A Multi-Objective Capacity-Constrained Optimization of Corn Planting Scheduling

Mingshi Cui

Rutgers University, Department of Statistics, New Brunswick, NJ USA

Kunting Qi

Miami University, Department of Statistics, Oxford, OH USA

Byran Smucker[1]

Miami University, Department of Statistics, Oxford, OH USA

Durai Sundarmoorthi

Washington University in St. Louis, Olin Business School, St. Louis, MO USA

Abstract

This article describes an improved set of solutions to the problems presented in the 2021 Syngenta Crop Challenge in Analytics. In particular, we use optimization and predictive modeling methods to produce a corn planting schedule that attempts to minimize the median and maximum absolute difference between weekly harvest quantity and capacity, the number of nonzero harvest weeks, and the total amount of corn wasted. This is accomplished while respecting planting windows, expected harvest amounts, the growing degree units required to bring seeds to harvest, and historical weather data. In one scenario, capacities for two sites are specified. For the other scenario, establishing the capacity is part of the requirement. We used a Long Short-Term Memory model to predict growing degree units for 2020 and 2021, based on historical data. Then, we used a genetic algorithm, and an extensive search of the tuning parameter space, to produce a Pareto front of solutions for three distinct optimization models related to the Challenge. We evaluate the quality of the Pareto fronts for each model, and use the results to choose a preferred model and final solution. For the second scenario, we augmented the optimization models with Capacity, used both as a decision variable and an objective. For both scenarios, we provide comparisons between our final solutions, the initial solutions given in the Syngenta Challenge, and previous solutions we submitted to the Challenge.

Keywords: Scheduling, Agricultural Optimization, Hypervolume, Genetic Algorithms

1. Introduction and Motivation

One of the UN's 17 Sustainable Development Goals is "Zero hunger." Corn is a key food commodity that provides benefits to human beings both directly through its consumption and indirectly through animal consumption. As commercial corn-growing technology has improved, the obvious benefits of food and energy availability are limited by challenges regarding storage capacity. The 2021 Syngenta Crop Challenge

[1]corresponding author, smuckebj@miamioh.edu

illustrates such a problem. Given a specified storage capacity, along with historical data about weather, potential planting intervals, the Growing Degree Units (GDUs) needed to bring the planted seeds to harvest, and the amount of harvest expected, the main question of interest is: when should a set of seed populations be planted in order to minimize waste and provide as consistent a weekly harvest quantity as possible? More specifically, we wish to minimize (a) the median difference between weekly harvest quantity and the capacity across all weeks; (b) the maximum difference between the weekly harvest quantity and the capacity across all weeks; and (c) the number of weeks that a non-zero harvest is realized. Though it was not a part of the original Challenge, we have added a fourth criterion to minimize: (d) the total amount of corn wasted because the weekly harvest exceeds the capacity. These criteria should be simultaneously optimized while respecting the assigned planting window for each seed population, expected harvest, GDUs required, and historical weather data. This is the first Scenario presented in the Challenge. The second Scenario generalizes the problem to require a reasonable capacity specification instead of taking it as given.

In this article, we report a principled solution to a challenging set of problems from industry. We formulate these problems as novel applications of multiobjective optimization, which we solve using a genetic algorithm. The problems have several challenging and unique aspects. First, in order to obtain a solution, we must use historical data to predict the Growing Degree Units (GDUs) over the planting year. This adds a predictive modeling aspect to the project, and we consider both a simple averaging model as well as a neural network-based approach. Second, we sharpen the basic optimization model by considering two related models that reflect the asymmetry inherent in the median and maximum difference objectives. We then construct Pareto fronts based on all three models but assess these Pareto fronts based on the original model, since it most explicitly reflects the problem outlined in the Challenge. Interestingly, we find that we always obtain better results using one of the related models, instead of using the original model. Third, on the multiobjective algorithm side, we perform an extensive round of parameter tuning to improve the optimization results, and use the hypervolume indicator to discriminate between competing Pareto fronts, before using another method from the literature to select a final solution. Fourth, for Scenario 2, we augment the models to include Capacity as both a decision variable and objective, establishing a reasonable Capacity requirement. We compare our solutions with initial solutions provided as part of the Challenge, as well as with solutions that we originally submitted as part of the second-place entry.

The rest of the article continues as follows. Section 2 surveys the literature related to optimization in agriculture. Section 3 provides a detailed description of the problem as well as the mathematical formulation of the three optimization models that we consider. Section 4 describes the methodological approach we took to solve the problem, including a description of the predictive model we used, the optimization procedure, and how we tuned the algorithm. In Section 5, we demonstrate our methodology and provide particular solutions, which we compare to previous solutions. Finally, we close with a short discussion in Section 6.

2. Literature Review

Our efforts are situated within the optimization literature as applied to agriculture; in particular, work which uses several elements of analytics, along with traditional optimization, to guide decision-making. Heady (1954) is one of the earliest operations research works that demonstrated the use of linear programming in farm management. Through empirical examples, this work presented how a farm can optimally allocate their land and labor resources between crops like corn, wheat, potatoes, and oats. They also showed how crop profitability can be estimated from the results of a linear programming model. Donaldson (1968) studied the interplay of uncertain weather during harvesting, cost and capacity of harvesting equipment, and moisture level of the harvested cereal using simulation. Corrie & Boyce (1972) optimized the harvesting schedule for certain fruits and vegetables which need multiple harvest passes during their maturation period. They utilized dynamic programming to optimize the harvesting schedule of cauliflower. Morey et al. (1972) formulated corn and soybean harvesting operations as a dynamic programming problem with each calendar week considered as a stage in the dynamic program. They optimized the number of combine harvester operation hours for each week to maximize the profit, considering also the moisture content of the harvest – a crucial factor which determines the revenue generated by the crop. Audsley & Boyce (1974) prescribed optimal harvesting policies to minimize equipment and grain loss. Miyake et al. (1979) solved a tobacco harvesting problem using stochastic dynamic programming. They prescribed the optimal starting time of harvest and the amount of labor needed for harvesting. They quantified the timeliness of farm operations by calculating the loss incurred when the harvest is delayed. Similar to the present work, Fokkens & Puylaert (1981) have tackled a single-objective harvesting optimization problem constrained by capacity and weather by using linear programming. However, the nature of the crop varieties considered in their work did not allow them to spread the planting window, whereas in our work there are multiple objectives and corn varieties come with flexibility in planting dates within a window. This flexibility allows us to improve the solution but at the cost of a more complex model. Glen (1987) provided a survey of operations research models utilized in agriculture up until that point. They broadly divided their survey into crop production and livestock production. They further divided the crop production area into determination of cropping policy, planning of harvesting operations, evaluating capital investments, and pest and disease control strategies. Their survey identified usage of mixed integer programming, linear programming, deterministic dynamic programming, stochastic dynamic programming, goal programming, decision theory, simulation, enumeration, portfolio theory, heuristics, and decision trees in solving these crop and livestock production problems. Glen (1987) notes that a lack of data is a common problem in this field, and we note that the present problem includes extensive data, described in the following section.

More recently, Lowe & Preckel (2004) provide an updated review of this literature, though they confined their review to research about crop planting, crop harvesting, and risk management, while discussing methods including linear programming, stochastic programming, risk programming, dynamic programming, and simulation. One key reference is Jones et al. (2001), who used a two-period sequential production scheduling

problem for an agribusiness which has operations in North America and South America. The agribusiness plants and harvests seeds in these two places for reselling in North America. In the first period, they have production operations in North America between April and September. In the second period, they produce in South America from October to March. Having production activities in North America and South America was a new phenomenon that began a couple of decades back. It is now a common practice of all major seed producers to produce corn in different countries and resell to farmers in the United States. Bansal et al. (2017) proposed an optimization-based approach to estimate parameters for yield distribution based on expert's judgement about the potential yield of different seed varieties. Estimation of these yield parameters enabled optimal production planning including allocation of land by an agribusiness for growing seed corn. Bansal et al. (2017) also indicate that a lack of data is a limiting factor in work in this area. This situation has been improved after Syngenta – an agribusiness - won the 2015 Edelman Award given by the Institute for Operations Research and the Management Sciences (INFORMS) for Byrum et al. (2016), which uses simulation and optimization in seed development and the production process. After winning the Edelman Award, Syngenta organized a series of annual analytics challenges to solve common problems in the agriculture domain using operations research. Through these INFORMS analytics challenges, Syngenta opened up their data for research that attracted participants from around the world to develop data-driven solutions to problems posed in these challenges. Zhong et al. (2017) constructed a hierarchical machine learning approach to predict yield of different soybean varieties and optimized a portfolio of varieties with the predicted yield serving as the input for the optimization model. Marko et al. (2017) used multiple linear regression, random forest, support vector machine, k-nearest neighbours, artificial neural network, and weighted histograms regression (Marko et al., 2016) to predict the yield of different varieties for a region and identifying the optimal varieties to be sold in the region. Sundaramoorthi & Dong (2022) used classification and regression trees to estimate parameters needed for an optimization model that chose the best varieties to be grown at a farm. Feng & Zhou (2021) used multi-task learning to predict the yield of multiple varieties simultaneously and used those predictions to optimize the varieties to be sold by a dealer. In another work that stemmed out of the analytics Challenge, Ansarifar et al. (2020) integrated random forest with optimization to predict the performance of new varieties. Our own research here elaborates upon the second-place entry of the 2021 Syngenta Crop Challenge, and presents a novel solution to a problem which to our knowledge has only recently been considered in the literature by two other participating teams (Sajid & Hu, 2022; Khalilzadeh & Wang, 2022).

3. Problem Description and Formulation

In this section we provide a general description of the problem and data that was provided, and then a mathematical formulation of the optimization problems we are trying to solve.

3.1. Description

The original motivation for this work was the 2021 Syngenta Crop Challenge. The problem was to provide a corn planting schedule that results in a weekly harvest quantity consistently close to capacity. Two datasets were provided. First, information about a set of 2,569 seed populations, including the site at which each population was to be planted (site 0 or 1); a window of time within which each population could be planted (early planting dates ranged from January 1, 2020 to December 29, 2020; late planting dates ranged from January 22, 2020 to February 16, 2021); the number of required GDUs needed to bring the seeds to harvest (range from 649 to 1,414); and the number of ears that the seeds would produce once they received the necessary GDUs, in Scenario 1 or Scenario 2 (Figure 1). Second, Syngenta provided the historical GDUs for each day from 2009 to 2019, at sites 0 and 1. The data provided is summarized in Table 1. Thus, each seed population must be planted at its specified site, but the seed population can be planted under either Scenario 1 or Scenario 2, which have differing harvest quantities. Furthermore, the date when a seed population is ready to harvest is determined by the site-specific daily GDUs and its own required GDUs. Note that the raw data provided to Challenge participants is proprietary and, based on a non-disclosure agreement, cannot be shared publicly.

Variables	**Description**
Population	1,375 seed population identifiers (Site 0), 1,194 seed population identifiers (Site 1)
Original Planting Date	Actual planting date of the population
Early Planting Date	Earliest date the population can be planted
Late Planting Date	Latest date the population can be planted
Required GDUs	GDUs needed to harvest a seed population (range: 649 to 1,414)
Harvest Quantity	Number of ears of corn for each seed population (Scenario 1 or Scenario 2)
Historical Daily GDUs	GDUs accumulated for each calendar day from 2009-2019 (Site 0 or Site 1)

Table 1: Description of the data provided in the 2021 Syngenta Crop Challenge.

As mentioned, the Challenge included two Scenarios, and for each Scenario participants were required to specify the planting date for each seed population. For Scenario 1, the weekly storage capacity was specified to be 7,000 ears at site 0 and 6,000 ears at site 1, with the sites to be optimized separately. The goal as specified in the Challenge was to minimize the first three quantities specified in Section 1, though as discussed we have added a fourth. For Scenario 2, the Challenge provided no capacity. The task in this scenario was to specify the schedule but also provide a meaningful capacity.

3.2. Mathematical Formulation

Suppose at site 0 there are n_0 seed populations and at site 1 there are n_1 seed populations. Without loss of generality, we will describe the optimization setting for site 0. We represent the n_0 seed populations as $s_1, s_2, \ldots, s_{n_0}$. Seed population s_i is associated with a known harvest quantity $h(s_i)$; a known amount of growing degree units (GDUs) necessary to trigger its harvest, $g(s_i)$; and an interval composed of two

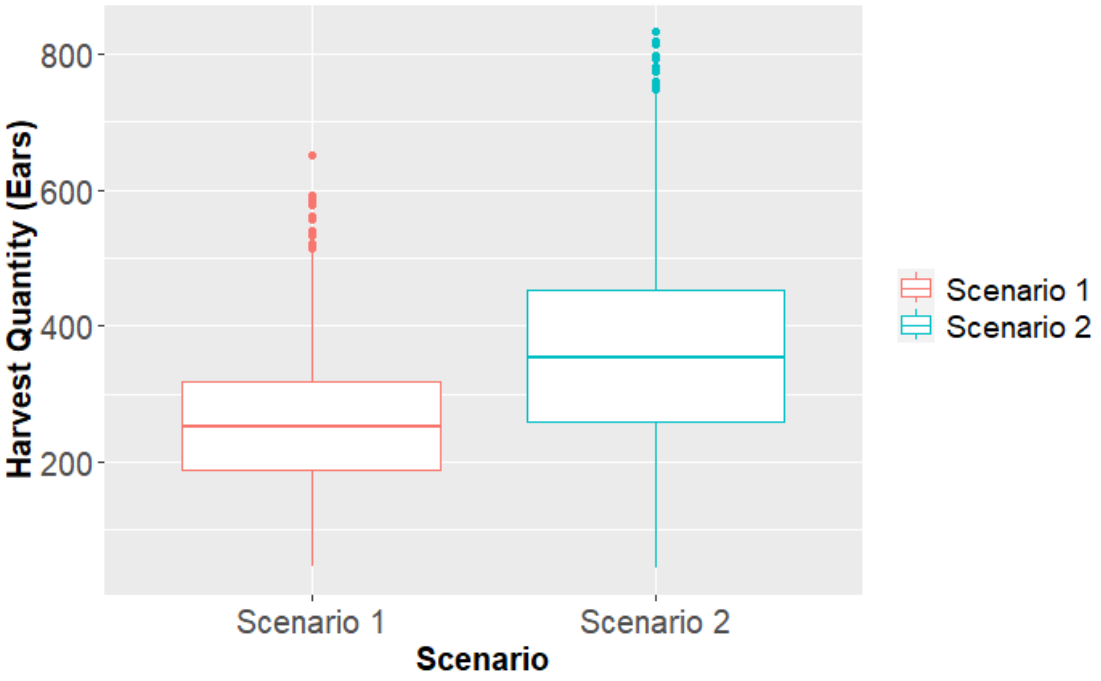


Figure 1: Provided seed population harvest quantities under both Scenarios. According to the Challenge, they roughly follow a $N(250, 100)$ distribution; for Scenario 2, they roughly follow $N(350, 150)$.

dates within which the population must be planted, $[L(s_i), U(s_i)]$. Also, let g_d be the GDU for day d for $d = 1, 2, ..., D$, where D is the number of days in the optimization period. Furthermore, $\hat{g}_d$ is the number of predicted GDUs for day d, as estimated by our predictive model (see Section 4.1). Finally, we use p_i to represent the planting day for seed population s_i. The $\mathbf{p} = p_i$ are the decision variables in our optimization and must be such that $L(s_i) \leq p_i \leq U(s_i)$, for $i = 1, 2, \ldots, n_0$. We then have the full harvest quantity on week j as $h_j(\mathbf{p}) = \sum_{i \in \mathcal{H}_j} h(s_i)$, $j = 1, 2, \ldots W$, where $\mathcal{H}_j$ includes the indices for seed populations that are harvested in week j, W is the number of weeks in the optimization period, which means W should be counted from the first week with corn planted. Mathematically, and to emphasize h_j's dependence upon $\mathbf{p}$, note that $i \in \mathcal{H}_j$ if $\sum_{d=p_i}^{d^*} \hat{g}_d > g(s_i)$ where $d^* = \arg\min_{d'=p_i, p_i+1, \ldots, D} \left\{ \sum_{d=p_i}^{d'} \hat{g}_d > g(s_i) \right\}$. Finally, let $\mathcal{J}$ be the set of all week indices for which $h_j > 0$ and $C_0 = 7{,}000$ be the weekly harvest capacity for site 0 in scenario 1 ($C_1 = 6{,}000$).

3.2.1. Model 1: Base

Given the problem specification above there are four criteria to minimize, and they can be represented as $\mathbf{f}_1(\mathbf{p}) = (f_{11}(\mathbf{p}), f_{12}(\mathbf{p}), f_{13}(\mathbf{p}), f_{14}(\mathbf{p}))$, where

- $f_{11}(\mathbf{p}) = \text{median}_{j \in \mathcal{J}}(|C_0 - h_j(\mathbf{p})|)$,
- $f_{12}(\mathbf{p}) = \max_{j \in \mathcal{J}}(|C_0 - h_j(\mathbf{p})|)$,
- $f_{13}(\mathbf{p}) = \sum_{j=1}^{W} I(h_j(\mathbf{p}) > 0)$,

- $f_{14}(\mathbf{p}) = \sum_{j=1}^{W}(h_j(\mathbf{p}) - C_0)^+$,

I is an indicator function, and $x^+ = \max(x, 0)$. The first and second criteria control the median and maximum absolute difference from capacity, respectively, while the third criterion minimizes the number of non-zero harvest weeks. The final criterion minimizes the total amount of wasted product.

This basic criteria set has the possible drawback that the penalty for being overcapacity is not very severe, which may lead to difficulty in finding an efficient Pareto front based on this model. We address this issue in the soft constraint models in the next sections.

3.2.2. Model 2: First Penalty Model

For a given planting schedule, let $a_j = h_j - C_0,\ j \in \mathcal{J}_a$ where $\mathcal{J}_a$ is the set of weeks for which the harvest quantity is above capacity with cardinality n_a, and $b_j = C_0 - h_j,\ j \in \mathcal{J}_b$ where $\mathcal{J}_b$ is the set of weeks for which the harvest quantity is below capacity with cardinality n_b. These quantities separately mark the positive part and negative part of the weekly harvest deviations from capacity. In order to impose a soft capacity constraint, we solve a new multiobjective optimization problem with the following criteria to be simultaneously minimized: $\mathbf{f}_2(\mathbf{p}) = (f_{21}(\mathbf{p}), f_{22}(\mathbf{p}), f_{23}(\mathbf{p}), f_{24}(\mathbf{p}))$, where

- $f_{21}(\mathbf{p}) = \text{median}_{j\in\mathcal{J}}|C_0 - h_j(\mathbf{p})|$,
- $f_{22}(\mathbf{p}) = \frac{\sum_{j\in\mathcal{J}_a} a_j^r(\mathbf{p})}{n_a}$,
- $f_{23}(\mathbf{p}) = \frac{\sum_{j\in\mathcal{J}_b} b_j(\mathbf{p})}{n_b}$,
- $f_{24}(\mathbf{p}) = \sum_{j=1}^{W} I(h_j(\mathbf{p}) > 0)$

As with the Base Model, this formulation controls both the median absolute deviation from capacity and the total number of nonzero harvest weeks. However, this model directly confronts the asymmetry between the costs of being overcapacity and undercapacity by minimizing the average of the r^{th} power of all overcapacity amounts as well as the unexponentiated average of all undercapacity amounts.

3.2.3. Model 3: Second Penalty Model

Finally, we consider an additional model that focuses on severely penalizing deviations from capacity, but also making the deviations as uniform as possible. In particular, Model 3 includes the following criteria, again to simultaneously minimize: $\mathbf{f}_3(\mathbf{p}) = (f_{31}(\mathbf{p}), f_{32}(\mathbf{p}), f_{33}(\mathbf{p}), f_{34}(\mathbf{p}))$

- $f_{31}(\mathbf{p}) = \text{median}_{j\in J}|C_0 - h_j(\mathbf{p})|^r$,
- $f_{32}(\mathbf{p}) = \max_{j\in J}|C_0 - h_j(\mathbf{p})|^r$,
- $f_{33}(\mathbf{p}) = \text{sd}_{j\in J}(C_0 - h_j(\mathbf{p}))$,
- $f_{34}(\mathbf{p}) = \sum_{j=1}^{W} I(h_j(\mathbf{p}) > 0)$,

where **sd** represents the standard deviation of the harvest deviations from capacity. Note that we also make the median and max criteria more severe by adding a power to each.

4. Solution Methodology

In Sections 4.1 and 4.2 we provide a description of our prediction of GDUs and optimization approach, respectively. Overall, our solution strategy includes two basic building blocks. First, we construct a predictive model to predict the GDUs for each relevant day in 2020 and 2021, at each site. Then, we build a multiobjective optimization framework to construct Pareto fronts of solutions for the four objectives, using a genetic algorithm. In order to enrich our search of the solution space, we consider the three model formulations discussed in Section 3.2, and for each we perform an extensive round of tuning. This leads to an assessment of the resulting Pareto fronts, and we use the hypervolume indicator to choose between the three Pareto fronts (one for each of the three optimization models of Section 3.2). Finally, we select a particular solution from the chosen Pareto front by using the Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS, Wang & Rangaiah, 2017). Note that when we compare the three Pareto fronts, we always do so by assessing their quality according to the Base Model (Section 3.2.1), since this is most reflective of the original Challenge objectives.

4.1. Predictive Models for Daily GDUs

A critical parameter within the optimization problem are the g_d's, which are the Growing Degree Units (GDUs) for day d within the planting window (January 1, 2020 to February 16, 2021) at site 0 and site 1. After seed population s_i is planted, it will be harvested once the accumulated GDUs are greater than $g(s_i)$. Since we don't know the g_d's in advance for both sites, we must estimate them from historical data, $\hat{g}_d$. As mentioned in Section 3 and Table 1, this data specifies the number of daily GDUs for each calendar day from 2009 to 2019 at site 0 and site 1. We compare two methods to predict the g_j's. The first one is a simple averaging procedure, and the second is a long short-term memory (LSTM) neural network.

4.1.1. Simple Averaging Model

This method simply takes the average of the GDUs for each specific day of the year. For instance, in order to predict the daily accumulated GDUs on January 1, 2020, we take the average for all of January 1's from 2009 to 2019. This average value is counted as the predicted value for January 1, 2020 and January 1, 2021. The same procedure is performed for the other 364 days of the year with all leap days being omitted.

4.1.2. Long Short-Term Memory Model

The second model we used for predicting daily GDUs is the Long Short-Term Memory (LSTM) neural network (Hochreiter & Schmidhuber, 1997). As a type of deep learning method, LSTM can model both linear and non-linear relationships within time-dependent observations, which means it is more flexible than traditional time-series modeling methods such as ARIMA (Siami-Namini et al., 2018; Box, 2013). In fact, Siami-Namini et al. (2018) compared LSTM and ARIMA using financial data, and they showed that the LSTM-based algorithm was superior to ARIMA and, on average, the LSTM-based algorithm improved predictions by around 85%.

LSTM is an improved version of a recurrent neural network (RNN, Medsker & Jain, 1999), the architecture of LSTM can address the limitations of traditional RNN on modelling long-term dependencies. The special "cells" in LSTM are called as LSTM units, in each hidden layer of the neural network, and these cells are connected sequentially. LSTM apply both hidden state and cell state in its interval architecture, where hidden state represents the current "memory" of the network based on the input and the information of the previous cell, and cell state is the memory component of LSTM for storing information over long-time period. As comparison, traditional RNN only utilizes hidden state in its architecture. The modification of LSTM over traditional RNN makes it able to establish a mechanism so that each cell has an input gate, a forget gate, and an output gate, and these gates will together determine the utilization of information from the previous cell, the memorization of information in current cell, the output of the current cell, and the information that should be passed to the next cell. This sequence of output is either passed into the next hidden layer for further processing, or processed by a regular deep learning output layer (called the Dense layer) to produce a final prediction. Compared to the traditional RNN, LSTM models remember a larger number of steps in the sequence via the memory gates (Goodfellow et al., 2016).

There are several hyperparameters that need to be specified for the LSTM model, including the number of epochs, batch size and number of hidden layers. An epoch is a complete pass through the entire training dataset, the batch size is the number of training samples randomly drawn from the training dataset and fed into the neural network model at one time (Siami-Namini et al., 2018), and the number of hidden layers is the number of layers in the network which neither directly receive data input nor output the final result Goodfellow et al. (2016). As a neural network model, LSTM uses gradient descent to optimize its internal parameters across epochs.

In our work, we used the LSTM network to predict GDU g_{ymd}, where y is the year, m is the month, and d is the day within month. (Note that our notation deviates slightly here, since previously as part of the problem specification we defined g_d as the GDU for day d of the optimization period; here, we use more flexible notation to capture days as far back as 2011.) We use this to define a sequence of GDU's comprised of the same day/month from the previous l years. That is, g_{ymd} is associated with the sequence $g_{y-l,md}$, $\ldots$, $g_{y-2,md}$, $g_{y-1,md}$. We denote this sequence $(g_{y-l,md}, \ldots, g_{y-2,md}, g_{y-1,md}; g_{ymd})$. We consider, as our training set, all such sequences for $y \in (2017, 2018, 2019)$ with $l = 8$ (a total of $365 \times 3 = 1{,}095$ sequences). For instance, a particular sequence would be $(g_{2011,7,13}, g_{2012,7,13}, \ldots, g_{2018,7,13}; g_{2019,7,13})$ and this would use information from July 13 in years $2011 - 2018$ to predict the GDUs on July 13, 2019. Based on informal testing, we chose the batch size to be 5 with 80 epochs. Tables 2 and 3 suggest that, compared with 3 hidden layers, 2 hidden layers result in a model that predicts well. Thus, the process of training the model would be as follows. Randomly select 5 sequences, and use them to train the model. Repeat this process until all of the sequences in the training set are exhausted. This constitutes one epoch. Repeat this process for a total of 80 epochs. Our LSTM framework, then, is composed of one Input Layer, followed by two LSTM hidden layers, in which the first LSTM hidden layer has K_1 LSTM units ("cells" in our informal description

above), the second LSTM hidden layer has K_2 LSTM units, and a Dense layer to produce predicted GDU from the output of the second LSTM hidden layer. As shown in Figure 2, the Input Layer takes the sequence $g_{y-l,md}, \ldots, g_{y-2,md}, g_{y-1,md}$ as input and feeds the input into the first LSTM hidden layer. Using the notation of James et al. (2021), the first LSTM hidden layer takes the input to generate the intermediate output of the first LSTM hidden layer, $A^{(1)}$ which is a two-dimensional matrix composed of l rows and K_1 columns. The second LSTM hidden layer uses the two-dimensional matrix $A^{(1)}$ as input to generate a vector $A^{(2)}$ which has K_2 elements. The Dense Layer uses the vector $A^{(2)}$ to compute the predicted $\hat{g}_{ymd} = \beta_0 + \sum_{k=1}^{K_2} \beta_k A_k^{(2)}, k = 1, \ldots, K_2$. Based in part on informal testing, for our implementation of the model with 2 hidden layers, we chose $K_1 = 64$ and $K_2 = 16$. (For the model implementation with 3 hidden layers, we used $K_1 = 200$, $K_2 = 64$ and $K_3 = 16$.)

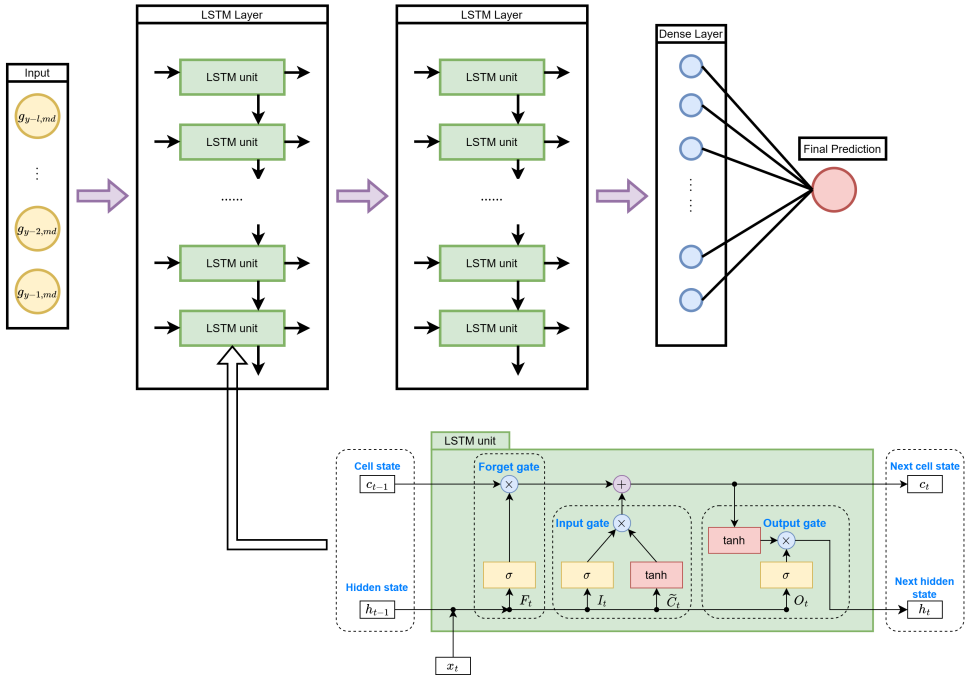


Figure 2: Compared to traditional RNN, LSTM units are composed of hidden state, cell state and different types of gates: input gates, forget gates and output gates. These states and gates cooperatively determine the flow of information within the sequentially connected LSTM units (Altché & de La Fortelle, 2018, Figure 4). The input layer takes the GDU sequence as input and passes the input into the first LSTM hidden layer. Then the first LSTM hidden layer produces a two-dimensional matrix as intermediate output and the second LSTM hidden layer uses the two-dimensional matrix to produce an intermediate vector. Finally, the Dense layer uses the intermediate vector to produce the final GDU prediction. More details regarding the LSTM units can be obtained in Altché & de La Fortelle (2018) and Gers et al. (1999)

The models were trained and evaluated on a holdout set using the mean absolute error (MAE) and mean squared error (MSE). Tables 2 and 3 show the performance of LSTM models for Site 0 and Site 1, respectively, computed by using the sequences implied by $y \in \{2017, 2018\}$ as the training set, with the sequences in 2019 as the holdout set. Based on the Tables and the previous informal testing, we chose to use the LSTM model with 2 hidden layers, a batch size of 5, and 80 epochs. The final version of the

Site 0	LSTM(2, 80)	LSTM(3, 80)	Simple Averaging
MAE	**0.548**	0.759	0.624
MSE	**0.682**	0.911	0.757

Table 2: Comparison of 2019 prediction results for Site 0 between LSTM Models and the simple averaging model. For LSTM(x, y) in the header, x is the number of layers of LSTM, and y is the number of epochs.

Site 1	LSTM(2, 80)	LSTM(3, 80)	Simple Averaging
MAE	**0.725**	0.754	0.825
MSE	**0.952**	0.992	1.062

Table 3: Comparison of 2019 prediction results for Site 1 between LSTM Models and the simple averaging model. For LSTM(x, y) in the header, x is the number of layers of LSTM, and y is the number of epochs.

model was trained using the sequences implied by $y \in \{2017, 2018, 2019\}$, and used to predict GDU's for 2020. When predicting time series, the further into the future predictions are made, the more variability is included in the predictions, and thus less seasonal structure is included in the predictions. In our case, instead of using the $y \in \{2017, 2018, 2019\}$-trained model to predict GDU's in 2021, we trained the model again with $y \in \{2018, 2019, 2020_{\hat{g}}\}$ and $l = 9$, where $2020_{\hat{g}}$ represents the predicted values for 2020. That is, we imputed GDU values for 2020 in order to make more variable predictions for 2021, so that the resulting predictions produced patterns more similar to past seasonal behavior.

4.1.3. Predictive Model Results

Figures 3 and 4 visualize the historical daily GDUs from 2009 to 2019 and the predicted daily GDUs from 2020 to 2021 for Site 0 and Site 1, respectively.

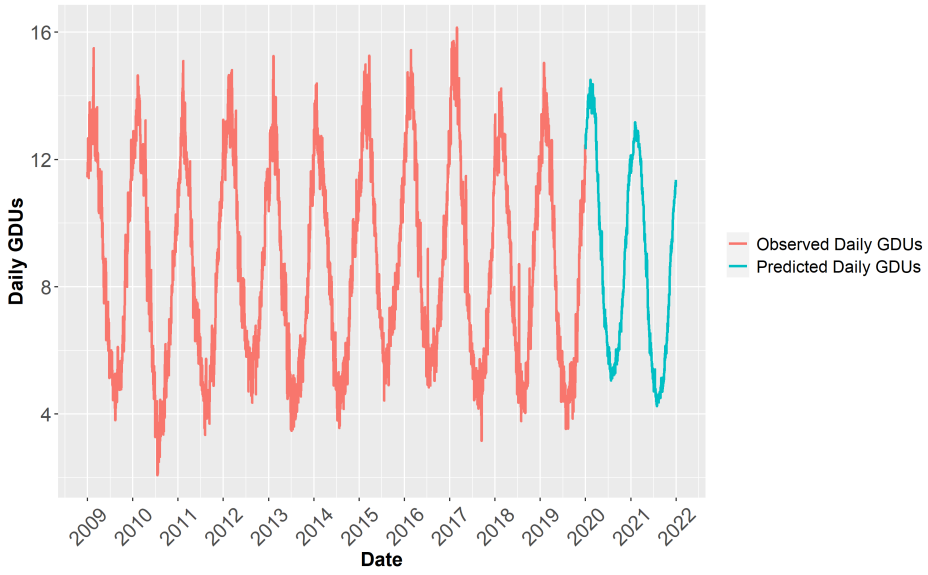


Figure 3: For Site 0, real GDU values between 2009 and 2019, and predicted GDU values for 2020 and 2021, using the LSTM(2,80) model.

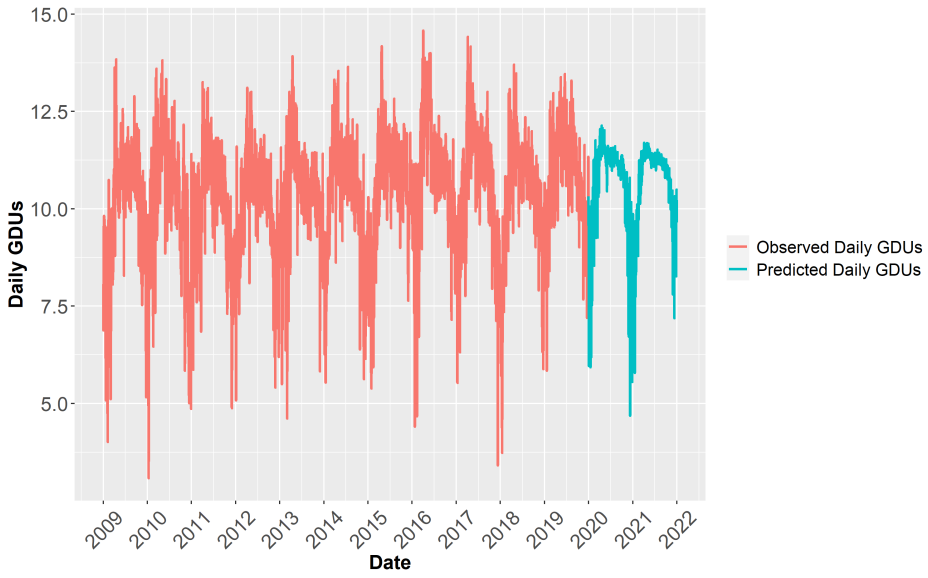


Figure 4: For Site 1, real GDU values between 2009 and 2019, and predicted GDU values for 2020 and 2021, using the LSTM(2,80) model.

4.2. Multi-objective Optimization Methodology

To solve each of the specified optimization problems in Section 3.2, we used an R implementation of the non-dominated sorting genetic algorithm II (NSGA-II) (Deb et al., 2002; Tsou, 2022). This procedure includes the elements of classical genetic algorithms (Whitley, 1994; Kumar et al., 2010; Mirjalili, 2019; Katoch et al., 2021), including the characterization of solutions as genes, as well as crossover and mutation. The overall idea is that a population is initialized and over a number of generations evolves to be more

fit, where fitness is measured by the objective functions. NSGA-II naturally handles multiple objectives in settings where objectives are treated as a black box, and attempts to provide a set of Pareto optimal solutions. In this section we provide a description of our solution strategy, including an extensive search of the tuning parameter space, details regarding Pareto optimality, the hypervolume measure of Pareto fronts, and the TOPSIS approach to choosing a final solution.

4.2.1. Solution Strategy

We characterize the solution of our multiobjective optimization problem in terms of Pareto optimality. Following Cao et al. (2015) and using notation from Sections 3.2.1-3.2.3, for optimization model $\mathbf{f}_k$, a solution $\mathbf{p}_1$ is said to dominate solution $\mathbf{p}_2$ if $f_{ko}(\mathbf{p}_1) \leq f_{ko}(\mathbf{p}_2)\ \forall\ o \in \{1,2,3,4\}$ with $f_{ko}(\mathbf{p}_1) < f_{ko}(\mathbf{p}_2)$ for at least one $o \in \{1,2,3,4\}$; whereas $\mathbf{p}_1$ weakly dominates $\mathbf{p}_2$ if $f_{ko}(\mathbf{p}_1) \leq f_{ko}(\mathbf{p}_2)\ \forall\ o \in \{1,2,3,4\}$. A set of solutions is Pareto optimal if the set consists of nondominated solutions, and this set of solutions is said to occupy the Pareto front. The hypervolume measure (Zitzler & Thiele, 1998) measures the volume of the criteria-space that is weakly dominated by the set of points composing a Pareto front, and we use this measure to compare the quality of different Pareto fronts. In order to bound the hypervolume measure, a reference point must be defined and we discuss that below.

As specified in Section 3.2, we have defined $\mathbf{f}_1(\mathbf{p})$ as the set of four basic criteria that we would like to optimize. We defined two additional sets of criteria in Sections 3.2.2 and 3.2.3, $\mathbf{f}_2(\mathbf{p})$ and $\mathbf{f}_3(\mathbf{p})$, but we only use these latter two sets as vehicles to explore the $\mathbf{f}_1$-solution space more fully. That is, though we solve the $\mathbf{f}_2$ and $\mathbf{f}_3$ problems, when considering possible solutions using Pareto fronts we first translate those solutions into the $\mathbf{f}_1$ criteria-space. This is in keeping with our goal of producing a practical solution to the provided Challenge problem, and it also provides the feasibility for us to use hypervolume measure to compare Pareto fronts between models in the later steps.

One difficulty with the NSGA-II approach is that a number of parameters must be specified in order to ensure an effective solution: the number of generations, the population size, and both crossover and mutation probabilities. In addition, for the two penalty models (Sections 3.2.2 and 3.2.3), we denote the exponent r as an additional parameter to be chosen. In the following subsection we provide more details regarding our study of these parameters, which explores a variety of parameter combinations for each of the three multi-objective problems. For now we simply denote the set of parameter combinations—for each of $\mathbf{f}_1$, $\mathbf{f}_2$, and $\mathbf{f}_3$—as $\mathcal{P}_1$, $\mathcal{P}_2$, and $\mathcal{P}_3$, respectively. Each implementation of NSGA-II for $\mathbf{f}_k$, for each parameter combination in $\mathcal{P}_k$, will include a final population of solutions, denoted $\text{POP}_{k\ell},\ k = 1,2,3;\ \ell = 1,2,...,c_k$, where c_k is the number of parameter combinations for $\mathcal{P}_k$. Each population $\text{POP}_{k\ell}$ includes $n_{k\ell}$ solutions, depending on the population size of the NSGA-II implementation for model k and parameter combination ℓ. Our basic solution strategy, then, is:

1. Solve $\mathbf{f}_k$ for each combination of tuning parameters in $\mathcal{P}_k$, $k = 1,2,3$, respectively, yielding $\text{POP}_{k\ell},\ k = 1,2,3;\ \ell = 1,2,...,c_k$.

2. Construct three populations $P_k = \bigcup\{\text{POP}_{k\ell} | \ell \in \{1, 2, \ldots, c_k\}\}$ with $k = 1, 2, 3$. Map each $\mathbf{p} \in P_k$ to the criteria in $\mathbf{f}_1$ as $\mathbf{f}_1(\mathbf{p})$, because we are evaluating all solutions, regardless of whether they solved $\mathbf{f}_1$, $\mathbf{f}_2$, or $\mathbf{f}_3$, in terms of the objective functions associated with $\mathbf{f}_1$.
3. From each P_k after mapping to the criteria f_1, construct Pareto fronts $PF_k, k = 1, 2, 3$. Then construct a meta-population $P_{PF} = \bigcup\{PF_k | k \in \{1, 2, 3\}\}$.
4. Let $F_{1o}^{\min} = \min_{\mathbf{p} \in P_{PF}} f_{1o}(\mathbf{p})$ and $F_{1o}^{\max} = \max_{\mathbf{p} \in P_{PF}} f_{1o}(\mathbf{p})$ be the minimum and maximum value of criterion f_{1o}, respectively, for model 1 and objective o, across all solutions in P_{PF}. For each $\mathbf{p} \in P_{PF}$, scale $\mathbf{f}_1(\mathbf{p})$ such that $\mathbf{f}_1^{\text{sca}}(\mathbf{p}) \in [0, 1]^4$, where $f_{1o}^{\text{sca}} = \frac{f_{1o} - F_{1o}^{\min}}{F_{1o}^{\max} - F_{1o}^{\min}}$ for $o = 1, 2, 3, 4$.
5. For each PF_k after scaling in Step 4, compute hypervolume v_k based on $[\mathbf{f}_{11}^{sca}, \mathbf{f}_{12}^{sca}, \mathbf{f}_{13}^{sca}, \mathbf{f}_{14}^{sca}]$, and choose $k^* = \text{argmax}_k v_k$. This defines the model which yields the best scaled Pareto front.
6. From PF_{k^*}, the best single solution $\mathbf{p}^*$ is chosen via TOPSIS (Wang & Rangaiah, 2017, see Section 4.2.3). The chosen solution $\mathbf{p}^*$ produces $\mathbf{f}_1(\mathbf{p}^*)$ as the final criteria values from the originally specified set of objectives.

Remark 1: We emphasize again that Models 2 and 3 are simply vehicles to improve the solutions obtained with respect to Model 1, because Model 1 most closely aligns with the original requirements of the Competition. This is why, in Step 2 and Step 6 above, we evaluate the solutions using the objectives defined in Model 1.

Remark 2: In order to compute the hypervolumes v_k, we not only need to scale the criteria, but we also must define a reference point. After some experimentation, we chose $[2, 2, 2, 2]$, recognizing that for all of the reference points we explored, the hypervolume ordering of the Pareto fronts stayed the same.

Remark 3: We have chosen to take a middle road regarding the pooling of solutions. On the one hand, one could choose to pool all solutions from all models, and construct a single Pareto front from which a final solution is chosen, so that the hypervolume technique can be ignored. On the other hand, one could form separate Pareto fronts for each parameter combination for each model, so that the best Pareto front is chosen based on both the best model and parameter combination. Instead, we have pooled across parameter combinations but preserving separate populations of solutions for each model. This allows some insight to be obtained regarding whether the alternative models result in more attractive solutions.

4.2.2. Tuning Parameter Selection

The NSGA-II heuristic requires the input of several parameters: the number of generations, the population size, and both crossover and mutation probabilities. In addition, for the two penalty models, we denote the exponent r as an additional parameter. In order to study the quality of the Pareto fronts across values of these parameters, we used NSGA-II for a number of different combinations of these parameter values, denoted $\mathcal{P}_k$ for model k above. Table 4 shows these sets, chosen based upon several works in the literature (Alander, 1992; Boyabatli & Sabuncuoglu, 2004; Najafi et al., 2009). Alander (1992) and Najafi et al. (2009) suggested exploring from N to $3N$, where N is the problem size (in our case, the total number

Table 4: For site 0, the three levels studied for each tuning parameter.

	Low	**Middle**	**High**
Crossover Rate	0.5	0.75	1.0
Mutation Rate	0.001	0.01	0.1
Population Size[1]	1376	2752	4128
Generation Size	8000	10000	12000
Penalty Power[2]	1	2	3

[1] Only for Models 2 and 3.

[2] For site 1, population sizes were 1,196, 2,392, and 3,588.

of seed populations). So, possible population sizes for site 0 are N = 1,376, $2N$ = 2,752 and $3N$ = 4,128, and similarly for site 1. For the other settings of tuning parameters, we are guided in part by the design of parameters in Najafi et al. (2009). In all, $\mathcal{P}_1$ consisted of $3^4 = 81$ parameter combinations, while $\mathcal{P}_2$ and $\mathcal{P}_3$ included $3^5 = 243$. This was a computationally intensive process, so we used Miami University's Redhawk Cluster, which has a CentOS 7.9.2009 operating system and uses an Intel Xeon Gold 6126 processor, with 2.6 GHz clock speed. Each node of the cluster has 96 GB available memory and the optimization problem for each parameter combination was solved using a single CPU core.

Thus, we use NSGA-II to optimize each model for each of the tuning parameter combinations in $\mathcal{P}_1$, $\mathcal{P}_2$ and $\mathcal{P}_3$, respectively. To determine which model, and which set of tuning parameter values, produced the best Pareto front, we implemented the strategy outlined in Section 4.2.1. Note that we tried several approaches to determine which parameter combination would optimize the hypervolume, include response surface methodology (Myers et al., 2016), which would even allow us to choose an interpolated parameter combination that was not necessarily explored in our formal study. However, these methods did not clearly improve upon the results observed directly by choosing the parameter combination, for each model, that produced the best hypervolume.

4.2.3. Choosing the Final Solution from Pareto Front

The Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS, Wang & Rangaiah, 2017) is a straightforward method to choose a particular Pareto solution. In Scenario 1, TOPSIS chooses the final solution $\mathbf{p}^*$ from the Pareto front PF_{k^*}. For simplicity, in this section we will explain the procedure using a generic Pareto front PF, composed of $i = 1, 2, \ldots, n$ solutions each evaluated on $o = 1, 2, 3, 4$ objectives. The basic idea of the method: TOPSIS first normalizes each solution in the Pareto front (Step 1 below), and then uses the Euclidean distance of a solution from both the positive and negative ideal to choose a particular solution (Steps 3-5 below).

- Step 1. For solution i and objective o, normalize the Pareto front to create the normalized Pareto front

F_{io} as follows:

$$F_{io} = \frac{PF_{io}}{\sqrt{\sum_{i=1}^{n} PF_{io}^2}}$$

- Step 2. Adjust each solution according to user-specified weights w_o:

$$V_{io} = F_{io} * w_o$$

For Scenario 1, we assume each objective function is equally important, thus the weights are constant.

- Step 3. Find the positive ideal solution A^+ and the negative ideal solution A^-. Here, our optimization goal is to minimize each objective; i.e. $min(\mathbf{f}_{1o}(p))\ \forall o \in \{1,2,3,4\}$, so that $A^+ = \{min(V_{io})|o \in \{1,2,3,4\}\}$ and $A^- = \{max(v_{io})|o \in \{1,2,3,4\}\}$.

- Step 4. Compute the Euclidean distance S_{i+} from each point V_i to the positive ideal solution A^+, and the Euclidean distance S_{i-} from each point V_i to the negative ideal solution A^-.

- Step 5. Calculate the score $C_i = \frac{S_{i-}}{S_{i-}+S_{i+}}$ and choose the solution with the largest C_i as the final recommended solution.

As can be seen, solutions close to the positive ideal and far from the negative ideal will be chosen.

There are several other approaches to choosing a final solution that might be used here. Both Linear Programming Technique for Multidimensional Analysis of Preference (LINMAP, Thakkar, 2021) and Viekriterijumsko Kompromisno Rangiranje (VIKOR, Opricovic & Tzeng, 2004) also use functions depending on the distance between the Pareto front and the ideal solutions. Other methods such as Simple Additive Weighting (SAW, Zionts & Wallenius, 1983) and Multiplicative Exponent Weighting (Wang & Rangaiah, 2017) use the sum of weighted objective values or the product of weighted objective values as the score to choose an optimal solution from Pareto-optimal front. Wang & Rangaiah (2017) introduces and compares several other methods as well. They suggest that TOPSIS is the most commonly used method for choosing the final single optimal solution of a multi-objective optimization problem. We chose it for several reasons. First, because it considers the distance between the Pareto front solutions and both the positive and negative ideal solutions. Second, TOPSIS requires relatively few user inputs. Finally, TOPSIS is computationally and conceptually straightforward to implement and understand.

4.3. Scenario 2 Description and Methodology

In contrast to Scenario 1, Scenario 2 requires a generalization of our methods in order to estimate and optimize the location capacity—a quantity we will denote $\hat{C}$—instead of assuming a fixed capacity, C_0, all while optimizing the planting schedule as well. While the predictive modeling approach for daily cumulative GDUs will stay the same (see Section 4.1), we generalize the optimization by treating both the location capacity and planting schedule as decision variables, while adding the location capacity as an objective

function as well. That is, for Scenario 2 we have decision variables p_i, $i = 1, 2, \ldots, n_0$, representing the planting day for seed population s_i, as well as $\hat{C}$, a decision variable representing the location capacity. As in Section 3.2, the problem requires $L(s_i) \leq p_i \leq U(s_i)$, but we introduce a new set of constraints having to do with the capacity. Specifically, we require $L(C) \leq \hat{C} \leq U(C)$, where $L(C)$ and $U(C)$ are a lower bound and upper bound that is user-specified based upon the particular setting. In addition to adding $\hat{C}$ as a decision variable, we also include it as its own objective function, in order that we might select a capacity as small as possible, while simultaneously balancing the number of harvest weeks, the deviation from capacity, and the amount of wasted corn. More specifically, for Models 1, 2, and 3 in Section 3.2, we augment the existing objective functions with a new one: $\mathbf{f}_{k5}(\mathbf{p}) = \hat{C}$, for $k = 1, 2, 3$. Hence, we are trying to simultaneously minimize $\mathbf{f}_k(\mathbf{p}) = (\mathbf{f}_{k1}(\mathbf{p}), \mathbf{f}_{k2}(\mathbf{p}), \mathbf{f}_{k3}(\mathbf{p}), \mathbf{f}_{k4}(\mathbf{p}), \mathbf{f}_{k5}(\mathbf{p}))$, $k = 1, 2, 3$, with decision variables $\mathbf{p}$ and $\hat{C}$.

Beyond these generalizations of the decision variables and objective functions, our methodology remains largely unchanged. The Pareto dominance description (Section 4.2.1) is the same, except it now accounts for the extra decision variable and objective function. The hypervolume computation simply incorporates the new objective and extends the reference point to $[2, 2, 2, 2, 2]$. As before, though we have defined $\mathbf{f}_2(\mathbf{p})$ (Model 2) and $\mathbf{f}_3(\mathbf{p})$ (Model 3), they only serve to aid in the construction of potentially improved Pareto fronts with respect to $\mathbf{f}_1(\mathbf{p})$ (Model 1). Our strategy to explore the tuning parameter space for Scenario 2 is also the same as for Scenario 1. Based on this structure, we use the same six-step solution strategy as outlined in 4.2.1, except that we account for the fifth objective and the additional decision variable.

We note that due to the curse of dimensionality, adding a fifth criterion makes the computation much more difficult. We also note that for the selection of the final single solution $\mathbf{p}^*$ from the best Pareto front PF_{k*}, using TOPSIS, instead of using equal weights for all five objective functions, we use weights $\{w_1 = 0.225, w_2 = 0.225, w_3 = 0.225, w_4 = 0.225, w_5 = 0.1\}$. The slightly lower weight for the new objective reflects our belief that the other objectives should be prioritized. Once $\mathbf{p}^*$ is chosen, the corresponding $\mathbf{f}_{15}(\mathbf{p}^*) = \hat{C}$ is our recommended location capacity.

5. Results

In this section, we provide solutions for Scenario 1, Site 0 and Scenario 1, Site 1 using the methodology provided in Section 4.2, as well as results for Scenario 2, Site 0, using the methodology presented in Section 4.3. We could use the same methodology to provide results for Scenario 2, Site 1, but we omit them due to the large Pareto fronts and associated computational cost.

5.1. Results for Scenario 1, Site 0

In Scenario 1, a storage capacity is provided, and for Site 0 it is 7,000 ears per week. For this initial set of results, we will provide details, cross-referenced with the steps outlined in Section 4.2.1. As in Step 3, PF_1 has 4,144 solutions in the constructed Pareto front, while PF_2 has 821 and PF_3 has 1,152. Combining them

together results in a meta-population, P_{PF}, which yields a 6117×4 matrix. In order to compute the hypervolume (Step 4), we found that $F_1^{min} = (6, 1027, 49, 12788)$ and $F_1^{max} = (1316, 8957, 54, 48764)$, which are used to scale each solution in PF_k, $k = 1, 2, 3$. As in Step 5, the hypervolume v_k is computed for each of the scaled Pareto fronts, PF_k, such that $v_1 = 11.5$, $v_2 = 15.6$, and $v_3 = 13.4$. This suggests that when evaluated on the Model 1 objectives, the Model 2 Pareto front produces the best set of solutions, as measured by the hypervolume. Thus, we will choose our final solution from PF_2, using TOPSIS (Step 6). That is, after the scaling and the weighting, we compute PF_2's positive ideal solution as $A^+ = (0.0002, 0.0023, 0.0085, 0.0053)$ and its negative ideal solution $A^- = (0.0259, 0.0182, 0.0090, 0.0142)$. Finally, then, we choose a final solution (Table 5; Figure 5). It is clear that our solution is improved; indeed it Pareto-dominates not only on the initial solution provided in the Contest, but also the solution we submitted. Notice that our preferred solution includes three fewer harvest weeks, is clearly less variable around the capacity, and results in less waste.

5.2. Results for Scenario 1 (Site 1)

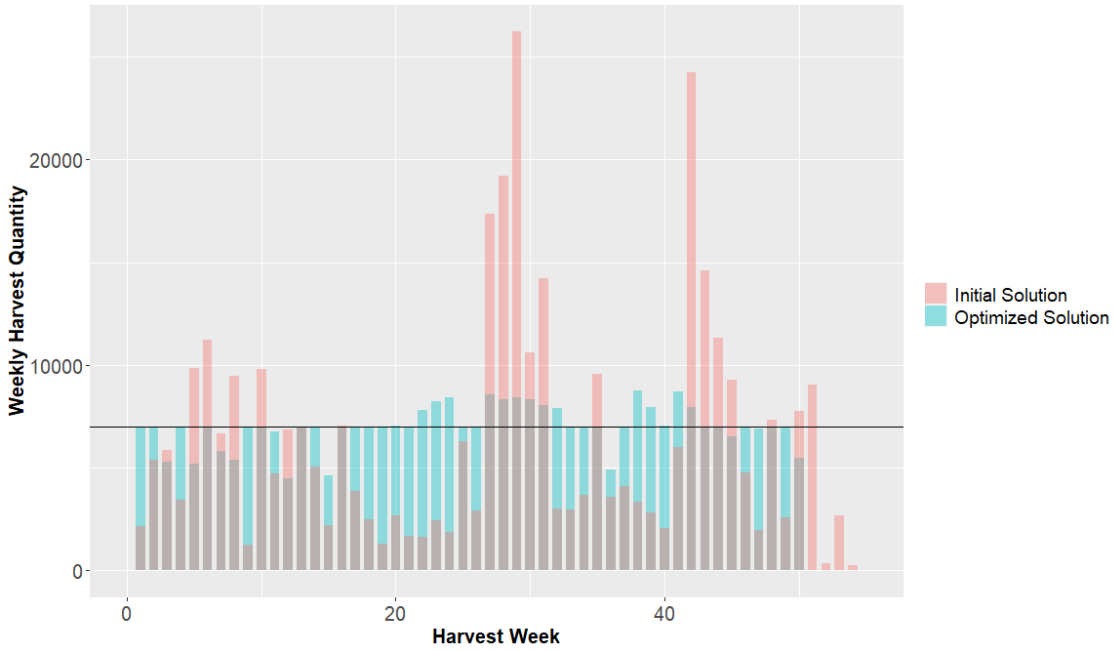
Here, we omit most details, instead focusing on the results themselves. For Site 1, Scenario 1, there is a designated storage capacity of 6,000 ears per week. Using the methodology outlined in Section 4, we find again that the Pareto front constructed using Model 2 yields the best hypervolume, leading to a final solution given in Table 6 and Figure 6.

Table 5: Criterion values for solutions $\mathbf{p}^*$ (final single optimal solution), $\mathbf{p}_{challenge}$ (solution submitted to the Challenge), $\mathbf{p}_{initial}$ (initial solution given by the Challenge), under Scenario 1, Site 0

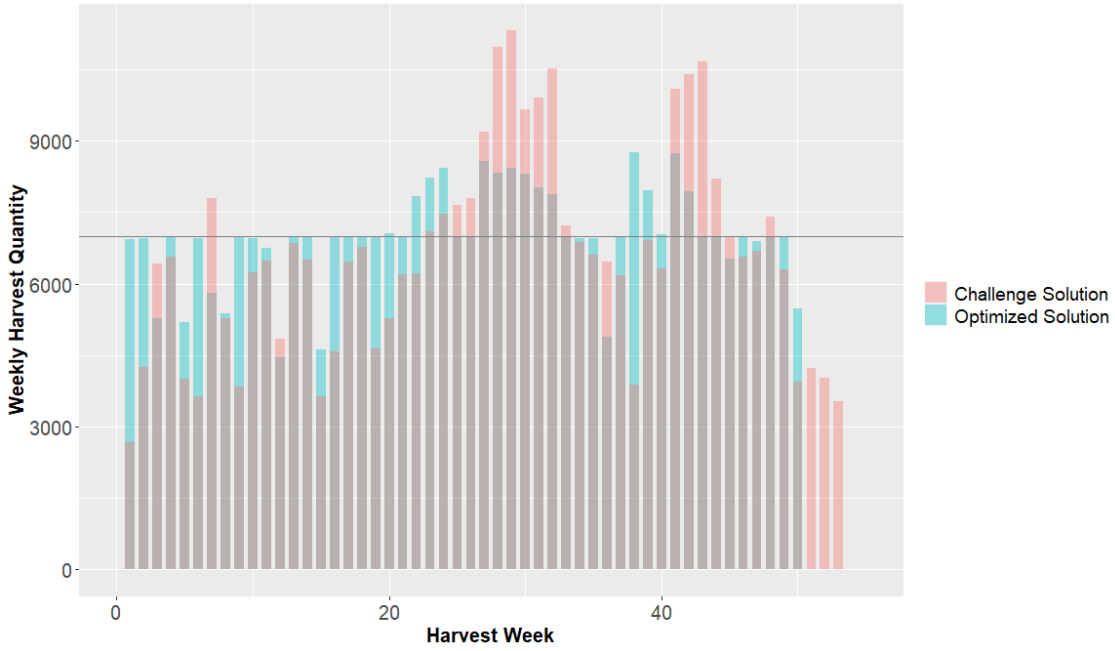
	$\mathbf{p}^*$	$\mathbf{p}_{challenge}$	$\mathbf{p}_{initial}$
Median Absolute Difference ($\mathbf{f}_{11}$)	53	823	4,012
Max Absolute Difference ($\mathbf{f}_{12}$)	2,537	4,320	19,222
# of Non-zero Harvest Week ($\mathbf{f}_{13}$)	50	53	54
Total Amount of Wasted Product ($\mathbf{f}_{14}$)	16,514	34,320	102,204

Table 6: Criterion values for solutions $\mathbf{p}^*$ (final single optimal solution), $\mathbf{p}_{challenge}$ (solution submitted to the Challenge), $\mathbf{p}_{initial}$ (initial solution given by the Challenge), under Scenario 1, site 1

	$\mathbf{p}^*$	$\mathbf{p}_{challenge}$	$\mathbf{p}_{initial}$
Median Absolute Difference ($\mathbf{f}_{11}$)	8	365	2,196
Max Absolute Difference ($\mathbf{f}_{12}$)	827	1,816	6,725
# of Non-zero Harvest Week ($\mathbf{f}_{13}$)	51	53	52
Total Amount of Wasted Product ($\mathbf{f}_{14}$)	4,809	12,290	59,549

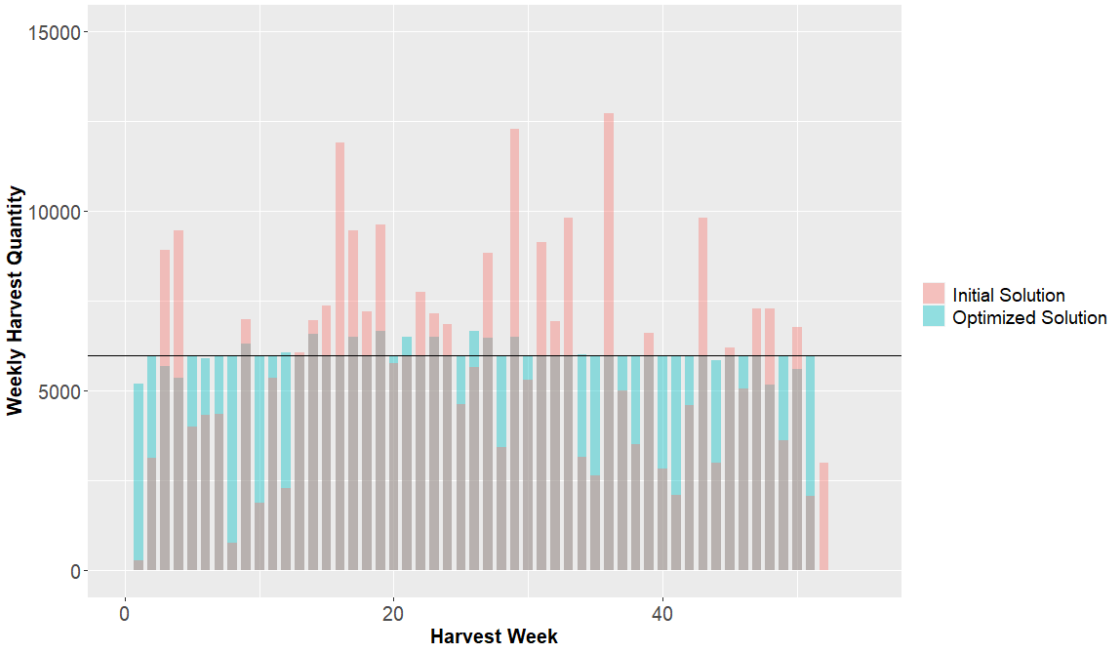


(a) Comparison between optimized and original planting schedule, where the original was provided as an initial solution for the Challenge.

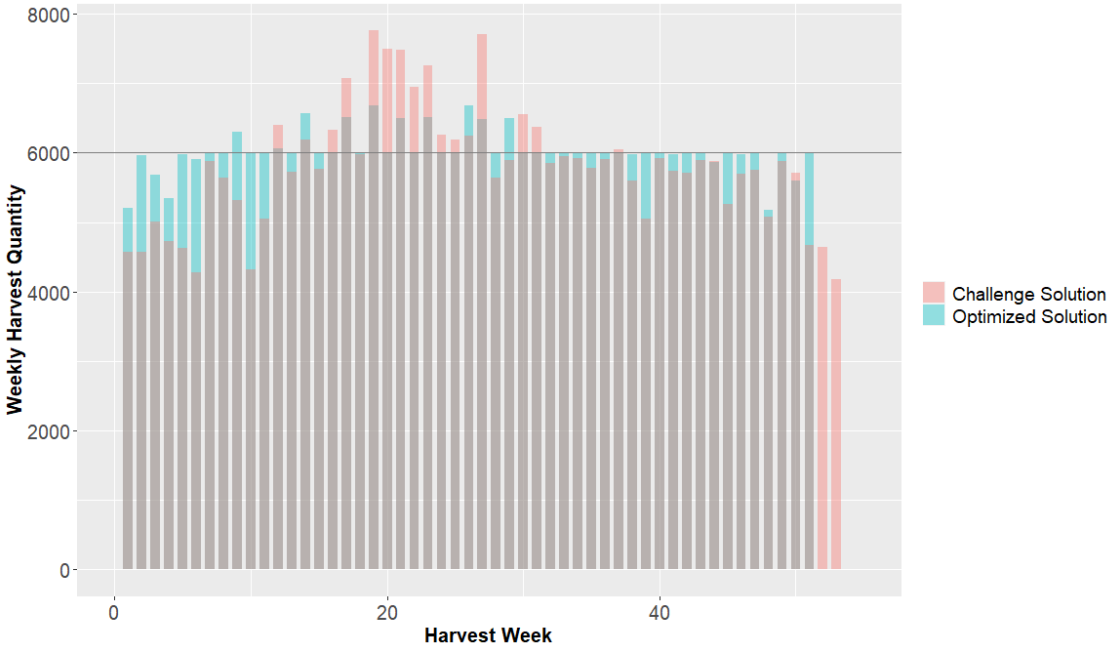


(b) Comparison between the solution submitted to the Challenge and the solution optimized using the methods described in this paper.

Figure 5: Results for Scenario 1, Site 0.



(a) Comparison between optimized and original planting schedule, where the original was provided as an initial solution for the Challenge.



(b) Comparison between the solution submitted to the Challenge and the solution optimized using the methods described in this paper.

Figure 6: Scenario 1, Site 1

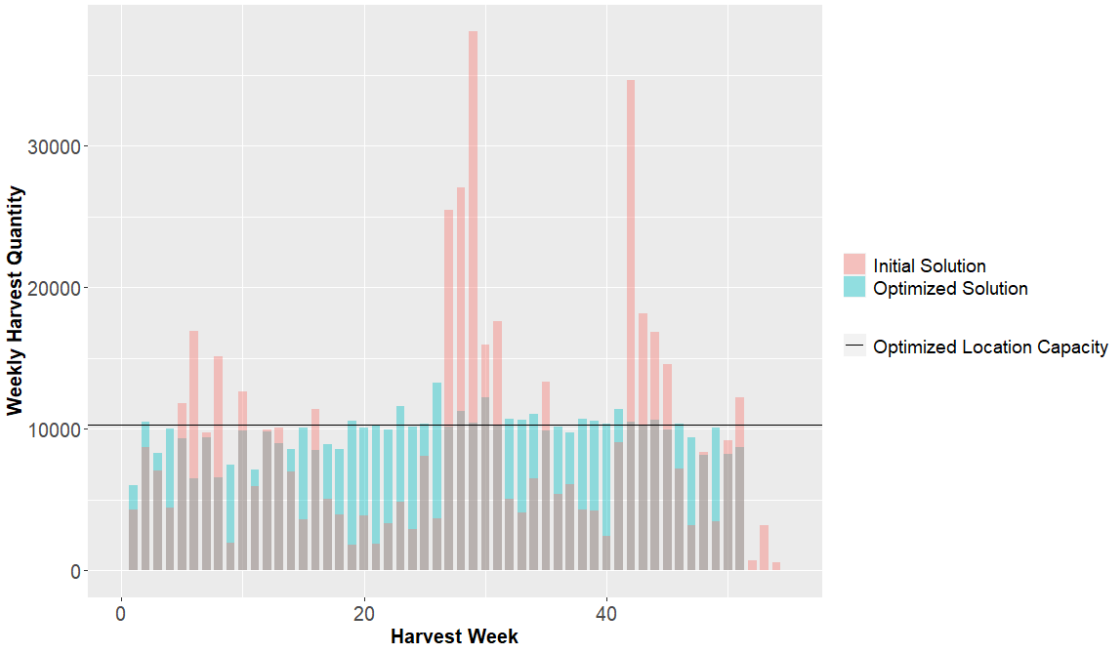
5.3. Results for Scenario 2 (Site 0)

As described in Section 4.3, Scenario 2 is like Scenario 1 except a capacity is not specified. Instead, as part of the solution, we need to provide a recommended capacity. Here we provide results for site 0; results for site 1 could be obtained similarly, but are omitted because the hypervolume computation cost for the resulting large Pareto front is high.

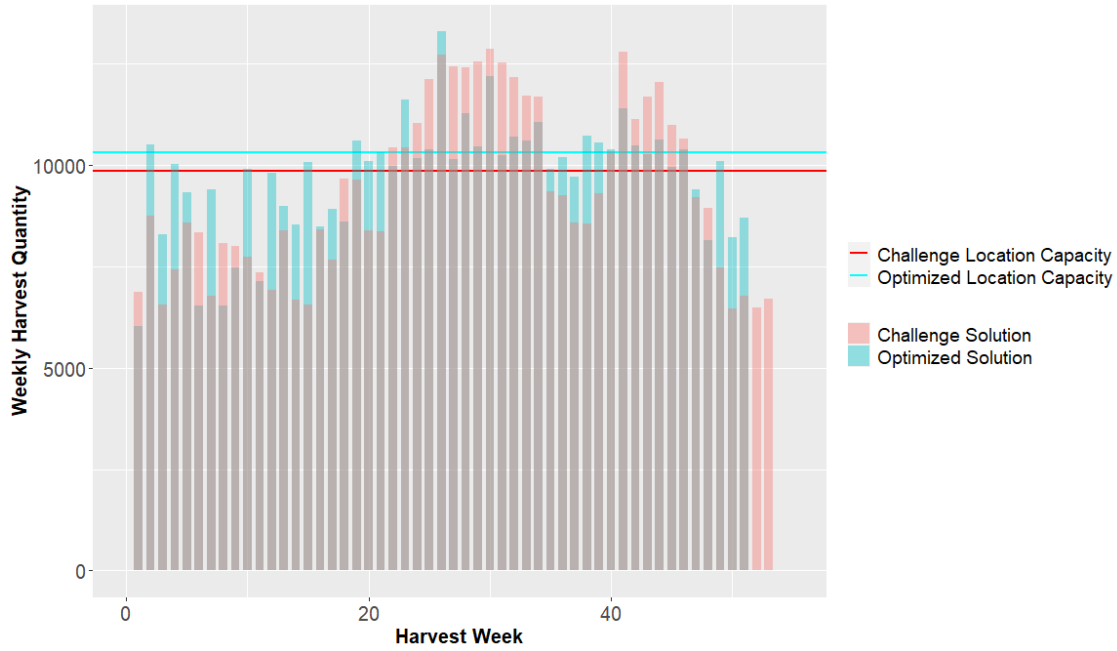
Based on Figure 1we observe that, on average, the population harvest quantity under Scenario 2 is larger than Scenario 1 which suggests that we need a relatively larger location capacity under Scenario 2. Thus, we specify $L(C) = 8{,}000$ and $U(C) = 20{,}000$, where the latter is chosen to be larger than we anticipate being necessary. Using the methodology described in Section 4.3, Table 7 shows our chosen $\mathbf{p}^*$. Since the initial Syngenta solution for Scenario 2 did not have a corresponding estimated location capacity, here we only compare the single optimal solution $\mathbf{p}^*$ with the solution we originally submitted to the Syngenta challenge. Compared to $\mathbf{p}_{challenge}$, we suggest a capacity of about 450 more ears of corn. This slight concession results in a much smaller median absolute difference and amount of wasted product, though our proposed solution has a larger maximum absolute difference. Our solution $\mathbf{p}^*$ also uses two fewer harvest weeks. The solutions can be compared in Figure 7.

Table 7: Criterion values for solutions $\mathbf{p}^*$ (final single optimal solution) and $\mathbf{p}_{challenge}$ (solution submitted to the Challenge), under Scenario 2, site 0

	$\mathbf{p}^*$	$\mathbf{p}_{challenge}$
Median Absolute Difference ($\mathbf{f}_{11}$)	409.599	1,836
Max Absolute Difference ($\mathbf{f}_{12}$)	4,286.599	3,402
# of Non-zero Harvest Week ($\mathbf{f}_{13}$)	51	53
Total Amount of Wasted Product ($\mathbf{f}_{14}$)	11,639.210	37,421
Estimated Location Capacity ($\mathbf{f}_{15}$)	10,312.6	9,864



(a) Comparison between optimized and original planting schedule, where the original was provided as an initial solution for the Challenge.



(b) Comparison between the solution submitted to the Challenge and the solution optimized using the methods described in this paper.

Figure 7: Scenario 2, Site 0

6. Discussion and Conclusion

In this paper, we have provided a solution to a challenging corn-planting problem presented in the 2021 Syngenta Crop Challenge. In particular, we have improved upon the solutions submitted to the Challenge, based upon the three criteria provided along with an additional "waste" criterion. This problem requires multiobjective optimization and predictive modeling, and we accomplished the improvements by considering two related optimization problems that exploit unique aspects of the problem in order to enhance solutions with respect to the original four criteria. We compare results across these three models using the hypervolume indicator, and choose a final solution using the TOPSIS method from the literature. In the end, we obtain solutions that clearly improve upon those our team submitted to the Challenge. (Note that the Challenge entry was submitted by the first three authors.) Compared to the Challenge entry, we extensively explored the tuning parameter space for the Genetic Algorithm, added an additional objective function to the base model ($\mathbf{f}_1$), modified the objective functions in Model 2 ($\mathbf{f}_2$), and added a new model ($\mathbf{f}_3$). In our original entry, we also chose a final solution using a weighted sum of normalized objectives, rather than the hypervolume measure and TOPSIS that we use here. Finally, for Scenario 2, in the method we submitted to the Challenge we treated the estimated location capacity as a decision variable but not as an objective function.

As far as we know, this problem has only been considered by two other Challenge teams (Sajid & Hu, 2022; Khalilzadeh & Wang, 2022). Sajid & Hu (2022) used a Convolutional Neural Network to predict GDUs, and developed a Mixed Integer Linear Programming-based algorithm to determine a final solution. Like us, Khalilzadeh & Wang (2022) uses LSTM for predictive modeling, whereas instead of optimizing the objectives individually, they focus on a single objective which minimizes the sum of the absolute difference between weekly harvest quantity and location capacity, reducing the problem to a single objective optimization problem with other objectives ignored. Table 8 shows the comparison of criteria values between our final optimized solution $\mathbf{f}_1(\mathbf{p}^*)$ and the optimized solutions made by Sajid & Hu (2022); Khalilzadeh & Wang (2022) under Scenario 1 as well as Scenario 2, Site 0. No group's solutions uniformly dominate any other, but for Scenario 1 especially our solutions are strong. We do uniformly outperform the other teams in the Number of Harvest Weeks criterion ($\mathbf{f}_{13}$).

There are several final points to make regarding this work. First, it represents a full solution to a challenging, real problem, which includes predictive modeling and multiobjective optimization. Note that we handled an inherently multiobjective optimization problem with multiobjective optimization tools, instead of combining into one objective or solving the various objectives separately. Instead, we used a genetic algorithm that explicitly handles the trade-off between objectives while considering them simultaneously. Secondly, we demonstrate the complexity inherent in employing multiobjective optimization in such a problem. Not only did we undertake substantial exploration of the tuning parameter space, we also proposed a procedure by which Pareto fronts were compared, and a final solution chosen. We also demonstrate that in this case, better solutions to the optimization problem of interest (Model 1) are obtained by estimating the solution to a related but distinct problem (Model 2).

Table 8: Comparison of final solutions between our proposed solution ($\mathbf{f}_1(\mathbf{p}^*)$) and the published results of two other teams.

Scenario and Site	Solution Source	$\mathbf{f}_{11}(\mathbf{p}^*)$	$\mathbf{f}_{12}(\mathbf{p}^*)$	$\mathbf{f}_{13}(\mathbf{p}^*)$	Estimated Capacity
Scenario 1, Site 0	$\mathbf{f}_1(\mathbf{p}^*)$	53	2,537	50	-
	Khalilzadeh & Wang (2022)	57	2,471	51	-
	Sajid & Hu (2022)	16.5	2,883	51	-
Scenario 1, Site 1	$\mathbf{f}_1(\mathbf{p}^*)$	8	827	51	-
	Khalilzadeh & Wang (2022)	25	2,633	52	-
	Sajid & Hu (2022)	29.5	394	52	-
Scenario 2, Site 0	$\mathbf{f}_1(\mathbf{p}^*)$	409.6	4,286.6	51	10,312.6
	Khalilzadeh & Wang (2022)	NA	NA	NA	10,795
	Sajid & Hu (2022)	49.5	2,057	52	9,800

Acknowledgements

The authors are grateful to Syngenta for providing the opportunity to work on this problem, as well as to the committee who judged the initial submissions to the Challenge.

Conflicts of Interest

The authors have no funding support or conflicts of interest to acknowledge.

References

Alander, J. (1992). On optimal population size of genetic algorithms. In *CompEuro 1992 Proceedings Computer Systems and Software Engineering* (pp. 65–70). doi:10.1109/CMPEUR.1992.218485.

Altché, F., & de La Fortelle, A. (2018). An lstm network for highway trajectory prediction, .

Ansarifar, J., Akhavizadegan, F., & Wang, L. (2020). Performance prediction of crosses in plant breeding through genotype by environment interactions. *Scientific Reports*, *10*, 11533. URL: https://doi.org/10.1038/s41598-020-68343-1. doi:10.1038/s41598-020-68343-1.

Audsley, E., & Boyce, D. (1974). A method of minimizing the costs of combine-harvesting and high temperature grain drying. *Journal of Agricultural Engineering Research*, *19*, 173–188. URL: https://www.sciencedirect.com/science/article/pii/0021863474900316. doi:https://doi.org/10.1016/0021-8634(74)90031-6.

Bansal, S., Gutierrez, G. J., & Keiser, J. R. (2017). Using experts' noisy quantile judgments to quantify risks: Theory and application to agribusiness. *Operations Research*,

65, 1115–1130. URL: https://doi.org/10.1287/opre.2017.1627. doi:10.1287/opre.2017.1627. arXiv:https://doi.org/10.1287/opre.2017.1627.

Box, G. (2013). Box and jenkins: Time series analysis, forecasting and control. In *A Very British Affair: Six Britons and the Development of Time Series Analysis During the 20th Century* (pp. 161–215). London: Palgrave Macmillan UK. URL: https://doi.org/10.1057/9781137291264_6. doi:10.1057/9781137291264_6.

Boyabatli, O., & Sabuncuoglu, I. (2004). Parameter selection in genetic algorithms. *J of Systemics, Cybernetics and Informatics*, *2*.

Byrum, J., Davis, C., Doonan, G., Doubler, T., Foster, D., Luzzi, B., Mowers, R., Zinselmeier, C., Kloeber, J., Culhane, D., & Mack, S. (2016). Advanced analytics for agricultural product development. *INFORMS Journal on Applied Analytics*, *46*, 5–17. URL: https://doi.org/10.1287/inte.2015.0823. doi:10.1287/inte.2015.0823. arXiv:https://doi.org/10.1287/inte.2015.0823.

Cao, Y., Smucker, B. J., & Robinson, T. J. (2015). On using the hypervolume indicator to compare pareto fronts: Applications to multi-criteria optimal experimental design. *Journal of Statistical Planning and Inference*, *160*, 60–74.

Corrie, W., & Boyce, D. (1972). A dynamic programming method to optimize policies for the multistage harvest of crops with an extended maturity period. *Journal of Agricultural Engineering Research*, *17*, 348–354. URL: https://www.sciencedirect.com/science/article/pii/S0021863472800428. doi:https://doi.org/10.1016/S0021-8634(72)80042-8.

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE transactions on evolutionary computation*, *6*, 182–197.

Donaldson, G. (1968). Allowing for wealth risk in assessing harvest machinery capacity. *American Journal of Agricultural Economics*, *50*, 24. URL: https://proxy.lib.miamioh.edu/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=fsr&AN=4626252&site=eds-live&scope=site.

Feng, Y., & Zhou, W. (2021). Seed stocking via multi-task learning. *CoRR*, *abs/2101.04333*. URL: https://arxiv.org/abs/2101.04333. arXiv:2101.04333.

Fokkens, B., & Puylaert, M. (1981). A linear programming model for daily harvesting operations at the large-scale grain farm of the ijsselmeerpolders development authority. *Journal of the Operational Research Society*, *32*, 535–547. URL: https://doi.org/10.1057/jors.1981.113. doi:10.1057/jors.1981.113. arXiv:https://doi.org/10.1057/jors.1981.113.

Gers, F., Schmidhuber, J., & Cummins, F. (1999). Learning to forget: continual prediction with lstm. In *1999 Ninth International Conference on Artificial Neural Networks ICANN 99. (Conf. Publ. No. 470)* (pp. 850–855 vol.2). volume 2. doi:10.1049/cp:19991218.

Glen, J. J. (1987). Feature article—mathematical models in farm planning: A survey. *Operations Research*, *35*, 641–666. URL: https://doi.org/10.1287/opre.35.5.641. doi:10.1287/opre.35.5.641. arXiv:https://doi.org/10.1287/opre.35.5.641.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press. http://www.deeplearningbook.org.

Heady, E. O. (1954). Simplified presentation and logical aspects of linear programming technique. *Journal of Farm Economics*, *36*, 1035–1048.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*, 1735–80. doi:10.1162/neco.1997.9.8.1735.

James, G., Hastie, T. J., Tibshirani, R., Witten, D., & James, G. (2021). *An introduction to statistical learning: With applications in R*. Springer.

Jones, P. C., Lowe, T. J., Traub, R. D., & Kegler, G. (2001). Matching supply and demand: The value of a second chance in producing hybrid seed corn. *Manufacturing & Service Operations Management*, *3*, 122–137. URL: https://doi.org/10.1287/msom.3.2.122.9992. doi:10.1287/msom.3.2.122.9992. arXiv:https://doi.org/10.1287/msom.3.2.122.9992.

Katoch, S., Chauhan, S. S., & Kumar, V. (2021). A review on genetic algorithm: past, present, and future. *Multimedia Tools and Applications*, *80*, 8091–8126.

Khalilzadeh, Z., & Wang, L. (2022). Corn planting and harvest scheduling under storage capacity and growing degree units uncertainty. *Scientific Reports*, *12*, 22482. doi:10.1038/s41598-022-25797-9.

Kumar, M., Husain, D., Upreti, N., Gupta, D. et al. (2010). Genetic algorithm: Review and application. *Available at SSRN 3529843*, .

Lowe, T. J., & Preckel, P. V. (2004). Decision technologies for agribusiness problems: A brief review of selected literature and a call for research. *Manufacturing & Service Operations Management*, *6*, 201–208. URL: https://doi.org/10.1287/msom.1040.0051. doi:10.1287/msom.1040.0051. arXiv:https://doi.org/10.1287/msom.1040.0051.

Marko, O., Brdar, S., Panic, M., Lugonja, P., & Crnojevic, V. (2016). Soybean varieties portfolio optimisation based on yield prediction. *Computers and Electronics in Agriculture*, *127*, 467–474.

Marko, O., Brdar, S., Panić, M., Šašić, I., Despotović, D., Knežević, M., & Crnojević, V. (2017). Portfolio optimization for seed selection in diverse weather scenarios. *PLOS ONE*, *12*, 1–27. URL: https://doi.org/10.1371/journal.pone.0184198. doi:10.1371/journal.pone.0184198.

Medsker, L., & Jain, L. C. (1999). *Recurrent neural networks: design and applications*. CRC press.

Mirjalili, S. (2019). Genetic algorithm. In *Evolutionary algorithms and neural networks* (pp. 43–55). Springer.

Miyake, Y., Wells, L. G., Duncan, G. A., & Rankin, J. (1979). Determination of strategy for harvesting burley tobacco. *Transactions of the ASAE*, *22*, 0251–0259. doi:10.13031/2013.35000.

Morey, R., Peart, R., & Zachariah, G. (1972). Optimal harvest policies for corn and soybeans. *Journal of Agricultural Engineering Research*, *17*, 139–148. URL: https://www.sciencedirect.com/science/article/pii/S0021863472800015. doi:https://doi.org/10.1016/S0021-8634(72)80001-5.

Myers, R. H., Montgomery, D. C., & Anderson-Cook, C. M. (2016). *Response surface methodology: process and product optimization using designed experiments*. John Wiley & Sons.

Najafi, A. A., Niaki, S. T. A., & Shahsavar, M. (2009). A parameter-tuned genetic algorithm for the resource investment problem with discounted cash flows and generalized precedence relations. *Computers Operations Research*, *36*, 2994–3001. URL: https://www.sciencedirect.com/science/article/pii/S030505480900029X. doi:https://doi.org/10.1016/j.cor.2009.01.016.

Opricovic, S., & Tzeng, G.-H. (2004). Compromise solution by mcdm methods: A comparative analysis of vikor and topsis. *European Journal of Operational Research*, *156*, 445–455. URL: https://www.sciencedirect.com/science/article/pii/S0377221703000201. doi:https://doi.org/10.1016/S0377-2217(03)00020-1.

Sajid, S. S., & Hu, G. (2022). Optimizing crop planting schedule considering planting window and storage capacity. *Frontiers in Plant Science*, *13*.

Siami-Namini, S., Tavakoli, N., & Siami Namin, A. (2018). A comparison of arima and lstm in forecasting time series. In *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)* (pp. 1394–1401). doi:10.1109/ICMLA.2018.00227.

Sundaramoorthi, D., & Dong, L. (2022). Machine learning and optimization based decision-support tool for seed variety selection. *Annals of Operations Research*, (pp. 1–35). URL: https://doi.org/10.1007/s10479-022-04995-8.

Thakkar, J. (2021). Linear programming techniques for multidimensional analysis of preference (linmap). (pp. 199–218). doi:10.1007/978-981-33-4745-8_12.

Tsou, C.-S. (2022). *nsga2R: Elitist Non-Dominated Sorting Genetic Algorithm*. URL: https://CRAN.R-project.org/package=nsga2R r package version 1.1.

Wang, Z., & Rangaiah, G. P. (2017). Application and analysis of methods for selecting an optimal solution from the pareto-optimal front obtained by multiobjective optimization. *Industrial & Engineering Chemistry Research*, *56*, 560–574. URL: https://doi.org/10.1021/acs.iecr.6b03453. doi:10.1021/acs.iecr.6b03453. arXiv:https://doi.org/10.1021/acs.iecr.6b03453.

Whitley, D. (1994). A genetic algorithm tutorial. *Statistics and computing*, *4*, 65–85.

Zhong, H., Li, X., Lobell, D. B., Ermon, S., & Brandeau, M. L. (2017). Hierarchical modeling of seed variety yields and decision making for future planting plans. *CoRR*, *abs/1711.05809*. URL: http://arxiv.org/abs/1711.05809. arXiv:1711.05809.

Zionts, S., & Wallenius, J. (1983). An interactive multiple objective linear programming method for a class of underlying nonlinear utility functions. *Management Science*, *29*, 519–529. URL: https://doi.org/10.1287/mnsc.29.5.519. doi:10.1287/mnsc.29.5.519. arXiv:https://doi.org/10.1287/mnsc.29.5.519.

Zitzler, E., & Thiele, L. (1998). Multiobjective optimization using evolutionary algorithms—a comparative case study. In *International conference on parallel problem solving from nature* (pp. 292–301). Springer.