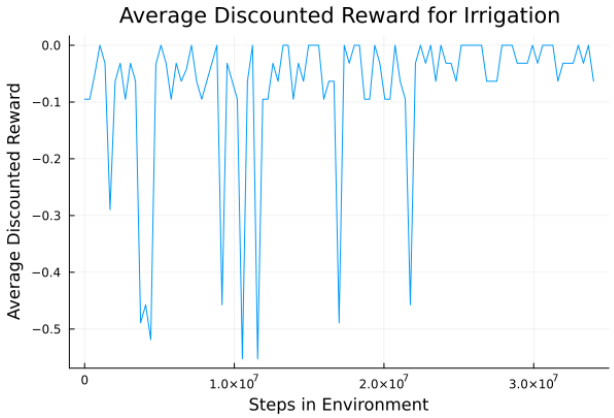

Fig. 1. Initial Irrigation Optimization Average Discounted Return

To fix this issue, the reward function had to be modified to incentivize planting empty spaces, watering dry spaces, and choosing harvest towards the end of the simulation as well as disincentivizing planting when there already is a plant. To do this, a positive reward of 0.1 was added to the $harvest$ action if it was near the end of the simulation, a negative reward of 0.0002 for any dry spaces, and a negative reward of double the seed cost if $plant$ was chosen on a space with a plant. These modifications resulted in the average discounted rewards shown in Figure 2 for the learned optimized irrigation policy.

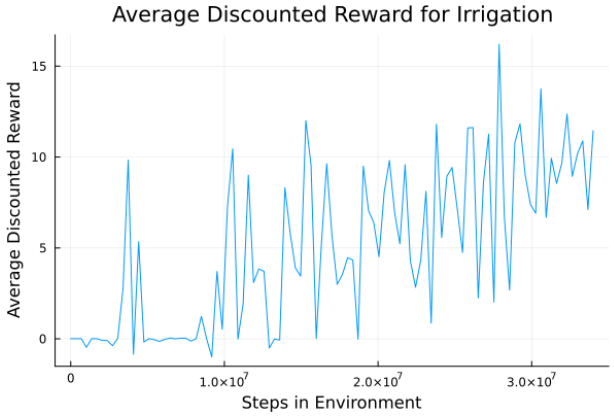

Fig. 2. Irrigation Optimization Average Discounted Return After Modifying the Reward Function

Table I shows the mean average discounted return and the standard error of the mean for the learned optimized irrigation policy in comparison to the heuristic expert policy.

Policy	Average Return	SEM
Learned DQN	10.31	0.2312
Expert	1.46	0.0241

TABLE I
Average Return and SEM of the Optimized Irrigation Policy and the Expert Policy

Following the success of the irrigation only optimization, the reinforcement learning environment was updated with the

additional states and actions to train and learn an optimized management policy for both irrigation and fertilization. Figure 3 shows the training of the DQN agent in this updated environment.

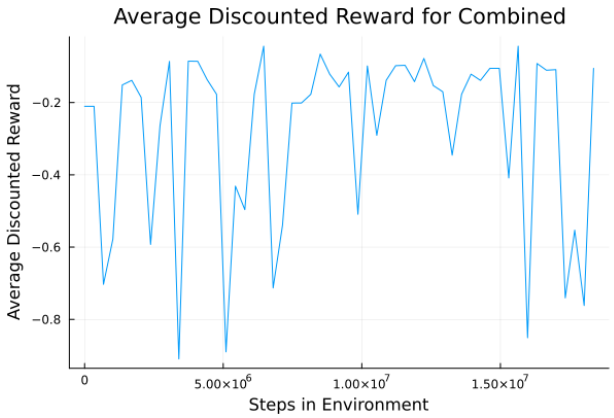

Fig. 3. Irrigation and Fertilization Optimization Average Discounted Return

Table II shows the mean average discounted return and the standard error of the mean for the learned irrigation and fertilization policy in comparison to the heuristic expert policy.

Policy	Average Return	SEM
Learned DQN	-0.04	6.691e-5
Expert	0.30	0.0250

TABLE II
AVERAGE RETURN AND SEM OF THE IRRIGATION AND FERTILIZATION POLICY AND THE EXPERT POLICY

Unfortunately, the DQN agent was unable to learn an optimized policy that outperformed the heuristic expert policy, instead performing 7.5 times worse than the informed policy from [6]. With over 17,000,000 steps executed in this environment, it is most likely that the reward function must be modified to incentivize more watering and fertilization as the implemented reward function penalizes them too much, leading to unproductive crops, or the training hyperparameters must be adapted to this larger and more complex environment. Additionally, optimizing over two management techinques instead of one introduces many more trade-offs that aren't immediately apparent. Further work on the formulation of this problem could possible lead to success.

VI. Future Work

As mentioned in [1], the states that the simulated environment use in this project are not readily available and observable measurements. Although these measurements are crucial to understanding and creating models of crop growth stages and dynamics, the policies the horticulturalists use to care for their plants must be predicated on observable states. In the future, this project would aim to reduce this simulation-real gap by formulating the problem as a Partially Observable Markov Decision Process (POMDP). This would allow the observations of the plant and its immediate environment to dictate the actions an agent takes as well as introduce the randomness that would be inherent to non-specific measurements a human can make unaided.

Although the simulated environment used in this project is informed from expert policies and research on growth stages and dynamics of Roma tomato plants, it has not been validated with respect to real-world experiments. In the future, this project would aim to develop a more accurate model such that the learned policies of the DQN agent can be applied to real-world agriculture. This would mean tweaking costs, environmental factors, and information about growth stage transitions as well as extending the state and action spaces to continuous functions that better represent a real-world environment. Large amounts of data about specific tomato growing seasons is not readily available, but the largest source would come from the crop simulator DSSAT. In the future it would be possible to modify the Python APIs for DSSAT used in [1] and [2] for use in tomato crop simulations given more experimental data.

VII. Conclusion

Finding an optimized irrigation and fertilization policy is crucial for agricultural operations around the globe. Through optimizing land and resource utilization, horticulturalists can increase their crop yields while reducing their impact on the environment. Although there are many other factors that influence the growth and development of a crop, the most useful practices to optimize are irrigation and fertilization as they drive a plant's ability to grow and produce fruit. In this project, a MDP formulation was created to simulate the growth stages and dynamics of a Roma tomato plant. This formulation consisted of constructing compound and complex state spaces, action spaces, and transition matrices from smaller, more specialized state spaces, action spaces, and transition matrices. After creating this MDP formulation, the DQN algorithm was selected to learn the optimal policies for irrigation management as well as irrigation and fertilization management. DQN was selected for its ability to learn in environments that have sparse rewards and for the ability to iterate upon the training parameters. When applied to the irrigation problem, the DQN agent was able to learn a policy that performed 7.06 times better than the heuristic expert policy and when applied to the combined irrigation and fertilization problem, the DQN agent was unable to learn an optimized policy and performed 7.5 times worse than the heuristic expert policy. Although this project has some areas in which it could improve for the application of its results to real-world practices, the results presented are a good starting point for what an optimized policy for land and resource utilization can do to improve both economic and environmental outcomes.

VIII. Contributions and Release

All work on this project was done by the sole contributor, Michael Jon Miller. This included researching other agricultural learning models, growth stages and characteristics of the Roma tomato cultivar, and prices of resources used in the

cultivation of tomatoes as well as writing the code for the tomato MDP, simulator, and Deep Q-Learning Network agent.

The authors grant permission for this report to be posted publicly.

REFERENCES

[1] R. Tao, P. Zhao, J. Wu, N. F. Martin, M. T. Harrison, C. Ferreira, Z. Kalantari, and N. Hovakimyan, "Optimizing Crop Management with Reinforcement Learning and Imitation Learning," arXiv.org, 27-Feb-2023. [Online]. Available: https://arxiv.org/abs/2209.09991. [Accessed: 28-Apr-2023].

[2] J. Wu, R. Tao, P. Zhao, N. F. Martin, and N. Hovakimyan, "Optimizing Nitrogen Management with Deep Reinforcement Learning and Crop Simulations," 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 1712–1720, Jun. 2022.

[3] L. Sun, Y. Yang, J. Hu, D. Porter, T. Marek, and C. Hillyer, "Reinforcement Learning Control for Water-Efficient Agricultural Irrigation," 2017 IEEE International Symposium on Parallel and Distributed Processing with Applications and 2017 IEEE International Conference on Ubiquitous Computing and Communications (ISPA/IUCC), pp. 1334–1341, Dec. 2017.

[4] M. Turchetta, L. Corinzia, S. Sussex, A. Burton, J. Herrera, I. N. Athanasiadis, J. M. Buhmann, and A. Krause, "Learning Long-Term Crop Management Strategies with CyclesGym," 36th Conference on Neural Information Processing Systems (NeurIPS 2022) Track on Datasets and Benchmarks, Nov. 2022.

[5] D. Elavarasan and P. M. Vincent, "Crop Yield Prediction Using Deep Reinforcement Learning Model for Sustainable Agrarian Applications," IEEE Access, vol. 8, pp. 86886–86901, May 2020.

[6] W. T. Kelley, G. E. Boyhan, K. A. Harrison, P. E. Sumner, D. B. Langston, A. N. Sparks, S. Culpepper, W. . Hurst, and E. G. Fonsah, Commercial Tomato Production Handbook. Athens, Georgia: University of Georgia, 2010.

[7] S.-L. Fang, Y.-H. Kuo, L. Kang, C.-C. Chen, C.-Y. Hsieh, M.-H. Yao, and B.-J. Kuo, "Using Sigmoid Growth Models to Simulate Greenhouse Tomato Growth and Development," Horticulturae, vol. 8, no. 11, p. 1021, Nov. 2022.

[8] J. Wang, L. Cui, B. Zhang, and C. He, "An Improved Simulation Model for Tomato Plant Based on Physiological Development Time," 2014 Seventh International Symposium on Computational Intelligence and Design, pp. 465–468, Dec. 2014.

[9] C. Janssen, S. Smith, R. Foster, R. Latin, S. Weller, and F. Whitford, Crop Profile for Tomatoes in Indiana. Raleigh, North Carolina: USDA, 1999.

[10] H. Bargel and C. Neinhuis, "Tomato (Lycopersicon Esculentum Mill.) Fruit Growth and Ripening as Related to the Biomechanical Properties of Fruit Skin and Isolated Cuticle," Journal of Experimental Botany, vol. 56, no. 413, pp. 1049–1060, Feb. 2005.