

# MASTER THESIS

## Multiobjective Optimization of Irrigation Scheduling Based on MIKE SHE

Maximilian Winderl

2020





Chair of Geoinformatics  
Department of Civil, Geo and Environmental Engineering  
Technical University of Munich

# MULTIOBJECTIVE OPTIMIZATION OF IRRIGATION SCHEDULING BASED ON MIKE SHE

Maximilian Winderl

For the degree of  
**Master of Science (M.Sc.)**

Supervisor	Univ.-Prof. Dr. rer. nat. Thomas H. Kolbe Bruno Willenborg
Advisor	Alexander Renz (DHI WASY GmbH)
Submitted on	16 <sup>th</sup> December 2020

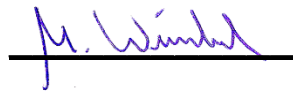


# Declaration of Authorship

I hereby declare that the thesis submitted is my own unaided work. All direct or indirect sources used are acknowledged as references.

Munich, 16<sup>th</sup> of December 2020

Signature:

A handwritten signature in blue ink is written over a solid black horizontal line. The signature is cursive and appears to read 'M. Winkler'.

# Acknowledgement

Foremost, I would like to express my sincere gratitude to my advisor Alexander Renz, who continuously supported me throughout my research and struggles with his immense knowledge, interest, and enthusiasm.

A special thanks also goes to my supervisor Bruno Willenborg, who made the collaboration between the Chair of Geoinformatics and the DHI WASY GmbH possible in the first place. His guidance and professional advice on how to approach problems have sustainably improved my skills in scientific working.

I also want to thank Patrick Keilholz for supporting me by sharing his extensive knowledge of the MIKE SHE software, and David Gackstetter for providing advice on the general agriculture-related questions I had.

My gratitude also goes to my fellow students, who continuously enriched my life throughout the entire Master studies, and I want to wish you all the best for the future.

Last, I would like to thank my family for supporting me during all the years of studying and for believing in me from the very beginning.

# Abbreviations

DFO	Derivative-Free Optimization
DFS	Data File System
DM	Decision Maker
DSM	Direct Search Methods
$ET_0$	Reference Evapotranspiration [ $LT^{-1}$ ]
$ET_a$	Actual Evapotranspiration [ $LT^{-1}$ ]
$ET_c$	Crop Evapotranspiration [ $LT^{-1}$ ]
FAO	Food and Agriculture Organization of the United Nations
FC	Field Capacity
GA	Genetic Algorithm
$K_c$	Crop Coefficient [-]
LAI	Leaf Area Index
MOEA	Multiobjective Evolutionary Algorithm
MOO	Multiobjective Optimization
MOOP	Multiobjective Optimization Problem
MPC	Model-Predictive-Control
NSGA-II	Nondominated Sorting Genetic Algorithm II
PM	Polynomial Mutation
POF	Pareto Optimal Front
PWP	Permanent Wilting Point
RAW	Readily Available Water
SBX	Simulated Binary Crossover
SO	Simulation Optimization
SOEA	Single Objective Evolutionary Algorithm
TAW	Total Available Water

# Table of Content

1	Introduction.....	1
2	Literature Review.....	2
3	Theoretical Background.....	7
3.1	Physical Background.....	7
3.1.1	Evapotranspiration.....	7
3.1.2	Modified Penman-Monteith Method.....	9
3.1.3	Soil-Water Retention Curve - Van Genuchten Equation.....	10
3.1.4	Deficit Irrigation.....	11
3.1.5	Phenology and Soil Moisture Conditions.....	11
3.1.6	Characteristics of Maize.....	16
3.2	MIKE SHE Software.....	18
3.2.1	Unsaturated Flow.....	18
3.2.2	Evapotranspiration.....	20
3.2.3	Irrigation Module.....	26
3.3	Optimization.....	28
3.3.1	Constraints.....	29
3.4	Multiobjective Optimization.....	30
3.4.1	Multiobjective Derivative-Free Algorithm Classes.....	33
3.4.2	Multiobjective Evolutionary Algorithms.....	34
3.5	Python Libraries.....	46
4	Implementation.....	47
4.1	MIKE SHE Model.....	49
4.2	Optimization.....	52
4.2.1	Analysis of the Optimization Problem.....	52
4.2.2	Objective Functions.....	56
4.2.3	Single-Field Optimization.....	58
4.2.4	Multi-Field Optimization.....	60
4.3	Python Code.....	63
5	Results.....	64
5.1	Single-Field Optimization.....	64



5.1.1	Optimization of Single Irrigation Event .....	64
5.1.2	Seasonal Optimization 2018 .....	68
5.1.3	Seasonal Optimization 2017 .....	70
5.1.4	Optimization with Pre-seasonal Watering of the Field.....	72
5.2	Comparison with Integrated MIKE SHE Optimization.....	74
5.3	Multi-Field Optimization .....	76
6	Discussion .....	77
6.1	System Model – MIKE SHE Simulation Model .....	77
6.2	Optimization Framework.....	79
6.2.1	Multi-Field Optimization .....	81
6.3	Computational Time Optimization.....	82
7	Conclusion and Outlook .....	82

# 1 Introduction

---

Water scarcity is a rising issue worldwide. Agriculture is responsible for 69% of annual water withdrawal worldwide, which makes it the largest water consumer (FAO, 2016). Global population growth and climate change are putting increasing stress on agricultural systems and water resources. Especially in large agricultural areas, droughts have had significant impacts. In the states of Texas and California, where agriculture is a crucial industry, climate change has led to severe droughts in recent years (S. B. Roy et al., 2012). The same can be observed in Spain where farmers are suffering from continuous dry periods (Peña-Gallardo et al., 2019). Likewise, in Germany drier summers are forcing farmers to install irrigation systems (Bundesinformationszentrum Landwirtschaft (BZL) & Bundesanstalt für Landwirtschaft und Ernährung, 2020). These are just a few examples of global agricultural systems under stress. Adaptation strategies are required for these regions and proactive measures must be taken. Varela-Ortega (2016), for example, outlines how irrigated agriculture can adapt to climate change in the Guadiana Basin in Spain.

Parallel to adaptation, technological progress in agriculture must also be driven forward to satisfy food demand without causing further water stress. Optimal water allocation is an urgent matter, and extensive research has been conducted in the field of optimization of irrigation systems, which is presented in the literature review. The goal of this work was to establish a novel optimization framework for irrigation scheduling based on a physics-based numerical model. The framework is built around two objectives: minimize water consumption and optimize soil moisture conditions. The aim was to successfully reduce water stress on agricultural systems by allocating water most efficiently. This was achieved by implementing a multiobjective evolutionary algorithm (MOEA), more precisely the modified Nondominated Sorting Genetic Algorithm (NSGA-II). The MIKE SHE software was used as a physics-based water-balance model to predict soil moisture content in the root zone. Single-irrigation events are optimized in alignment with the weather forecast while at the same time embedding seasonal specific factors. The approach was then extended to multiple fields where sequential irrigation events are optimized.

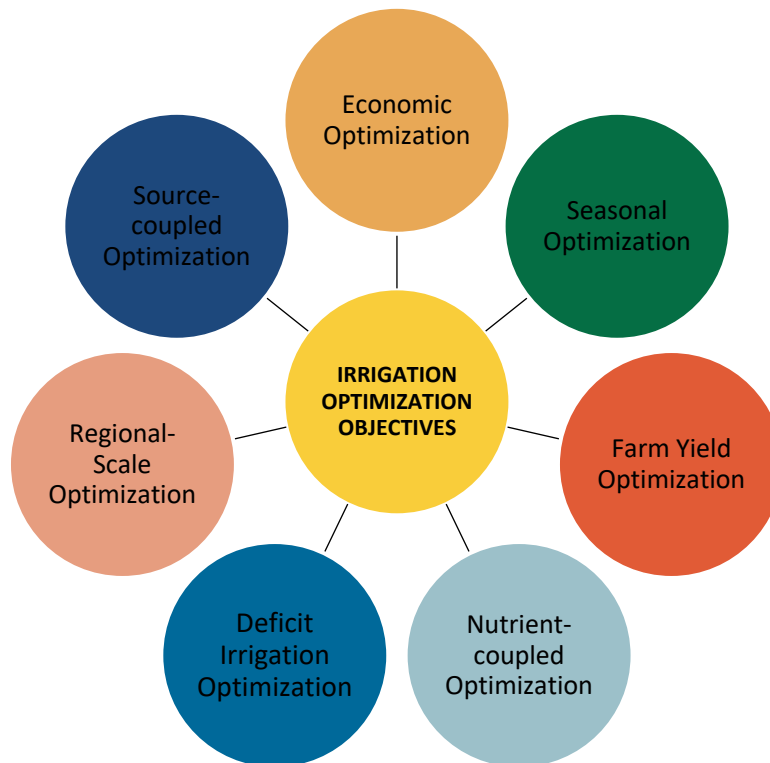
The NSGA-II has been proven effective to solve the multiobjective optimization problem. The optimization framework has been successfully tested for a single event and two full seasonal optimizations. The results are further compared to the integrated irrigation scheduler of the MIKE SHE software. Besides, an example of how to use the framework to test different irrigation strategies is presented.

## 2 Literature Review

---

Many researchers have been working on irrigation automation and various approaches have been investigated to achieve optimal water allocation. The approaches can be divided categorically according to the optimization objectives and the applied optimization method. First, I will review the different optimization strategies from the perspective of the objectives. Second, the frameworks are reviewed based on the applied optimization methods. Last, I will focus on research that investigated similar approaches to the optimization framework of this thesis.

Figure 1 shows the various optimization objectives that researchers have investigated in the context of irrigation. The objectives either differ in a spatial (e.g., single, or multiple fields) or temporal sense (short-term or seasonal), address specific aspects of the irrigation physics (e.g., nutrients or salts), or tackle economic incentives.



*Figure 1: Overview of optimization objectives in the field of irrigation.*

Seasonal optimization focuses on the optimization of irrigation over the whole crop season. Here, expected dry periods, water resources and growth strategies play a major role. Several researchers have followed this optimization strategy (Brown et al., 2010; Fu et al., 2014; Kassing et al., 2020). Kassing et al. (2020) proposed a two-level control

strategy including a seasonal planner to account for long-term effects while optimizing on a daily level. The seasonal planner is based on input data for historic weather data, water availability, crop and field information, and seasonal constraints. Fu et al. (2014) focused on the crop growth stages, optimizing by allocating the limited amount of water to the crucial time of the season.

Economic optimization aims to maximize the profit of the farm, considering all economic factors. Beside fluctuations of product and water prices, García and Fereres (2012) also took fluctuations in agricultural policies into account. Their tool is mainly designed to help the farmer to make a pre-seasonal decision. Ortega Álvarez et al. (2004) developed a model to maximise profit on a farm, based on comparisons of gross margins of crops, cost functions, irrigation depths, and the applied irrigation water. Further researchers followed similar approaches (Kuo & Liu, 2003).

Moreover, there are concepts where the regional water allocation is optimized, which is a more holistic approach that takes large scale parameter into account, such as water supply. One approach is to couple the modelling of reservoir systems to the irrigation schedule (conjunctive management). This is often done on a regional scale, too. By applying this, fluctuation of water tables and ecological incentives are embedded in the irrigation context. (Belaineh et al., 1999; Jiang et al., 2016; M. G. Kang & Park, 2014; Singh, 2014)

Other researchers simply focus on the optimization of the growing conditions for the plants. Due to the many factors that play a role in this, different strategies exist. Some researchers solely aim to keep the soil moisture content in a stress-free level (Delgoda, Malano, et al., 2016). However, nutrients and water stand in a close relationship, as will be discussed in 3.1.5. Thus, Roy et al. (2019) modelled the nutrient transport additionally to optimize the placement of a subsurface water retention technology. Li et al. also concentrated on the water-nutrient coupled modelling to optimize cotton yield, income, and nitrogen and water use efficiency (X. Li et al., 2019).

One strategy that has been discussed by many researchers is deficit irrigation. Here, the plant is deliberately exposed to water stress for a certain time of the growing period or the whole season to optimize the crop-water production function (Kirda, 2002). This concept is crucial in arid or semi-arid regions and will be further discussed in Subsection 3.1.4. Lopez et al. (2017) applied deficit irrigation to conduct a seasonal optimization under water limitation, based on the growth stages of soya and maize.

After reviewing the various optimization objectives, I will now delve into the different optimization methods that have been applied in the field of irrigation optimization. The available literature on irrigation scheduling covers a vast number of optimization approaches, which are illustrated in Figure 2.

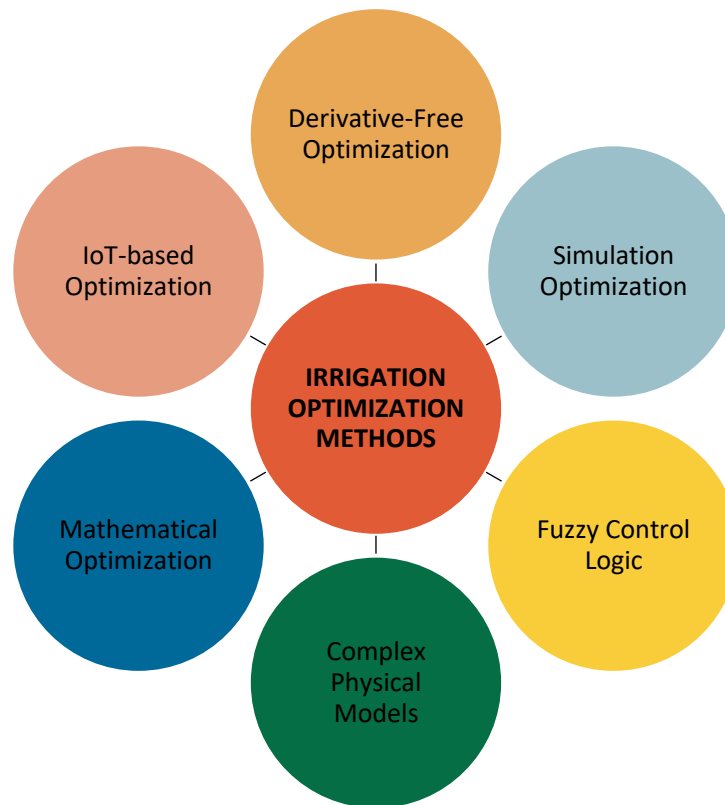


Figure 2: Overview of optimization methods in the field of irrigation.

The methods can be fundamentally divided into four optimization techniques. In mathematical optimization and stochastic programming an underlying algebraic model exists, while derivative-free and simulation optimization rely on a black-box model (Amaran et al., 2016). Furthermore, the approaches can be split up into deterministic and stochastic methods. Optimization algorithms must be further separated into single-objective and multiobjective solvers. In Section 3.3, I will explain the theory of optimization in more detail. fphy

Among others, mathematical optimization strategies comprise Model-Predictive-Control (MPC), Linear Quadratic Regulator (LQR), and Proportional-integral-derivative (PID) controller (McCarthy et al., 2014). Delgoda et al. (2016) implemented a MPC-based strategy to minimize root zone soil moisture deficit under limited water resources. In their approach, direct measurements can be included and uncertainty of weather forecasts are dealt with by introducing robust MPC techniques. Hence, they achieve feasibility and stability, which are crucial aspects of the application of MPC. The optimization is based on a linear system model, developed beforehand, which represents the physics of the system (Delgoda, Saleem, et al., 2016). In this context, the root zone soil moisture content is optimized in every timestep.

Fu et al. (2014) also followed this approach in their case study. They developed an integrated interval nonlinear programming model for optimal allocation of water under

uncertainty during the whole season for different crops. Here, the focus lies on the different growth stages of the plant. What made their model special is that the results are obtained in intervals and uncertainties of the agricultural system can be considered. The water resources were allocated based on 18 stages over the growing season and according to the rainfall within the stages.

Alternative control methods include fuzzy control logic (Giusti & Marsili-Libelli, 2015; M. Li et al., 2019; Mendes et al., 2019; Souza et al., 2020) or complex physics-based models (Jones et al., 2003; Mannini et al., 2013). AquaCrop is one example of such an irrigation model. It is an open-source software developed by the Food and Agriculture Organization (FAO) of the United Nations. It is used in several studies (Delgoda, Malano, et al., 2016; García-Vila & Fereres, 2012; Hsiao et al., 2009; Linker et al., 2016). Linker et al. (2016) based their optimization on an AquaCrop model, whereas García-Vila and Fereres (2012) combined an AquaCrop model with an economic model to manage irrigation at farm level.

As another complex model example, McCarthy et al. (2014) combined MPC with an irrigation control simulation framework, which comprises a crop model called OZCOT. With their method, they established a strategy that includes spatial and temporal varying requirements by defining zones. The control algorithm is applied to each zone and a holistic objective function is applied to achieve optimal overall efficiency. The crop model is calibrated upfront and then continuously calibrated during application.

The zoning approach of McCarthy et al. (2014) will be a fundamental idea of this work. The MIKE SHE model is highly simplified for a 1D simple soil-column scenario and therefore characteristic zones need to be defined, each requiring an adjusted simulation model. This is needed in crop fields with heterogeneous soil or varying crops. Parts of the farm are split up into zones and the individual zones are optimized separately.

Simulation Optimization is an approach to optimize under consideration of uncertainty in a derivative-free context (Amaran et al., 2016). Alizadeh and Mousavi (2013) created a model built on the relationship of stochastic rainfall and irrigation, including shallow water table effects. They further considered salt effects on the irrigation, an important aspect in arid regions. Linker et al. (2016) published a paper in 2020, which describes the implementation of a two-stage explicit stochastic optimization for seasonal optimization.

Furthermore, I discovered that many researchers and companies integrate or base their approach on Internet of Things (IoT) technologies (Fan TongKe, 2013; *IoT Based Smart Irrigation System*, 2020; Togneri et al., 2019). In the scope of these systems, sensors measure the soil moisture parameters to ensure optimal growth conditions. Together with IoT systems, Machine Learning algorithms and Artificial Intelligence are generally getting more popular in the agricultural sector (Liakos et al., 2018). Goldstein et al. (2018) used soil moisture sensor data, climate data, and records of actual irrigation strategies that have been performed by an expert agronomist to give irrigation recommendations. Different regression and classification algorithms were tested.

The progress in the field of derivative-free and especially evolutionary algorithms (EA) has also generated interest in the agricultural sector. Schütze et al. (2006) employed an

evolutionary algorithm to optimize deficit irrigation systems. They further compared their results with outcomes from simulated annealing, shuffled complex evolution algorithm, and differential evolution and found an increase of efficiency and a reduction of computational time. Belaqziz et al. (2013, 2014) used a Covariance Matrix Adaptation Evolutionary Strategy (CMA-ES) to schedule irrigation based on an Irrigation Priority Index and certain constraints, such as canal capacity, tasks timing, distances, and canal flow rates. A genetic algorithm (GA) was compared to a linear programming approach by Azamathulla et al. (2008). The GA showed a better yield and outperformed the linear programming approach. A detailed literature review on irrigation optimization through evolutionary algorithms is given by Ikudayisi and Adeyemo (2015).

In the last part of the literature review, I want to examine approaches which bear resemblance to the framework of the optimization approach implemented in this work. Many optimization objectives and methods have been discussed but solely in a single-objective manner. Now, I concentrate on multiobjective optimization. I employed two objective functions, which resemble the root zone soil moisture deficit and water consumption. The Nondominated Sorting Genetic Algorithm II (NSGA-II) was chosen as the multiobjective optimization algorithm. Fanuel et al. (2018) analysed 40 papers on agricultural water management, which use a multiobjective optimization technique in the form of the multiobjective genetic algorithm, NSGA-II, or multiobjective differential evolution. Ikudayisi et al. (2018) applied a similar approach to this works but employed the multiobjective differential evolution algorithm instead. The two objectives were maximising the total crop net benefit over a season while minimizing water use. Another similar approach was taken by Roy et al. (2019). In their study, they used the HYDRUS-2D software to simulate water and nutrient flow plus the DSSAT crop simulation software. HYDRUS-2D is a software to model two-dimensional water, heat, and solute flow, by solving the Richards equation, analogous to MIKE SHE. However, it does not provide a crop prediction module. Therefore, DSSAT is used to simulate crop growth over time and the overall crop yield. Built on top of these simulation models, they also use NSGA-II for optimization. The goal of their study was to integrate a subsurface water retention technology, which is often used in sandy soils. The main objectives of the study are comparable to the ones of this work: maximising the crop yield, whilst maximising water efficiency. (P. C. Roy et al., 2019)

During the literature review, I found that the list of scientific papers on irrigation scheduling is long and the theory behind all the approaches can be challenging. There is a lack of research that compares the different approaches based on computational time, success, and applicability to the optimization problem.

## 3 Theoretical Background

---

In this chapter, I first explain the physical processes of the unsaturated zone, which are relevant to understand the optimization framework. Special attention will be paid to the phenology and soil moisture conditions that impact the plant growth. Second, the MIKE SHE software is described, with its main features that are important to the optimization. Subsequently, I will examine the vast theory of relevant optimization strategies and the NSGA-II solver algorithm. Last, a short overview of the Python libraries that were used is given.

### 3.1 Physical Background

---

This section deals with the physics of the unsaturated zone. The principle of evapotranspiration and the Penman-Monteith method are outlined. The soil moisture retention curve is a crucial concept to define the water-soil relationship. It will be discussed subsequently. Furthermore, the method of deficit irrigation is explained, and a short overview of general plant phenology is given with maize as an example crop.

#### 3.1.1 Evapotranspiration

Some definitions regarding the physics of water must be discussed to understand the principles MIKE SHE is based upon. Evaporation is the process where water changes its phase from liquid to gas caused by energy. The main energy source that affects evaporation is solar radiation, followed by evaporation caused by air temperature (DHI, 2019). Furthermore, when water evaporates and the humidity rises, the evaporation will decrease. Wind is crucial for the exchange of moist and dry air. If vegetation shades the soil, evaporation will decrease due to diminished solar radiation.

Transpiration is a term used to describe the evaporation from the plant tissues. Pores in the plant leaves are the predominant cause of water loss. These pores are called stomata and various plants can open or close their pores to regulate the water loss. Wind, solar radiation, and humidity also influence the overall transpiration of the plant. (DHI, 2019)

Evapotranspiration describes the combination of both processes on vegetated soils. While shortly after sowing the seed the soil evaporation dominates, the crop transpiration causes the main part of water evaporation as the plant growth proceeds. This relationship is depicted in Figure 3, where the leaf area index (LAI) indicates the stage of the plant growth. The LAI is a dimensionless parameter that quantifies the total upper leaf area of a plant per  $\text{m}^2$  ground area (Allen, 1998).

Another process related to the water balance in the unsaturated zone is interception. While canopy interception defines the retention of precipitation on leaves, branches, and stems of plants, the more general term interception further comprises the interception



caused by forest floor (Gerrits & Savenije, 2011). The retained water does not add to the soil moisture but is evaporated directly.

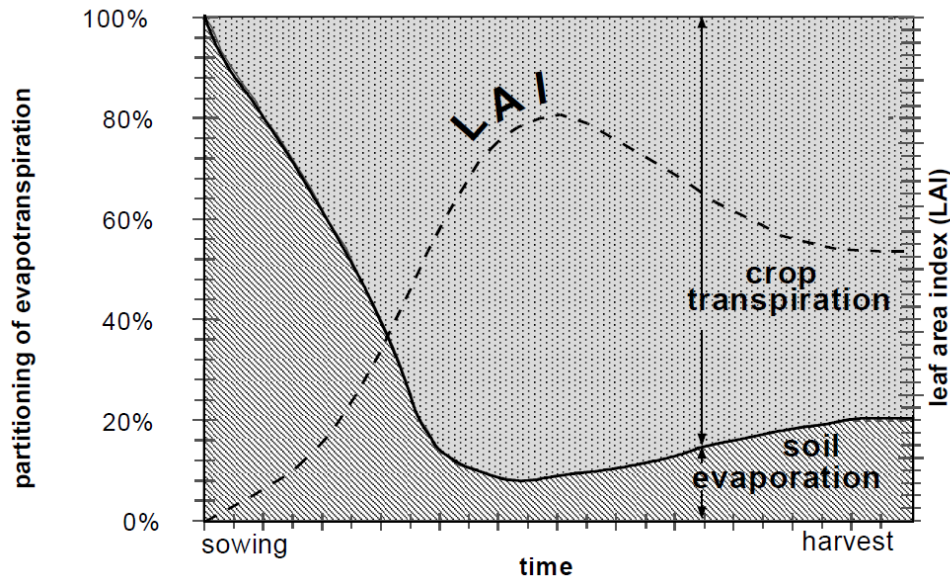


Figure 3: Partitioning of evapotranspiration over the growth period of a plant. (Allen, 1998)

Different definitions are in use to describe evapotranspiration. Potential evapotranspiration is mostly defined as the evapotranspiration that would occur on a surface that has unlimited water supply, such as a lake. However, the FAO strongly recommends not to use this denomination due to ambiguities in different definitions. This is the reason they introduced the so-called reference crop evapotranspiration or simply reference evapotranspiration, which is denoted as  $ET_0$ . This parameter is based on a hypothetical grass crop, of 12 cm height, with a surface resistance of  $70 \text{ s m}^{-1}$  (indicating a dry soil surface) and an albedo of 0.23. In this approach, one assumes that the grass crop is fully watered and, hence,  $ET_0$  characterizes the maximum ET that can be extracted from the reference grass surface. (Allen, 1998; DHI, 2019)

Reference evapotranspiration solely depends on climate data. Actual evapotranspiration, on the other hand, relies on different crop characteristics, such as resistance to transpiration, crop height, crop roughness, reflection, ground cover, and crop rooting characteristics. Hence, distinct crops can have different ET levels even though they grow under identical environmental and climatic conditions.

Crop evapotranspiration under standard conditions ( $ET_c$ ) is a plant-specific term that is used to express evapotranspiration of crops under optimal water, management, environmental, and climatic conditions, grown in a large field.  $ET_c$  is calculated by multiplying  $ET_0$  by the so-called crop coefficient  $K_c$ . This factor describes the physical and

physiological differences between the actual crop and the reference crop. The FAO provides values of  $K_c$  for all commonly cultivated crops. (Allen, 1998)

Various stress factors may influence plant growth in a way that prevents optimal conditions. This can be in the form of pests and diseases, salinity, low fertility, water stress, and spatial limitations of root development (Allen, 1998). These factors depend partly on the phenology of a plant, including its water and nutrient storage ability, resilience, root depths, and LAI. Further details on this are given in Subsection 3.1.5.

To improve the accuracy of the crop evapotranspiration the crop coefficient can be adapted to field and management practises and growth stages. Additionally, water stress is described through a water stress factor  $K_s$ . By multiplying  $ET_0$  with this factor and the modified crop coefficient, one receives the adjusted crop evapotranspiration ( $ET_{c, adj}$ ). In the rest of this work, this parameter will be referred to as actual evapotranspiration ( $ET_a$ ). (Allen, 1998)

A combined approach exists where the water stress factor is adjusted to further account for salinity stress, based on the yield-salinity relationship (Allen, 1998). Because salinity is neglected in my model, the details of this estimation are not further elaborated. (Allen, 1998)

### 3.1.2 Modified Penman-Monteith Method

The modified Penman-Monteith method describes a way to estimate reference evapotranspiration from climate data. Under various methods, it is the approach that delivers the best results with the minimum error. However, it was found that often the values estimated exceed the measurements. Local calibration of the wind function may improve this issue. (Allen, 1998)

The FAO Penman-Monteith method is based on the following equation (Allen, 1998):

$$ET_0 = \frac{0.408\Delta(R_n - G) + \gamma \frac{900}{T + 273} u_2 (e_s - e_a)}{\Delta + \gamma(1 + 0.34u_2)} \quad (1)$$

where

$ET_0$  = reference evapotranspiration [mm day<sup>-1</sup>],

$\Delta$  = slope vapour pressure curve [kPa °C<sup>-1</sup>],

$R_n$  = net radiation at the crop surface [MJ m<sup>-2</sup> day<sup>-1</sup>],

$G$  = soil heat flux density [MJ m<sup>-2</sup> day<sup>-1</sup>],

$\gamma$  = psychrometric constant [kPa °C<sup>-1</sup>],  
 $e_s$  = saturation vapour pressure [kPa],  
 $e_a$  = actual vapor pressure [kPa],  
 $u_2$  = wind speed at 2 m height [m s<sup>-1</sup>],  
 $T$  = mean daily air temperature at 2 m height [°C].

To ensure accuracy, the weather measurements should be taken at two meters above the ground with well-watered green grass shading the soil (Allen, 1998). Zotarelli et al. (2018) provide a detailed step-by-step guide to calculate the reference evapotranspiration according to Penman-Monteith.

### 3.1.3 Soil-Water Retention Curve - Van Genuchten Equation

Soil-water retention curve indicates how the volumetric water content behaves to changes of the matric potential. This depends on the soil type and grain size. Van Genuchten (1980) established an equation to describe this behaviour:

$$\theta(\Psi) = \theta_r + \frac{\theta_s - \theta_r}{(1 + (\alpha|\Psi|^n)^{1-\frac{1}{n}})} \quad (2)$$

where

$\theta(\Psi)$  = water content according to matric potential [L<sup>3</sup> L<sup>-3</sup>],

$\Psi$  = suction pressure [hPa],

$\theta_s$  = saturated water content [L<sup>3</sup> L<sup>-3</sup>],

$\theta_r$  = residual water content [L<sup>3</sup> L<sup>-3</sup>],

$\alpha$  = shape parameter [L<sup>-1</sup>],

$n$  = shape parameter [-],

shape parameter  $m = 1 - \frac{1}{n}$  [-].

The empirical shape parameters of this approach depend on the pore size distribution and the soil type (van Genuchten, 1980). Another factor that influences the soil moisture retention curve is the organic content in the soil. Jong et al. (1983) discovered that although the texture of the soil is the main influence on the shape and position of the

curve, the organic content affects predominantly the point at which a break of the curve occurs. Moreover, rock fragments in the soil are causing variability. Thus, an approach to set soil parameters for soil-gravel mixtures is addressed by Wang et al. (2013).

#### 3.1.4 Deficit Irrigation

Increasing water scarcity has led to the establishment of a new water optimization strategy, called deficit irrigation (DI). Instead of maximizing the crop yield per unit area by satisfying evapotranspiration needs, deficit irrigation aims to maximize the production per water unit consumed whilst stabilizing crop yield (Fererres & Soriano, 2006). This means that crops are grown under water stress conditions below the evapotranspiration needs. It was found that deficit irrigation can be applied during growth stages with low sensitivities without influencing the crop yield (FAO, 2002). This approach is becoming more popular, especially in arid climates as water becomes less abundant. When deficit irrigation is applied the salt-balance in arid climates needs to be managed well to minimize water use, while preventing salt problems. Some water loss is unavoidable because salt leaching must be achieved by applying excess water (Fererres & Soriano, 2006).

#### 3.1.5 Phenology and Soil Moisture Conditions

Phenology is the study of periodic plant and animal life cycle events correlated with climatic conditions and biological phenomena (Demarée & Rutishauser, 2011). In irrigation optimization, various phenological aspects are to be considered, such as root development, growth stages, threshold air temperatures, flowering, development of canopy cover, salinity, and nutrient stress. Not only the development of the roots themselves is important, but which specific root parts absorb how much water (Ahmed et al., 2018). In Figure 4, a rule of thumb is given for the relationship root depth to water uptake according to the Natural Resource Conservation Service of the United States (Waller & Yitayew, 2016).

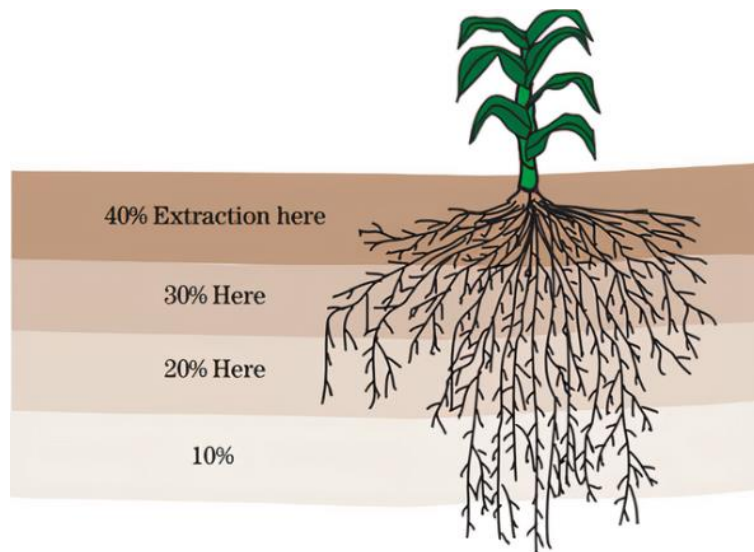


Figure 4: Rule of thumb for root water uptake over depth. (Waller & Yitayew, 2016)

Another significant factor is the interaction between water and nutrients. While water makes nutrients available to the plant, the nutrients can also affect the soil water uptake and the plants' resistance to water stress. Droughts lead to a reduction of nutrient uptake mainly because nutrients are transported to the roots via water films. If the water film is discontinuous, the transport will be disrupted. Moreover, the lack of water causes diminished microbial activity, which is responsible for the degradation of organic matter, releasing nitrogen, phosphorus, and sulphur. (Shaxson & Barber, 2003)

Nutrients can also be flushed out of the root zone by advection. While plants can generally only store a limited amount of water within them, nutrient storage is much higher. Thus, it is suggested to prioritize water supply as a limiting factor. (Shaxson & Barber, 2003)

A lack of nutrients places constraints on water uptake by preventing root growth. Predominantly phosphorus is the limiting nutrient for root development. An appropriate strategy is to apply phosphorus-based fertilizers in lower rainfall periods to improve root growth and water uptake capability of the plant and to prevent high flow rates that can flush out the fertilizer. An approach to modify root growth is described by Aibara and Miwa (2014). They propose the modulation of the root system according to the nutrient conditions. Phosphorus, for example, is stored predominantly in the topsoil. Putting the plant under phosphorus starvation will lead to excessive root development in the upper soil. The opposite case would be nitrate and sulphate, which are more water-soluble. Here, the lack of them will lead to root growth towards larger depths. In general, it is hard to assess if nutrients or water is the limiting factor because it depends on the sensitivity to water stress or nutrient stress at the current crop growth stage. (Shaxson & Barber, 2003)

The water content available for plants depends on the soil texture. By determining the soil water retention curve, the relationship between matrix potential and water content can be established. If no measured data is available, a model can be used. The most widely used

one is the van Genuchten model (van Genuchten, 1980), which was presented before. It is also used in the MIKE SHE software. With the help of this model, the volumetric water content at the permanent wilting point (PWP) and field capacity (FC) are determined. The PWP is the point where the matrix potential is so large that the remaining water cannot desorb from the soil matrix and is hence not available to the plants. The PWP was defined to be at a matric potential of -15 bar (Veihmeyer & Hendrickson, 1928). The FC is the other extreme where the gravity force of the water exceeds the soil matrix holding capacity. The field capacity is the available water after 2-3 days of drainage with no evapotranspiration occurring as bulk water content remaining in soil at -0.33 bar of suction pressure (Israelson, O.W. & West, F.L, 1922). This value has become the default over the years. However, other matric potential values have been proposed, such as -0.25 bar, -0.1 bar, and -0.05 bar (Nemes et al., 2011; Romano & Santini, 2002; Salter & Haworth, 1961). In MIKE SHE, a value of -0.1 bar is set (DHI, 2017b). As drained water is not available to plants either, the amount of total available water is hence defined as (Allen, 1998):

$$TAW = 1000 (FC - PMP) Z_r \quad (3)$$

where

TAW = total available water content [mm],

FC = field capacity [ $\text{m}^3 \text{m}^{-3}$ ],

PWP = permanent wilting point [ $\text{m}^3 \text{m}^{-3}$ ],

$Z_r$  = rooting depth [m].

Optimal growing conditions are fulfilled when the crop evapotranspiration equals the available water amount in the root zone of the crop. The difference between these two parameters is also defined as root zone soil moisture deficit (RZSMD) (Delgoda, Saleem, et al., 2016). Because it is unlikely to achieve a constant RZSMD of zero, parameters are needed to describe how the plant reacts when this criterion is not fulfilled.

Theoretically, all of TAW can be used by plants, however, the plant will be under stress before the two thresholds PWP and FC are reached. This has several reasons. Most plants need oxygen for cell respiration of the root cells (van Bodegom et al., 2008). If too much water is introduced into the soil, it will possibly lead to reduced aeration of the soil and anaerobic conditions causing the plant stress (Ministerium für Ländliche Entwicklung, Umwelt und Verbraucherschutz des Landes Brandenburg, 2005). Hence, upper thresholds for the soil moisture content exist to limit stressful conditions. When the root zone soil water content drops, plants experience stress already before the PWP is reached. This is caused by increasing soil matric potential, which reduces water transport velocity towards the plant. The rate at which water is transported to the plant drops with

lowering water content and transpiration demand cannot be maintained. In summary, plants will experience water stress before PWP, or FC is reached. This is expressed in terms of the readily available water content (RAW) (Allen, 1998):

$$RAW = p TAW \quad (4)$$

where

TAW = total available water content [mm],

RAW = readily available water content [mm],

p = depletion fraction [0-1].

The depletion fraction defines the average fraction of TAW that can be depleted before deficit water stress occurs. The ministry of agriculture, environment and spatial planning of the state Brandenburg in Germany (2005) recommends the following water stress thresholds:

*Table 1: Guideline for water stress thresholds. (Ministerium für Ländliche Entwicklung, Umwelt und Verbraucherschutz des Landes Brandenburg, 2005)*

<b>Share of TAW [%]</b>	<b>Growth development</b>
< 30	plant under water stress.
30-50	sufficient water supply.
50-80	optimal water supply.
80-100	excessive water, danger of oxygen depletion.
> 100	oversupply

Even though Table 1 is a practical first approach, it is not sufficiently accurate. As elaborated before, plants show different characteristics regarding water stress sensibility. This depends on the crop itself plus the growth stage. Furthermore, climate-dependent potential crop evapotranspiration plays a major role in the depletion fraction. Root depth and weather conditions, therefore, affect the depletion fraction greatly. The factor varies between 0.30 for shallow-rooted plants at high crop evapotranspiration rates, while deep-rooted crops at low  $ET_c$  show depletion fractions of 0.70. Crop specific values for the depletion fraction of different crops can be found in Allen et al. (1998). A common value used for many crops is 0.50. A common simplified approach is to use one depletion fraction per growth stage. (Allen, 1998)

Adjustments for weather conditions must be made (Allen, 1998):

$$p = p_{ET=5 \text{ mm/d}} + 0.04 * (5 - ET_c) \quad (5)$$

where

$p$  = weather-corrected depletion fraction [mm] [0.1-0.8],

$p_{ET=5 \text{ mm/d}}$  = depletion fraction for no stress,

$ET_c$  = crop evapotranspiration [mm/d].

The water uptake further depends directly on the soil matric potential and the corresponding hydraulic conductivity of the soil and therefore on the soil. Hence, it is also recommended to adjust the depletion fraction to the soil type. Depletion factors of fine-textured soils should be reduced by a factor by 5-10%, whereas coarse textures an increase by 5-10% is suitable. (Allen, 1998)

The yield-moisture stress relationship was established by the FAO in the Irrigation and Drainage paper N°33 (Doorenbos & Kassam, 1979). This so-called crop-water production function is the following (Doorenbos & Kassam, 1979):

$$\left(1 - \frac{Y_a}{Y_m}\right) = K_y \left(1 - \frac{ET_{c,adj}}{ET_c}\right) \quad (6)$$

where

$K_y$  = yield response factor [-],

$Y_a$  = actual yield [kg ha<sup>-1</sup> d<sup>-1</sup>],

$Y_m$  = maximum yield [kg ha<sup>-1</sup> d<sup>-1</sup>],

$ET_{c,adj}$  = adjusted (actual) crop evapotranspiration [mm d<sup>-1</sup>],

$ET_c$  = crop evapotranspiration for standard conditions (no water stress) [mm d<sup>-1</sup>].

The yield response factor  $K_y$  is a crop dependent parameter and may vary over the growth stages. It is an indicator of crop sensitivity. Generally,  $K_y$  is large during the flowering and yield formation period and low during the vegetative and ripening period, which means



that plants are more resilient at the beginning of the growing season. Seasonal yield response factors are given in Doorenbos & Kassam (1979). (Allen, 1998)

### 3.1.6 Characteristics of Maize

Maize was used as a test crop for the optimization framework. Thus, the crop characteristics are discussed in this section. Maize is one of the most important cereals for human and animal consumption and is grown in temperate to tropic climates during the period where temperatures are above 15°C. Maize has adapted to various climates and the right choice of variety is crucial for successful cultivation. It can withstand high temperatures up to 45°C, provided that sufficient water is supplied. However, the plant is susceptible to frost during the seedling stages. Maize growth is very reactive to radiation and grows on most soils apart from very dense clays and very sandy soils. (FAO, 2020)

The sensitivity of maize to water shortages is generally medium to high (Allen, 1998). Furthermore, the crop is sensitive to waterlogging. Especially, during flowering waterlogging must be avoided. Moreover, maize is moderately susceptible to salinity. The root depth maximum of maize ranges from 60 cm to 100 cm. A detailed paper was published by Ahmed et al. (2018) about root water uptake of maize. It was found that the crown roots are the part of the root system that takes up the most water. Seminal roots and their laterals also contribute to a lesser extent to the overall water uptake. This is in accordance with the rule of thumb in Subsection 3.1.5 and is implemented in the objective function. (FAO, 2020; Ministerium für Ländliche Entwicklung, Umwelt und Verbraucherschutz des Landes Brandenburg, 2005)

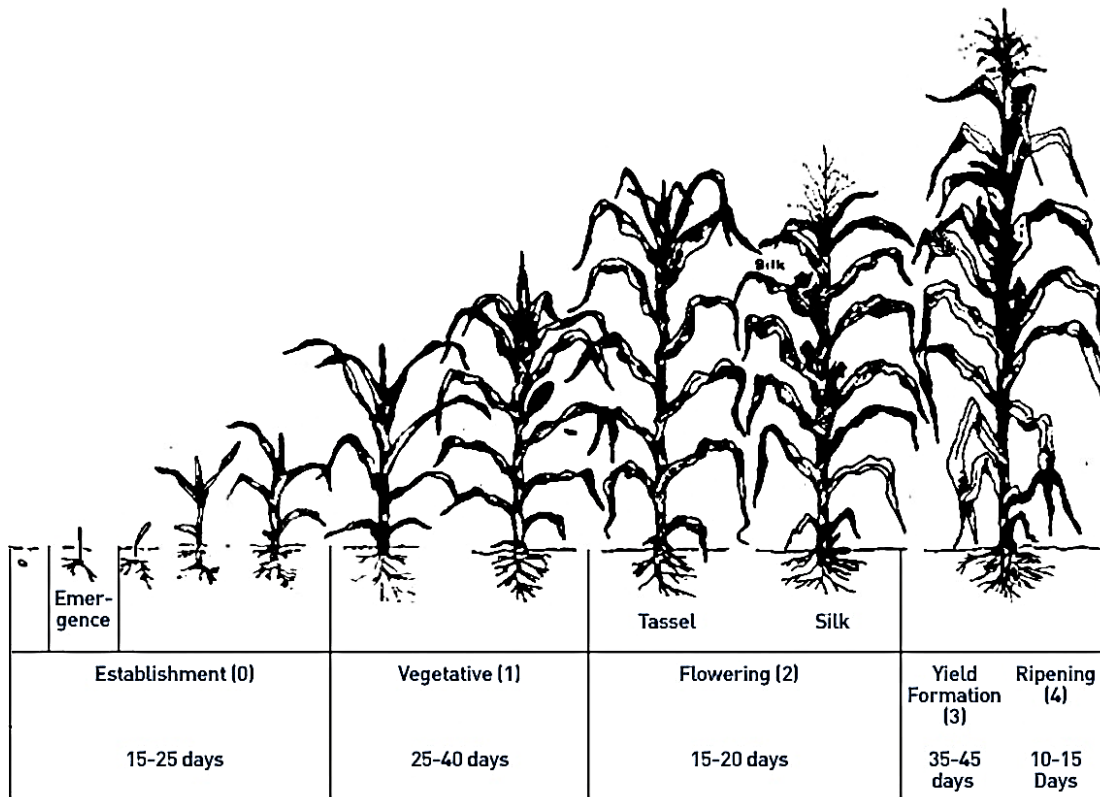


Figure 5: Growth stages of maize. (FAO, 2020)

The growth stages of maize are shown in Figure 5. The yield response factors, root depth, crop coefficients, and depletion coefficients over the growth stages are given in Table 2. The sensitivity to water stress is high during the flowering stage, while it is moderate during the establishment and vegetative periods. This is represented by the yield response factor. During the flowering stage, RAW must not drop below 50% (Bundesinformationszentrum Landwirtschaft, 2017; Ministerium für Ländliche Entwicklung, Umwelt und Verbraucherschutz des Landes Brandenburg, 2005).

Table 2: Summary of seasonal crop-characteristic coefficients of maize. (FAO, 2020)

Crop characteristic	Growth Stage				
	Initial	Crop Development	Mid-season	Late	Total
Stage length [days]	25	40	40	35	135
Depletion coefficient p [-]	0.50	0.50	0.50	0.80	-
Root Depth [m]	0.30	>>	>>	1.00	-
Crop Coefficient Kc [-]	0.30	>>	1.20	0.50	-
Yield Response Factor Ky [-]	0.40	0.40	1.30	0.50	1.25

Maize is a suitable crop for deficit irrigation. No significant reduction of the crop yield was found when maize was kept at 30-40% depletion of TAW between irrigation events (Stegman, 1982). Kang et al. (2000) confirmed that when applying deficit irrigation during certain periods, the maize crop yield can be maintained.

## 3.2 MIKE SHE Software

---

The MIKE software products are developed by the company DHI. One of the products is the MIKE SHE software, which is used for water balance modelling and simulation of surface-groundwater interactions. It integrates all significant processes at the catchment scale, such as evapotranspiration, infiltration, overland flow, unsaturated flow, groundwater flow and channel flow (*MIKE SHE*, 2020). Physical, stochastic, and data-driven methods are provided to model individual processes. MIKE SHE models can hence be considered as a holistic approach for the implementation of the whole hydrological cycle with its influences on the crop field.

All MIKE products are using the proprietary Data File System (DFS) format to handle spatial time series data. DFS is a binary format that can be split into three parts: A header section comprises general information such as start time, geographic map projection, etc. The static section saves time-independent static data for certain items. An item could be, for example, a parameter such as soil moisture content. The section that takes the most storage space within the file architecture is the dynamic data section, which contains time-dependent data. The Data File System format can be sub-divided into spatial-dimension-dependent sub-formats. (DHI, 2020)

The MIKE SHE model itself has the DHI proprietary file format PFS (DHI, 2017a). It can be understood as a setup file where all the information of the model is stored.

In the following chapters, I will provide a theoretical overview of the MIKE SHE software functionalities that are crucial in the irrigation optimization context and which are part of the setup of the inherent model.

### 3.2.1 Unsaturated Flow

MIKE SHE is a software to simulate hydraulic processes with its focus on the unsaturated zone. However, an iterative coupling approach is established to account for processes related to the saturated zone. Further, the software neglects horizontal flow due to strong vertical gravity forces during infiltration processes. This can become a problem for steep hill areas but it is stated that in most cases this approach is sufficiently accurate. (DHI, 2017b)

Three options are available to calculate the flow in the unsaturated flow through MIKE SHE: the Richards' equation, gravity flow, and a two-layer water balance. I will focus on

the Richards equation, which is the most sophisticated method that was applied in the underlying model of this work. (DHI, 2017b)

The Richards equation was established in 1931 and is based on the soil moisture retention curve and the hydraulic conductivity function (Richards, 1931). The tension-based equation is defined by (DHI, 2017b):

$$C \frac{\partial \psi}{\partial t} = \frac{\partial}{\partial z} \left( K(\theta) \frac{\partial \psi}{\partial z} \right) + \frac{\partial K(\theta)}{\partial z} - S \quad (7)$$

where

$C = \frac{\partial \theta}{\partial \psi}$ , which is the slope of the soil moisture retention curve,

$K(\theta)$  = hydraulic conductivity function,

$S$  = root extraction sink term,

$\psi$  = matrix potential,

$\theta$  = volumetric soil moisture content.

The extraction sink term  $S$  comprises the water loss through root transpiration in the entire root zone and the direct soil evaporation. The direct soil evaporation is computed in the top node of the discretization (DHI, 2017b).

MIKE SHE solves the differential Richards equation by applying a fully implicit finite-difference method (DHI, 2017b). The spatial derivatives are described at time level  $t+1$ , whereas  $C(\theta)$  and  $K(\theta)$  are defined at time level  $t+1/2$ . The numerical solution of the Richards equation is the following (DHI, 2017b):

$$C_x^{t+1} \left( \frac{\psi_x^{t+1} - \psi_x^t}{\Delta t} \right) = \left| K_{x+\frac{1}{2}}^{t+\frac{1}{2}} * \left( \frac{\psi_{x+1}^{t+1} - \psi_x^{t+1}}{\Delta z_{x+1}} \right) - K_{x-\frac{1}{2}}^{t+\frac{1}{2}} * \left( \frac{\psi_x^{t+1} - \psi_{x-1}^{t+1}}{\Delta z_x} \right) \right| * \frac{1}{\frac{1}{2} * (\Delta z_{x+1} + \Delta z_x)} - S_x^{t+1} \quad (8)$$

where

$C = \frac{\partial \theta}{\partial \psi}$  , which is the slope of the soil moisture retention curve,

$K$  = hydraulic conductivity,

$S$  = root extraction sink term,

$\psi$  = matrix potential,

$\Delta z$  = physical distance between neighbour nodes,

$\Delta t$  = time step interval.

The spatially centred hydraulic conductivity  $K$  is calculated as follows (DHI, 2017b):

$$K_{x+1/2}^{t+1/2} = \frac{K_{x+1}^{t+1/2} + K_x^{t+1/2}}{2}$$

$$K_{x-1/2}^{t+1/2} = \frac{K_x^{t+1/2} + K_{x-1}^{t+1/2}}{2}$$

### 3.2.2 Evapotranspiration

MIKE SHE performs the computation of the actual evapotranspiration through an evapotranspiration (ET) model based on the method Kristensen and Jensen (DHI, 2017b; Kristensen & Jensen, 1975). This approach is built on the assumption that actual evapotranspiration ( $ET_a$ ) cannot exceed  $ET_0$  and that the temperature remains above the frost point. Due to this limitation, it is not required to differentiate between transpiration and evaporation, a separation that can be difficult. The dominating influences that reduce the actual evapotranspiration are assumed to be lack of water in the root zone and density of the vegetation, defined by the leaf area index. Kristensen and Jensen developed this method based on empirical equations derived from field data. (Kristensen & Jensen, 1975)

### 3.2.2.1 Plant Transpiration

The transpiration of the crop depends on the density of the vegetation, which is a function of the LAI, the root zone soil moisture content, and the root density. The underlying equation is (DHI, 2017b):

$$E_{at} = f_1(LAI) * f_2(\theta) * RDF * ET_C \quad (9)$$

where

$E_{at}$  = actual transpiration,

$RDF$  = root distribution function,

$ET_C$  = crop evapotranspiration,

$$f_1(LAI) = C_2 + C_1 * LAI \quad (10)$$

where

$C_1$  and  $C_2$  are empirical values [-],

$LAI$  = leaf area index [-],

$$f_2(\theta) = 1 - \left( \frac{\theta_{FC} - \theta}{\theta_{FC} - \theta_W} \right)^{\frac{C_3}{ET_C}} \quad (11)$$

where

$\theta_{FC}$  = volumetric moisture content at field capacity [-],

$\theta_W$  = volumetric moisture content at wilting point [-],

$\theta$  = actual volumetric moisture content [-],

$C_3$  = empirical value [-],

$ET_C$  = crop evapotranspiration.

The function  $f_1$  expresses the dependency relation of the transpiration on the leaf area, whereas  $f_2$  describes the soil moisture content in the root zone.  $C_1$  and  $C_2$  are empirically derived and dimensionless parameters.  $C_1$  is the slope of the linear function  $f_1$  at the

current value of LAI. It influences the ratio of soil evaporation to transpiration and is a plant-dependent value. The MIKE SHE handbook proposes a  $C_1$ -value of 0.3 for agricultural crops and grass, whereas Kristensen and Jensen defined a more specific value of 0.31 for barley, fodder sugar beets, and grass. Large values of LAI cause the function to be 1.0. The dependency is depicted in Figure 6.  $C_2$  is the basic evaporation that occurs when the soil moisture content in the root zone is above PWP due to diffusion processes between the moist soil atmosphere and the drier atmosphere above the soil. The parameter does neither depend on vegetation density nor soil dryness. A value of 0.2 is used in MIKE SHE based on the assumption that the crop is cultivated in clayed loamy soils. However, in the paper of Kristensen and Jensen, a value of 0.15 is proposed for these soils.  $C_2$  is reduced linearly for soil moisture values below the PWP. (DHI, 2017b; Kristensen & Jensen, 1975)

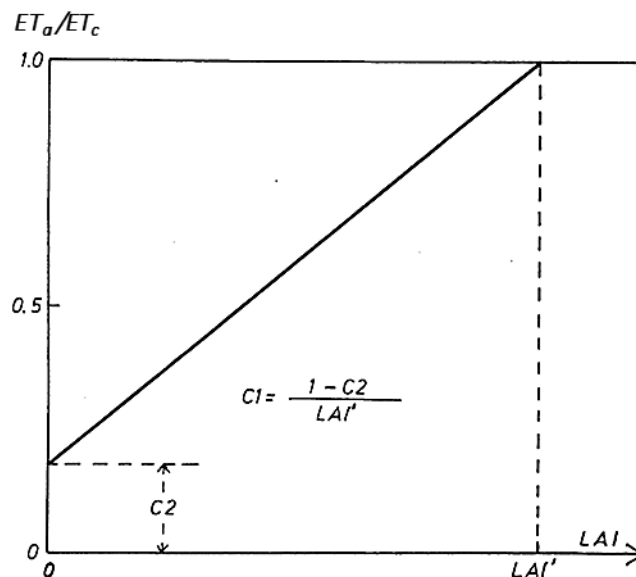


Figure 6:  $ET_a/ET_c$  relationship to LAI regarding the empirical constants  $C_1$  and  $C_2$ . (Kristensen & Jensen, 1975)

The influence of  $C_2$  in MIKE SHE is best understood if we look at how the software handles evapotranspiration within the discretization. Soil evaporation is only considered in the top node of the unsaturated zone (UZ) soil profile. This is a crucial detail because it implies that the soil evaporation can solely occur in the top node. Therefore, this node strongly influences the total actual evapotranspiration in dry conditions. If the ratio soil evaporation - transpiration is shifted towards soil evaporation by increasing  $C_2$ , it will lead to an overall drop of total actual evapotranspiration, because of higher extraction rates from the topsoil node and lower extraction at the bottom nodes (see Figure 7). This is caused by the topsoil drying out, while the bottom moisture content is not evaporated. Thus, by increasing the constant  $C_2$ , the model is made dependent on capillary action that brings water up to the upper nodes. (DHI, 2017b)

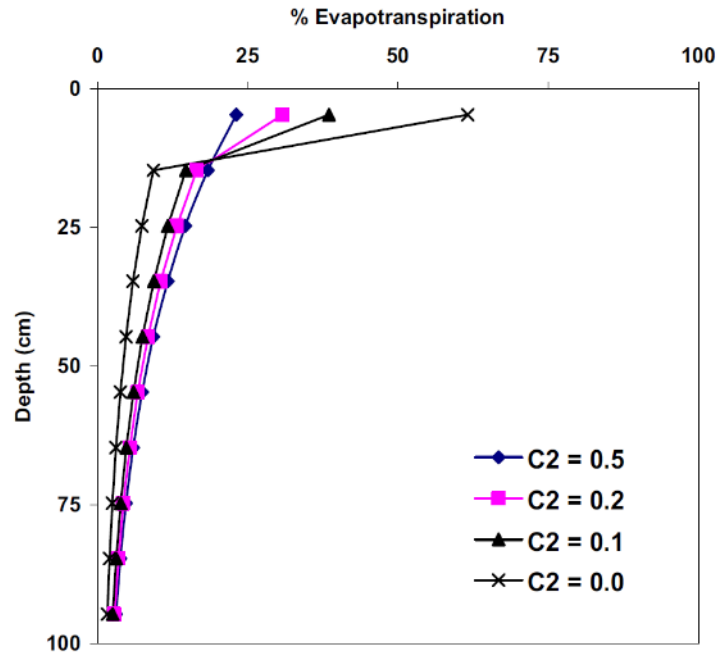


Figure 7: Distribution of actual evapotranspiration over depth for different values of  $C_2$ .  $C_2=0$  corresponds to pure transpiration. (DHI, 2017b)

The function  $f_2$  returns a factor for the current volumetric soil moisture content in the root zone. The factor depends on the parameters field capacity, permanent wilting point, crop evapotranspiration and the constant  $C_3$ . No experimental data is available for  $C_3$ . However, the parameter depends on the root density and the soil type. A great value of  $C_3$  should be chosen for soils with high water release at low matrix potential. MIKE SHE sets a default value of 20 mm/day for  $C_3$ , which is double the rate suggested by Kristensen and Jensen. After all, the value should be set according to experience. (DHI, 2017b; Kristensen & Jensen, 1975)

The root distribution function (RDF) specifies the root development over depth. This is a complex process that depends on climatic, environmental, and phenology conditions. Root development has been discussed in 3.1.5. MIKE SHE offers to choose between user-defined inputs or to use a vertical root density function (DHI, 2017b):

$$\log R(z) = \log(R_0) - AROOT * z \quad (12)$$

where

$R(z)$  = root extraction at depth  $z$ .

$R_0$  = root extraction at the soil surface.

$AROOT$  = root mass distribution factor.

$z$  = depth below the surface.



The root mass distribution factor (AROOT) controls the root distribution. As shown in Figure 8, the closer AROOT is to zero, the flatter the distribution of ET extraction becomes. As in the case of the constant  $C_2$ , AROOT also influences the total actual evapotranspiration strongly. If it is chosen to be high, which means that most water extracted by the roots is done in the upper part of the plant, the topsoil will dry out fast and the total actual transpiration will drop because the remaining water below is not used by the roots. (DHI, 2017b)

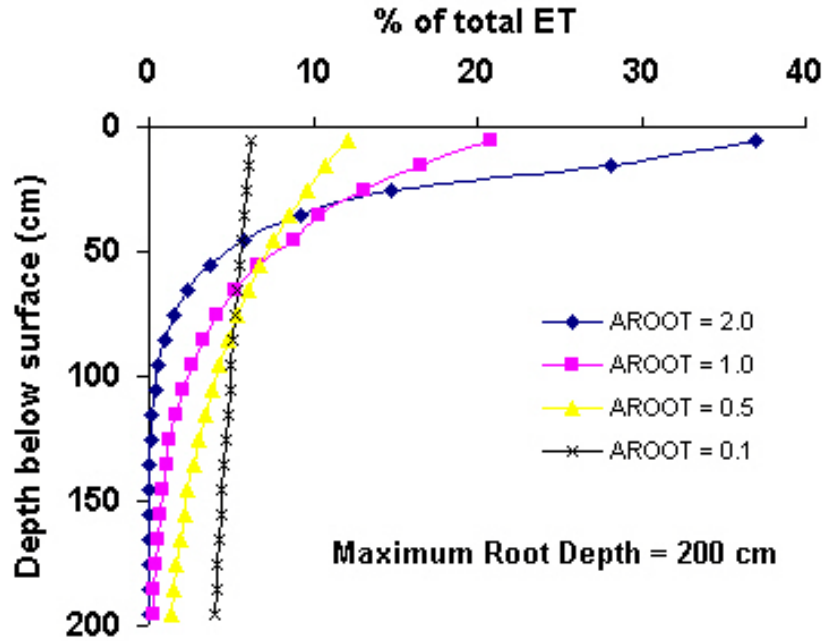


Figure 8: Fraction of ET extracted as a function of depth for different values of AROOT. (DHI, 2017b)

The value for the root distribution function is layer-specific and is estimated by dividing the extracted water in the layer by the total amount extracted by the roots (DHI, 2017b):

$$RDF_i = \frac{\int_{z_1}^{z_2} R(z) dz}{\int_0^{L_R} R(z) dz} \quad (13)$$

where

$z_1 / z_2$  = layer boundaries.

$R(z)$  = root extraction at depth  $z$ .

$L_R$  = maximum root depth.

### 3.2.2.2 Soil Evaporation

As mentioned before, MIKE SHE considers the soil evaporation only in the upper node of the discretised model. The soil evaporation comprises two parts: the basic amount of evaporation ( $ET_{ref} * f_3$ ) and evaporation from excess soil water as the soil moisture content exceeds field capacity (DHI, 2017b):

$$E_s = ET_{ref} * f_3(\theta) + \left( ET_{ref} - E_{at} - ET_{ref} * f_3(\theta) \right) * f_4(\theta) * (1 - f_1(LAI)) \quad (14)$$

where

$E_s$  = soil evaporation.

$ET_{ref}$  = reference evapotranspiration.

$E_{at}$  = total actual transpiration.

$LAI$  = leaf area index.

$$f_3(\theta) = \begin{cases} C_2, & \theta \geq \theta_W \\ C_2 * \frac{\theta}{\theta_W}, & \theta_r \leq \theta \leq \theta_W \\ 0, & \theta \leq \theta_r \end{cases} \quad (15)$$

$$f_4(\theta) = \begin{cases} \frac{\theta - \frac{\theta_W + \theta_{FC}}{2}}{\theta_{FC} - \frac{\theta_W + \theta_{FC}}{2}}, & \theta \geq \frac{\theta_W + \theta_{FC}}{2} \\ 0, & \theta < \frac{\theta_W + \theta_{FC}}{2} \end{cases} \quad (16)$$

### 3.2.2.3 Evapotranspiration from Canopy Interception

MIKE SHE models the canopy interception as storage. Before the storage is not full, no water is added to the soil moisture budget. The maximum storage capacity depends on the growth stage of the plant and the type of the plant per se. The equation is the following (DHI, 2017b):

$$I_{max} = C_{int} * LAI \quad (17)$$

where

$I_{max}$  = interception storage capacity [mm].

$C_{int}$  = interception coefficient [mm].

$LAI$  = leaf area index [-].

The interception coefficient is typically set to about 0.05 mm, but calibration is recommended to increase accuracy (DHI, 2017b). An important aspect of how the interception is displayed in MIKE SHE is that the interception storage is set as a unit of length, not a rate. The full interception storage capacity is added for every timestep, given that precipitation is available. This means that the total intercepted water depends on the timestep defined, as stated in the MIKE SHE handbook (DHI, 2017b). Hence, doubling the number of timesteps in a simulation will also double the intercepted water, provided that the interception storage does not fill up completely. The evaporation from the canopy is estimated as (DHI, 2017b):

$$E_{can} = \min (I_{max}, ET_{ref} * \Delta t) \quad (18)$$

where

$E_{can}$  = canopy evaporation [LT<sup>-1</sup>].

$\Delta t$  = time step length.

MIKE SHE offers an alternative method to estimate the actual evapotranspiration where the unsaturated zone is split up by applying a Two-Layer UZ/ET model established by Yan and Smith into the root zone and the zone below where no ET occurs (DHI, 2017b; Yan & Smith, 1994). This method was not applied in the MIKE SHE model of this work. Hence it is not further elaborated.

### 3.2.3 Irrigation Module

The MIKE SHE software offers a module to include irrigation. The user can define where the water originates from and how it is applied to the field. Various sources are available, such as a river, single well, shallow well, and external. The available water amount can be limited by adding a licence limited irrigation file that sets the maximum amount of water available per timestep. This is usually applied to account for water permits but it can be generally applied to set an upper limit for irrigation rates. Irrigation sources can also be limited by a constant maximum rate. Important to mention is that unused water in one timestep is not accumulated and added to the next timestep. (DHI, 2019)

Three irrigation systems are applicable in MIKE SHE: sprinkler, drip, and sheet irrigation. According to the chosen method, the introduction of water to the field modelled differently. If the user sets sprinkler as the water application method, the irrigation water will be added to the precipitation, whereas for drip irrigation it will be appended to the ground as pond water. The main difference between these two methods is the canopy interception. Sheet irrigation requires an additional data item for defining where the water is added within the

command area, under the objective to calculate overland flow. The water is then added to the cells as ponded water. (DHI, 2019)

The irrigation demand module lets the user define a deficit compensation method. Four options are available: user-specified, crop stress factor, ponding depth, and maximum allowed deficit. User-specified means that no optimization strategy is applied. The user can set a constant value or create an irrigation time series. This setup was used in the optimization framework. The crop stress factor method is based on  $ET_c$  and the  $ET_a$ . This factor describes the allowed fraction  $ET_a/ET_c$  before irrigation is started. The ponding depth method is specially developed for irrigation systems where crops are under water, such as rice fields.

The method that was used to compare the MIKE SHE model to the here developed multi-objective optimization approach is the maximum allowed deficit method. This method is based on the readily available water content, a parameter that has been explained in Subsection 3.1.5. Deficit limits can be set, which define at what soil moisture levels the irrigation starts and ends. The user chooses a reference value, which can be either field capacity or saturation. If the field capacity is chosen, the difference between the moisture deficit start and end corresponds to the depletion factor, described in Subsection 3.1.5. (DHI, 2019)

### 3.3 Optimization

---

The goal of optimization is to find the best available solution from a set of feasible solutions. An optimization problem consists of a real function that is either minimized or maximized. Input values, also known as control variables, are systemically chosen and the output of the function is computed. The function of the optimization problem can be found under various names in literature. It has been denoted as objective function, loss function, or cost function for minimization and utility function or fitness function for maximization. (Boyd & Vandenberghe, 2004; Erwin Diewert, 2008)

In this work, it will be referred to as objective function. The standard continuous deterministic optimization problem is defined as follows (Larson et al., 2019):

$$\underset{x}{\text{minimize}} f(x) \tag{19}$$

subject to

$$x \in A \subseteq \mathbb{R}^n$$

where

$f(x)$  = objective function.

$A$  = domain search space.

The domain search space is some subset of the Euclidean space that lies in the boundaries of optimization constraints. A feasible set is characterized as the elements within this search space. One can distinguish optimization problems according to the variable type. An optimization that is based on discrete variables is called discrete optimization, whereas continuous optimization addresses optimization problems with continuous variables. One challenge in optimization is to find the global minimum. A local minimum is defined as at least as good as any nearby elements, whereas the global minimum is at least as good as every feasible element. In the case of a convex objective function, there is only one global minimum, while nonconvex objective functions may have various local minima. (Boyd & Vandenberghe, 2004)

The underlying optimization framework is based on the MIKE SHE soil-moisture prediction. Hence, no derivative information is available, which makes it a black-box optimization problem. The lack of this information makes it difficult to find descent directions, to proof convergence, and generally to identify characteristics of optimal points (Amaran et al., 2016).

Different approaches exist to solve this kind of optimization problem. One option is numerical differentiation, where the derivative of every timestep is computed, with subsequent algorithms that rely on derivative information, such as the gradient descent algorithm. Alternative solutions are derivative-free optimization (DFO) and simulation optimization (SO). Here, the algorithms do not rely on mathematical information and are therefore suitable for black-box optimization. As depicted in Table 3, the fundamental difference between DFO and SO is that DFO is built on deterministic outputs while SO is applied for uncertainties in the output data.

Table 3: Terminology of optimization problems. (Amaran et al., 2016)

	<b>Algebraic model available</b>	<b>Unknown/complex problem structure</b>
<i>Deterministic</i>	Traditional math programming (linear, integer, and nonlinear programming)	Derivative-free optimization
<i>Uncertainty present</i>	Stochastic programming, robust optimization	Simulation optimization

The present irrigation scheduling optimization problem relies on two objective functions to account for water scarcity and optimal soil moisture conditions. Hence, it is a multiobjective optimization problem (MOOP). Due to their strength in solving MOOPs and their ability to handle black-box optimization, I chose DFO as the strategy to develop an irrigation scheduling multiobjective optimization framework based on MIKE SHE. The mentioned concepts are elaborated in detail within this chapter.

### 3.3.1 Constraints

In an optimization problem, constraints can be set, which makes it a constrained optimization. Constraints can be defined as soft or hard constraints. Hard constraints imply that some boundaries cannot be exceeded, while soft constraints are based on penalization within the objective function. In other words, soft constraints can be exceeded but a penalization method will be applied for the values that exceed the boundaries. (Larson et al., 2019)

Constraints can be also divided into algebraic constraints and simulation-based constraints. Algebraic constraints are used if the constraints are algebraically available, which means that gradient information is available to the optimization method. If the constraint depends on a black-box simulation, simulation-based constraints will be applied. (Larson et al., 2019)

There are several methods to deal with both algebraic and simulation-based constraints. For example, filter approaches attempt to synchronously minimize the objective and the constraint violation. Another method is penalization, as mentioned already. The original penalization strategy was an extreme-barrier approach, where the output of the objective function is set to infinite when values are out of the set constraints (Audet & Dennis, 2006). Later, Audet and Dennis (2009) developed the progressive-barrier method, which employs a quadratic constraint penalty for simulation-based soft constraints and an extreme-barrier for simulation-based hard constraints.

### 3.4 Multiobjective Optimization

---

Multiobjective problems are a vast occurrence in many sectors and our daily lives. Imagine a situation where you are in a supermarket comparing various products of the same type. Your objective is to buy the product that is cheap but also healthy and eco-friendly. Since all these criteria usually cannot be fulfilled at the same time, a compromise must be found. For example, the cheapest product is usually not produced eco-friendly, but buying the product with the highest eco-standards comes at a higher price. These kinds of problems require multiobjective optimization.

As in the case of shopping groceries, the farmer has various objectives for irrigation. Every farmer has the incentive to save water but must ensure that his crops remain in good conditions. Another factor is the nutrient-water interaction in the soil. Water is one limiting factor, but as described in 3.1.5, nutrient availability is crucial for plant growth. This leads to another objective to minimize nutrient usage while creating optimal nutrient availability for the crop. In deficit irrigation, the objective differs inherently since here the aim is the highest possible farm yield given the available total water, tolerating some crop water stress.

Several review papers about multiobjective optimization methods in different fields are available (Althaimen & Arditi, 2019; Cui et al., 2017; Gunantara, 2018). The mathematical formulation of a continuous deterministic multiobjective optimization problem is defined as (Ehrgott, 2005):

$$\underset{x}{\text{minimize}} f_1(x), f_2(x), f_3(x), \dots, f_n(x) \quad (20)$$

subject to

$$x \in A \subseteq \mathbb{R}^n$$

where

$f(x)$  = objective function.

$n$  = number of objective functions.

$A$  = domain search space.

Gunantara (2018) points out two methods that are most commonly applied in MOO. These methods are the Pareto method and Scalarization. Scalarization describes the transformation of the multiobjective functions into a single objective, by applying weights to the individual objective functions. The Pareto optimization on the other hand is a performance indicator component that solves the multiple objectives separately and gives a compromise solution. The optimal solutions can be displayed as the so-called Pareto optimal front (POF). An example POF is visualized in Figure 9. The optimal front is the set of parameterizations, which are Pareto optimal solutions, also called Pareto efficient or Pareto non-dominated. Optimal solutions are defined as points where no objective can be improved further without impacting another objective. Points that do not provide a Pareto optimal solution are called Pareto dominated solutions or Pareto non-efficient. These points represent solutions where an objective function can still be improved without influencing another objective. Some researchers define the solution of a multiobjective optimization problem as computing of all or a representative set of the POF values (Ehrgott, 2005). (Gunantara, 2018)

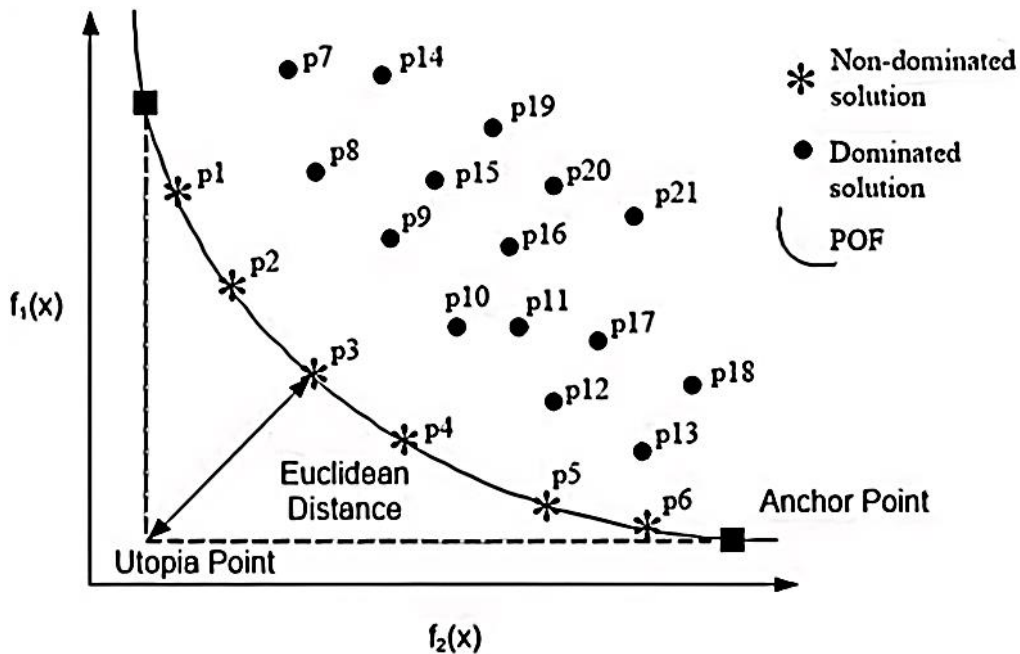


Figure 9: Pareto optimal front for two objective functions with characteristic points. (Gunantara, 2018)



Characteristic points of the Pareto method are the Anchor Point, which is the best value of a corresponding objective function, and the Utopia Point, depicted in Figure 9. The solution that shows the shortest Euclidean distance from the Utopia point to the POF is also known as knee point, which in Figure 9 is  $p_3$ . (Gunantara, 2018; Ramirez-Atencia et al., 2017)

In multiobjective optimization methods, the decision-maker (DM) plays a crucial role. The decision-maker is expected to be an expert in his field. According to the role of the DM, multiobjective optimization methods are generally distributed into four classes (Zhang, 2014). The first is the no-preference methods, which assume that no DM is accessible (Miettinen, 1999). In this method, a neutral compromise is determined without any preferences. Second, are the *a priori* methods. They are based on preferences of the DM, which are established beforehand and a solution is found accordingly. Third are the *a posteriori*, which rely on estimating a set of Pareto optimal values. Upon this information, the DM can decide according to her/his preferences. The last method is the interactive one, meaning that the DM is part of the process and the Pareto front is estimated iteratively. Applying this method has the advantage that the DM can explore his options and focus on feasible solutions. (Braun, 2018; Zhang, 2014)

As mentioned before, scalarization functions are one simplified method to aggregate multiple objective functions into multiple single-objective problems. This represents an *a priori* method because the weights are set beforehand and only one optimal output is estimated. Different approaches exist to achieve this, for example, equal weights, rank order centroid (ROC) weights, and rank-sum (RS) weights (Gunantara, 2018). For scalarization the objective functions must occur in the same range, thus, normalization must be applied (Blank & Deb, 2020; Marler & Arora, 2010).

Weighting methods can also be used *a posteriori* to receive one optimal solution from the Pareto optimal front. The naïve linear weighted-sum method was used in the implemented optimization framework. In this method, relative scalar weights are applied. A disadvantage of this method is that the weighted-sum method only works for convex Pareto fronts (Blank & Deb, 2020).

The equation is the following (Yang, 2014):

$$U(x) = \sum_{i=1} w_i f_i(x) \quad (21)$$

subject to

$$\sum_{i=1} w_i = 1, w_i > 0$$

where

$U(x)$  = weighted sum.

$f_i(x)$  = objective function.

$x$  = control variable.

$w$  = weight.

### 3.4.1 Multiobjective Derivative-Free Algorithm Classes

Many solvers for MOOP that are based on the described Pareto and Scalarization method have been developed over the years. As stated in Custódio et al. (2012), derivative-free algorithms can be subdivided into the following groups: meta-heuristic algorithms, direct search methods (DSM), line-search algorithms for DFO, and trust-region interpolation-based methods. Meta-heuristic algorithms and DSM are the most advanced in multiobjective optimization, thus, the other approaches are not discussed.

#### 3.4.1.1 *Heuristics, Meta-heuristics, and Hyper-heuristics*

In contrast to mathematical optimization or iterative methods, a heuristic method defines any strategy that seeks to find a solution to a problem, which is not necessarily optimal but delivers a sufficient approximation (Bianchi et al., 2009). Well-known examples are the rule of thumb or trial and error. While heuristics are a problem-dependent technique that takes advantage of the problem characteristics, meta-heuristics are problem-independent (Bianchi et al., 2009). Another difference lies in the greediness of the algorithms. A greedy algorithm is defined as an algorithm that makes a distinct optimal choice at each stage, which can cause a reduction of diversity and randomness (Paul E. Black, 2005). Since heuristic techniques are greedier than meta-heuristics, this leads to them tending to get stuck in a local optimum. Meta-heuristics outperform simple heuristics in their computational time and because many meta-heuristics implement some stochastic

optimization with sets of random variables, they are more successful in finding the global optimum (Bianchi et al., 2009).

Evolutionary Algorithms (EA) are the predominant meta-heuristic approach for solving MOOPs (Braun, 2018; Deb, 2001). They are described in an individual chapter. A sub-field of evolutionary algorithms is swarm intelligence. Examples of algorithms that apply swarm intelligence are the Ant Colony Optimization (ACO) and Particle Swarm Optimization (PSO). Moreover, other meta-heuristic approaches to solve MOOP are simulated annealing, tabu search, and extremal optimization.

Hyper-heuristic can be imagined as a process of optimizing the operators of simpler heuristics to solve an optimization problem. A hyper-heuristic itself is inherently a heuristic search method that often embeds machine learning. The idea is to organize and apply simpler heuristics according to their strength and weaknesses and build a system that can handle classes of problems instead of solving merely one problem. While meta-heuristics search in the problem-solution space, hyper-heuristics search in the search space of heuristics. (Burke et al., 2003)

Apart from these methods, other strategies can be used to solve a MOOP. In contrast to DFO, various gradient-descent-based methods exist for MOOPs that provide derivative information (Désidéri, 2012; Giacomini et al., 2014; Montonen et al., 2018; Peitz & Dellnitz, 2018). If uncertainties are inherent to the problem, multiobjective simulation optimization is applied (Hunter et al., 2019). Furthermore, a relatively new field is to achieve MOO through machine learning. Li et al. introduced a deep reinforced learning strategy to solve MOOPs (K. Li et al., 2020).

### 3.4.2 Multiobjective Evolutionary Algorithms

The underlying principle of Multiobjective Evolutionary Algorithms (MOEA) is the use of search agents that collectively estimate the Pareto-optimal front. EAs are bio-inspired meta-heuristic methods capable of solving non-convex numerical optimization problems (Emmerich & Deutz, 2018). They apply principles of Darwin's evolution theory, such as selection, recombination, and mutation to lead the search agents towards the optimal solutions (Emmerich & Deutz, 2018). One must distinguish between single-objective evolutionary algorithms (SOEA) and MOEA. The selection schemes between these two differ because, in contrast to single-objective optimization algorithms, multiobjective optimization algorithms intend to find multiple nondominated optimal solutions in one single optimization run (Deb et al., 2002).

Standard solvers for MOOP are Non-dominated Sorting Genetic Algorithms-II (NSGA-II) (Deb et al., 2002), Strength Pareto Evolutionary Algorithm Version 2 (SPEA2) (Zitzler et al., 2001), Multiobjective selection based on dominated hypervolume (SMS-EMOA) (Beume et al., 2007), and Multiobjective Evolutionary Algorithm Based on Decomposition (MOEA/D) (Emmerich & Deutz, 2018; Qingfu Zhang & Hui Li, 2007).

The main distinction between different classes of MOEA is the method used to define selection operators. Emmerich and Deutz (2018) define three classes of MOEAs, Pareto-based, indicator-based, and decomposition-based:

1. Pareto-based MOEAs (e.g., NSGA-II): These algorithms found on a two-level ranking scheme, where Pareto dominance controls the first ranking and diversity the second.
2. Indicator-based MOEAs (e.g., SMS-EMOA): The indicator-based MOEAs are governed by an indicator that assesses the performance of a set. The indicator is the base of the selection or ranking of an individual.
3. Decomposition-based MOEAs (e.g., MOEA/D, NSGA-III): The underlying theory of these algorithms is to split up the optimization problem into subproblems. Each of these tackle distinct parts of the Pareto front. These algorithms are based on scalarization and weighting. Every subproblem is assigned with a weight that specifies what part of the Pareto front is laid focus on.

#### 3.4.2.1 Genetic Algorithm Theory

Genetic Algorithms (GA), like genetic programming, evolutionary programming, evolution strategy, and others, belong to the category of evolutionary algorithms. I briefly present the method of the standard genetic algorithm to provide fundamental knowledge for the more complex nondominated sorting genetic algorithm (NSGA-II).

Some terms must be defined when discussing genetic algorithms. A standard genetical algorithm is initialized by choosing a randomly selected population within the full range of the search space. These candidate solutions will evolve over certain generations. They are also referred to as chromosomes. With every new generation, the fitness of every individual chromosome is estimated, based on the objective function. The size of the initial population and the offspring is a user-defined parameter, which should correspond to the problem characteristic (Mitchell, 1996). (Baluja & Caruana, 1995)

The new generations are created by applying the concepts of selection, crossover, and mutation. Selection describes the process of choosing the individuals that will create children for the generation (Mitchell, 1996). Crossover defines the merging of information from the selected parents by transferring random characteristics of them to the offspring. Mutation is a principle applied to maintain diversity in the population by adding some random changes to the population. (Baluja & Caruana, 1995)

According to Mitchell (1996), crossover can be interpreted as a variation and innovation tool, whereas mutation prevents permanent fixation on any particular point. While it has been argued about which operator is the most important, the right balance between selection, crossover, and mutation and how it corresponds to the objective function is crucial for success.

### 3.4.2.2 Operator Balance and Parameter Setting

After outlining the basic principle of a standard genetic algorithm, parameter setting strategies, which are crucial to establish a well-designed operator balance, are now discussed. Selection has the purpose to amplify the impact of the fitter individuals. However, it can lead to populations consisting of the local fittest individuals without considering other areas of the search space. A certain diversity needs to be maintained. On the other hand, a too weak selection will cause slow evolution. (Mitchell, 1996)

Apart from the selection, crucial parameters to balance the operators are population size, crossover rate, and mutation rate. Unfortunately, there are no conclusive reports on the effectiveness of parameter settings (Mitchell, 1996). In 1986, a GA was applied to find the optimal parameter settings of another GA (Grefenstette, 1986). The results showed an optimal performance when setting population size to 30, crossover rate to 0.95, the mutation rate to 0.01, and using an elitist selection. Despite its popularity, this setting is not optimal for all fitness functions (Mitchell, 1996). Schaffer et al. (1989) proved the problem-independence of their optimal parameter setting and suggested a population size of 20-30, crossover rate of 0.75-0.95, and a mutation rate of 0.005-0.01. However, the common view on EA is that distinct problems require specific EA parameter setups for good performances (Lobo et al., 2007).

This is the reason why in more recent years, the research has been shifted towards adaptive parameter settings. Lobo et al. (2007) name the two forms of parameter setting “parameter tuning” and “parameter control”. Parameter tuning describes what I have discussed until now: the search for good parameter values before the run of the algorithm. This can be very computationally expensive if it is done by experimenting (Lobo et al., 2007). Parameter control on the other side is done during the run. The approaches here can be split up into deterministic, adaptive, and self-adaptive. In this context, deterministic means that the value of a parameter is changed by some deterministic rule that is predefined and does not take any intermediate outputs into account. Adaptive parameter control is defined by some feedback-based function that uses the information of the current state to adjust the parameter values. Self-adaptive control is built upon the idea of evolution, meaning that the parameters themselves undergo an optimization process with mutation and crossover and are embedded within the encoding of the chromosomes. (Lobo et al., 2007)

### 3.4.2.3 Operators

Over the years, vast amounts of operator methods have been established to solve issues of operator balances. Among others, developed selection strategies include Roulette Wheel Selection (RWS), stochastic universal sampling (SUS), linear rank selection (LRS), sigma scaling, Boltzmann selection, exponential rank selection (ERS), tournament selection (TOS), and truncation selection (TRS) (Jebari & Madiafi, 2013; Mitchell, 1996). Similarly, the number of crossover methods has increased drastically. Simulated binary crossover (SBX), 1-point crossover, K-point crossover, reduced surrogate crossover,

shuffle crossover, average crossover, discrete crossover, flat crossover, uniform crossover, half uniform crossover, exponential crossover, and differential crossover are just a few to mention in a much wider spectrum, summarized by Umbarkar and Sheth (2015). Popular mutation methods are mirror mutation, binary bit-flipping mutation, random uniform mutation, directed mutation, and polynomial mutation (Lim et al., 2017). The number here is comparatively high and corresponding review papers are recommended for further details (Lim et al., 2017). The operators that are used in the implemented algorithm are described in this chapter.

### *Simulated Binary Crossover:*

Simulated Binary Crossover (SBX) is one popular recombination approach for real-value parameters. It was originally introduced by Deb and Agrawal (1995). Deb et al. (2007) extended the operator to a self-adaptive form. First, I will explain the original operator before I refer to self-adaptive SBX that was used within the NSGA-II.

The SBX method uses two parent vectors and a so-called blending operator to produce two offspring solutions (Deb & Agrawal, 1995). A crucial parameter in this method is the fixed distribution index  $\eta_c$ . A large distribution index value will lead to a resulting offspring that is similar to the parent, whereas a smaller value causes the offspring to differ more strongly from the parents (Deb & Agrawal, 1995). The computation of this method is described as follows.

A spreading factor  $\beta_i$  is defined as a measure for the spreading of the parents and the children:

$$\beta_i = \left| \frac{x_i^{(2,t+1)} - x_i^{(1,t+1)}}{x_i^{(2,t)} - x_i^{(1,t)}} \right| \quad (22)$$

Here, the denominator shows the difference of the offspring solutions, while the nominator represents the difference of the parents. As a first step, a random number  $u_i$  between zero and one is created. A specific probability distribution that is used to create offspring is given as follows (Deb & Agrawal, 1995):

$$P(\beta_i) = \begin{cases} 0.5(\eta_c + 1)\beta_i^{\eta_c}, & \beta_i \leq 1 \\ 0.5(\eta_c + 1)\frac{1}{\beta_i^{\eta_c+2}}, & \text{otherwise} \end{cases} \quad (23)$$

where

$\eta_c$  = distribution index.

$\beta_i$  = spread factor.

Based on this probability distribution, an ordinate  $\beta_{q_i}$  is estimated that sets the area under the probability curve from zero to  $\beta_{q_i}$  to the chosen  $u_i$ . The offspring is then calculated with the following equation (K. Deb et al., 2007):

$$x_i^{(1,t+1)} = 0.5 \left[ (1 + \beta_{q_i}) x_i^{(1,t)} + (1 - \beta_{q_i}) x_i^{(2,t)} \right] \quad (24)$$

$$x_i^{(2,t+1)} = 0.5 \left[ (1 - \beta_{q_i}) x_i^{(1,t)} + (1 + \beta_{q_i}) x_i^{(2,t)} \right] \quad (25)$$

where

$\beta_{q_i}$  = ordinate value that sets the area under the probability curve from zero to  $\beta_{q_i}$  to the chosen  $u_i$ .

Figure 10 clarifies the influence of the distribution index. The higher  $\eta$ , the more likely offspring is close to the parent value of 0.2 and 0.8. Setting the index close to zero will lead to a random distribution.

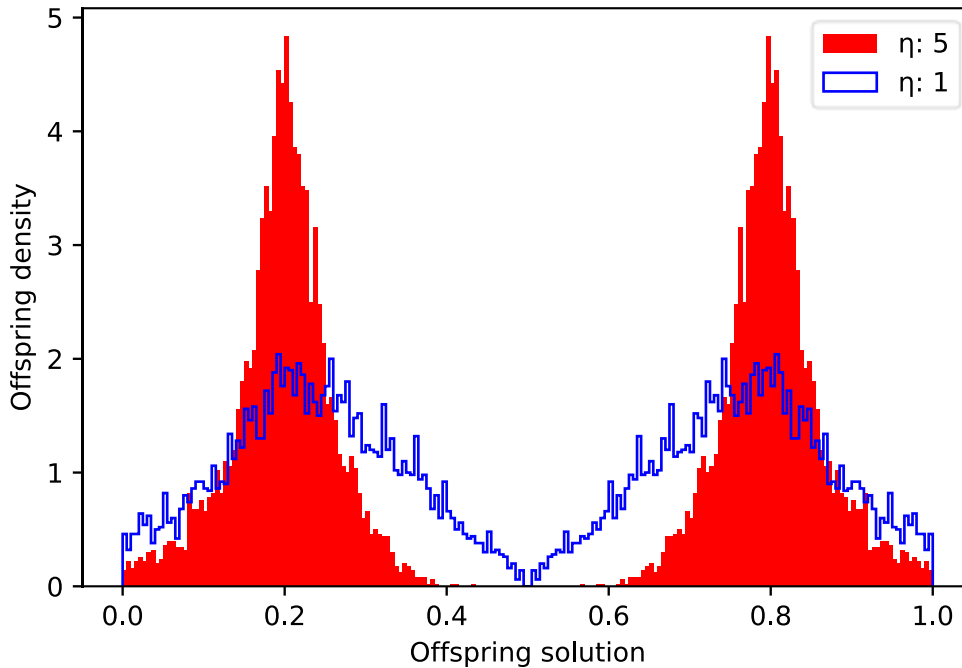


Figure 10: Influence of distribution index on the offspring probability density.  
 $\eta$ : distribution index.

In conclusion, the SBX operator favours solutions near to the parents, because of the probability density. Furthermore, the spread between the offspring solutions is proportional to the spread of the parent solution (Deb et al., 2007). The consequence is that, once the solutions converge towards an optimum, more distant solutions are neglected (Deb et al., 2007). Hence, it was implied that the SBX operator is not satisfactory alone to solve complex, large set problems.

In the self-adaptive version of the operator, a procedure is introduced to update the distribution index based on the extension-contraction concept. This means that if the offspring has better objective results than the parent, the child solution is embraced by increasing the distribution index whilst maintaining the same probability (extension). If worse results occur, the distribution index will be decreased. (Deb et al., 2007)

#### *Polynomial Mutation:*

This mutation operator was established in 1996 by Deb and Goyal (Deb & Goyal, 1996). The name polynomial mutation (PM) is based on the polynomial probability distribution in the method. As in SBX, PM also relies on a user-defined distribution index  $\eta$ . Additionally, a perturbation factor  $d$  is used (Deb & Goyal, 1996):



$$\delta = \frac{c-p}{\Delta max}. \quad (26)$$

where

$c$  = mutated offspring value.

$p$  = parent value.

$\Delta max$  = maximum allowed perturbation in the parent value.

$\delta$  = perturbation factor.

Following the principle of the crossover operator SBX, a probability distribution is applied, which is dependent on the distribution index and the perturbation factor (Deb & Goyal, 1996):

$$P(\delta) = 0.5(\eta + 1)(1 - |\delta|)^{\eta}. \quad (27)$$

where

$\delta$  = perturbation factor.

$\eta$  = distribution index.

A random value  $u$  within the range zero to one is generated. The perturbation factor is calculated with this value: (Deb & Goyal, 1996)

$$\bar{\delta} = \begin{cases} (2u)^{\frac{1}{\eta+1}} - 1, & u < 0.5. \\ 1 - (2(1-u))^{\frac{1}{\eta+1}}, & u \geq 0.5. \end{cases} \quad (28)$$

where

$u$  = randomly generated number in range 0 - 1.

$\eta$  = distribution index.

$\bar{\delta}$  = perturbation factor.

The mutated child value  $c$  is estimated by transforming Equation 26 to: (Deb & Goyal, 1996)

$$c = p + \bar{\delta} * \Delta max. \quad (29)$$

#### 3.4.2.4 Elitism

Elitism is an important principle to be able to understand and compare MOEA. Elitism ensures that a number of optimal individuals are preserved, which otherwise could be lost because they are not selected to reproduce or if they are corrupted by crossover or mutation effects (Mitchell, 1996). It can be understood as a storage method that guarantees that solution quality cannot decrease from generation to generation (Grosan et al., 2003). Different modalities are applied by algorithms to make use of elitism. SPEA2 is an example of an elitist EA. The algorithm stores all nondominated solutions found in an external archive (Zitzler et al., 2001).

#### 3.4.2.5 Applicability of Genetic Algorithms

There is no straight answer to the question when genetic algorithms are a good method to be used. Nevertheless, most researchers share the intuitive opinion that GAs are a suitable method if the problem space is large and known not to be perfectly smooth and unimodal. Furthermore, it deems applicable for noisy fitness functions, to find local optima or if the solution space is not well understood. In case the space of the problem is well understood, search methods that use domain-specific heuristics may produce outputs that surpass the solution of a GA. (Mitchell, 1996)

Even though these intuitions may be helpful, the performance will depend on the implementation details itself. The success depends on the chosen algorithm, the operators, the parameter settings, and the termination. (Mitchell, 1996)

#### 3.4.2.6 NSGA-II

The non-dominated sorting algorithm (NSGA), developed in 1994, was one of the first MOEA algorithms (Srinivas & Deb, 1994). However, some aspects were criticised, which eventually led to the establishment of a more elaborated approach that was named NSGA-II. It tackles the issues of NSGA, which are high computational complexity of non-dominated sorting, lack of elitism, and need for specifying the sharing parameter. Furthermore, with NSGA-II Deb et al. introduced a technique to embed constraints into the optimization paradigm. (Deb et al., 2002)

NSGA-II falls into the previous mentioned Pareto-based MOEAs category. It follows the general outline of a genetic algorithm with a modification in mating and survival selection. The algorithm's procedure is the following (Deb et al., 2002; Emmerich & Deutz, 2018):

The first step is the random initialization of a population of points. This first population is sorted based on non-domination, also called non-dominated sorting. Every solution receives a rank corresponding to its non-domination. Then variation operators are applied to create an offspring population. The procedure differs for subsequent steps of the algorithm because elitism is introduced by always merging previous populations with the offspring population.

A loop is started consisting of two parts. In the first, the population is subjected to variation. Then a selection process is conducted in the second part, which leads to a new generation-population. The loop is run until a certain criterion is reached, which can be generation amounts, a time limit, or a convergence criterion.

The variation part consists of the generation of offspring based on two parents. This is achieved through binary tournament selection. The simulated binary crossover was suggested to recombine the parents (Deb et al., 2002). However, the self-adaptive simulated binary crossover is a more elaborate option today and was therefore used in this work's optimization framework (Blank & Deb, 2020). As mutation strategy, a polynomial mutation is used. After the variation, the offspring population is merged with the parent generation and the second part of the loop is started.

The selection part is what makes the NSGA-II special. The ranking process comprises two parts: Non-dominated sorting and crowding distance sorting, which is depicted in Figure 11.  $P_t$  is the parent generation and  $Q_t$  the offspring that are both merged into  $R_t$ . The goal is to select a new generation  $P_{t+1}$  of the same size as the parent population.

The non-dominated sorting works as follows. Two parameters are estimated for each individual: the domination count, which provides the information of how many solutions dominate the individual, and a list of the set of solutions that are dominated by the individual. This method splits up all solutions into different fronts. In Figure 11,  $F_{1-3}$  are the fronts that are obtained by the sorting process. First, all individuals are compared with each other. The first front will comprise only solutions with a domination count of 0. From there, the algorithm continues going individual by individual through all sets of solutions of the first front. The domination count of the individuals that are found in the sets is reduced by one for every appearance in a set. After this process, all the individuals that have a domination count of zero, excluding the first front solutions, will form the second front. The procedure is continued until the last front is obtained.

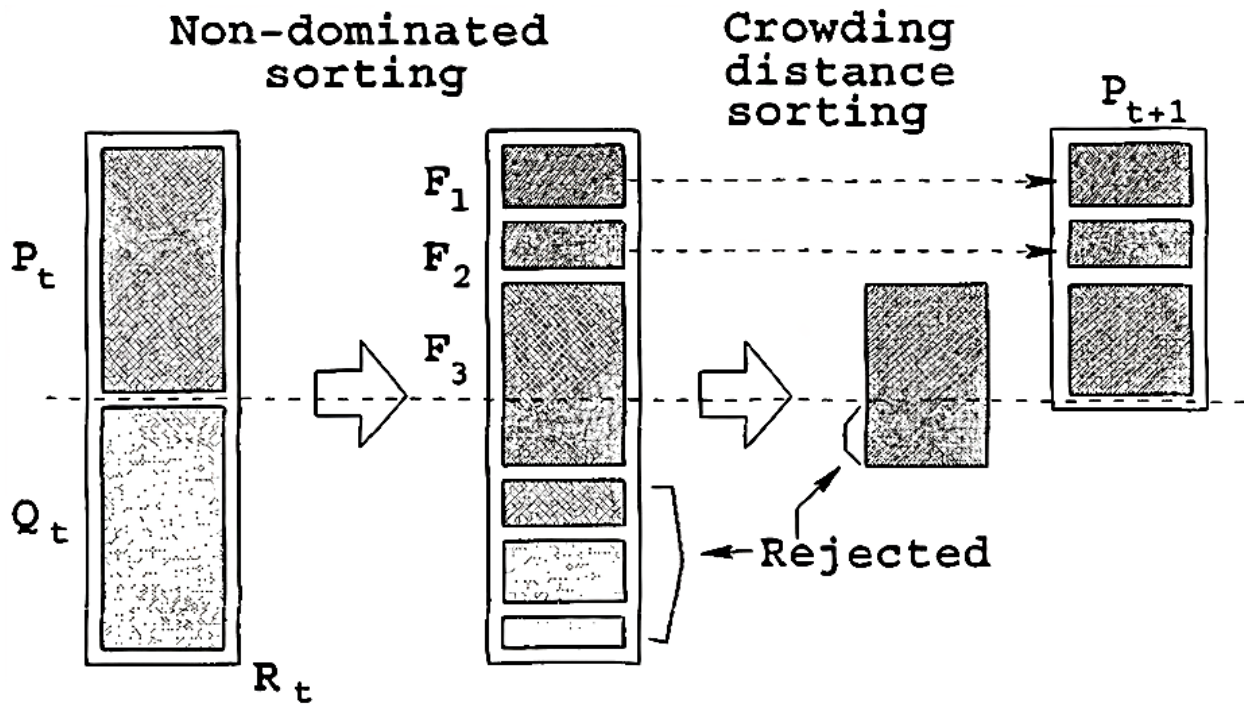


Figure 11: Ranking procedure of NSGA-II. (Deb et al., 2002)

After obtaining the fronts, crowding distance sorting is applied. The crowding distance is a solution density estimation that implies how closely other solutions are surrounding an individual. The nearest neighbours are used to calculate the average distance between the closest solutions of the same front. The procedure is the following: (Deb et al., 2002)

First, all distances of the front are set to zero. Then a loop through all the objectives is initialized. Within the loop, the solutions are first sorted according to the objective function values in ascending magnitude and a distance value of infinite is assigned to the maximum and minimum values of the solution set. The values  $f_m^{max}$  and  $f_m^{min}$  are the maximum and minimum values of the  $m$ -th objective function. By adding up the distances of  $S[i]$ , the overall crowding distance of  $S[i]$  will be the sum of individual objective solution distances. The pseudo-code of the computational estimation of the crowding distance is given below:

Algorithm 1: Estimation of crowding distance.

---

$l =$  number of solutions

for  $i$ : set  $S[i]_{distance} = 0$

for each objective  $m$ :

$S = \text{sort}(S, m)$

$S[0]_{distance} = S[l]_{distance} = \text{inf}$

for  $i = 1$  to  $(l - 1)$ :

$$S[i]_{distance} = S[i]_{distance} + \frac{S[i+1]_m - S[i-1]_m}{f_m^{\max} - f_m^{\min}}$$


---

Figure 12 shows a visual example of the crowding distance calculation for a problem with two objectives. The crowding distance of the solution  $i$  is the average side length of the cuboid depicted as a dashed box. (Deb et al., 2002)

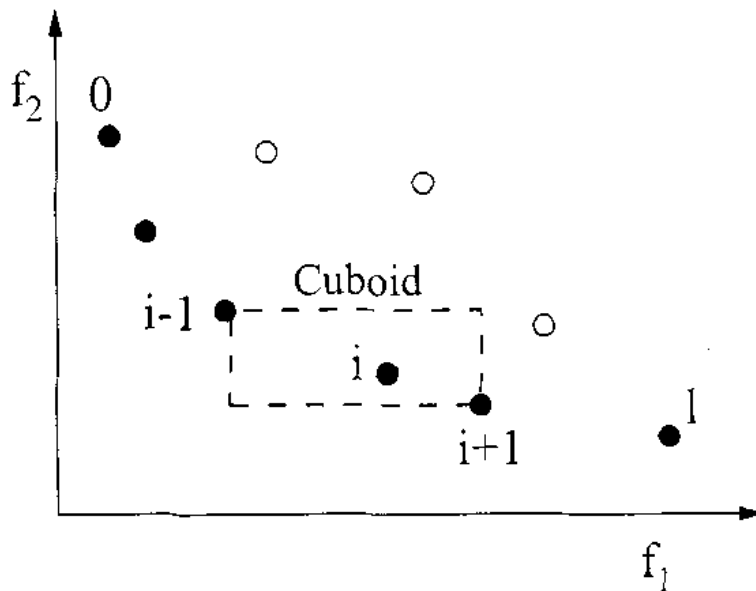


Figure 12: Estimation schema of the crowding distance in a 2-dimensional objective space. Filled circles are solutions of the same nondominated front. The axis  $f_1$  and  $f_2$  are the objective function values. (Deb et al., 2002)

After receiving the selection parameters, which are the nondomination rank (front number) and the crowding distance, the new population  $P_{t+1}$  can be formed. This is done with the crowded-comparison operator, which fundamentally sets the preferences of solutions: the

dominant parameter is the nondomination rank. Thus, solutions with a better rank are selected for the new generation. Only if two solutions are within the same front, the crowding distance is considered and the lesser crowded individual is chosen. (Deb et al., 2002)

With the selection part, the main loop of the algorithm is complete. The working scheme of the NSGA-II algorithm is shown below: (Emmerich & Deutz, 2018)

*Algorithm 2: Non-dominated Sorting Genetic Algorithm II (NSGA-II).*

---

```

initialize population  $P_0 \subset \mathcal{X}^\mu$ 
while not terminate do
  {Begin variate}
   $Q_t \leftarrow \emptyset$ 
  for all  $i \in \{1, \dots, \mu\}$  do
     $(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}) \leftarrow \text{select\_mates}(P_t)$  {select two parent individuals  $\mathbf{x}^{(1)} \in P_t$  and  $\mathbf{x}^{(2)} \in P_t$ }
     $\mathbf{r}_t^{(i)} \leftarrow \text{recombine}(\mathbf{x}^{(1)}, \mathbf{x}^{(2)})$ 
     $\mathbf{q}_t^{(i)} \leftarrow \text{mutate}(\mathbf{r})$ 
     $Q_t \leftarrow Q_t \cup \{\mathbf{q}_t^{(i)}\}$ 
  end for
  {End variate}
  {Selection step, select  $\mu$ -"best" out of  $(P_t \cup Q_t)$  by a two step procedure:}
   $(R_1, \dots, R_\ell) \leftarrow \text{non-dom\_sort}(\mathbf{f}, P_t \cup Q_t)$ 
  Find the element of the partition,  $R_{i_\mu}$ , for which the sum of the cardinalities  $|R_1| + \dots + |R_{i_\mu}|$  is for the first time  $\geq \mu$ . If  $|R_1| + \dots + |R_{i_\mu}| = \mu$ ,  $P_{t+1} \leftarrow \cup_{i=1}^{i_\mu} R_i$ , otherwise determine set  $H$  containing  $\mu - (|R_1| + \dots + |R_{i_\mu-1}|)$  elements from  $R_{i_\mu}$  with the highest crowding distance and  $P_{t+1} \leftarrow (\cup_{i=1}^{i_\mu-1} R_i) \cup H$ .
  {End of selection step.}
   $t \leftarrow t + 1$ 
end while
return  $P_t$ 

```

---

## 3.5 Python Libraries

---

In this section, I briefly want to present the most important code libraries that are used in the implemented Python code. An overview of the code itself is given in Section 4.3.

*Mikeio* (<https://github.com/DHI/mikeio>)

This library, which was developed and is further improved by DHI, represents a MIKE SHE model wrapper. It enables Python developers to interact with the model, set parameters, and get outputs. In this work, the library is used to update and DFS files, which is relevant for the weather forecast data and the simulation results.

*Pfsreader* (<https://github.com/red5alex/pfsreader>)

As mentioned in 3.2, the MIKE SHE model setup is fundamentally a PFS-file. Unfortunately, *mikeio* does not implement functionality yet to interact with this file. Thus, certain model parameter, mostly when creating the model, cannot be adjusted. Therefore, another library was developed within DHI WASY to deal with this issue. The *pfsreader* library is used to assign setup parameters to the MIKE SHE models.

*Pymoo* (<https://pymoo.org>)

The *pymoo* multiobjective optimization framework for Python was developed by Blank & Deb (2020). K. Deb is also the main developer of the NSGA-II algorithm and the self-adaptive SBX (Deb et al., 2002, 2007). The *pymoo* optimization framework offers implemented algorithms, operators, visualizations, and test problems. It is especially effective for the optimization framework of this thesis because it is easily extendable and customizable (Blank & Deb, 2020).

## 4 Implementation

---

In this chapter, I explain the specifics of the chosen optimization approach and how it was implemented. A short overview is given before going into detail in the individual sections.

In the literature review, many objectives were discussed that have been applied in the past in the context of irrigation optimization. Various aspects were picked out to develop a novel optimization framework. An overview of this framework is shown in Figure 13. As discussed before, the MIKE SHE software was used as a system model to predict soil moisture content. The optimization approach is, thus, based on a physics-based model with no derivative information available. Because uncertainties are neglected, a deterministic derivative-free optimization is a suitable approach in this scenario (compare Table 3).

Two objective functions were chosen, which makes the irrigation scheduling a multiobjective optimization problem. On the one side, one objective function aims to keep water consumption at a minimum. On the other side, optimal soil moisture conditions in the root zone are to be maintained, which is addressed by another objective function. The Pareto optimal front is computed with the multiobjective evolutionary algorithm NSGA-II. Moreover, an *a posteriori* decision-making strategy was chosen. According to preferences, the optimal solution is obtained by applying either the weighted-sum method or by computing the knee point solution.

Many researchers focused on seasonal optimization, considering growth stages and phenological aspects. Seasonal planning can result in misleading strategies because of uncertainties in long-term weather forecasts. Hence, I decided to aim at optimizing over a shorter time horizon in alignment with reliable weather forecast data. Nevertheless, seasonal factors are included in the method. Using the MIKE SHE software offers huge advantages here. On the one side, soil and crop attributes may be adjusted over the season, possibly through measurements or sensor data, thus improving the accuracy of the model. On the other side, parameters such as root development, LAI development, water stress, actual evapotranspiration, etc. are estimated as part of the MIKE SHE simulation runs. In addition to the soil moisture data, this data offers further options for the objective functions and the general optimization approach. For instance, the root depth output is established in the objective function to define in what depth the soil moisture conditions are optimized and the estimations of  $ET_a$  and  $ET_c$  are used to develop a water stress indicator. The optimization approach is further brought into a seasonal context by putting seasonal specific weights on the objectives. According to the crop growth stage and the water stress sensitivity during the corresponding stage, either saving water or ensuring optimal soil moisture condition is prioritized in the weighted-sum *a posteriori* decision-making process. For example, due to the high sensitivity of maize during the flowering period, the weight in the weighted-sum method is put on maintaining optimal soil moisture conditions during this time.

In general, the focus was laid on single-field optimization. However, an extended multi-field approach was implemented and tested for a two-field scenario. The idea is to have



one model for every individual field or crop. By adjusting the control variables and weighting the objective function outputs of the individual models *a priori*, a multifield-optimization has been successfully established. More detailed information on the multifield approach is given in Subsection 4.2.4.

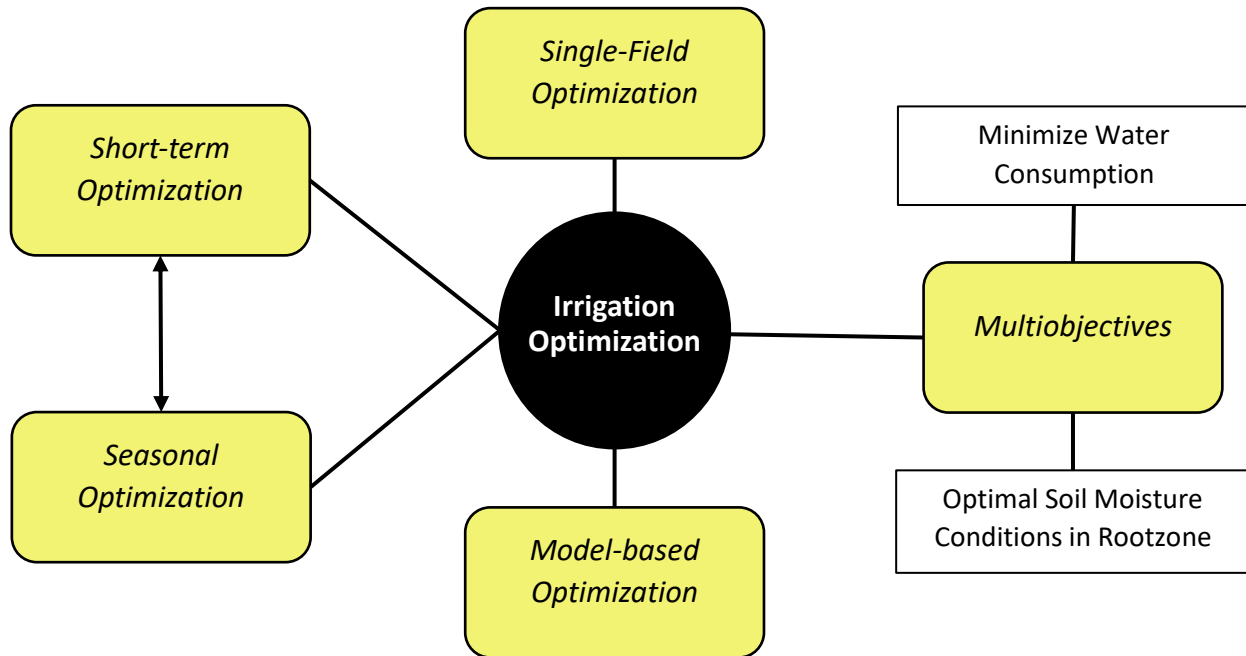


Figure 13: Overview implemented optimization framework.

Before talking about the optimization, the black-box MIKE SHE system model is described. Second, the optimization framework with an analysis of the system behaviour, single-field optimization, objective functions, and multi-field optimization is explained. Last, the structure of the implemented code is briefly presented.

## 4.1 MIKE SHE Model

---

In this section, I will explain the parameter and the general setup of the test model that was created to validate the optimization. This also serves as a general example on how to set up a MIKE SHE models for the optimization purpose in other locations. An overview of the crucial model parameters is given in Table 4. It is important to stress that all these parameters can easily be changed and adapted for other scenarios. The model is completely decoupled from the optimization itself and could be theoretically replaced by a different system model that predicts soil moisture content.

Table 4: Overview setup of the physics-based system model created in MIKE SHE.

<b>Model Parameter</b>	
Simulation Period	7 days
Simulation Timestep	6 min
Vegetation Start Date	1st of May
Groundwater Table	4.1 m
Irrigation Type	Sprinkler
Soil Type	Brown Earth
Crop Type	Maize
Initial Conditions	Flexible - based on excel sheet

<b>Climate Inputs:</b>	
Reference Evapotranspiration	Penman-Monteith based on DWD weather data
Precipitation Rate	DWD weather data

<b>Discretization:</b>	
Depth	Cell height
0 - 0.2	0.05
0.2 - 1.0	0.1
1.0 - 2.0	0.25
2.0 - 10	0.5

A one-week optimization horizon was chosen to test optimization in alignment with sufficiently reliable weather forecasts. At the same time, this is the simulation period because in the objective function all simulation outputs within the simulation period are evaluated. The longer the simulation period is set, the higher the uncertainty in the optimization due to the increasing uncertainty of the weather forecast. Furthermore, the chosen framework only optimizes the next best irrigation event. Therefore, the simulation

period must be adjusted to how the physics-based system reacts to an event, which means how fast the water infiltrates and distributes in the soil. The infiltration velocity depends highly on soil characteristics. This must be kept in mind when choosing the optimization horizon in cases of different soils with deviating hydraulic conductivities.

Two irrigation methods were tested in this work: sprinkler and drip irrigation. Furthermore, brown earth (German: “Braunerde”) was selected as a test soil. The Van Genuchten soil-water retention curve for brown earth was computed based on the soil database of MIKE SHE and is plotted in Figure 14. The saturation is reached at a volumetric soil moisture content of 38.5 %. Using the default field capacity matric potential of MIKE SHE (-0.1 bar), it was computed that FC is reached at 16.54 %, while the permanent wilting point starts from 2.65 % volumetric water content at a matric potential of -15 bar. This means that according to Equation 3, TAW is estimated to be 13.89 cm per meter root depth.

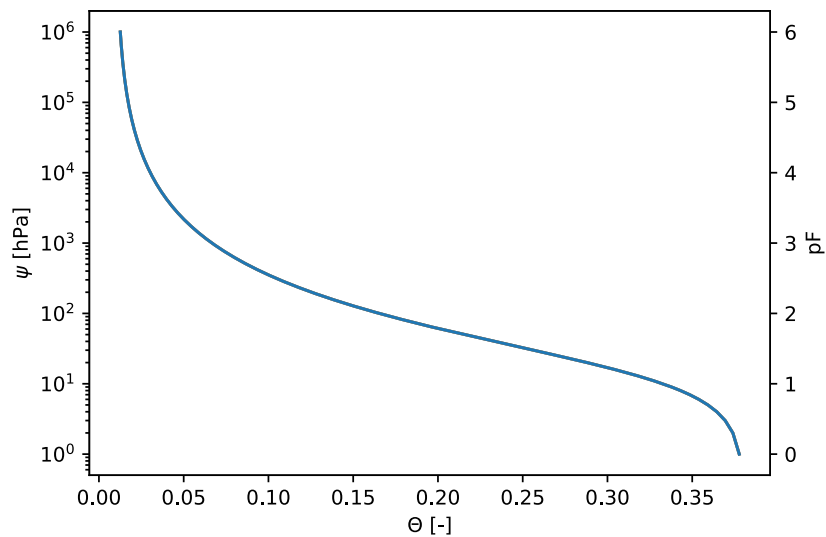


Figure 14: Van Genuchten soil-water retention curve of soil type: brown earth.

Maize was the crop of choice in the optimization test scenario. The phenology of maize was discussed in detail in Subsection 3.1.6. In Figure 15, the development of the plant is plotted in the form of root depth and LAI development. These progressions are predefined by the MIKE SHE database itself but could be manually adjusted.

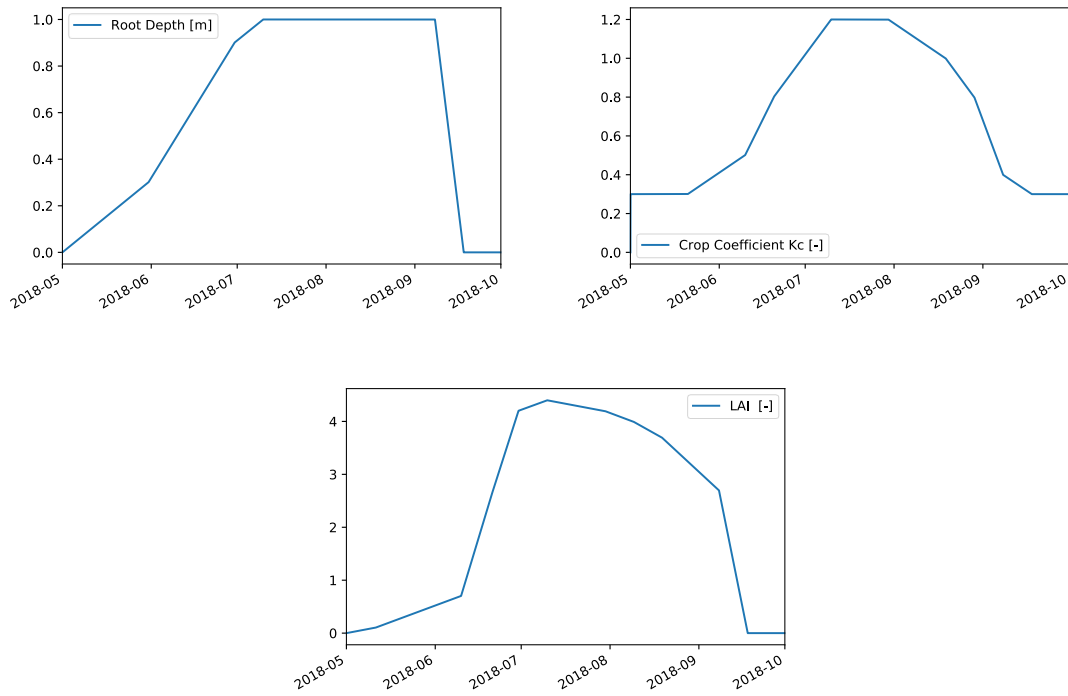


Figure 15: Root depth/LAI/crop coefficient development of maize over a crop season computed with MIKE SHE.

The soil depth is discretised according to Table 4. In the upper part, the resolution is chosen higher because of a stronger variation of moisture content. I implemented a Microsoft Excel spreadsheet to assign the initial soil moisture content. Measurements in any depths can be filled in and are linearly interpolated to fit the discretization of the model.

The input weather data is taken from the German meteorological service (DWD). They provide weather forecast for ten days in their Model Output Statistics-Mix (MOSMIX). This data can be downloaded from their server (<https://opendata.dwd.de/>) manually. This process has been automated for the optimization framework. Furthermore, historical data was downloaded from their server for the testing and validation of the optimization. Based on this data, the Penman-Monteith method was applied to estimate the hourly reference evapotranspiration.

## 4.2 Optimization

---

### 4.2.1 Analysis of the Optimization Problem

The implementation of a successful optimization framework depends on various factors, such as scientific expertise, knowledge of customer demands, and knowledge of operational and general limitations. The behaviour of the system must be well understood to choose the right approach and adapt the optimization to the system. Therefore, I will discuss the behaviour of the system model and then the optimization problem in this section.

Figure 16 shows the results of a seasonal simulation in 2018. No irrigation was set in this scenario and it was expected that at the time of seeding, the soil moisture content in the root zone was close to field capacity, representing an ideal scenario. Further details on this simulation run are given in Appendix A.1.

I observed that the soil moisture varies most in the top node in 5 cm depth, which is due to the soil evaporation that is estimated solely in the upper node (see 3.2.2). With increasing depth, the soil moisture lines are smoother. At the beginning immediately after seeding, the soil moisture content remains relatively stable, decreasing mainly due to soil evaporation. As soon as the roots and leaves begin to develop, a general drop of the soil moisture content is observed and the evaporation ratio shifts from soil evaporation to transpiration, as elaborated in Figure 3. Having a closer look at the average water content in the root zone, it is noticeable that even though the soil water content drops in the upper root zone, the average root zone soil moisture content is equalized. This is caused by the root growth of the plant. In the first stages, the soil remains close to field capacity below the root zone. With further expansion of the roots, this water is available.

Once the leaves and roots are fully developed in July, the soil water content in 5 cm depth has reached the permanent wilting point. Since now the roots are not expanding into new depths where water is still untouched, the overall water content in the root zone starts dropping. Furthermore, soil evaporation is decreased because of the LAI reaching its maximum, which leads to canopy interception.

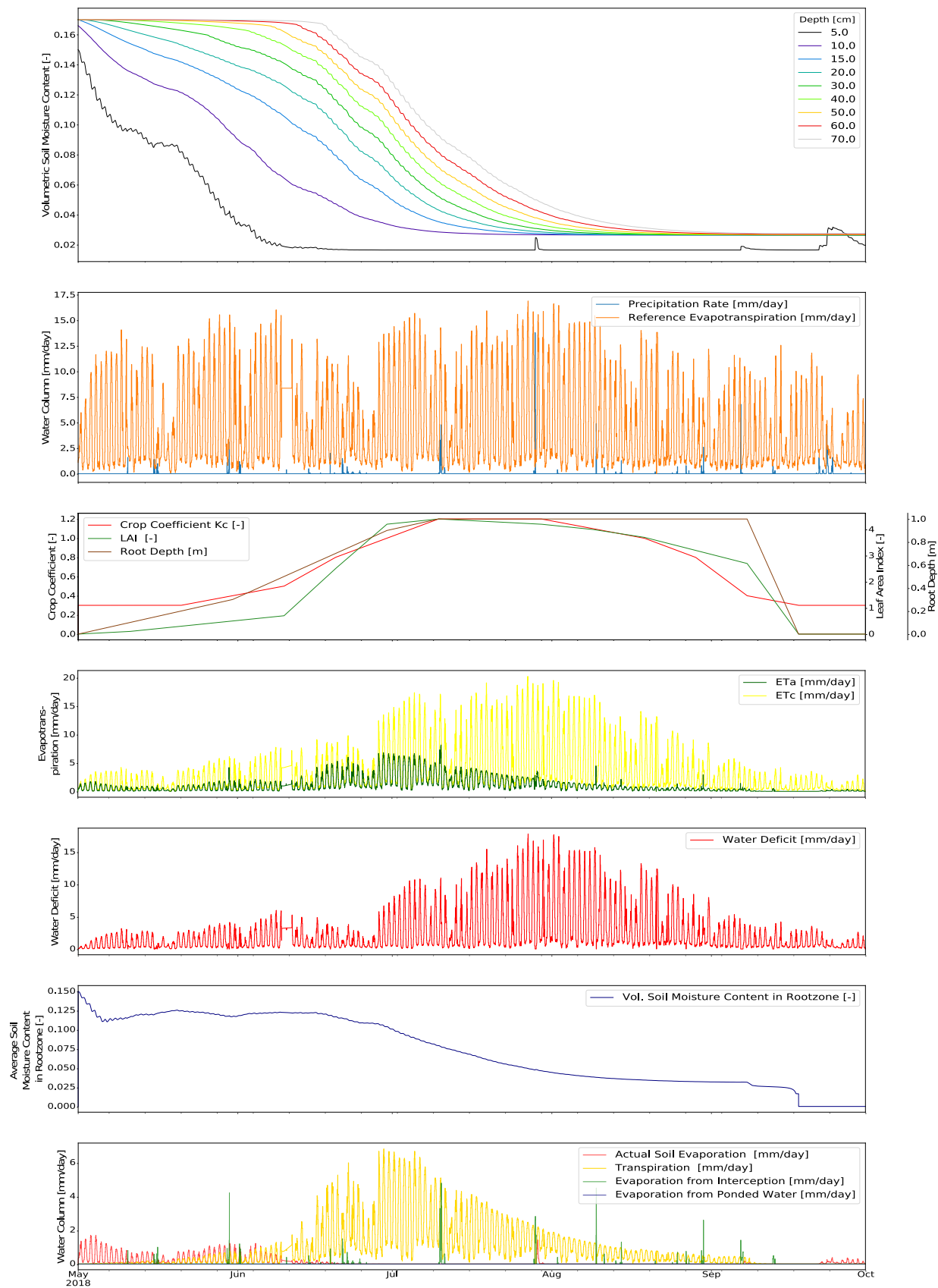


Figure 16: Analysis system model - 2018 seasonal simulation without irrigation. Initial conditions were set approximately to field capacity.

The water stress can be expressed in form of root zone water deficit, which is the difference between  $ET_c$  and  $ET_a$ , also shown in

Figure 16. The deficit is mainly found during the flowering and in mid-season once the water stored in the soil is exhausted and when the crop has developed all its roots and leaves. At this point in late June, irrigation becomes crucial to assure optimal growing conditions, especially because maize is sensitive to water stress during flowering. The total water deficit from seeding until the 1<sup>st</sup> of October lies at 363 mm. In other words, the maize requires this amount of water for optimal yield, which must be added in form of irrigation. By considering seasonal sensitivities and specific crop growing strategies, the allocation of the water can be optimized. As I have discussed the system's behaviour without irrigation, I now want to examine how the system reacts to different irrigation events.

First, I compared the soil moisture contents of an irrigation event that occurs during a precipitation event to a preceding irrigation event. The results are shown in Figure 17 and indicate that asynchronous irrigation leads to water reaching deeper into the soil, whereas a simultaneous irrigation event shows higher soil moisture contents in the upper soil. The whole analysis with the model parameters is in Appendix A.2.

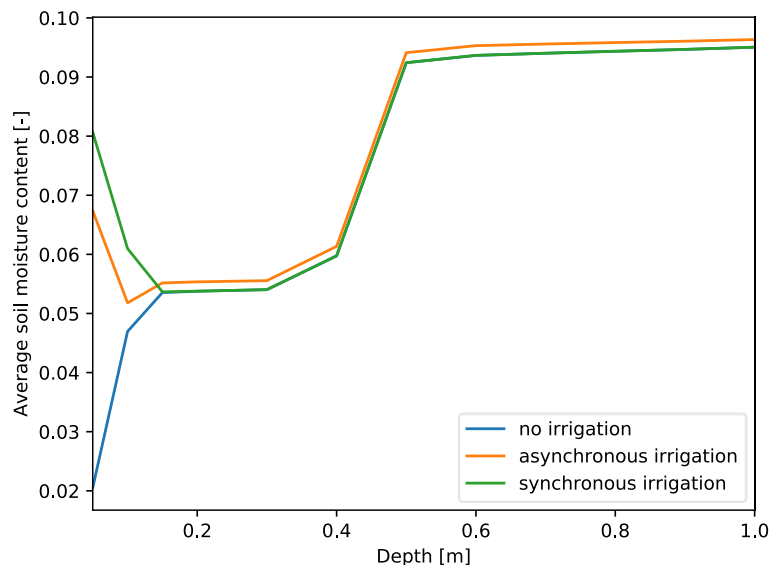


Figure 17: Comparison of irrigation events that occur synchronous vs. asynchronous to a precipitation event. Y-axis: average soil moisture for a weekly prediction, X-axis: depth below the soil surface.

At the same time, this implies that gradual irrigation instead of short-term flooding should have the same effect. To prove this, I conducted three test runs: one scenario where no irrigation method was applied, another wherein a short time a lot of water is introduced into the field, and third, a long-term irrigation event over a period 4 times larger than the short-term scenario. The total amount of water in the irrigation scenarios was kept at the

same level. The results are given in Figure 18 and show that indeed the long-term irrigation leads to a better distribution of water in the ground. The parameter settings and detailed plots of the test are in Appendix A.3.

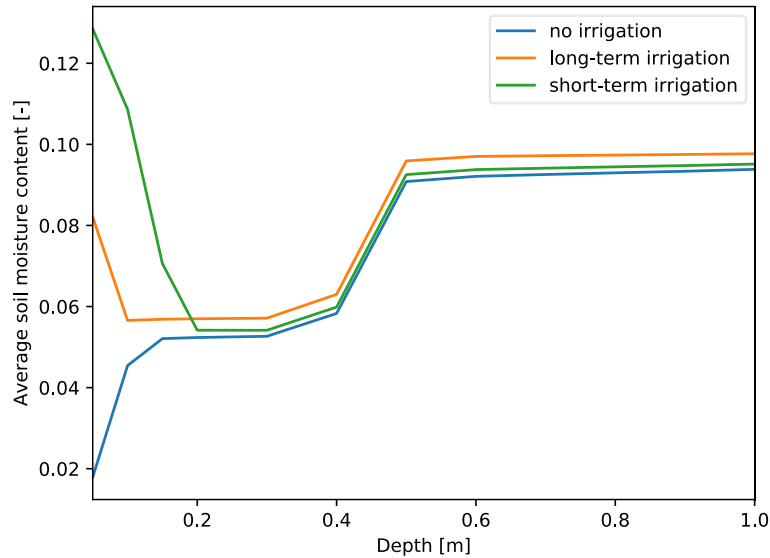


Figure 18: Comparison of long-term vs. short-term irrigation as average soil moisture content over depth. y-axis: average soil moisture for a weekly prediction, x-axis: depth below the soil surface.

After giving an overview of the system's behaviour, I will now talk about further aspects of the optimization problem. A general question is how to deal with seasonal characteristics. One issue here is how to address the initial stage, where no leaves and roots exist, yet. The soil evaporation is high during this stage and unnecessary water losses may occur. Furthermore, a problem that was observed during the mid-season is that it is not possible to allocate water in a way that deeper soil parts of the root zone are replenished without exceeding the upper water stress constraint in the root zone close to surface. Hence, during this time the soil moisture levels cannot be maintained at an optimal level in all depths. It is a trade-off between placing the top roots under stress due to excessive water or allowing water depletion in the bottom root zone.

The impacts of irrigation over a longer period than a week are hard to qualify but are crucial for the irrigation schedule. One of my research objectives was to analyse if it is more efficient to fill up the water storage before the season starts or if more water is stored and effectively used by applying weekly irrigation. This analysis is presented in 5.1.4.

Other factors that influence the optimization paradigm are operational limits. How often can the farmer water one field in a week and how long is the duration of an irrigation period? Furthermore, the optimization depends on the available pump and irrigation system on site. A solar pump will show fluctuating pumping rates depending on solar radiation and limits the irrigation to daytime. It is favourable to make the optimization



parameters flexible so that the framework can be scaled and customized to distinct preferences. This was achieved by applying an objective-oriented code structure and generally decoupling the optimization from the MIKE SHE software. The parameters can be easily adjusted, and constraints can be set individually. This is further elaborated in Subsection 4.2.3.

#### 4.2.2 Objective Functions

As discussed in the theoretical background in Section 3.3, an optimization problem is solved by minimizing one or in the context of MOO multiple objective functions. In the context of irrigation, the objective functions must inherit all necessary irrigation parameters that require evaluation for optimal scheduling. The main goal of this optimization framework was to ensure optimal growth conditions under limited water supply. Therefore, the parameters soil moisture, root zone depth, sensitivity to water stress, and water consumption are considered. The objectives were split up into two functions, one addressing the water use and the other the optimal moisture conditions in the soil. The water consumption objective function is the following:

$$\min_{Q, IP \in \Theta} f(Q, IP) = \frac{Q * IP}{Q_{max} * t} \quad (30)$$

where

$f(Q, IP)$  = water consumption objective function.

$IP$  = Irrigation duration period.

$Q$  = irrigation rate.

$Q_{max}$  = maximum possible irrigation rate.

$t$  = total amount of timesteps in the simulation.

This objective function simply quantifies the water consumption in the form of a linear relationship between irrigation duration period and irrigation rate. The amount is further normalized to keep it in an interpretable range between 0 and 1.

The objective of optimal conditions is formed by implementing a function that comprises the resulting soil moisture data from a whole simulation run. Every estimated soil moisture value of the simulation period within the root zone is evaluated by taking the difference from the predefined optimal value. Following the progressive-barrier method, the function is built with the cost quadratically increasing with the said difference. This way, soil moisture value results that are further away from the optimum are penalized to a higher

extent. The root depth itself depends on the growth stage and is automatically adjusted based on the MIKE SHE output. The equation is as follows:

$$\min_{\theta \in \Theta} f(\theta) = \frac{\sqrt{\sum_t \sum_i^n f_i(\theta_{i,t} - \theta_{opt})^e}}{t * n * \theta_{opt}} \quad (31)$$

where

$f(\theta)$  = soil moisture condition objective function.

$n$  = current root depth index according to discretization.

$t$  = simulation time step.

$i$  = depth index within the root zone.

$\theta$  = volumetric soil moisture content.

$\theta_{opt}$  = optimal volumetric soil moisture content for maize.

$\theta_l$  = lower constraint of volumetric soil moisture content.

$\theta_u$  = upper constraint of volumetric soil moisture content.

$$\text{penalty exponent: } e = \begin{cases} 2, & \theta_l \geq \theta_{i,t} \leq \theta_u \\ 2, & \theta_{i,t} > \theta_u \text{ and } i = 1 \\ 4, & \theta_{i,t} > \theta_u \text{ and } i \leq \frac{1}{2}n \\ 4, & \theta_{i,t} < \theta_l \text{ and } i \leq \frac{1}{2}n \\ 3, & \theta_{i,t} < \theta_l \text{ and } i > \frac{1}{2}n \\ 3, & \theta_{i,t} < \theta_l \text{ and } i > \frac{1}{2}n \end{cases} .$$

$$\text{root extraction factor: } f_i = \begin{cases} 0.4, & i \leq n/4 \\ 0.3, & \frac{1}{2}n \geq i > \frac{n}{4} \\ 0.2, & \frac{3}{4}n \geq i > \frac{1}{2}n \\ 0.1, & n \geq i > \frac{3}{4}n \end{cases} .$$

The soft constraints are established by using a penalty factor. This is achieved by increasing the exponent when a soil-moisture value exceeds the water stress levels.

Because maize extracts most water at the upper root zone the focus is laid more on optimizing in this depth. In the lower half of the root zone, the exponent is therefore only increased by one if the water stress levels are exceeded. A special case is the top node of the soil. As discussed in 4.2.1, it was found that limiting the water content in the topsoil prevents replenishing lower depths of the root zone. To deal with this issue the upper constraint penalty is lifted for the top node. This strategy showed good results where neither the roots close to the surface nor the lower roots are experiencing extensive water stress. Moreover, a general weighting of the root zone is applied based on the rule of thumb, discussed in Subsection 3.1.5. The root extraction factor embeds this by splitting up the root zone into four parts and weighting it accordingly. The term is further normalized to reduce the output to a similar range as the water minimization objective function outputs.

### 4.2.3 Single-Field Optimization

The presented approach is a deterministic multiobjective optimization. Three control variables were optimized: the irrigation rate, the start time of irrigation, and the irrigation duration. The approach is a mixed discrete-continuous optimization because the irrigation rate was set as a continuous control variable, whereas the start and interval are set as integer to represent hourly values. The solver NSGA-II was implemented to compute the Pareto optimal front. The framework of this single-field optimization is explained in this subsection.

A set of variables need to be defined for the optimization process. These can be divided into seasonal constants and seasonally varying variables. This way, phenological changes and growth stage-specific changes, as discussed in the section before, can be considered. Table 5 shows all significant variables of the optimization framework with example values of the test simulation of 2018 (see 5.1.2). I want to stress that all given values in the table can be changed freely according to application or needs. The individual parameters of the table are discussed now.

The soil-water characteristics are described in the section before. However, the water stress levels have not been discussed because they are not relevant for the MIKE SHE model itself but the optimization as variables in the objective functions. According to the guideline values found in the literature (see Subsection 3.1.6), the water stress levels were set as shown in Table 5. The optimal value was defined as 65% of the field capacity. The optimal value, the lower, and the upper threshold were constant apart from the late season, where I penalize only very low water contents to avoid waste of water close to harvest.

The optimization interval was set to one week for the test runs. This means that every week the optimization algorithm was executed in the seasonal test optimization. The minimum watering depth is crucial for the initial crop stage immediately after seeding. It defines at what depth the soil moisture is optimized before the plant has even started

growing. The range of the control parameters sets the domain search space. On the one side, this is the irrigation rate minimum and maximum. On the other side, the possible values for the irrigation start time, which is limited by the total simulation period, and the range of the irrigation duration are defined. The range of irrigation duration is set growth stage-specific and according to the guideline provided by the German Association for Water, Wastewater and Waste (2019). This has proven to be a successful approach. After winter, water is abundant in the soil and during the initial growth stage, only the topsoil is slowly depleted of water. As examined in Subsection 4.2.1, shorter irrigation events lead to a higher increase in the topsoil as compared to long irrigation intervals. Thus, short irrigation events are the right approach in the initial stage, whereas with increasing root depth and transpiration, longer irrigation intervals are favourable to replenish soil moisture content in the whole root zone.

Table 5. A general overview of the optimization parameters with example values.

<b>Constants</b>					
Optimal Soil Moisture Level	11.68%	<u>NSGA-II:</u>			
Optimization Interval	7 days	Population Size	10		
Maximum Pumping Rate	2 mm/h	Offspring Size	10		
Minimum Watering Depth	10 cm	SBX distribution index $\eta$	1		
		PM distribution index $\eta$	1		
<u>Control Variables Ranges:</u>		<u>Manual Sampling:</u>			
Upper Limit Start Time	$\triangleq$ total timesteps - irrigation interval maximum	Start Time Range	0 - 12 hours		
Lower Limit Start Time	0 ( $\triangleq$ current time)	Range Irrigation Interval	$\triangleq$ seasonal value		
Upper Limit Pumping Rate	$\triangleq$ max pumping rate	Pumping Rate Range	0.3 - 1.0 mm/h		
Lower Limit Pumping Rate	0	Stress Indicator	0.2 mm/h		
<b>Seasonal Dependent Variables</b>					
Parameter	Initial Stage	Crop Development	Flowering	Mid-season	Late season
Days after seeding	30	70	90	105	130
Depletion Factor	0.5	0.5	0.5	0.5	0.8
Lower Stress Level	8.27%	8.27%	8.27%	8.27%	3.31%
Upper Stress Level	13.76%	13.76%	13.76%	13.76%	13.76%
RAW [cm/m root depth]	8.27	8.27	8.27	8.27	13.23
Range Irrigation Interval [h]	4-36	48-96	48-96	48-96	48-96
No. of generations	4	7	8	7	4
Objective Weights f1/f2	0.5/0.5	0.3/0.7	0/1	0.3/0.7	0.4/0.6

The parameters that are specific to the algorithm NSGA-II have been tested and the values were chosen under consideration of the computational time and the recommendation in the literature (see 3.4.2.2). Population size and offspring size were set to 10 in the test runs. Distribution indices of SBX and PM were both set to 1 to increase the randomness in each new generation.

I also implemented the option of an initial manual sampling that differs from the defined control variable ranges. This means that the first generation that is evaluated by the NSGA-II is set to a narrower range of values to direct the algorithm to the range where the expected global minimum lies. Manual sampling is used in the following way. Before running the NSGA-II, the model is simulated without irrigation. Based on a water deficit indicator, it is then decided if the manual sampling is activated or not. If the field is predicted to be under water stress, the optimal irrigation start point will not lie far ahead in the future. The range of the initial parent generation is set accordingly so that the NSGA-II focuses on values close to the current time. By including this domain knowledge, manual sampling helps the algorithm to work more efficiently by starting the search in the presumably correct range.

A way to minimize the computational time during the test runs was to set the termination criterium of NSGA-II according to the growth stage. While in the initial stage and the late-season fewer generations were computed, the amount was increased during the other stages. During the flowering stage, a lot of factors influence each other, and the optimization is more complex than at previous crop stages, thus the total generation amount is set higher.

I implemented two methods to deal with the *a posteriori* weighting of the objective functions. One is to receive the knee point of the Pareto Optimal Front by computing all Euclidean distances from the utopia point to the solutions and choosing the one with the shortest. The other approach is the weighted-sum method, which was deployed in the test optimization runs. The weighting is seasonally adjusted to the sensitivity of maize to water stress. For example, during the flowering stage, the solution is picked that delivers the best results for the optimal conditions objective.

#### 4.2.4 Multi-Field Optimization

Another objective was to optimize the schedule in the case of multiple fields with different crops. For this purpose, the approach was altered. The control variables that are optimized are different now. The irrigation rate is predefined and set as constant to reduce complexity. Instead, multiple irrigation intervals are optimized sequentially. This means that for a two-field optimization there are 3 control variables: start time and irrigation interval of the first field, and the irrigation interval of the second field. Since all the values are discrete integers that represent hourly intervals, it is a discrete optimization. The framework is depicted in

Figure 19. Every field is represented in an individual MIKE SHE model file, which must be set up according to Section 4.1.

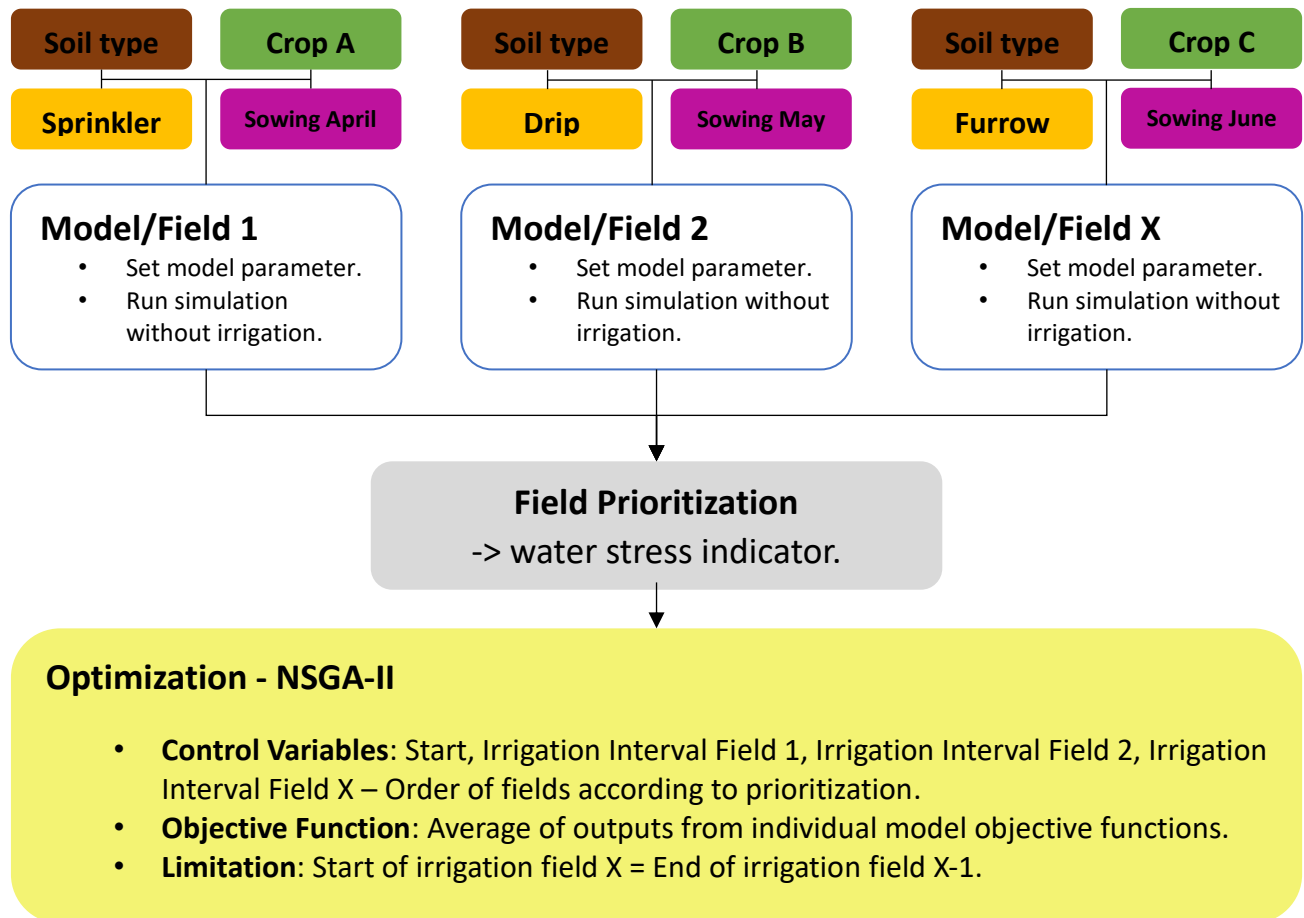


Figure 19: Multi-field optimization framework with example values.

Before the optimization, the irrigation urgency of each field is evaluated. This is achieved by assigning a field water stress indicator. Like the single-field manual sampling approach, the water deficit average is used for this purpose. Each model is simulated once without considering any irrigation. According to the water needs of each field, one is prioritized. One assumption in this framework is that the fields are watered consecutively without stopping. Furthermore, the simulation horizon must be extended according to how many fields exist. In the case of more than two fields, one-week simulation horizon is not sufficient because during the flowering period each field should be irrigated for a time interval of two to four days.

In this approach, one control variable is added per extra field. This limits this approach. Because of this, I recommend splitting up the optimization for more than two fields and optimize sequentially. For instance, in the case of a farm with four fields, prioritization would be done based on the stress indicator. According to this, the fields are ranked. The first two in line are then optimized together. Towards the end of the irrigation period of the second-ranked field, the other two fields can be optimized. This also ensures that sufficiently reliable weather forecast data is used.

The objective functions were defined as follows:

$$F_1 = \frac{\sum_i^n w_i f_i(Q, IP)}{\sum w_i} \quad (32)$$

$$F_2 = \frac{\sum_i^n w_i f_i(\theta)}{\sum w_i} \quad (33)$$

where

$F_1$  = overall water use objective function,

$F_2$  = overall soil-moisture condition objective function,

$f_i(Q, IP)$  = water use objective function of model  $i$ ,

$f_i(\theta)$  = soil moisture condition objective function of model  $i$ ,

$w_i$  = weights of model  $i$ ,

$IP$  = Irrigation duration period,

$Q$  = irrigation rate [mm/h],

$n$  = number of fields that are optimized,

$\theta$  = volumetric soil moisture content [-].

This enables the user to weigh the model-specific objectives. The weights are to be oriented on the sensitivities of the corresponding crop growth stage of the corresponding model. As in the single-field optimization, the NSGA-II algorithm is applied to compute the Pareto optimal front and the algorithm-specific parameter can be freely chosen.

## 4.3 Python Code

Three python files were created with a total of approximately 2700 lines of code that contain all functionality to set up a MIKE SHE model entirely in Python, run optimizations, get weather forecast data and plot results. An object-oriented approach was taken to make it scalable and to make it easy to set model and optimization parameters for other scenarios. An overview of the structure and each file's purpose is depicted in Figure 20. The `she_model.py` file comprises all functions to set up the water-balance model with all parameters described in Table 4. The code is based on the `pfsreader` library. Furthermore, within the `she_model.py` file, functions were established to run simulations and to receive and visualize simulation results. The precipitation rate, reference evapotranspiration, and irrigation are handled in MIKE SHE with `dfs0` – files. These can be accessed and adjusted with the `mikeio` library, which was also integrated into the `she_model.py` code. The import and wrangling of weather forecast data are done in the `weather_forecast.py` file. Data from the German meteorological service was imported. The calculations of the Penman-Monteith method are implemented and applied to estimate hourly reference evapotranspiration.

The `optimizer.py` file contains all functionality of the multiobjective evolutionary optimization. It is built on top of the `pymoo` library that implements the NSGA-II algorithm with all necessary operators. The `optimizer.py` file also makes use of the functionality of the `she_model.py` file. All optimization relevant inputs must be defined. Furthermore, functions are implemented to visualize the optimization results.

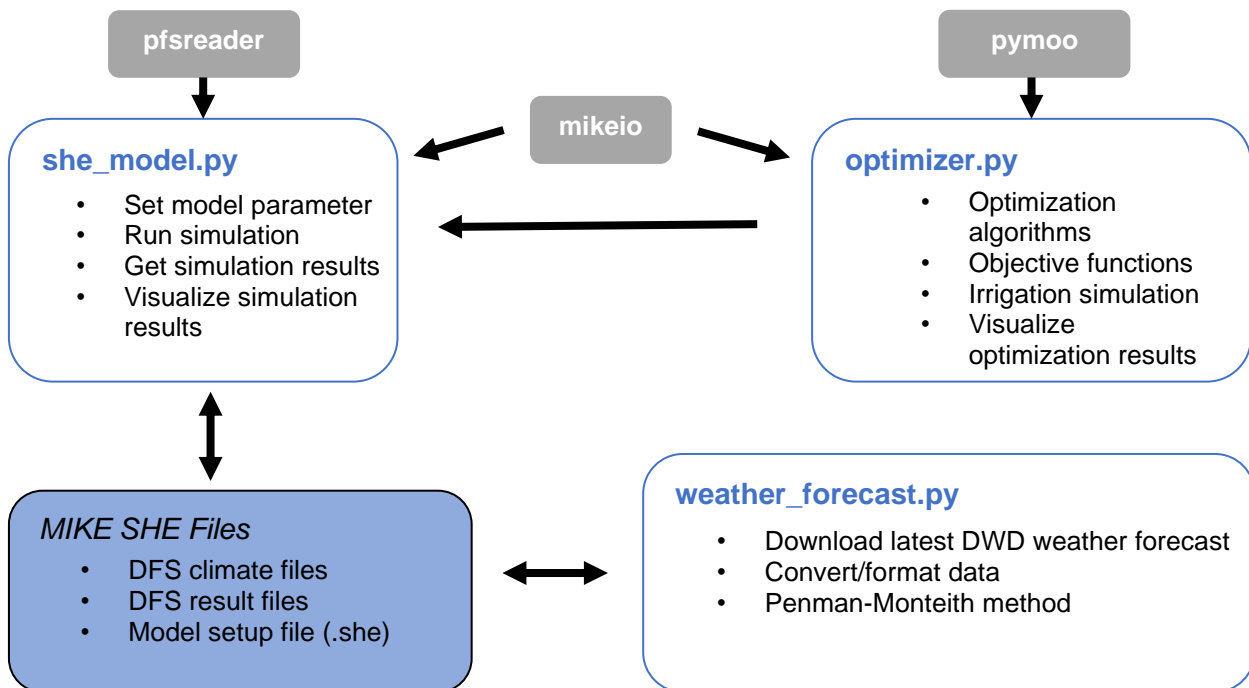


Figure 20: Overview of the code structure.



# 5 Results

---

In this chapter, I present the results of the optimization tests. These are split up into three main parts. First, I will give an overview of the single-field optimization including single event analysis, whole seasonal optimizations for the year 2017 and 2018 and a comparison to pre-seasonal watering. Then, the framework is compared to the integrated irrigation optimizer module of the MIKE SHE software. Third, multi-field optimization results are shown.

A few constants were set for all test runs. As soil type, I picked brown earth and maize was chosen as a test crop. The groundwater level was set to 4.1 meters below the surface and is, hence, not significantly impacting the root zone.

## 5.1 Single-Field Optimization

---

### 5.1.1 Optimization of Single Irrigation Event

I conducted a test run to optimize a single sprinkler irrigation event. The detailed information of the model and optimization setup are given in Appendix A.5. The simulation start time was set to the 15<sup>th</sup> of June, 46 days after seeding. At that time, the maize is in the crop development stage with a root depth of approximately 60 cm. An artificial weather forecast was set up with a long rain event starting 15 hours after simulation start and ending 70 hours after, with a total rainfall of 2.92 mm. The precipitation plus the reference evapotranspiration for the simulation period are shown in Figure 22. The initial soil moisture conditions are set as critical with the top 20 cm depth under significant water stress, which is depicted in Figure 21.

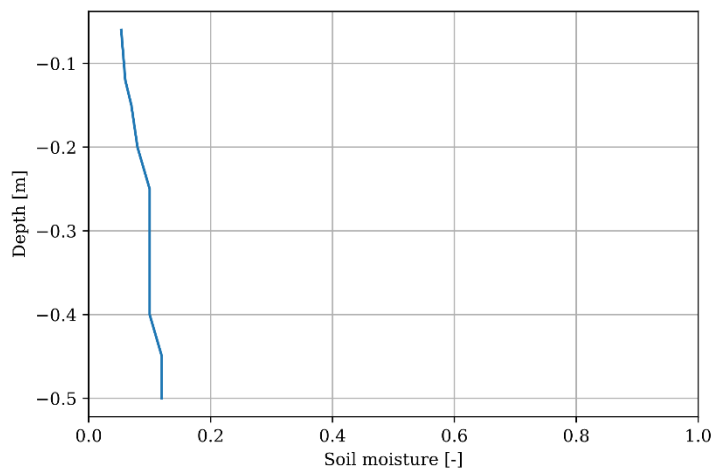


Figure 21: Single irrigation event test optimization - initial soil moisture conditions.

The optimization horizon is set to 7 days with an irrigation duration constraint of 4 to 96 hours, while the irrigation rate is limited to a range of 0 to 2 mm/hour. Manual sampling was deactivated during this test run. The termination criterium of NSGA-II was set to 50 generations to test how the algorithm behaves over a large number of generations.

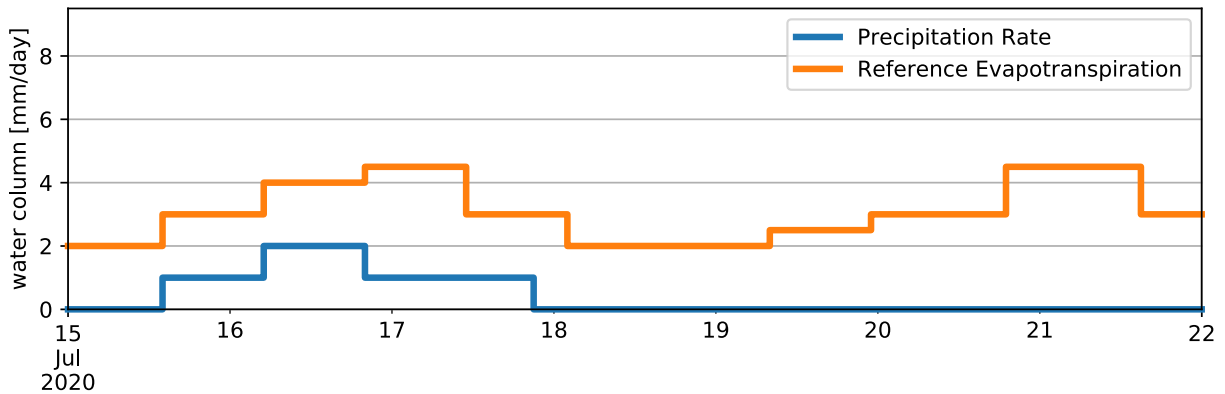


Figure 22: Single irrigation event test optimization - artificially set precipitation and reference evapotranspiration rates of the simulation period.

In Figure 23, the running metric of the optimization is plotted. It shows how the objective functions evolve over the optimization process by estimating the difference of the objectives from one generation to another. The slope is steepest during the first three generations. Nevertheless, the objective functions can still be further minimized even after many generation runs. The more generations are set, the better is the resulting quality of the solution.

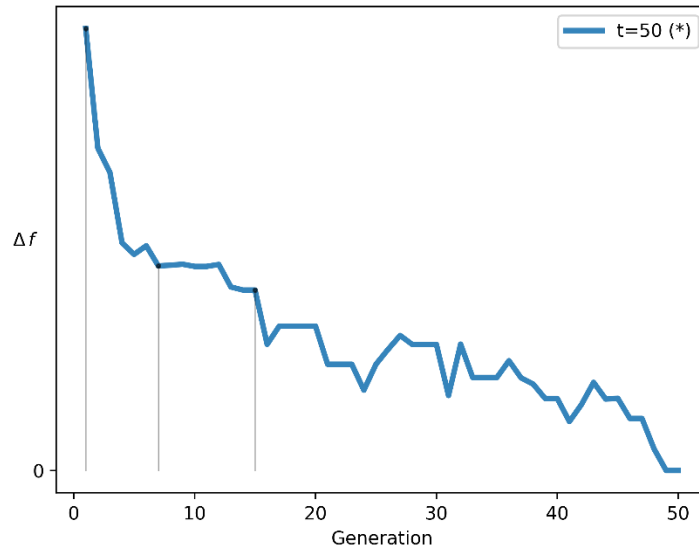


Figure 23: Running metric of the single-event optimization.  $t$ : termination criterium in the number of generations.

Figure 24 visualizes the resulting Pareto optimal front. It can be observed that the front is convex apart from two outliers. The corresponding control variables, pictured as a schedule plot, are shown in Figure 24. Most solutions of the Pareto Optimal Front are irrigation events occurring before the rain. This makes sense when considering the advantages of asynchronous irrigation, analysed in 4.2.1. Nevertheless, the solution that delivers the best soil moisture conditions in the soil is an irrigation rate of 0.35 mm/hour from midnight the 15<sup>th</sup> to 7 pm on the 17<sup>th</sup>. This is a reasonable solution because the initial conditions are critical and water in larger depths must be replenished.

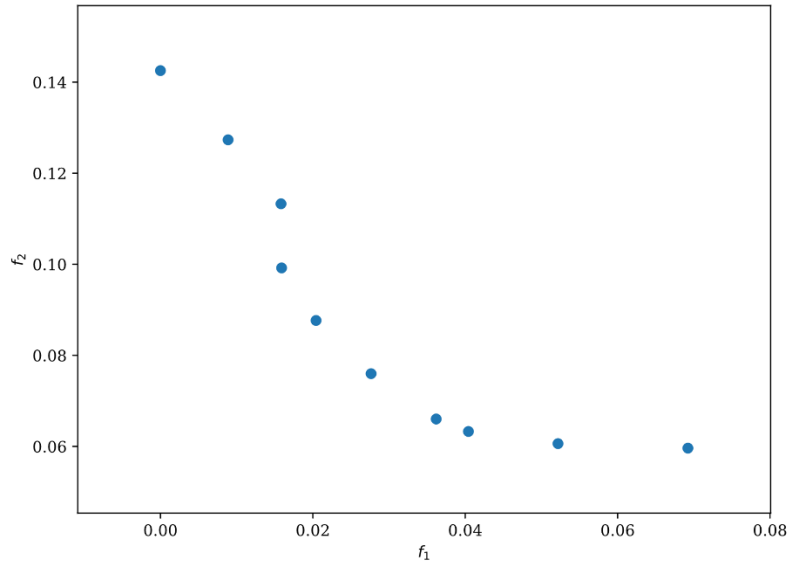


Figure 24: Result Pareto optimal front of the single-event test optimization.  $f_1$ : water consumption objective function,  $f_2$ : soil moisture objective function.

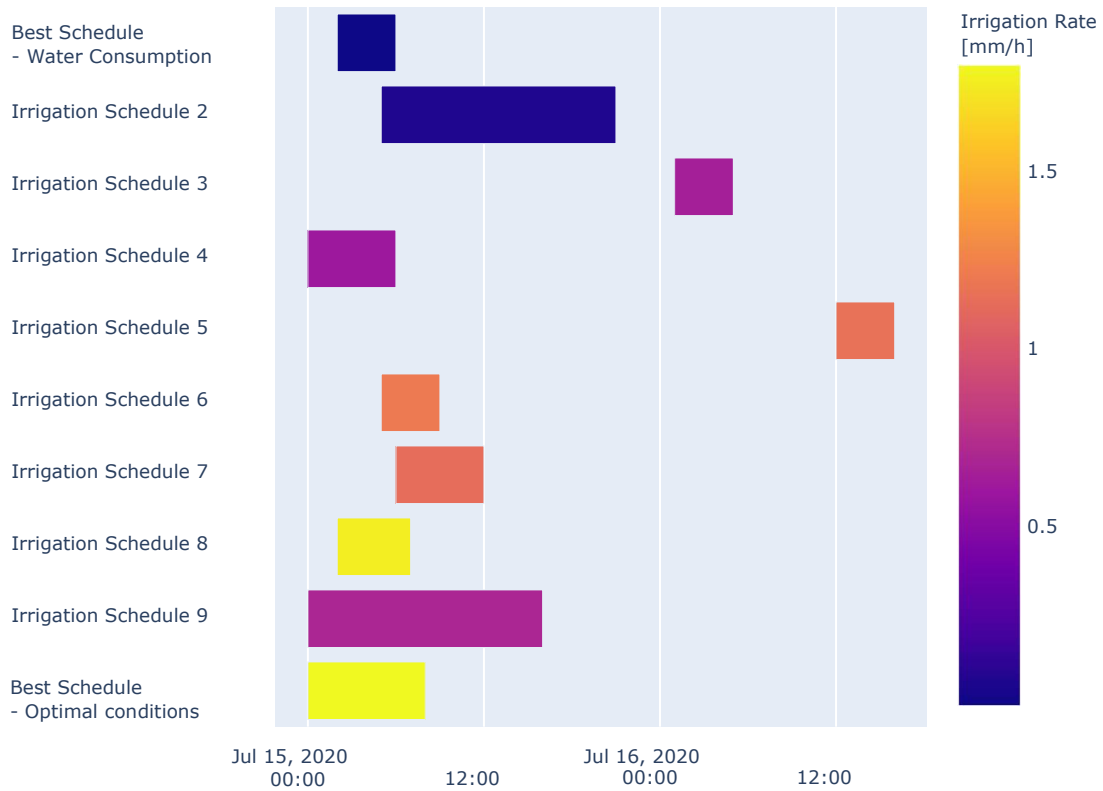


Figure 25: Result Pareto optimal control variables of the single-event test optimization.

### 5.1.2 Seasonal Optimization 2018

After the single irrigation had been tested successfully, the next step was an optimization of a whole season based on historic climate data from a climate station of the German Meteorological Service in Peine, Germany. The whole setup of model and optimization with detailed results is available in Appendix A.6. Here, the approach that was explained in 4.2.3 was implemented with seasonally changing variables. The result is shown in Figure 26.

The year 2018 was the warmest in Germany since the beginning of records in 1881 (Friedrich & Kaspar, 2019). Additionally, precipitation was very low and the total days of sunshine reached all-time highs. As initial soil moisture conditions, values close to the field capacity were assigned to all depths in this scenario, which represent ideal initial soil moisture conditions. The objective weights were set conservatively in alignment with the water stress sensitivity.

The total amount of irrigation water that was used added up to 247.6 mm. It can be observed that over the season the duration of the irrigation events lengthens. This is due to the constraints that are set during the corresponding growth stages and, on the other side, the decrease of water content in lower parts of the root zone that the optimizer tries to replenish. The plot of the soil moisture content shows that even though a rather high amount of water is added over the whole season, it is not possible to maintain an optimal water level in all depths. This is due to the water needs of the plant and the evaporation from the soil that prevents water from reaching further into the ground. Only high infiltration rates that exceed the evapotranspiration rate would achieve an increase. As discussed before (see 4.2.1), these high rates are constrained by the upper soil moisture levels of the optimization framework. Therefore, the total remaining water deficit is at 139.2 mm.

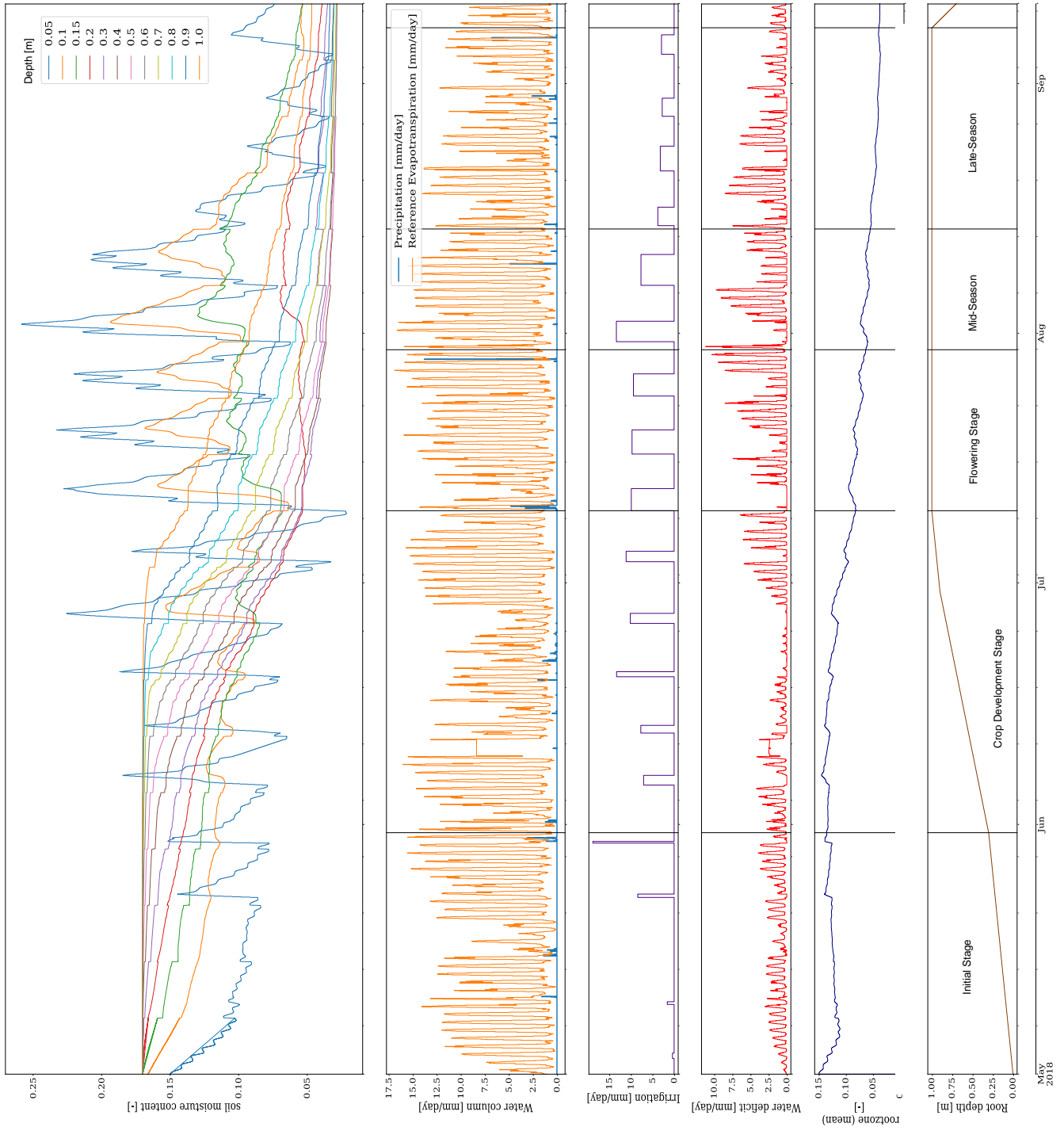


Figure 26: Overview optimization result of a seasonal simulation of the dry year 2018.

### 5.1.3 Seasonal Optimization 2017

I performed another seasonal optimization test for the year 2017 to test the algorithm under different conditions. The year 2017 was also a warm year with a high amount of total sunshine hours but with a surplus of rain (Deutscher Wetterdienst, 2017). A different approach was tested at the same location but with critical initial soil moisture conditions at the beginning of the season. This represents a scenario, where a dry winter failed to replenish a sufficient amount of water in the soil. The objective weights were adjusted more in favour of saving water compared to the optimization of the dry year of 2018. The results are plotted in Figure 27 and the detailed simulation can be found in Appendix A.7.

This setup resulted in total water consumption of 168.2 mm between the 1<sup>st</sup> of May and the 25<sup>th</sup> of August. The optimizer achieved to maintain good soil moisture levels even though the initial soil moisture content was low. After the crop development stage, the water supply for the roots comes mainly from the top 20 cm. The optimizer managed to keep the levels in the topsoil at an optimal level without wasting too much water. The overall water deficit is at approximately 117 mm. As in the optimization before, this is due to the impossibility of replenishing water at higher depths during mid-season.

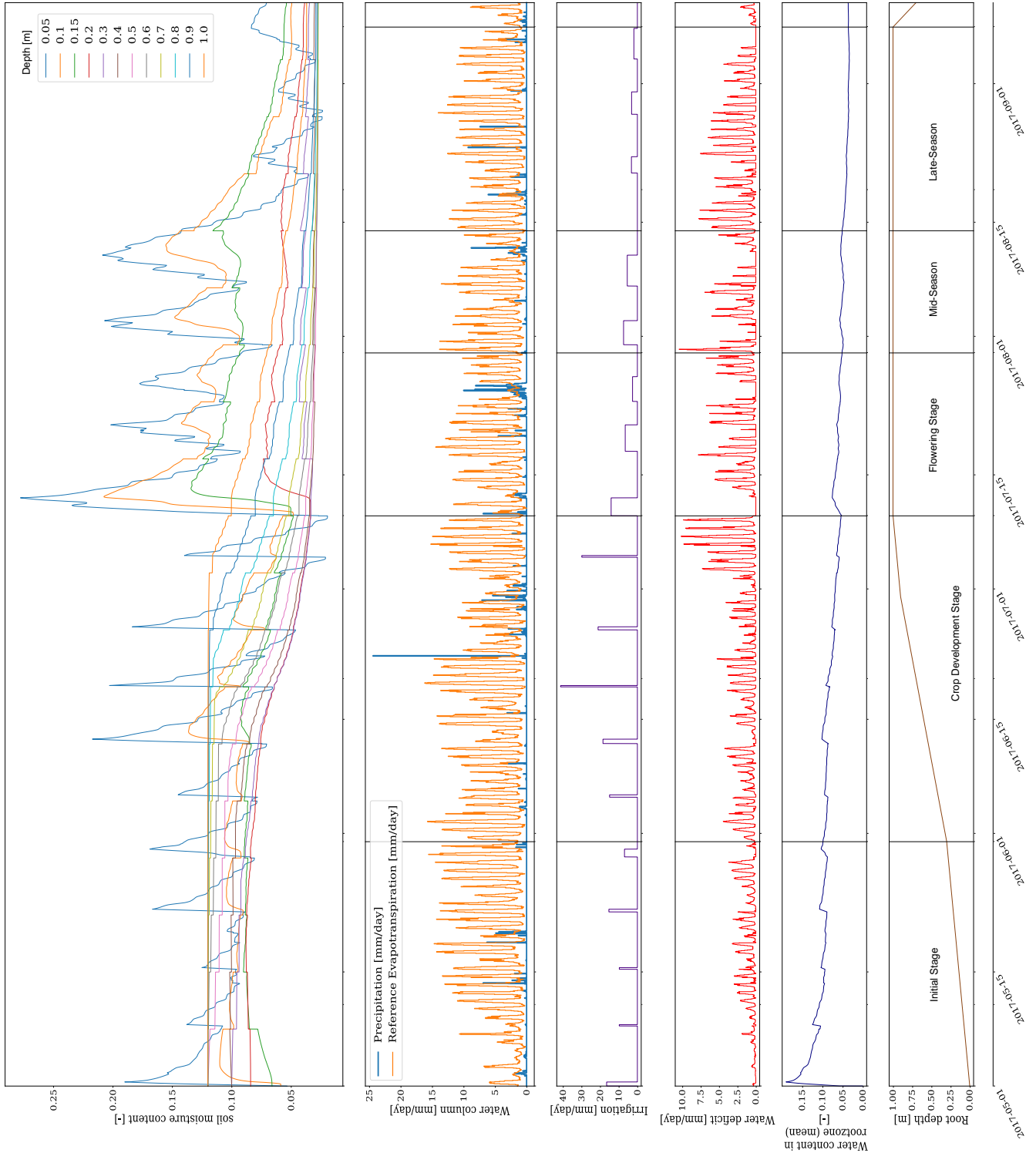


Figure 27: Overview optimization result of a seasonal simulation of the year 2017.



#### 5.1.4 Optimization with Pre-seasonal Watering of the Field

In the previous seasonal test runs, I discovered that the soil moisture content in lower parts of the root zone is generally difficult to raise. Therefore, I analysed a scenario where the field is irrigated before seeding. This analysis is presented here and a detailed overview is given in Appendix A.8. The parameter setup of the model and optimization is identical to the seasonal optimization 2017 but in this case, the field was watered for two weeks in a row before planting the maize, as shown in the results in Figure 28.

The total water used for this pre-seasonal watering lies at 60.5 mm. The total irrigation water that the optimizer recommends after seeding is 144.7 mm. This means that overall, only an additional 37 mm of water is used in this pre-seasonal watering strategy. However, looking at the average water content in the root zone, this approach shows by far better results. The total water deficit ( $ET_a - ET_c$ ) of the optimization without pre-seasonal watering lies at 116.9 mm, while by bringing up the water content before seeding the overall water deficit was lowered to 77.4 mm. In conclusion, 37 mm water used in a sophisticated way of pre-seasonal watering led to a strong reduction of water deficits over the whole season. Especially in the flowering period, the pre-seasonal watering had a strong impact. The deficit during this period dropped from 21.2 mm to 13.2 mm.

A problem with this approach is that during the initial stage the water level in the top part of the root zone is above the upper water stress threshold, which leads to water stress due to oxygen depletion and other factors described in Subsection 3.1.5. This excess water is, however, evaporated or transpired briefly after the seeding and should therefore not impact the overall yield. Hence, in cases where the initial soil conditions are dry, a pre-seasonal watering strategy should be considered but the amount of pre-seasonal watering must be modelled to prevent long-term stress in the initial stage due to too high moisture levels.

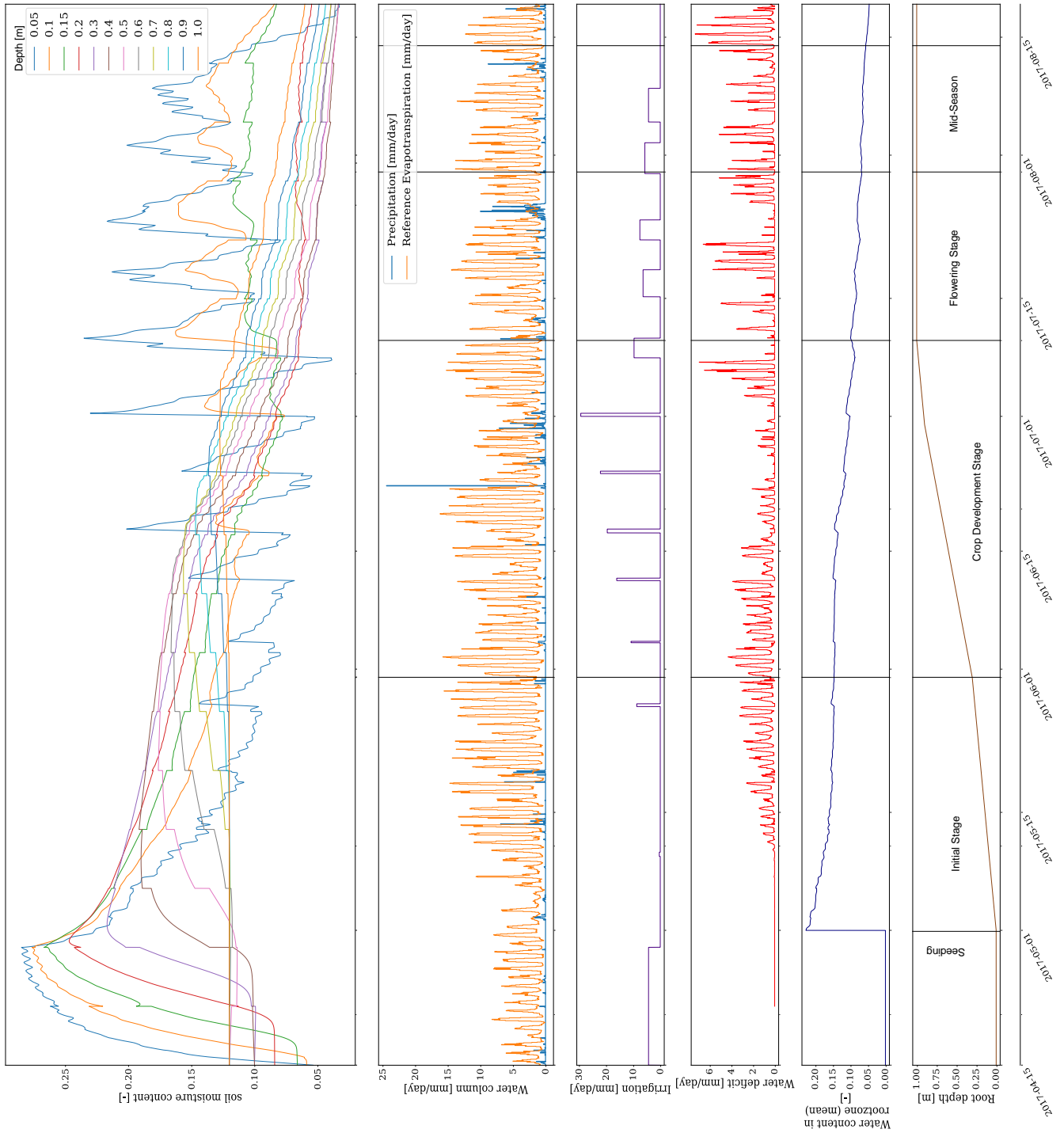


Figure 28: Overview optimization result of a seasonal simulation of the year 2017 with pre-seasonal watering.

## 5.2 Comparison with Integrated MIKE SHE Optimization

---

The MIKE SHE software provides a module to recommend an irrigation schedule to balance the water deficit. Different methods are available, which were discussed in 3.2.3. The deficit method was used, which is the most comparable approach, also based on water stress limits. The same conditions were set as in the optimization of 2018 with a constant maximum allowed irrigation rate of 2 mm/hour, an upper water stress constraint of 80 % TAW, and a lower water stress constraint of 50 % TAW.

The results in Figure 29 show that the optimizer recommends only three irrigation events in the whole season. The total water consumed by applying this schedule is 350.8 mm. This is 103.2 mm more than the schedule that my optimization framework recommended. The total water deficit over the season is 131 mm. In this approach, the upper and lower stress constraints and water losses are not considered by the MIKE SHE irrigation scheduler. The soil moisture content is increased to saturation during the three irrigation events, as lack of oxygen and general water stress factors are disregarded.

MIKE SHE offers solely a static irrigation scheduling, in which for every timestep the root zone soil moisture deficit is evaluated and compensated accordingly. This has the disadvantage that no future or past events or uncertainties are considered. In the case that a rain event occurs shortly after the irrigation was started, the MIKE SHE irrigation optimization approach would result in overwatering.

Comparing my framework to the embedded scheduler of MIKE SHE, significantly higher efficiency in the allocation of water was achieved with a similar overall deficit. The flexibility of adjusting operational preferences, seasonal variables, and root depth prioritization are favouring my method over the MIKE SHE optimization. Furthermore, the general objective of saving water is not part of the MIKE SHE irrigation module. All these advantages come at the price of higher computational cost. While the optimization with MIKE SHE requires only one simulation run, my optimizer relies on multiple soil moisture predictions runs. The details of the optimization with MIKE SHE are given in Appendix A.10.



Figure 29: Overview optimization with MIKE SHE - result of a seasonal simulation for the dry year 2018.

### 5.3 Multi-Field Optimization

Two models were set up for the test of the multi-field optimization. Different initial conditions and vegetation start dates were chosen to characterise different fields. The detailed description of both models and optimization parameters are in Appendix 1.10. An artificial weather forecast was set with constant reference evapotranspiration of 4 mm/day and no predicted rain event. The population size and offspring size were doubled to 20 for the multi-field optimization because of the rising complexity that is caused by the increased number of influencing objectives.

The Pareto optimal schedule solutions are visualized in Figure 30. Field 1 was prioritized with an average water deficit of approximately 2 mm/day, while field 2 was in a better condition with 0.4 mm/day average water deficit. The resulting schedules indicate that for optimal overall conditions field 1 urgently needs to be irrigated. At the same time, field 2 needs irrigation from the beginning, too. Therefore, short irrigation of field 1 is recommended but not for very long to prevent field 2 from dropping below the water stress threshold. Field 2 is irrigated for a longer time since it is not restricted by another field. The results validate the applicability of this approach for multi-field optimization. However, an efficient strategy for more than two fields must be put in place. This is further discussed in 6.2.1.

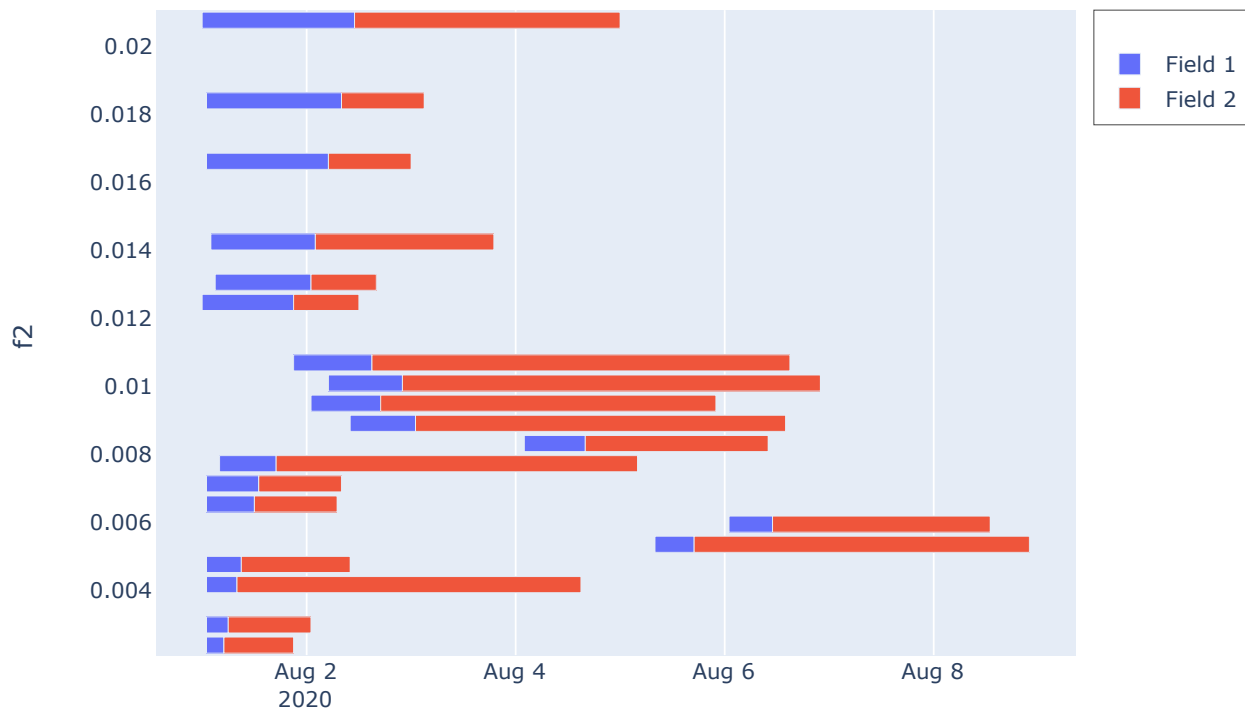


Figure 30: Multi-field test optimization – Pareto optimal solutions.  $f_2$ : soil moisture objective function.

## 6 Discussion

---

In this chapter, I will discuss the strengths, weaknesses, and possible improvements in the implemented optimization framework. First, the accuracy of the system model is evaluated. Then, I review the whole optimization framework. Last, possible improvements of the computational time are discussed.

### 6.1 System Model – MIKE SHE Simulation Model

---

The MIKE SHE software was described in detail in the theoretical part and the test model was presented in Chapter 4. In this section, I examine general weaknesses and discrepancies of, first, the software and, second, the chosen simplified model approach.

MIKE SHE has proven to be a viable system model for the optimization of irrigation schedules. Nevertheless, inherent weaknesses and shortcomings of the software exist. The way MIKE SHE estimates interception storage is timestep-dependent (DHI, 2017b). This means that the smaller the chosen time step, the more influence interception will have, which is not physically correct. This is especially important for the sprinkler irrigation simulation, which is strongly influenced by canopy interception. The functionality of the software should therefore be changed to avoid this dependency. Another shortcoming is the neglect of hysteresis effects (DHI, 2017b). Hysteresis in the context of water flow in the unsaturated zone means that the soil water retention curve behaves differently for wetting than it does for drying processes. Small pores control drainage, while large pores control wetting processes. Other factors that cause hysteresis are changing liquid-solid contact angles, air entrapment during wetting, and potential shrink and swell events of distinct soils. Various hysteresis models for soil-water characteristics are available and may be implemented in the MIKE SHE software. An overview is given by Pham et al. (2005). (Sławiński, 2011)

I conducted a comparison between the sprinkler and drip irrigation methods in MIKE SHE to analyse how the software embeds the irrigation system into the model. The detailed output and setup of the test simulation are shown in Appendix 1.4. The results show that the canopy interception storage for drip irrigation is verifiable zero, as expected. However, this does not cause a visible change of average soil moisture content in the root zone. I discovered the cause of this by examining the water balance. The water, which is intercepted when sprinkler irrigation is used, leads to a decrease of transpiration rate. In the case of drip irrigation, the total amount of water is transpired. The water evaporated either way just through different mechanisms. Logically, sprinkler irrigation would therefore cause the plant to be under more water stress due to interception losses. However, I found that at the same time for sprinkler irrigation the actual evapotranspiration increases. It seems as if the interception evaporation is added to the actual evapotranspiration. Consequently, sprinkler irrigation shows lower water deficits than drip irrigation. This is not plausible and must be investigated.

Some characteristics of the irrigation system are partly neglected in MIKE SHE. In sprinkler irrigation, the water will not be equally spread over the field due to wind and imprecision of the sprinklers. Furthermore, some water evaporates in the air before it reaches a surface. Drip irrigation on the other side is a very precise technique where almost no water is lost. However, it is a point irrigation system, and the soil is watered only in a small radius around the emitters. These factors are usually summarized as the efficiency factor of the irrigation system and shall be included in the system model to improve its accuracy.

A further possible improvement of the software would be the consideration of agricultural soil practises. Tillage and mulching cause the soil to show strongly different characteristics than presumed in the simplified model. This is for example considered in other simulation software, such as DSSAT (Jones et al., 2003). Allen et al. (1998) provide guidelines on how to embed agricultural practices into the crop coefficient.

All the issues that have been discussed and the suggested improvements to the computational time (see Section 6.3) have been reported to the development department of DHI. They will be addressed in the further development of the MIKE SHE software. The enhancement of MIKE SHE will also lead to an improvement in optimization.

After evaluating weaknesses and discrepancies within the software, itself, I want to talk about the test model that has been set up and general limitations. A successful optimization relies on the accuracy of the system model. Apart from the weather forecast data, the test model is not referring to a distinct location yet. In a real-life test scenarios calibration is crucial to improve the accuracy of the model. Furthermore, I recommend that soil probes are taken and evaluated to adjust the soil characteristics. Other options to increase the quality of the MIKE SHE system model are to embed satellite information or data from remote sensing to adjust, for instance, LAI and  $K_c$ . Moreover, I have not yet talked about the significant impact of accurate initial conditions. Until now, values have been set artificially. In a real case, the idea is to update the initial conditions before every optimization run according to sensor measurements on-site. However, this means the system would rely on hardware. Alternatives are soil moisture data from nearby climate stations or satellite data.

In my approach, I used a very simple water-balance model. However, there are specific scenarios where the system model must be more complex to ensure accuracy. For example, if a subsurface water membrane is applied in the field, as described by Roy et al. (2019), the system model will not be accurate. Neither will it be applicable to advancing farming technologies, like agroforestry or permaculture. The water processes that occur in these modern farming systems are hard to model and computationally expensive. Burgess et al. (2019) discussed agroforestry models and gave an overview of existing models. One of the existing models could be used to replace the MIKE SHE model in an agroforestry optimization context. Another issue arises for fields located on steep slopes, for example, vineyards. In this case, the horizontal flow would occur and lead to a profoundly different water flow behaviour in the system.

In the implemented system model nutrient transport and solubility have been neglected. However, nutrient availability is a crucial point to ensure optimal growth conditions for the crop. As discussed in Subsection 3.1.5, water and nutrients are closely related. Therefore, a system model that embeds this relationship is desirable and the optimization should address nutrient availability as another objective. MIKE SHE possesses with ECOLAB a related water quality module, which may enable such simulations (DHI, 2017b). Another viable option is the HYDRUS-2D software that was already used in a similar optimization context to simulate nutrient and water flow (P. C. Roy et al., 2019).

A crucial issue that has not been addressed is how heterogeneities are incorporated in the simplified model. The general idea is to have one model per crop field, as in the multi-field optimization. If different soil characteristics exist between fields, the model will be adjusted in the corresponding model. However, if heterogeneities exist within a field, these must be addressed differently. Since the model should not be extended to further dimensions due to computational expenses, another approach is needed. Geostatistical methods are an option to correlate the different soil-water behaviours in a field, which can be based on soil moisture sensor data, satellite data, or various simplified models that compute soil moisture in different areas of the field.

## 6.2 Optimization Framework

---

Many aspects of the system model have been analysed. Now, I want to discuss the complete optimization framework with its strengths and weaknesses. One important advantage of the proposed framework is flexibility. The MIKE SHE software is treated as a black-box system model within the optimization. The parameters of the model can be changed without impacting the optimization but solely the behaviour of the system. Exchanging crop type, soil characteristics, and all physically relevant factors are therefore made very easy. Using a complex physics-based model based on numerical methods is more accurate than a simplified water-balance system model, as applied in other frameworks (Delgoda, Saleem, et al., 2016; Kassing et al., 2020), but comes with the price of higher computational time.

The timeframe of the optimization can theoretically be chosen freely. This means that the optimization can be shifted from a weekly run with a monthly optimization horizon to a daily run with weekly optimization horizon simply by adjusting two parameters. The limitation of the approach in this context is, however, that only one irrigation event can be predicted. This makes the method only applicable to an optimization horizon that resembles the reaction of the system to the irrigation event, as explained in 4.2.1. For example, in the case of a monthly optimization horizon, the optimizer will search for the optimal time to water the crop within that month. This means that most probably excessive irrigation will be scheduled at the beginning of the month to deliver a long-term water supply. This is not the optimal approach. Thus, the optimization horizon must be in



alignment with the reaction of the system. Besides, a monthly weather forecast shows high uncertainties, limiting the optimization horizon further.

The optimization interval can be chosen according to the preference of the user. Theoretically, the shorter the chosen interval, the better because alterations and updates on the weather forecast are considered. However, the computational time is a limiting factor here. Therefore, I recommend analysing the weather forecast before every optimization run to check if significant changes in the forecast are present that may lead to significantly different soil moisture predictions. If this is not the case, the previously estimated schedule should be maintained. I recommend ranges from one to three days for the interval to take new weather predictions into account.

Another strength of the framework is the handling of seasonal variables. Water stress levels and weighting of the objectives can be adjusted freely. This allows the adjustment of these factors to the crop phenology including water stress sensitivity during certain growth stages. This approach of the adaptable seasonal parameter could be extended following the approach of Kassing et al. (2020). In their framework, a seasonal planner is implemented that estimates the water allocation over the season based on historical data and experience. This way, overall maximum water use can be set. By implementing a similar approach in this work's approach, this maximum water constraint could be considered. Especially in arid climates, this would be a significant extension. The optimization parameters could generally be adapted to deficit irrigation practices by lowering the stress values and adjusting the seasonal variables. However, in arid climates, the objective of eliminating salt problems in the soil should be addressed.

As described in Subsection 4.2.3, the control variables are constrained. The start time of the irrigation is limited by the simulation period of the MIKE SHE model. I successfully tested limiting the start time to certain times of the day. This can be useful for farmers that cannot irrigate their fields during the daytime, as it is the case in the state of Bavaria, Germany, or simply to set preferred times.

I chose a bi-objective optimization approach to minimize water use while optimizing soil moisture conditions. I reviewed various research papers that focused on different objectives in their irrigation scheduling approach (see Chapter 2). Potentially, the alternative objectives of these researchers may be added to the optimization paradigm of this work. For instance, economic objectives could be included, such as minimization of energy use on the farm, thus reducing fuel use. By considering various cost factors that come hand in hand with irrigation (nutrient price, fuel price, investment cost, etc.), the overall farm income can be optimized.

### 6.2.1 Multi-Field Optimization

The implemented multi-field approach is computationally expensive. With every additional field, another control variable is added, and the optimization relies on an extra simulation run. Hence, computational time optimization (see 6.3) is crucial for the successful application of this approach.

In 5.3, I explained that every field carries with it two objective functions that need to be minimized. The total amount of objectives therefore is rising linearly with every additional field. The implemented strategy is to weigh the water consumption objective and the soil moisture condition objective of all fields *a priori* to maintain two general objective functions. The nondominated results are then weighted *a posteriori* as in the single-field optimization. Weighting all the objectives *a posteriori* instead is difficult and a multi-dimensional Pareto optimal front would be obtained. The opposite strategy may be more suitable by weighting all objective functions *a priori*. This would make it easier for the decision-maker to prioritize fields or incentives. The resulting single objective function that considers all the fields would be the following:

$$F = \frac{\sum_i^n w_{i,1} f_{i,1}(Q, IP) + w_{i,2} f_{i,2}(\theta)}{\sum w_{i,1} + w_{i,2}} \quad (34)$$

where

$F$  = total objective function.

$f_{i,1}$  = water use objective function of model  $i$ .

$f_{i,2}$  = soil moisture condition objective function of model  $i$ .

$w_i$  = weights of model  $i$ .

$IP$  = Irrigation duration period.

$Q$  = irrigation rate [mm/h].

$n$  = amounts of fields that are optimized.

$\theta$  = volumetric soil moisture content [-].

By applying scalarization, we only have one objective function to minimize, which would make this a single-objective optimization problem. For this purpose, other algorithms than the NSGA-II are required. Particle Swarm Optimization (PSO) is an example of a suitable single-objective solver algorithm. Following this approach, an option of how to deal with the weights is to relate them to the water stress indicator. This way, more emphasis can be put on the fields under stress in addition to prioritizing them in the irrigation sequence.

## 6.3 Computational Time Optimization

---

As proven in 5.1.1, there exists a distinct trade-off between the solution quality and the computational time, which rises linearly with the generations set. If this optimization is applied on a large scale, it will become very computational and energy demanding. This trade-off must be generally considered when debating optimization in the water-energy-nexus. The run time of the optimization must therefore be minimized. In the context of this framework, different strategies can be pursued to achieve this. One is the parallelization of the model runs. For a population size of 10, also ten simulation runs must be conducted for every new generation. These simulations could be run simultaneously. Another potential to reduce numerical effort is to optimize the runtime of the MSHE system model itself for repeatedly executed short model runs. At this point, this is not the case and initializing the software every run requires unnecessary computational resources. Moreover, the way the algorithm computes the results may be adjusted from an elementwise evaluation to a vectorized evaluation, as described in the *pymoo* getting started guide (Blank & Deb, 2020). Furthermore, the discretization of the depth in MIKE SHE and the simulation timestep are the crucial parameters for not only the numerical stability but also the computational time. Hence, the model could be further simplified to make it faster.

## 7 Conclusion and Outlook

---

A multiobjective optimization framework based on the software MIKE SHE has been developed and tested successfully. The solver algorithm NSGA-II has proven to be effective in the context of irrigation scheduling. Good results can be obtained already with a low number of simulation runs and domain knowledge was embedded in the algorithm to make it more efficient. Seasonal and short-term optimization aspects were put into a combined approach. The results showed that by adjusting the optimization parameters the objective focus could be shifted from either saving more water or ensuring optimal soil moisture conditions.

Due to the decoupling of the system model and optimization, and the object-oriented code implementation the approach is highly flexible. This offers huge advantages for testing different scenarios and further extending the framework for other purposes than shown in the tests. The remaining task is to optimize the computational time so that the general result quality can be further improved without consuming excessive energy and time. Besides, with the reduction of computational time, the multi-field approach can be applied and further expanded.

The presented approach outperformed the software integrated irrigation scheduler. Through its dynamic approach with consideration of future weather events and growth stage-specific phenological changes, the optimizer delivers a realistic irrigation schedule

that is adjustable to the farmers' preferences. Furthermore, it offers a framework to perform pre-seasonal assessments to compare different irrigation strategies due to its flexibility and physics-based system model.

All the test optimizations that have been performed were based on fixed historical data. Thus, the optimization must be still tested in a real-life scenario with fluctuations of the weather forecast prediction. Furthermore, the effectiveness of the approach must be validated on a real farm.

# References

- Ahmed, M. A., Zarebanadkouki, M., Meunier, F., Javaux, M., Kaestner, A., & Carminati, A. (2018). Root type matters: Measurement of water uptake by seminal, crown, and lateral roots in maize. *Journal of Experimental Botany*, *69*(5), 1199–1206. <https://doi.org/10.1093/jxb/erx439>
- Aibara, I., & Miwa, K. (2014). Strategies for Optimization of Mineral Nutrient Transport in Plants: Multilevel Regulation of Nutrient-Dependent Dynamics of Root Architecture and Transporter Activity. *Plant and Cell Physiology*, *55*(12), 2027–2036. <https://doi.org/10.1093/pcp/pcu156>
- Alizadeh, H., & Mousavi, S. J. (2013). Coupled stochastic soil moisture simulation-optimization model of deficit irrigation: Stochastic Optimization Model of Deficit Irrigation. *Water Resources Research*, *49*(7), 4100–4113. <https://doi.org/10.1002/wrcr.20282>
- Allen, R. G. (Ed.). (1998). *Crop evapotranspiration: Guidelines for computing crop water requirements*. Food and Agriculture Organization of the United Nations.
- Alothaimen, I., & Arditi, D. (2019). Overview of Multi-Objective Optimization Approaches in Construction Project Management. In *Multi-criteria Optimization—Pareto-optimal and Related Principles [Working Title]*. IntechOpen. <https://doi.org/10.5772/intechopen.88185>
- Amaran, S., Sahinidis, N. V., Sharda, B., & Bury, S. J. (2016). Simulation optimization: A review of algorithms and applications. *Annals of Operations Research*, *240*(1), 351–380. <https://doi.org/10.1007/s10479-015-2019-x>
- Audet, C., & Dennis, J. E. (2006). Mesh Adaptive Direct Search Algorithms for Constrained Optimization. *SIAM Journal on Optimization*, *17*(1), 188–217. <https://doi.org/10.1137/040603371>
- Audet, C., & Dennis, J. E. (2009). A Progressive Barrier for Derivative-Free Nonlinear Programming. *SIAM Journal on Optimization*, *20*(1), 445–472. <https://doi.org/10.1137/070692662>
- Baluja, S., & Caruana, R. (1995). Removing the Genetics from the Standard Genetic Algorithm. In *Machine Learning Proceedings 1995* (pp. 38–46). Elsevier. <https://doi.org/10.1016/B978-1-55860-377-6.50014-1>
- Belaineh, G., Peralta, R. C., & Hughes, T. C. (1999). Simulation/Optimization Modeling for Water Resources Management. *Journal of Water Resources Planning and Management*, *125*(3), 154–161. [https://doi.org/10.1061/\(ASCE\)0733-9496\(1999\)125:3\(154\)](https://doi.org/10.1061/(ASCE)0733-9496(1999)125:3(154))
- Belaqziz, S., Khabba, S., Er-Raki, S., Jarlan, L., Le Page, M., Kharrou, M. H., Adnani, M. E., & Chehbouni, A. (2013). A New Irrigation Priority Index Based on Remote Sensing Data for Assessing the Networks Irrigation Scheduling. *Agricultural Water Management*, *119*, 1–9. <https://doi.org/10.1016/j.agwat.2012.12.011>
- Belaqziz, S., Mangiarotti, S., Le Page, M., Khabba, S., Er-Raki, S., Agouti, T., Drapeau, L., Kharrou, M. H., El Adnani, M., & Jarlan, L. (2014). Irrigation Scheduling of a Classical Gravity Network Based on the Covariance Matrix Adaptation – Evolutionary Strategy Algorithm. *Computers and Electronics in Agriculture*, *102*, 64–72. <https://doi.org/10.1016/j.compag.2014.01.006>

- Beume, N., Naujoks, B., & Emmerich, M. (2007). SMS-EMOA: Multiobjective selection based on dominated hypervolume. *European Journal of Operational Research*, 181(3), 1653–1669. <https://doi.org/10.1016/j.ejor.2006.08.008>
- Bianchi, L., Dorigo, M., Gambardella, L. M., & Gutjahr, W. J. (2009). A survey on metaheuristics for stochastic combinatorial optimization. *Natural Computing*, 8(2), 239–287. <https://doi.org/10.1007/s11047-008-9098-4>
- Blank, J., & Deb, K. (2020). pymoo: Multi-objective Optimization in Python. *IEEE Access*, 8, 89497–89509. <https://doi.org/10.1109/ACCESS.2020.2990567>
- Boyd, S. P., & Vandenberghe, L. (2004). *Convex Optimization*. Cambridge University Press.
- Braun, M. A. (2018). *Scalarized Preferences in Multi-objective Optimization*. <https://doi.org/10.5445/IR/1000082700>
- Brown, P. D., Cochrane, T. A., & Krom, T. D. (2010). Optimal on-farm irrigation scheduling with a seasonal water limit using simulated annealing. *Agricultural Water Management*, 97(6), 892–900. <https://doi.org/10.1016/j.agwat.2010.01.020>
- Bundesinformationszentrum Landwirtschaft. (2017). *Agrarmeteorologie* (Vol. 1651). Bundesanstalt für Landwirtschaft und Ernährung (BLE).
- Bundesinformationszentrum Landwirtschaft (BZL), & Bundesanstalt für Landwirtschaft und Ernährung. (2020). *Bewässerung in der Landwirtschaft*. <https://www.praxis-agrar.de/pflanze/bewaesserung/bewaesserung-in-der-landwirtschaft/?L=0>
- Burgess, P., Graves, A., de Jalón, S. G., Palma, J., Dupraz, C., & van Noordwijk, M. (2019). Modelling Agroforestry Systems. In Universidade de Santiago de Compostela, Spain, M. R. Mosquera-Losada, R. Prabhu, & World Agroforestry Centre (ICRAF), Kenya (Eds.), *Burleigh Dodds Series in Agricultural Science* (pp. 209–238). Burleigh Dodds Science Publishing. <https://doi.org/10.19103/AS.2018.0041.13>
- Burke, E., Kendall, G., Newall, J., Hart, E., Ross, P., & Schulenburg, S. (2003). Hyper-Heuristics: An Emerging Direction in Modern Search Technology. In F. Glover & G. A. Kochenberger (Eds.), *Handbook of Metaheuristics* (Vol. 57, pp. 457–474). Kluwer Academic Publishers. [https://doi.org/10.1007/0-306-48056-5\\_16](https://doi.org/10.1007/0-306-48056-5_16)
- Cui, Y., Geng, Z., Zhu, Q., & Han, Y. (2017). Review: Multi-objective optimization methods and application in energy saving. *Energy*, 125, 681–704. <https://doi.org/10.1016/j.energy.2017.02.174>
- Custódio, A. L., Emmerich, M., & Madeira, J. F. A. (2012). Recent Developments in Derivative-Free Multiobjective Optimisation. *Computational Technology Reviews*, 5, 1–30. <https://doi.org/10.4203/ctr.5.1>
- Deb, K. (2001). *Multi-objective optimization using evolutionary algorithms* (1st ed). John Wiley & Sons.
- Deb, K., & Agrawal, R. B. (1995). Simulated Binary Crossover for Continuous Search Space. *Complex Syst.*, 9.
- Deb, K., & Goyal, M. (1996). A Combined Genetic Adaptive Search (GeneAS) for Engineering Design. *Computer Science and Informatics*, 26, 30–45.
- Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2), 182–197. <https://doi.org/10.1109/4235.996017>
- Deb, K., Sindhya, K., & Okabe, T. (2007). Self-adaptive simulated binary crossover for real-parameter optimization. *Proceedings of the 9th Annual Conference on*

- Genetic and Evolutionary Computation - GECCO '07*, 1187.  
<https://doi.org/10.1145/1276958.1277190>
- Delgoda, D., Malano, H., Saleem, S. K., & Halgamuge, M. N. (2016). Irrigation control based on model predictive control (MPC): Formulation of theory and validation using weather forecast data and AQUACROP model. *Environmental Modelling & Software*, 78, 40–53. <https://doi.org/10.1016/j.envsoft.2015.12.012>
- Delgoda, D., Saleem, S. K., Malano, H., & Halgamuge, M. N. (2016). Root zone soil moisture prediction models based on system identification: Formulation of the theory and validation using field and AQUACROP data. *Agricultural Water Management*, 163, 344–353. <https://doi.org/10.1016/j.agwat.2015.08.011>
- Demarée, G. R., & Rutishauser, T. (2011). From “Periodical Observations” to “Anthochronology” and “Phenology” – the scientific debate between Adolphe Quetelet and Charles Morren on the origin of the word “Phenology”. *International Journal of Biometeorology*, 55(6), 753–761. <https://doi.org/10.1007/s00484-011-0442-5>
- Désidéri, J.-A. (2012). Multiple-gradient descent algorithm (MGDA) for multiobjective optimization. *Comptes Rendus Mathématique*, 350(5–6), 313–318. <https://doi.org/10.1016/j.crma.2012.03.014>
- Deutscher Wetterdienst. (2017). *Wetter und Klima—Deutscher Wetterdienst—Archiv 2017—Deutschlandwetter im Jahr 2017*. [https://www.dwd.de/DE/presse/pressemitteilungen/DE/2017/20171229\\_deutschlandwetter\\_jahr2017.html;jsessionid=1A40F8E1B22B434BC5DA541B1142360E.liv e21072?nn=618918](https://www.dwd.de/DE/presse/pressemitteilungen/DE/2017/20171229_deutschlandwetter_jahr2017.html;jsessionid=1A40F8E1B22B434BC5DA541B1142360E.liv e21072?nn=618918)
- DHI. (2017a). *Class library documentation for the DHI.PFS*. [http://doc.mikepoweredbydhi.help/webhelp/2017/DHI\\_PFS/html/N\\_DHI\\_PFS.htm](http://doc.mikepoweredbydhi.help/webhelp/2017/DHI_PFS/html/N_DHI_PFS.htm)
- DHI. (2017b). *MIKE SHE Handbook*.
- DHI. (2019). *MIKE SHE Help Topics*. <http://doc.mikepoweredbydhi.help/webhelp/2019/MIKESHE/index.htm>
- DHI. (2020). *DFS file system—MIKE for developers docs*. [http://docs.mikepoweredbydhi.com/core\\_libraries/dfs/dfs-file-system/](http://docs.mikepoweredbydhi.com/core_libraries/dfs/dfs-file-system/)
- Doorenbos, J., & Kassam, A. (1979). *FAO Irrigation and Drainage Paper 33*.
- DWA-Fachausschuss GB-4 „Bewässerung“. (2019). *Grundsätze und Richtwerte zur Beurteilung von Anträgen zur Entnahme von Wasser für die Bewässerung* (DWA-M 590).
- Ehrgott, M. (2005). *Multicriteria optimization* (2nd ed). Springer.
- Emmerich, M. T. M., & Deutz, A. H. (2018). A tutorial on multiobjective optimization: Fundamentals and evolutionary methods. *Natural Computing*, 17(3), 585–609. <https://doi.org/10.1007/s11047-018-9685-y>
- Erwin Diewert, W. (2008). Cost Functions. In Palgrave Macmillan (Ed.), *The New Palgrave Dictionary of Economics* (pp. 1–12). Palgrave Macmillan UK. [https://doi.org/10.1057/978-1-349-95121-5\\_659-2](https://doi.org/10.1057/978-1-349-95121-5_659-2)
- Fan TongKe. (2013). Smart Agriculture Based on Cloud Computing and IOT. *Journal of Convergence Information Technology*, 8(2), 210–216. <https://doi.org/10.4156/jcit.vol8.issue2.26>
- Fanuel, I. M., Mushi, A., & Kajunguri, D. (2018). Irrigation water allocation optimization using multi-objective evolutionary algorithm (MOEA) – a review. *International Journal for Simulation and Multidisciplinary Design Optimization*, 9, A3. <https://doi.org/10.1051/smdo/2018001>

- FAO (Ed.). (2002). *Deficit Irrigation Practices*. Food and Agriculture Organization of the United Nations.
- FAO. (2016). *AQUASTAT Main Database*. <http://www.fao.org/aquastat/en/overview/methodology/water-use>
- FAO. (2020). *Database Crop Information*. <http://www.fao.org/land-water/databases-and-software/crop-information/maize/en/>
- Fereres, E., & Soriano, M. A. (2006). Deficit Irrigation for Reducing Agricultural Water Use. *Journal of Experimental Botany*, 58(2), 147–159. <https://doi.org/10.1093/jxb/erl165>
- Friedrich, K., & Kaspar, F. (2019). *Rückblick auf das Jahr 2018 – das bisher wärmste Jahr in Deutschland*. 4.
- Fu, Y., Li, M., & Guo, P. (2014). Optimal Allocation of Water Resources Model for Different Growth Stages of Crops under Uncertainty. *Journal of Irrigation and Drainage Engineering*, 140(6), 05014003. [https://doi.org/10.1061/\(ASCE\)IR.1943-4774.0000724](https://doi.org/10.1061/(ASCE)IR.1943-4774.0000724)
- García-Vila, M., & Fereres, E. (2012). Combining the simulation crop model AquaCrop with an economic model for the optimization of irrigation management at farm level. *European Journal of Agronomy*, 36(1), 21–31. <https://doi.org/10.1016/j.eja.2011.08.003>
- Gerrits, A. M. J., & Savenije, H. H. G. (2011). Interception. *Treatise on Water Science*, Vol. 2, 89–101.
- Giacomini, M., Désidéri, J.-A., & Duvigneau, R. (2014). *Comparison of multiobjective gradient-based methods for structural shape optimization* (p. 30).
- Giusti, E., & Marsili-Libelli, S. (2015). A Fuzzy Decision Support System for irrigation and water conservation in agriculture. *Environmental Modelling & Software*, 63, 73–86. <https://doi.org/10.1016/j.envsoft.2014.09.020>
- Goldstein, A., Fink, L., Meitin, A., Bohadana, S., Lutenberg, O., & Ravid, G. (2018). Applying Machine Learning on Sensor Data for Irrigation Recommendations: Revealing the Agronomist's Tacit Knowledge. *Precision Agriculture*, 19(3), 421–444. <https://doi.org/10.1007/s11119-017-9527-4>
- Grefenstette, J. (1986). Optimization of Control Parameters for Genetic Algorithms. *IEEE Transactions on Systems, Man, and Cybernetics*, 16(1), 122–128. <https://doi.org/10.1109/TSMC.1986.289288>
- Grosan, C., Oltean, M., & Oltean, M. (2003). The role of elitism in multiobjective optimization with evolutionary algorithms. *Acta Universitatis Apulensis. Mathematics - Informatics*, 5.
- Gunantara, N. (2018). A review of multi-objective optimization: Methods and its applications. *Cogent Engineering*, 5(1). <https://doi.org/10.1080/23311916.2018.1502242>
- Hsiao, T. C., Heng, L., Steduto, P., Rojas-Lara, B., Raes, D., & Fereres, E. (2009). AquaCrop-The FAO Crop Model to Simulate Yield Response to Water: III. Parameterization and Testing for Maize. *Agronomy Journal*, 101(3), 448–459. <https://doi.org/10.2134/agronj2008.0218s>
- Hunter, S. R., Applegate, E. A., Arora, V., Chong, B., Cooper, K., Rincón-Guevara, O., & Vivas-Valencia, C. (2019). An Introduction to Multiobjective Simulation Optimization. *ACM Transactions on Modeling and Computer Simulation*, 29(1), 1–36. <https://doi.org/10.1145/3299872>



- Ikudayisi, A. (2015). Irrigation water optimization using evolutionary algorithms. *Environmental Economics*, 6(1), 6.
- Ikudayisi, A., Adeyemo, J., Odiyo, J., & Enitan, A. (2018). Optimum Irrigation Water Allocation and Crop Distribution Using Combined Pareto Multi-Objective Differential Evolution. *Cogent Engineering*, 5(1).  
<https://doi.org/10.1080/23311916.2018.1535749>
- IoT Based Smart Irrigation System*. (2020). Bosch Global.  
<https://www.bosch.com/stories/iot-based-smart-irrigation-system/>
- Israelson, O.W., & West, F.L. (1922). *Water holding capacity of irrigated soils*.  
<https://doi.org/10.1097%2F00010694-193109000-00003>
- Jebari, K., & Madiafi, M. (2013). *Selection Methods for Genetic Algorithms*. 13.
- Jiang, Y., Xu, X., Huang, Q., Huo, Z., & Huang, G. (2016). Optimizing regional irrigation water use by integrating a two-level optimization model and an agro-hydrological model. *Agricultural Water Management*, 178, 76–88.  
<https://doi.org/10.1016/j.agwat.2016.08.035>
- Jones, J. W., Hoogenboom, G., Porter, C. H., Boote, K. J., Batchelor, W. D., Hunt, L. A., Wilkens, P. W., Singh, U., Gijsman, A. J., & Ritchie, J. T. (2003). The DSSAT cropping system model. *European Journal of Agronomy*, 18(3–4), 235–265.  
[https://doi.org/10.1016/S1161-0301\(02\)00107-7](https://doi.org/10.1016/S1161-0301(02)00107-7)
- Jong, R. D., Campbell, C. A., & Nicholaichuk, W. (1983). Water Retention Equations and their Relationship to Soil Organic Matter and Particle Size Distribution for Disturbed Samples. *Canadian Journal of Soil Science*, 63(2), 291–302.  
<https://doi.org/10.4141/cjss83-029>
- Kang, M. G., & Park, S. W. (2014). Combined Simulation-Optimization Model for Assessing Irrigation Water Supply Capacities of Reservoirs. *Journal of Irrigation and Drainage Engineering*, 140(5), 04014005.  
[https://doi.org/10.1061/\(ASCE\)IR.1943-4774.0000726](https://doi.org/10.1061/(ASCE)IR.1943-4774.0000726)
- Kang, S., Shi, W., & Zhang, J. (2000). An Improved Water-Use Efficiency for Maize Grown Under Regulated Deficit Irrigation. *Field Crops Research*, 67(3), 207–214.  
[https://doi.org/10.1016/S0378-4290\(00\)00095-2](https://doi.org/10.1016/S0378-4290(00)00095-2)
- Kassing, R., de Schutter, B., & Abraham, E. (2020). *Optimal seasonal water allocation and model predictive control for precision irrigation* [Other]. display.  
<https://doi.org/10.5194/egusphere-egu2020-11270>
- Kirda, C. (2002). Deficit Irrigation Scheduling Based on Plant Growth Stages Showing Water Stress Tolerance. *Deficit Irrigation Practices*, 22, 3–10.
- Kristensen, K. J., & Jensen, S. E. (1975). A Model for Estimating Actual Evapotranspiration from Potential Evapotranspiration. *Hydrology Research*, 6(3), 170–188. <https://doi.org/10.2166/nh.1975.0012>
- Kuo, S.-F., & Liu, C.-W. (2003). Simulation and optimization model for irrigation planning and management. *Hydrological Processes*, 17(15), 3141–3159.  
<https://doi.org/10.1002/hyp.1269>
- Larson, J., Menickelly, M., & Wild, S. M. (2019). Derivative-free optimization methods. *Acta Numerica*, 28, 287–404. <https://doi.org/10.1017/S0962492919000060>
- Li, K., Zhang, T., & Wang, R. (2020). Deep Reinforcement Learning for Multiobjective Optimization. *IEEE Transactions on Cybernetics*, 1–12.  
<https://doi.org/10.1109/TCYB.2020.2977661>

- Li, M., Sui, R., Meng, Y., & Yan, H. (2019). A real-time fuzzy decision support system for alfalfa irrigation. *Computers and Electronics in Agriculture*, 163, 104870. <https://doi.org/10.1016/j.compag.2019.104870>
- Li, X., Liu, H., He, X., Gong, P., & Lin, E. (2019). Water–Nitrogen Coupling and Multi-Objective Optimization of Cotton Under Mulched Drip Irrigation in Arid Northwest China. *Agronomy*, 9(12), 894. <https://doi.org/10.3390/agronomy9120894>
- Liakos, K., Busato, P., Moshou, D., Pearson, S., & Bochtis, D. (2018). Machine Learning in Agriculture: A Review. *Sensors*, 18(8), 2674. <https://doi.org/10.3390/s18082674>
- Lim, S. M., Sultan, A. B. Md., Sulaiman, Md. N., Mustapha, A., & Leong, K. Y. (2017). Crossover and Mutation Operators of Genetic Algorithms. *International Journal of Machine Learning and Computing*, 7(1), 9–12. <https://doi.org/10.18178/ijmlc.2017.7.1.611>
- Linker, R., Ioslovich, I., Sylaos, G., Plauborg, F., & Battilani, A. (2016). Optimal model-based deficit irrigation scheduling using AquaCrop: A simulation study with cotton, potato and tomato. *Agricultural Water Management*, 163, 236–243. <https://doi.org/10.1016/j.agwat.2015.09.011>
- Lobo, F. G., Lima, C. F., & Michalewicz, Z. (Eds.). (2007). *Parameter Setting in Evolutionary Algorithms* (Vol. 54). Springer Berlin Heidelberg. <https://doi.org/10.1007/978-3-540-69432-8>
- Lopez, J. R., Winter, J. M., Elliott, J., Ruane, A. C., Porter, C., & Hoogenboom, G. (2017). Integrating growth stage deficit irrigation into a process based crop model. *Agricultural and Forest Meteorology*, 243, 84–92. <https://doi.org/10.1016/j.agrformet.2017.05.001>
- Mannini, P., Genovesi, R., & Letterio, T. (2013). IRRINET: Large Scale DSS Application for On-farm Irrigation Scheduling. *Procedia Environmental Sciences*, 19, 823–829. <https://doi.org/10.1016/j.proenv.2013.06.091>
- Marler, R. T., & Arora, J. S. (2010). The Weighted Sum Method for Multi-Objective Optimization: New Insights. *Structural and Multidisciplinary Optimization*, 41(6), 853–862. <https://doi.org/10.1007/s00158-009-0460-7>
- McCarthy, A. C., Hancock, N. H., & Raine, S. R. (2014). Simulation of irrigation control strategies for cotton using Model Predictive Control within the VARlwise simulation framework. *Computers and Electronics in Agriculture*, 101, 135–147. <https://doi.org/10.1016/j.compag.2013.12.004>
- Md. Azamathulla, H., Wu, F.-C., Ghani, A. A., Narulkar, S. M., Zakaria, N. A., & Chang, C. K. (2008). Comparison Between Genetic Algorithm and Linear Programming Approach for Real Time Operation. *Journal of Hydro-Environment Research*, 2(3), 172–181. <https://doi.org/10.1016/j.jher.2008.10.001>
- Mendes, W. R., Araújo, F. M. U., Dutta, R., & Heeren, D. M. (2019). Fuzzy control system for variable rate irrigation using remote sensing. *Expert Systems with Applications*, 124, 13–24. <https://doi.org/10.1016/j.eswa.2019.01.043>
- Miettinen, K. (1999). *Nonlinear multiobjective optimization*. Kluwer Academic Publishers.
- MIKE SHE. (2020). <https://www.mikepoweredbydhi.com/download/mike-2017-sp2/mike-she?ref=%7B40160C10-5509-4460-A36F-FA2759EAC02F%7D>
- Ministerium für Ländliche Entwicklung, Umwelt und Verbraucherschutz des Landes Brandenburg. (2005). *Leitfaden zur Berechnung landwirtschaftlicher Kulturen. Schriftenreihe des Landesamtes für Verbraucherschutz, Landwirtschaft und Flurneuordnung Abteilung Landwirtschaft und Gartenbau Teltow, Groß Kreutz*,

- Güterfelde, Paulinenaue, Wünsdorf Reihe Landwirtschaft, Heft VI (Band 6 (2005)), 16.
- Mitchell, M. (1996). *An introduction to genetic algorithms*. MIT Press.
- Montonen, O., Karmitsa, N., & Mäkelä, M. M. (2018). Multiple subgradient descent bundle method for convex nonsmooth multiobjective optimization. *Optimization*, 67(1), 139–158. <https://doi.org/10.1080/02331934.2017.1387259>
- Nemes, A., Pachepsky, Y. A., & Timlin, D. J. (2011). Toward Improving Global Estimates of Field Soil Water Capacity. *Soil Science Society of America Journal*, 75(3), 807–812. <https://doi.org/10.2136/sssaj2010.0251>
- Ortega Álvarez, J. F., de Juan Valero, J. A., Tarjuelo Martín-Benito, J. M., & López Mata, E. (2004). MOPECO: An economic optimization model for irrigation water management. *Irrigation Science*, 23(2), 61–75. <https://doi.org/10.1007/s00271-004-0094-x>
- Paul E. Black. (2005, February 2). *Greedy algorithm*. <https://www.nist.gov/dads/HTML/greedyalgo.html>
- Peitz, S., & Dellnitz, M. (2018). Gradient-Based Multiobjective Optimization with Uncertainties. *ArXiv:1612.03815 [Math]*, 731, 159–182. [https://doi.org/10.1007/978-3-319-64063-1\\_7](https://doi.org/10.1007/978-3-319-64063-1_7)
- Peña-Gallardo, M., Vicente-Serrano, S. M., Domínguez-Castro, F., & Beguería, S. (2019). The impact of drought on the productivity of two rainfed crops in Spain. *Natural Hazards and Earth System Sciences*, 19(6), 1215–1234. <https://doi.org/10.5194/nhess-19-1215-2019>
- Pham, H. Q., Fredlund, D. G., & Barbour, S. L. (2005). A Study of Hysteresis Models for Soil-Water Characteristic Curves. *Canadian Geotechnical Journal*, 42(6), 1548–1568. <https://doi.org/10.1139/t05-071>
- Qingfu Zhang, & Hui Li. (2007). MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6), 712–731. <https://doi.org/10.1109/TEVC.2007.892759>
- Ramirez-Atencia, C., Mostaghim, S., & Camacho, D. (2017). A Knee Point Based Evolutionary Multi-Objective Optimization for Mission Planning Problems. *Proceedings of the Genetic and Evolutionary Computation Conference*, 1216–1223. <https://doi.org/10.1145/3071178.3071319>
- Richards, L. A. (1931). Capillary Conduction of Liquids Through Porous Mediums. *Physics*, 1(5), 318–333. <https://doi.org/10.1063/1.1745010>
- Romano, N., & Santini, A. (2002). *Water Retention and Storage: Field*. (pp. 721–738).
- Roy, P. C., Guber, A., Abouali, M., Nejadhashemi, A. P., Deb, K., & Smucker, A. J. M. (2019). Crop yield simulation optimization using precision irrigation and subsurface water retention technology. *Environmental Modelling & Software*, 119, 433–444. <https://doi.org/10.1016/j.envsoft.2019.07.006>
- Roy, S. B., Chen, L., Girvetz, E. H., Maurer, E. P., Mills, W. B., & Grieb, T. M. (2012). Projecting Water Withdrawal and Supply for Future Decades in the U.S. under Climate Change Scenarios. *Environmental Science & Technology*, 46(5), 2545–2556. <https://doi.org/10.1021/es2030774>
- Salter, P. J., & Haworth, F. (1961). The Available-Water Capacity of a Sandy Loam Soil: I. a Critical Comparison of Methods of Determining the Moisture Content of Soil at Field Capacity and at the Permanent Wilting Percentage. *Journal of Soil Science*, 12(2), 326–334. <https://doi.org/10.1111/j.1365-2389.1961.tb00922.x>

- Schaffer, J., Caruana, R., Eshelman, L., & Das, R. (1989). *A Study of Control Parameters Affecting Online Performance of Genetic Algorithms for Function Optimization*.
- Schütze, N., Wöhling, T., Paly, M. D., & Schmitz, G. H. (2006). Global Optimization of Deficit Irrigation Systems using Evolutionary Algorithms. *Proceedings of the XVI International Conference on Computational Methods in Water Resources*, 8.
- Shaxson, T. F., & Barber, R. G. (2003). *Optimizing soil moisture for plant production: The significance of soil porosity*. Food and Agriculture Organization of the United Nations.
- Singh, A. (2014). Simulation–optimization modeling for conjunctive water use management. *Agricultural Water Management*, 141, 23–29. <https://doi.org/10.1016/j.agwat.2014.04.003>
- Sławiński, C. (2011). Hysteresis in Soil. In J. Gliński, J. Horabik, & J. Lipiec (Eds.), *Encyclopedia of Agrophysics* (pp. 385–385). Springer Netherlands. [https://doi.org/10.1007/978-90-481-3585-1\\_72](https://doi.org/10.1007/978-90-481-3585-1_72)
- Souza, G., Aquino-Jr, P. T., Maia, R. F., Kamienski, C., & Soininen, J.-P. (2020). *A fuzzy irrigation control system*. 5.
- Srinivas, N., & Deb, K. (1994). Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms. *Evolutionary Computation*, 2(3), 221–248. <https://doi.org/10.1162/evco.1994.2.3.221>
- Stegman, E. C. (1982). Corn Grain Yield as Influenced by Timing of Evapotranspiration Deficits. *Irrigation Science*, 3(2), 75–87. <https://doi.org/10.1007/BF00264851>
- Togneri, R., Kamienski, C., Dantas, R., Prati, R., Toscano, A., Soininen, J.-P., & Cinotti, T. S. (2019). Advancing IoT-Based Smart Irrigation. *IEEE Internet of Things Magazine*, 2(4), 20–25. <https://doi.org/10.1109/IOTM.0001.1900046>
- Umbarkar A.J., & P.D. Sheth. (2015). Crossover Operators in Genetic Algorithms: A Review. *ICTACT Journal on Soft Computing*, 06(01), 1083–1092. <https://doi.org/10.21917/ijsc.2015.0150>
- van Bodegom, P. M., Sorrell, B. K., Oosthoek, A., Bakker, C., & Aerts, R. (2008). Separating the Effects of Partial Submergence and Soil Oxygen Demand on Plant Physiology. *Ecology*, 89(1), 193–204. <https://doi.org/10.1890/07-0390.1>
- van Genuchten, M. Th. (1980). A Closed-form Equation for Predicting the Hydraulic Conductivity of Unsaturated Soils. *Soil Science Society of America Journal*, 44(5), 892–898. <https://doi.org/10.2136/sssaj1980.03615995004400050002x>
- Varela-Ortega, C., Blanco-Gutiérrez, I., Esteve, P., Bharwani, S., Fronzek, S., & Downing, T. E. (2016). How can irrigated agriculture adapt to climate change? Insights from the Guadiana Basin in Spain. *Regional Environmental Change*, 16(1), 59–70. <https://doi.org/10.1007/s10113-014-0720-y>
- Veihmeyer, F. J., & Hendrickson, A. H. (1928). Soil Moisture at Permanent Wilting of Plants. *Plant Physiology*, 3(3), 355–357. <https://doi.org/10.1104/pp.3.3.355>
- Waller, P., & Yitayew, M. (2016). *Irrigation and Drainage Engineering*. Springer International Publishing. <https://doi.org/10.1007/978-3-319-05699-9>
- Wang, H., Xiao, B., Wang, M., & Shao, M. (2013). Modeling the Soil Water Retention Curves of Soil-Gravel Mixtures with Regression Method on the Loess Plateau of China. *PLOS ONE*, 8(3), e59475. <https://doi.org/10.1371/journal.pone.0059475>
- Yan, J., & Smith, K. R. (1994). Simulation of Integrated Surface Water and Ground Water Systems—Model Formulation. *Journal of the American Water Resources Association*, 30(5), 879–890. <https://doi.org/10.1111/j.1752-1688.1994.tb03336.x>

- Yang, X.-S. (2014). Multi-Objective Optimization. In *Nature-Inspired Optimization Algorithms* (pp. 197–211). Elsevier. <https://doi.org/10.1016/B978-0-12-416743-8.00014-2>
- Zhang, L. (2014). Multi-attribute Decision Making. In A. C. Michalos (Ed.), *Encyclopedia of Quality of Life and Well-Being Research* (pp. 4164–4166). Springer Netherlands. [https://doi.org/10.1007/978-94-007-0753-5\\_1863](https://doi.org/10.1007/978-94-007-0753-5_1863)
- Zitzler, E., Laumanns, M., & Thiele, L. (2001). *SPEA2: Improving the strength pareto evolutionary algorithm* [Working Paper]. Eidgenössische Technische Hochschule Zürich (ETH), Institut für Technische Informatik und Kommunikationsnetze (TIK). <https://doi.org/10.3929/ethz-a-004284029>
- Zotarelli, L., Dukes, M. D., Romero, C. C., & Migliaccio, K. W. (2018). *Step by Step Calculation of the Penman-Monteith Evapotranspiration (FAO-56 Method)*. 10.

# List of Figures

FIGURE 1: OVERVIEW OF OPTIMIZATION OBJECTIVES IN THE FIELD OF IRRIGATION. ....	2
FIGURE 2: OVERVIEW OF OPTIMIZATION METHODS IN THE FIELD OF IRRIGATION. ....	4
FIGURE 3: PARTITIONING OF EVAPOTRANSPIRATION OVER THE GROWTH PERIOD OF A PLANT. (ALLEN, 1998) .....	8
FIGURE 4: RULE OF THUMB FOR ROOT WATER UPTAKE OVER DEPTH. (WALLER & YITAYEW, 2016) .....	12
FIGURE 5: GROWTH STAGES OF MAIZE. (FAO, 2020) .....	17
FIGURE 6: $ET_A/ET_C$ RELATIONSHIP TO LAI REGARDING THE EMPIRICAL CONSTANTS $C_1$ AND $C_2$ . (KRISTENSEN & JENSEN, 1975) .....	22
FIGURE 7: DISTRIBUTION OF ACTUAL EVAPOTRANSPIRATION OVER DEPTH FOR DIFFERENT VALUES OF $C_2$ . $C_2=0$ CORRESPONDS TO PURE TRANSPIRATION. (DHI, 2017B) .....	23
FIGURE 8: FRACTION OF ET EXTRACTED AS A FUNCTION OF DEPTH FOR DIFFERENT VALUES OF $AROOT$ . (DHI, 2017B) .....	24
FIGURE 9: PARETO OPTIMAL FRONT FOR TWO OBJECTIVE FUNCTIONS WITH CHARACTERISTIC POINTS. (GUNANTARA, 2018) .....	31
FIGURE 10: INFLUENCE OF DISTRIBUTION INDEX ON THE OFFSPRING PROBABILITY DENSITY. $\eta$ : DISTRIBUTION INDEX .....	39
FIGURE 11: RANKING PROCEDURE OF NSGA-II. (DEB ET AL., 2002) .....	43
FIGURE 12: ESTIMATION SCHEMA OF THE CROWDING DISTANCE IN A 2-DIMENSIONAL OBJECTIVE SPACE. FILLED CIRCLES ARE SOLUTIONS OF THE SAME NONDOMINATED FRONT. THE AXIS $F_1$ AND $F_2$ ARE THE OBJECTIVE FUNCTION VALUES. (DEB ET AL., 2002) .	44
FIGURE 13: OVERVIEW IMPLEMENTED OPTIMIZATION FRAMEWORK .....	48
FIGURE 14: VAN GENUCHTEN SOIL-WATER RETENTION CURVE OF SOIL TYPE: BROWN EARTH. .....	50
FIGURE 15: ROOT DEPTH/LAI/CROP COEFFICIENT DEVELOPMENT OF MAIZE OVER A CROP SEASON COMPUTED WITH MIKE SHE .....	51
FIGURE 16: ANALYSIS SYSTEM MODEL - 2018 SEASONAL SIMULATION WITHOUT IRRIGATION. INITIAL CONDITIONS WERE SET APPROXIMATELY TO FIELD CAPACITY. ....	53
FIGURE 17: COMPARISON OF IRRIGATION EVENTS THAT OCCUR SYNCHRONOUS VS. ASYNCHRONOUS TO A PRECIPITATION EVENT. Y-AXIS: AVERAGE SOIL MOISTURE FOR A WEEKLY PREDICTION, X-AXIS: DEPTH BELOW THE SOIL SURFACE .....	54
FIGURE 18: COMPARISON OF LONG-TERM VS. SHORT-TERM IRRIGATION AS AVERAGE SOIL MOISTURE CONTENT OVER DEPTH. Y-AXIS: AVERAGE SOIL MOISTURE FOR A WEEKLY PREDICTION, X-AXIS: DEPTH BELOW THE SOIL SURFACE .....	55
FIGURE 19: MULTI-FIELD OPTIMIZATION FRAMEWORK WITH EXAMPLE VALUES. ....	61
FIGURE 20: OVERVIEW OF THE CODE STRUCTURE. ....	63
FIGURE 21: SINGLE IRRIGATION EVENT TEST OPTIMIZATION - INITIAL SOIL MOISTURE CONDITIONS. ....	64
FIGURE 22: SINGLE IRRIGATION EVENT TEST OPTIMIZATION - ARTIFICIALLY SET PRECIPITATION AND REFERENCE EVAPOTRANSPIRATION RATES OF THE SIMULATION PERIOD .....	65
FIGURE 23: RUNNING METRIC OF THE SINGLE-EVENT OPTIMIZATION. T: TERMINATION CRITERIUM IN THE NUMBER OF GENERATIONS. ....	66
FIGURE 24: RESULT PARETO OPTIMAL FRONT OF THE SINGLE-EVENT TEST OPTIMIZATION. $F_1$ : WATER CONSUMPTION OBJECTIVE FUNCTION, $F_2$ : SOIL MOISTURE OBJECTIVE FUNCTION. .....	67
FIGURE 25: RESULT PARETO OPTIMAL CONTROL VARIABLES OF THE SINGLE-EVENT TEST OPTIMIZATION. ....	67
FIGURE 26: OVERVIEW OPTIMIZATION RESULT OF A SEASONAL SIMULATION OF THE DRY YEAR 2018. ....	69

FIGURE 27: OVERVIEW OPTIMIZATION RESULT OF A SEASONAL SIMULATION OF THE YEAR 2017.....	71
FIGURE 28: OVERVIEW OPTIMIZATION RESULT OF A SEASONAL SIMULATION OF THE YEAR 2017 WITH PRE-SEASONAL WATERING. ....	73
FIGURE 29: OVERVIEW OPTIMIZATION WITH MIKE SHE - RESULT OF A SEASONAL SIMULATION FOR THE DRY YEAR 2018.....	75
FIGURE 30: MULTI-FIELD TEST OPTIMIZATION – PARETO OPTIMAL SOLUTIONS. F2: SOIL MOISTURE OBJECTIVE FUNCTION. ....	76

## List of Tables

TABLE 1: GUIDELINE FOR WATER STRESS THRESHOLDS. (MINISTERIUM FÜR LÄNDLICHE ENTWICKLUNG, UMWELT UND VERBRAUCHERSCHUTZ DES LANDES BRANDENBURG, 2005) .....	14
TABLE 2: SUMMARY OF SEASONAL CROP-CHARACTERISTIC COEFFICIENTS OF MAIZE. (FAO, 2020) .....	17
TABLE 3: TERMINOLOGY OF OPTIMIZATION PROBLEMS. (AMARAN ET AL., 2016).....	29
TABLE 4: OVERVIEW SETUP OF THE PHYSICS-BASED SYSTEM MODEL CREATED IN MIKE SHE. ....	49
TABLE 5. A GENERAL OVERVIEW OF THE OPTIMIZATION PARAMETERS WITH EXAMPLE VALUES. ....	59

## List of Algorithms

ALGORITHM 1: ESTIMATION OF CROWDING DISTANCE. ....	44
ALGORITHM 2: NON-DOMINATED SORTING GENETIC ALGORITHM II (NSGA-II). ....	45





# Appendix

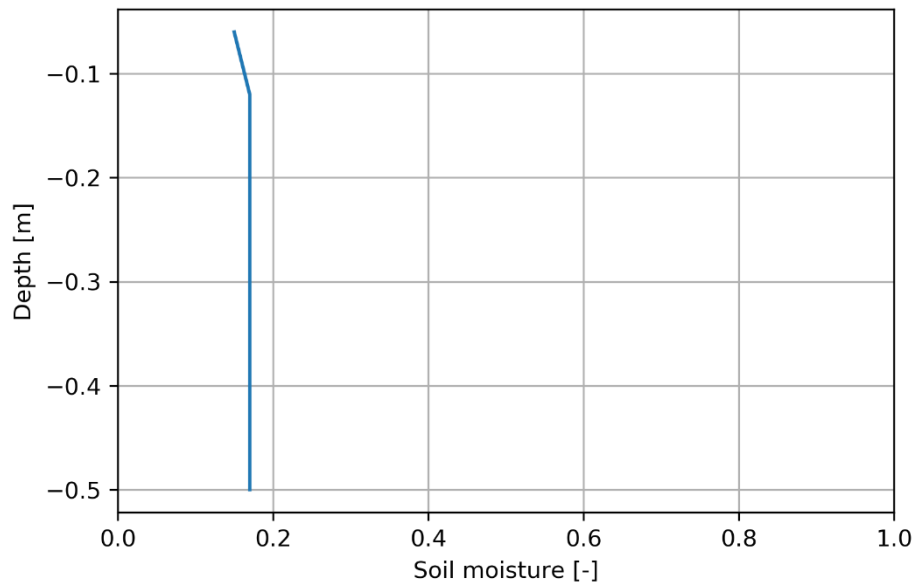
## A.1 System Behaviour Analysis: Seasonal Simulation without Irrigation

### ***Model Setup***

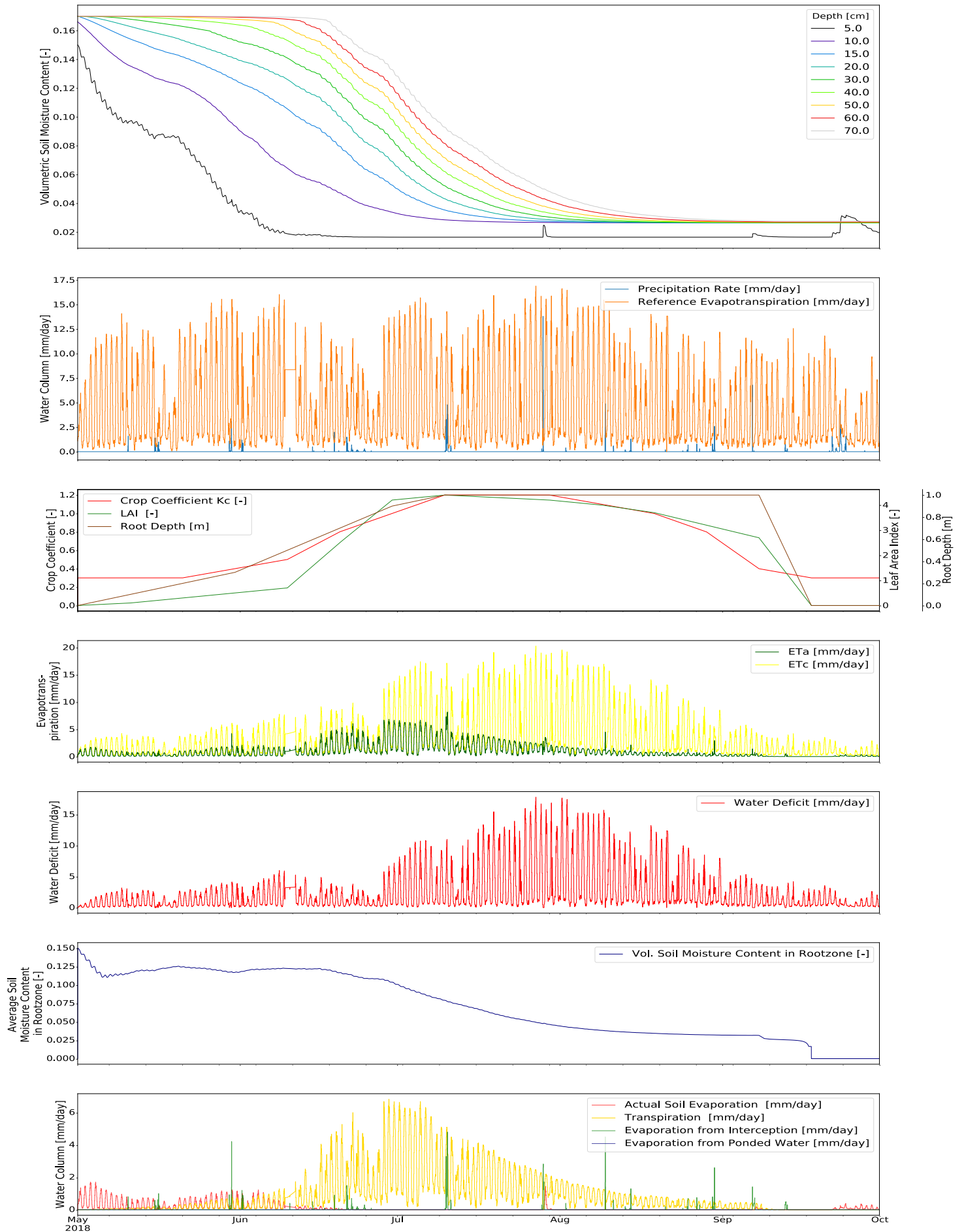
---

Irrigation Method	Sprinkler
Groundwater Table	4.1 m
Crop	Maize
Vegetation Start Date	1st of May, 2018
Simulation Start Date	1st of May, 2018
Simulation End Date	1st of October, 2018

Initial volumetric soil moisture content:



# Full simulation result overview:



# A.2 System Behaviour Analysis: Asynchronous and Synchronous Irrigation

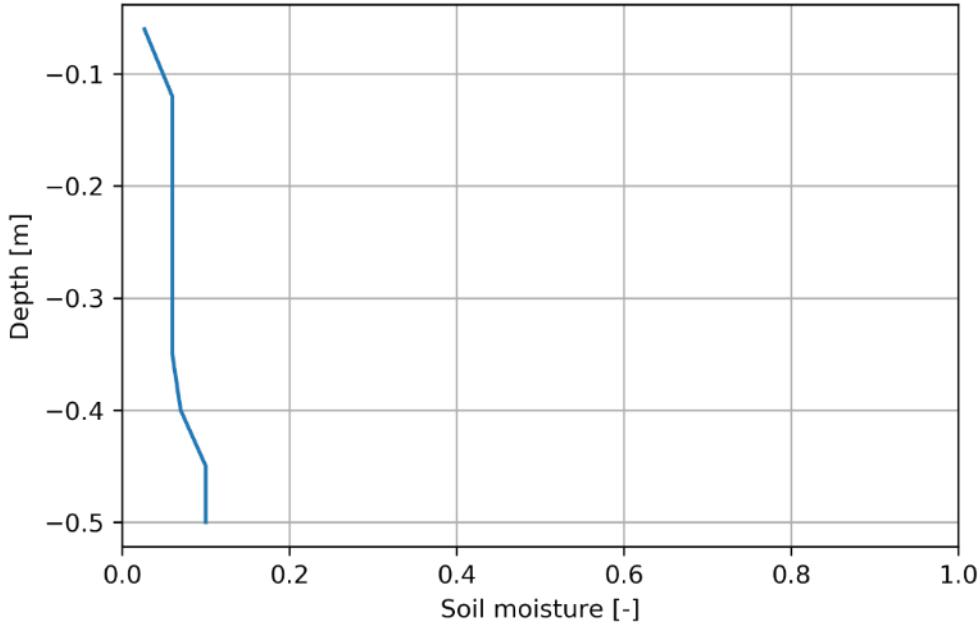
## A.2.1 Parameter Setting

### **Model Setup**

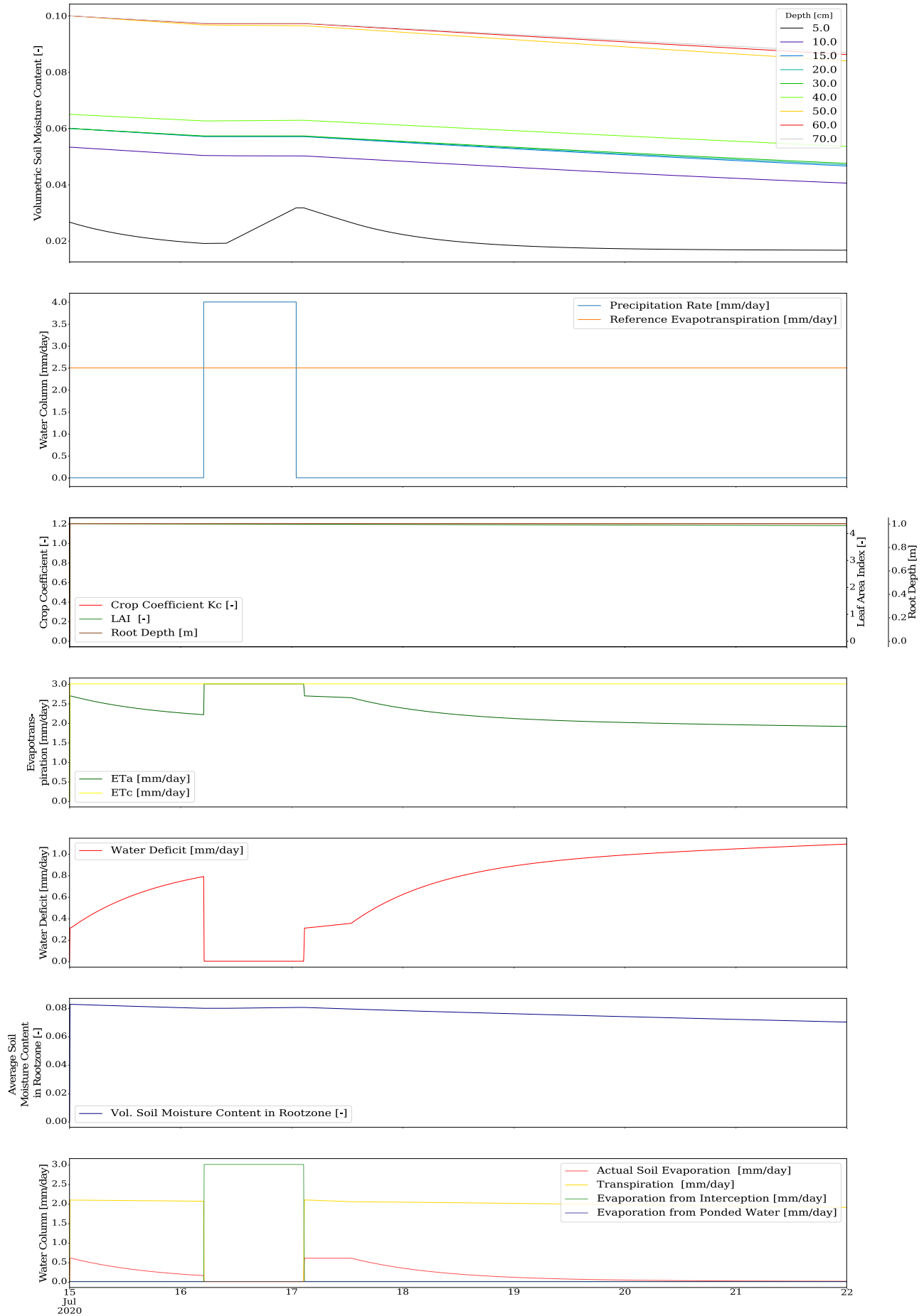
---

Irrigation Method	Sprinkler
Crop	Maize
Vegetation Start Date	1st of May
Simulation Start Date	15th of June
Simulation End Date	22 of June
Groundwater Table	4.1 m

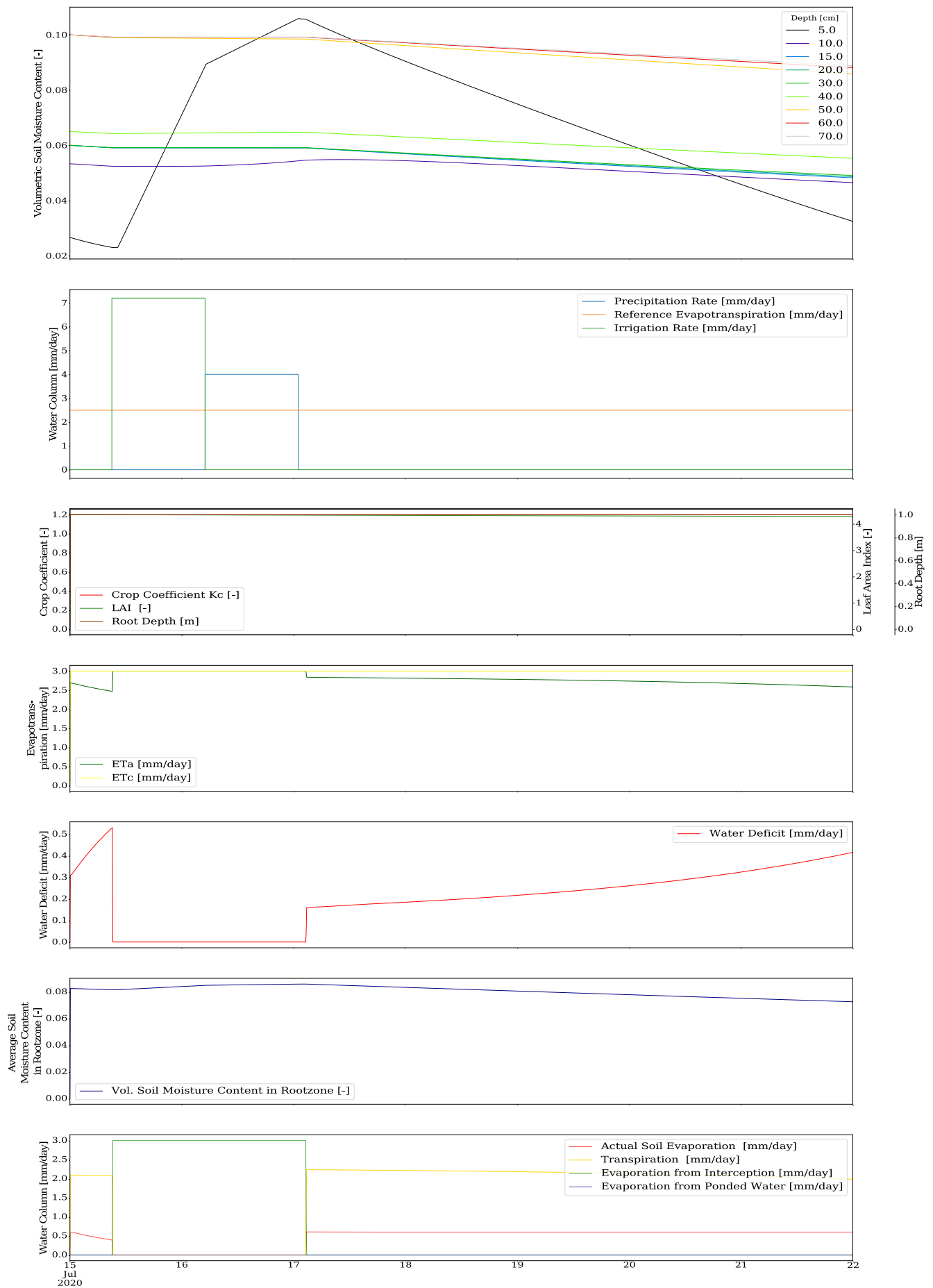
Initial Soil Moisture Content:



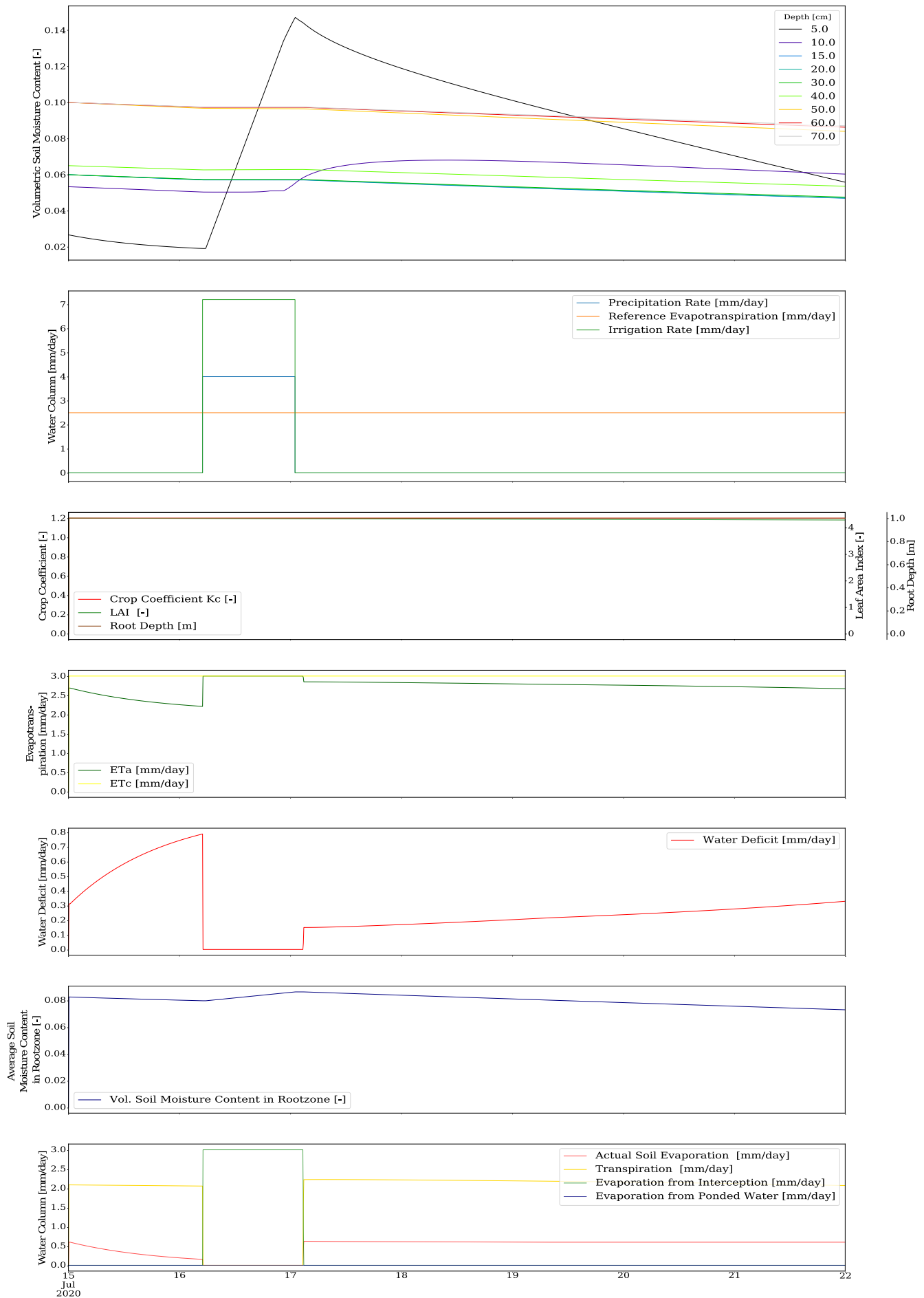
## A.2.2 Simulation Run without Irrigation



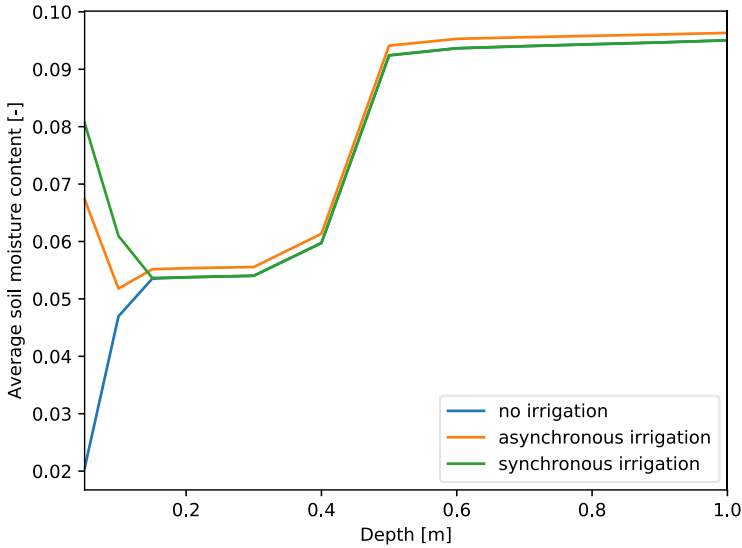
### A.2.3 Simulation Run with Irrigation just before Precipitation (Asynchronous)



## A.2.4 Simulation Run with Irrigation during Precipitation (Synchronous)



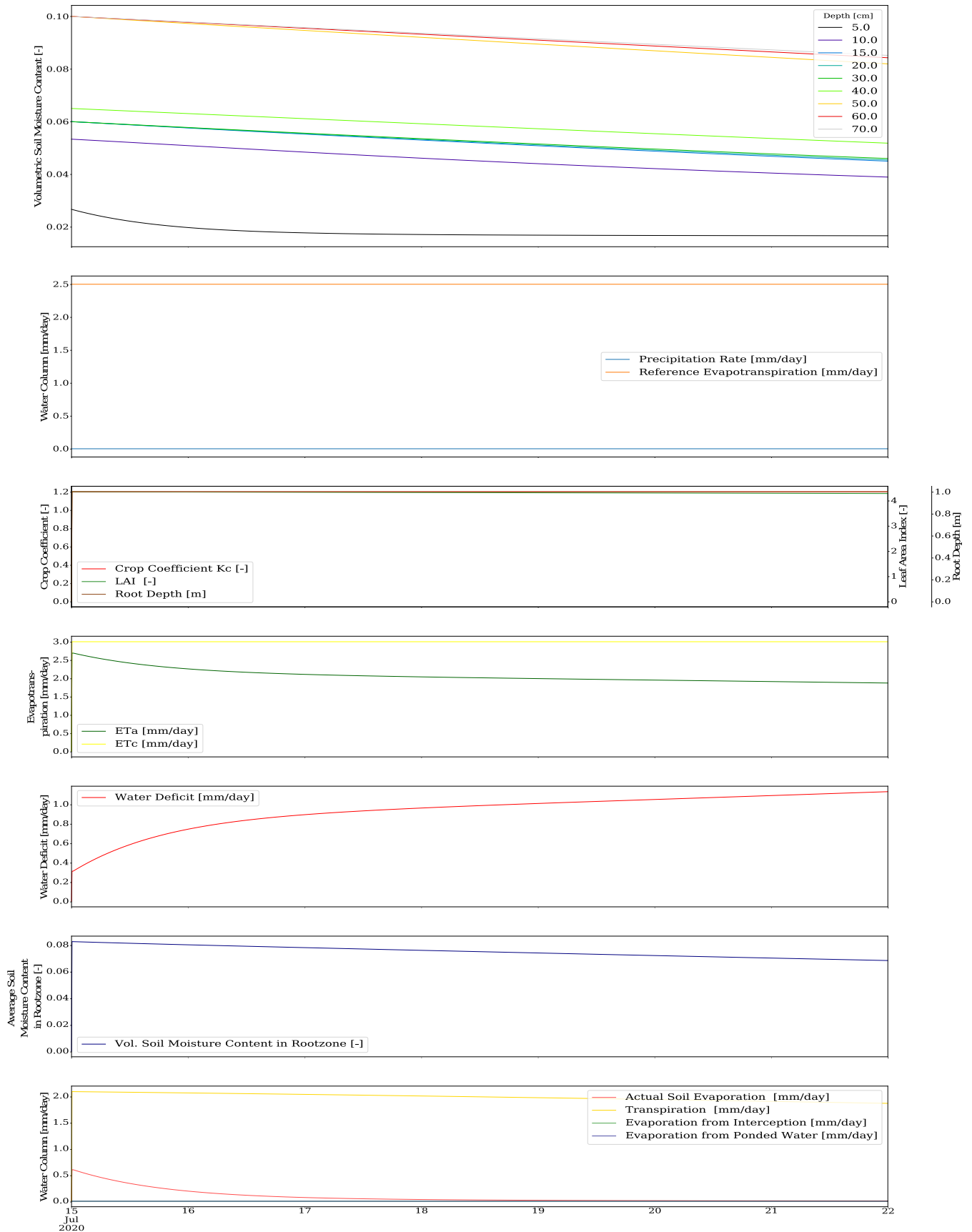
### A.2.5 Comparison of Behaviours - Average Soil Moisture Content over Depth



### A.3 System Behaviour Analysis: Large vs. Small Irrigation Intervals

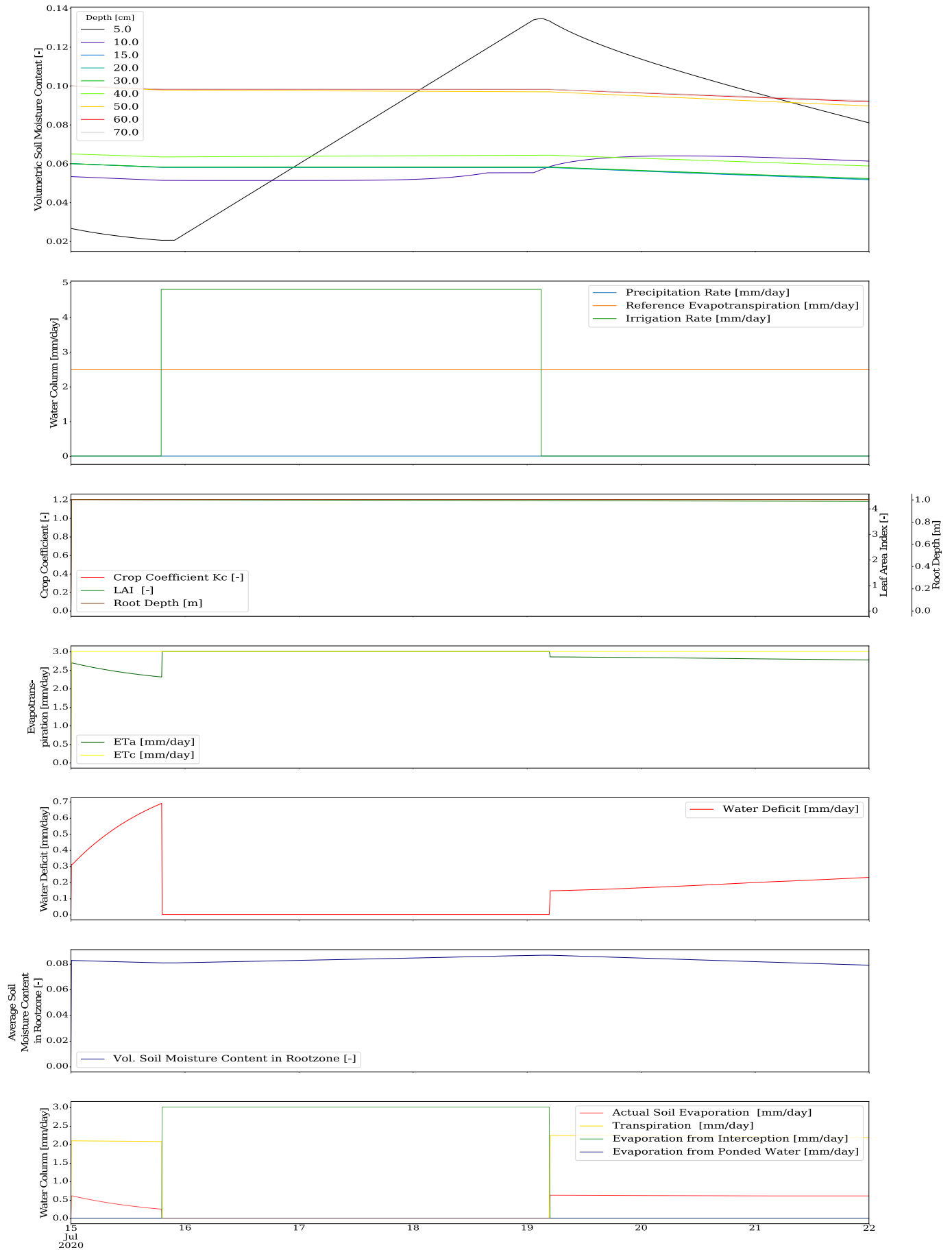
Model parameters are identical to A.1.2.

#### A.3.1 Simulation Run without Irrigation

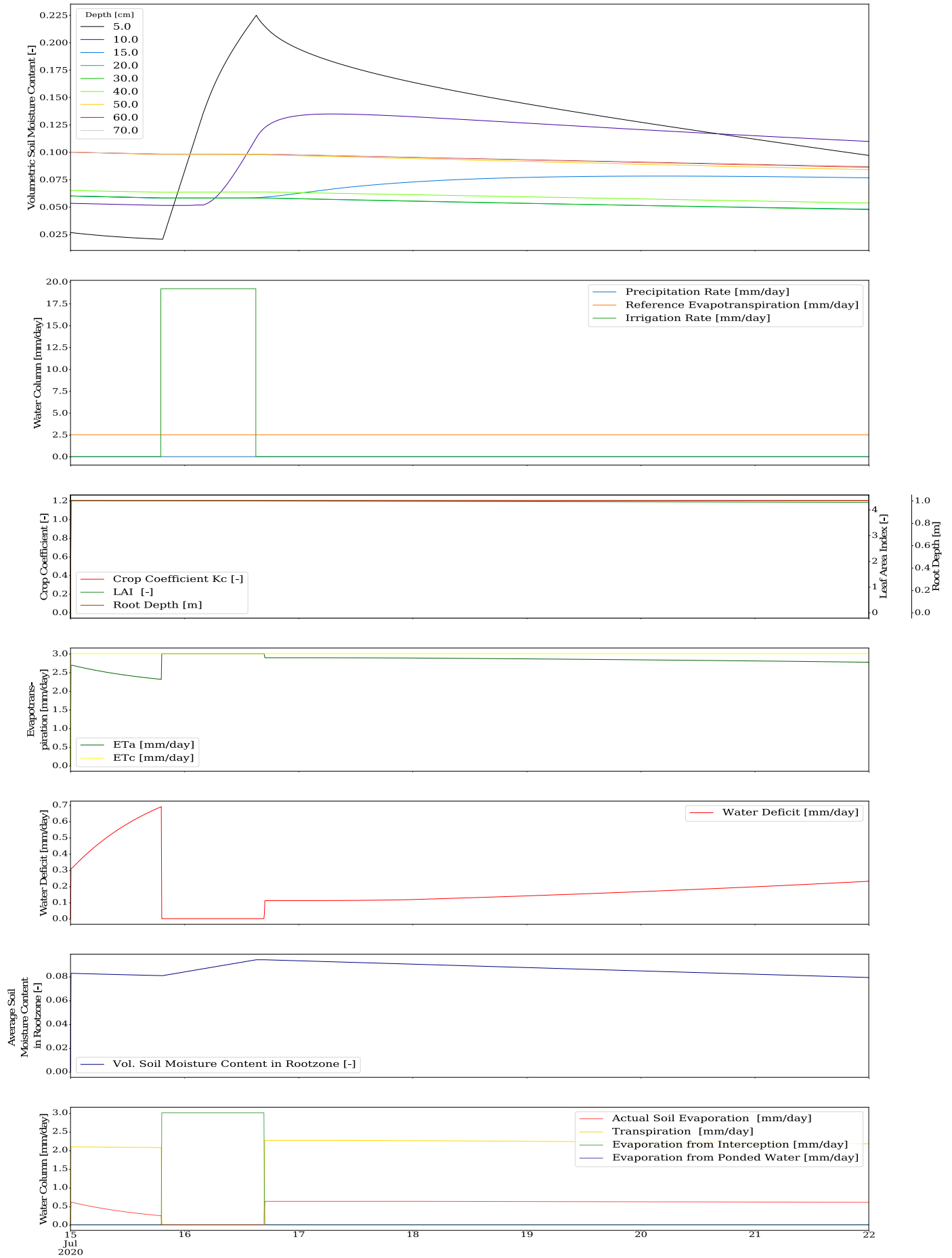




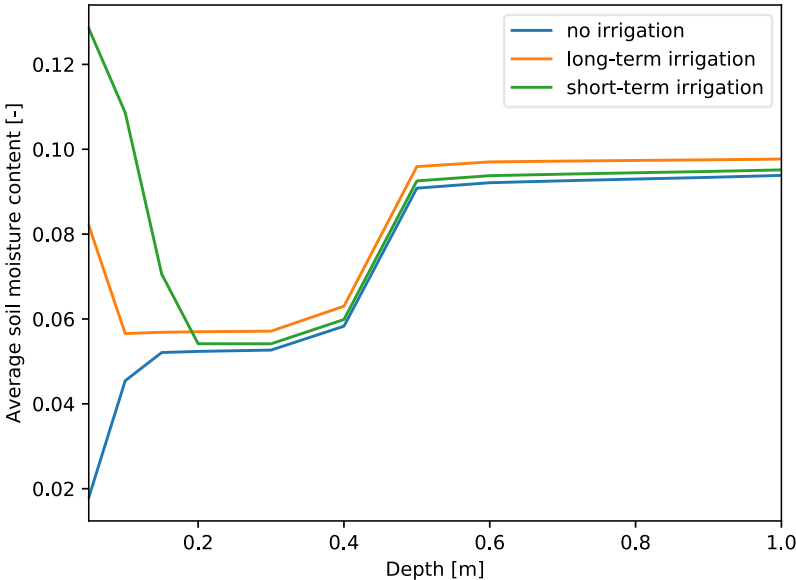
### A.3.2 Simulation Run – Large Irrigation Interval



### A.3.3 Simulation Run – Short Irrigation Interval



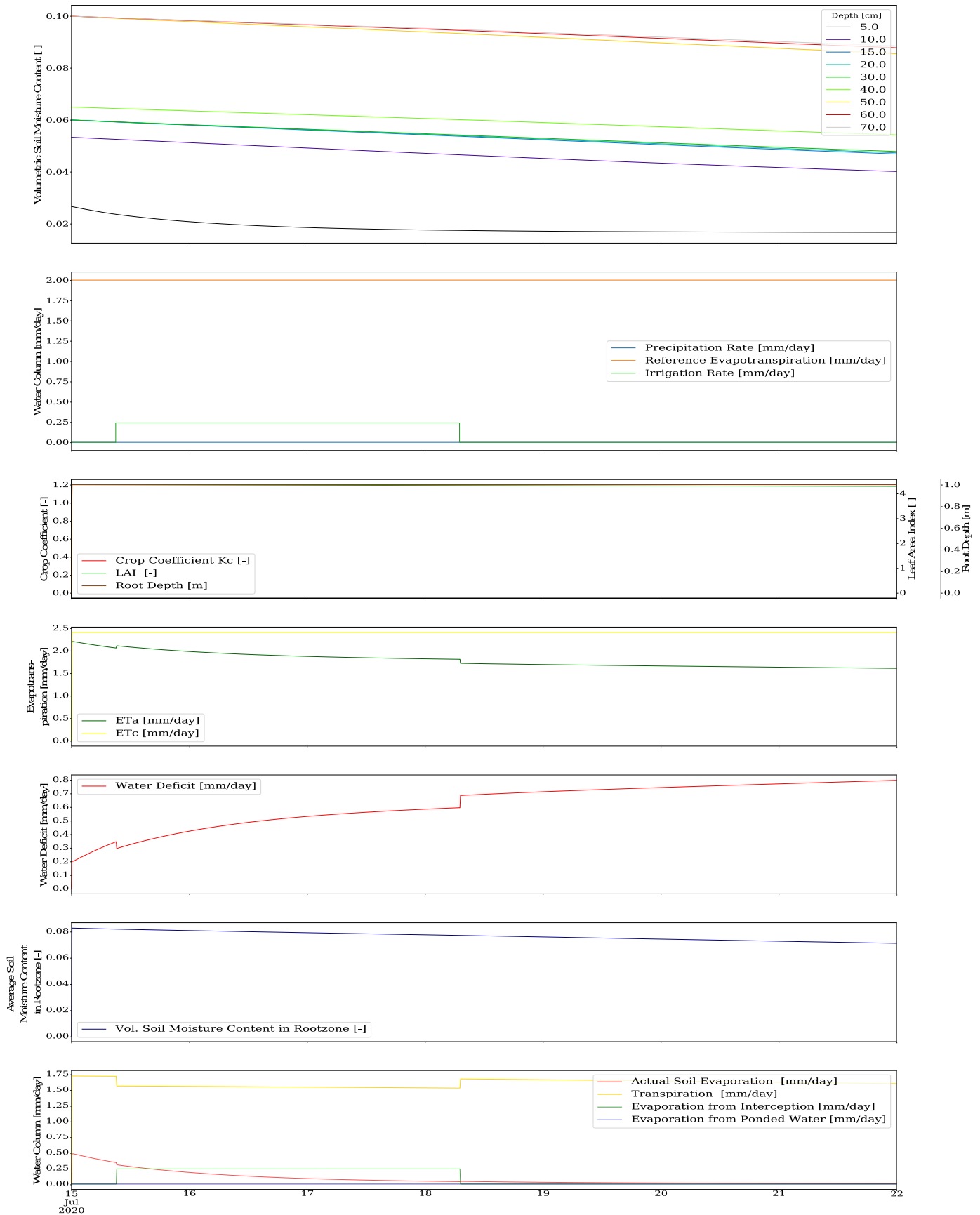
### A.3.4 Comparison of Behaviours - Average Soil Moisture Content over Depth



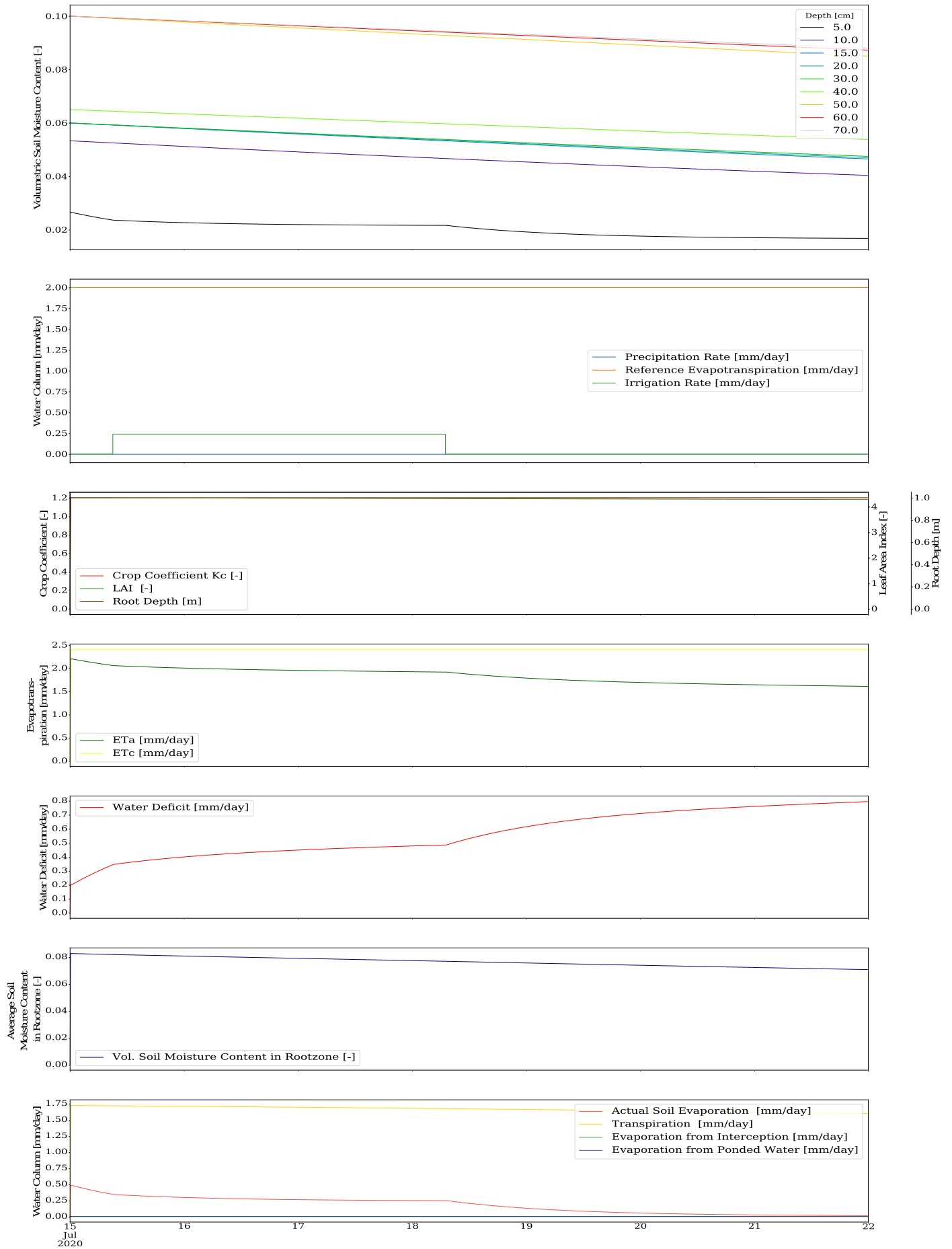
## A.4 System Behaviour Analysis: Sprinkler vs. Drip Irrigation

Model parameters as in A.1.2.

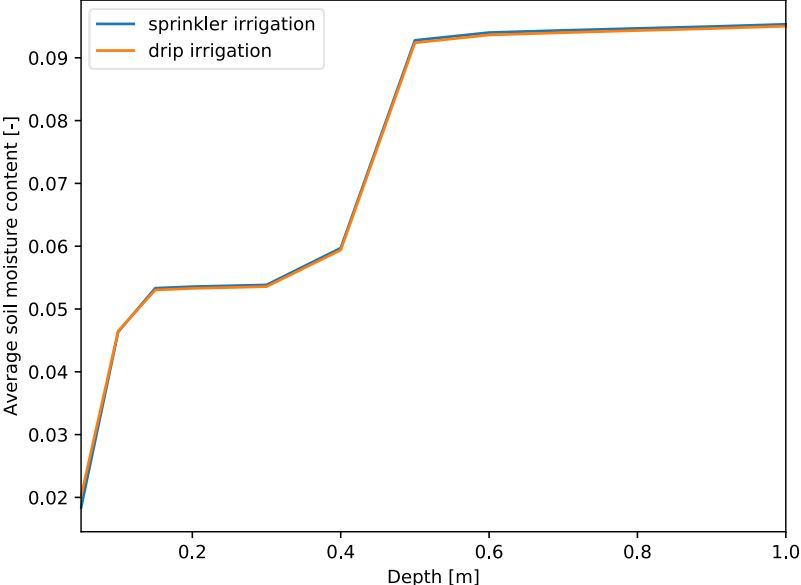
### A.4.1 Simulation Run – Sprinkler Irrigation



## A.4.2 Simulation Run – Drip Irrigation



### 7.1.1 Comparison of Behaviours – Sprinkler vs. Drip Irrigation

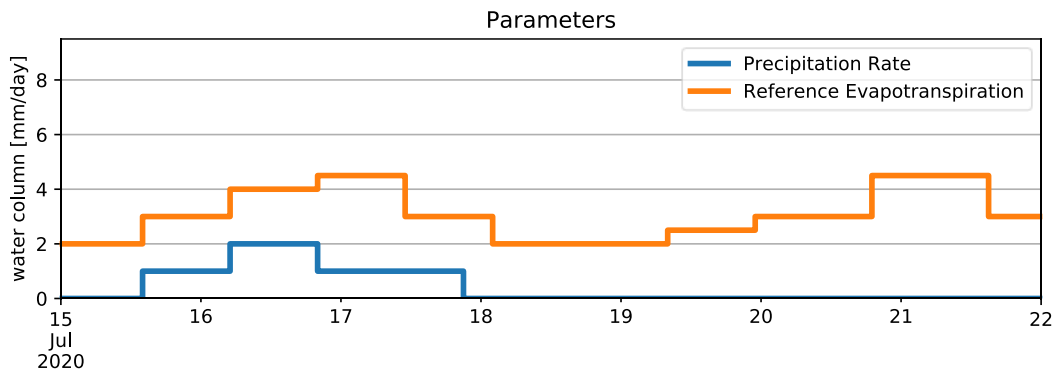


## A.5 Optimization – Single Irrigation Event

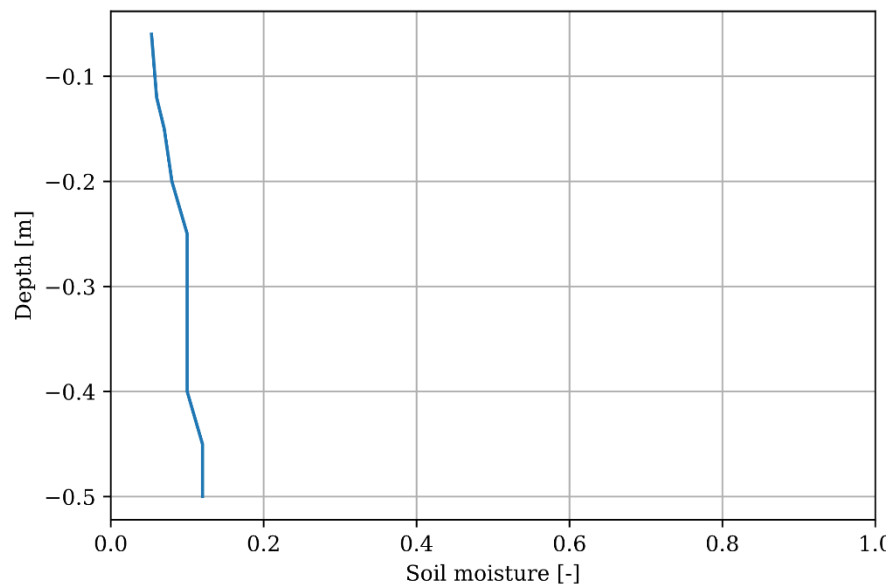
### Model Setup

Irrigation Method	Sprinkler
Crop	Maize
Vegetation Start Date	1st of May
Simulation Start Date	15th of June
Simulation End Date	22 of June
Groundwater Table	4.1 m

### Weather Data – Artificial:



### Initial soil moisture conditions:



## Optimization Setup

---

Optimal Soil Moisture Level	11%
Optimization Interval	7 days
Lower Stress Level	14%
Upper Stress Level	8%

### Control Variables Ranges:

Upper Limit Start Time	$\triangleq$ total timesteps - irrigation interval maximum
Lower Limit Start Time	0 h
Upper Limit Pumping Rate	2 mm/h
Lower Limit Pumping Rate	0
Range Irrigation Interval [h]	4-96

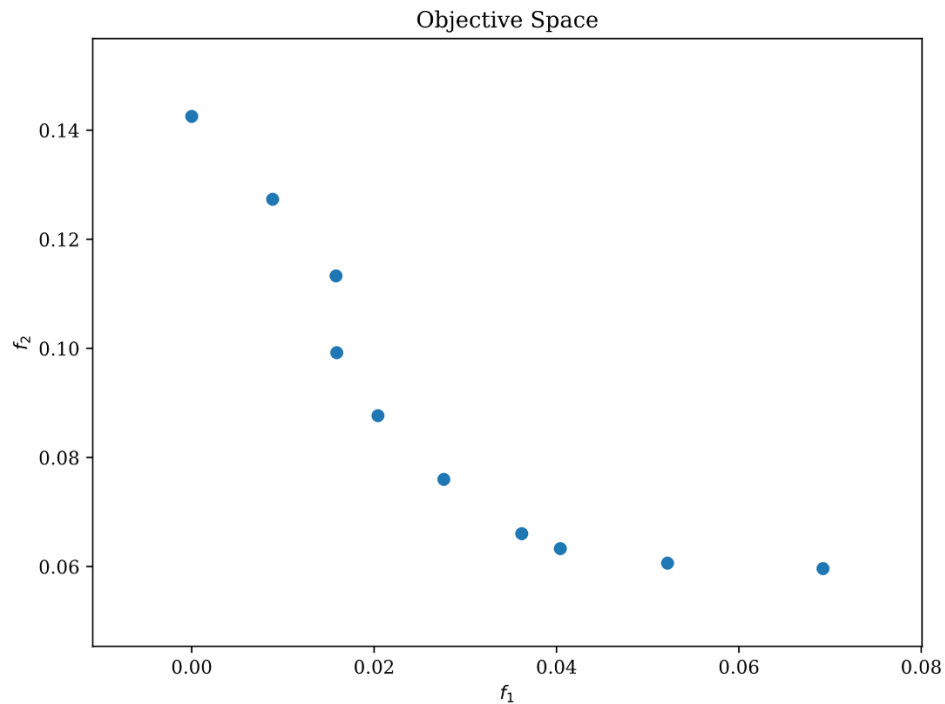
### NSGA-II:

Population Size	10
Offspring Size	10
SBX distribution index $\eta$	1
PM distribution index $\eta$	1
Termination Criterium	50 generations

## Results:

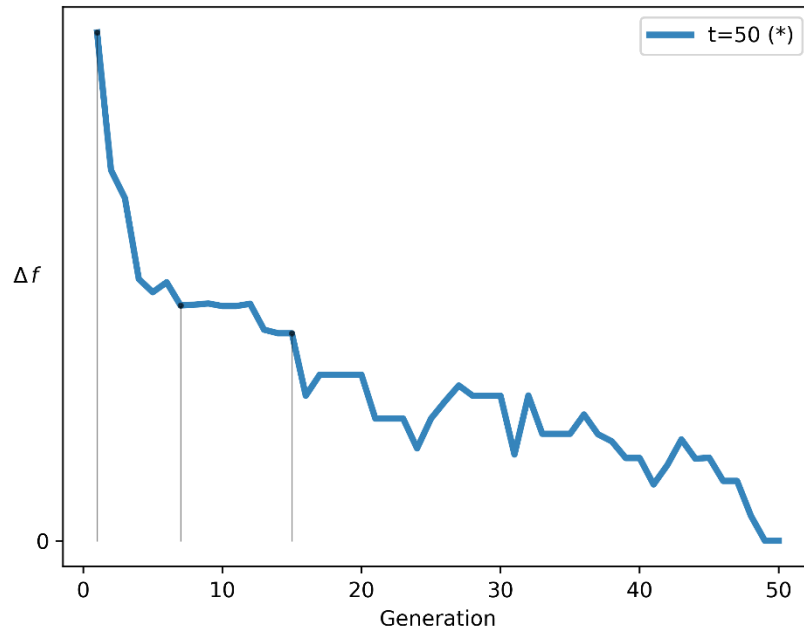
Pareto optimal front:

$f_1$  refers to water consumption objective function (Equation 30),  $f_2$  equals soil-moisture conditions objective function (Equation 31)

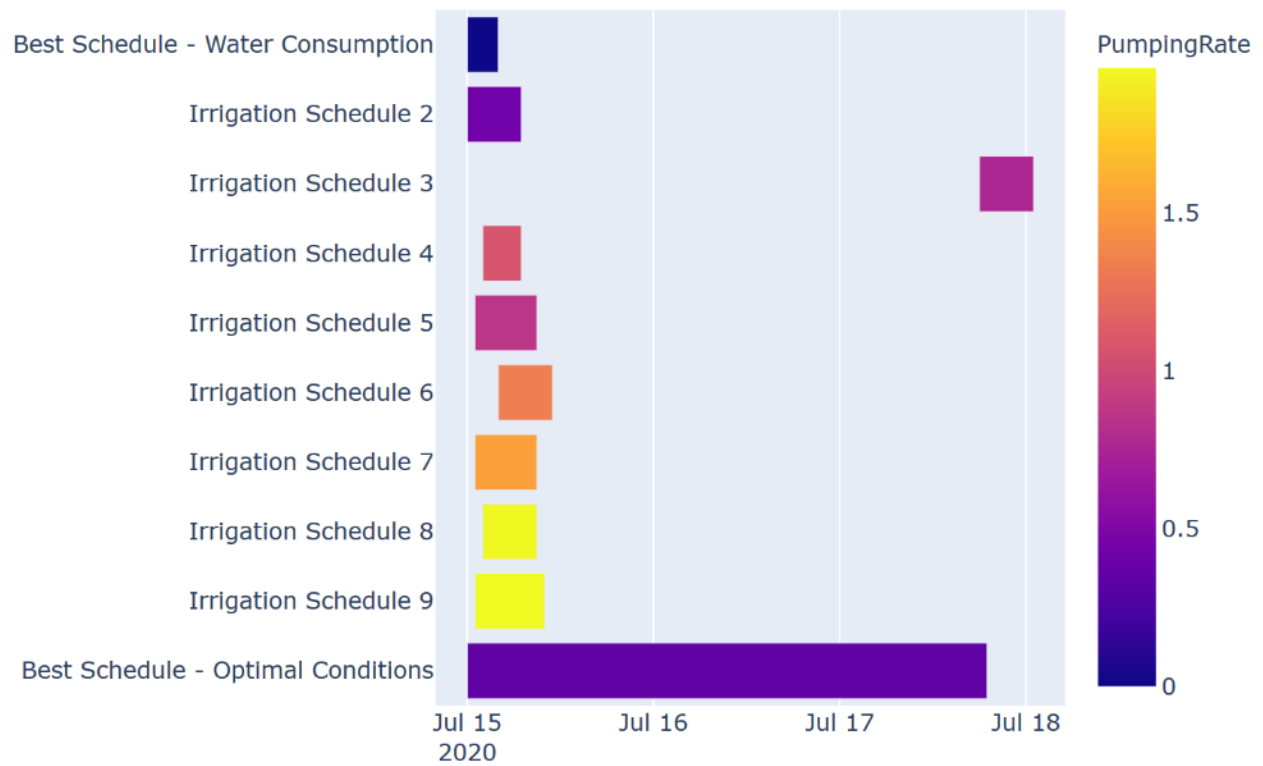




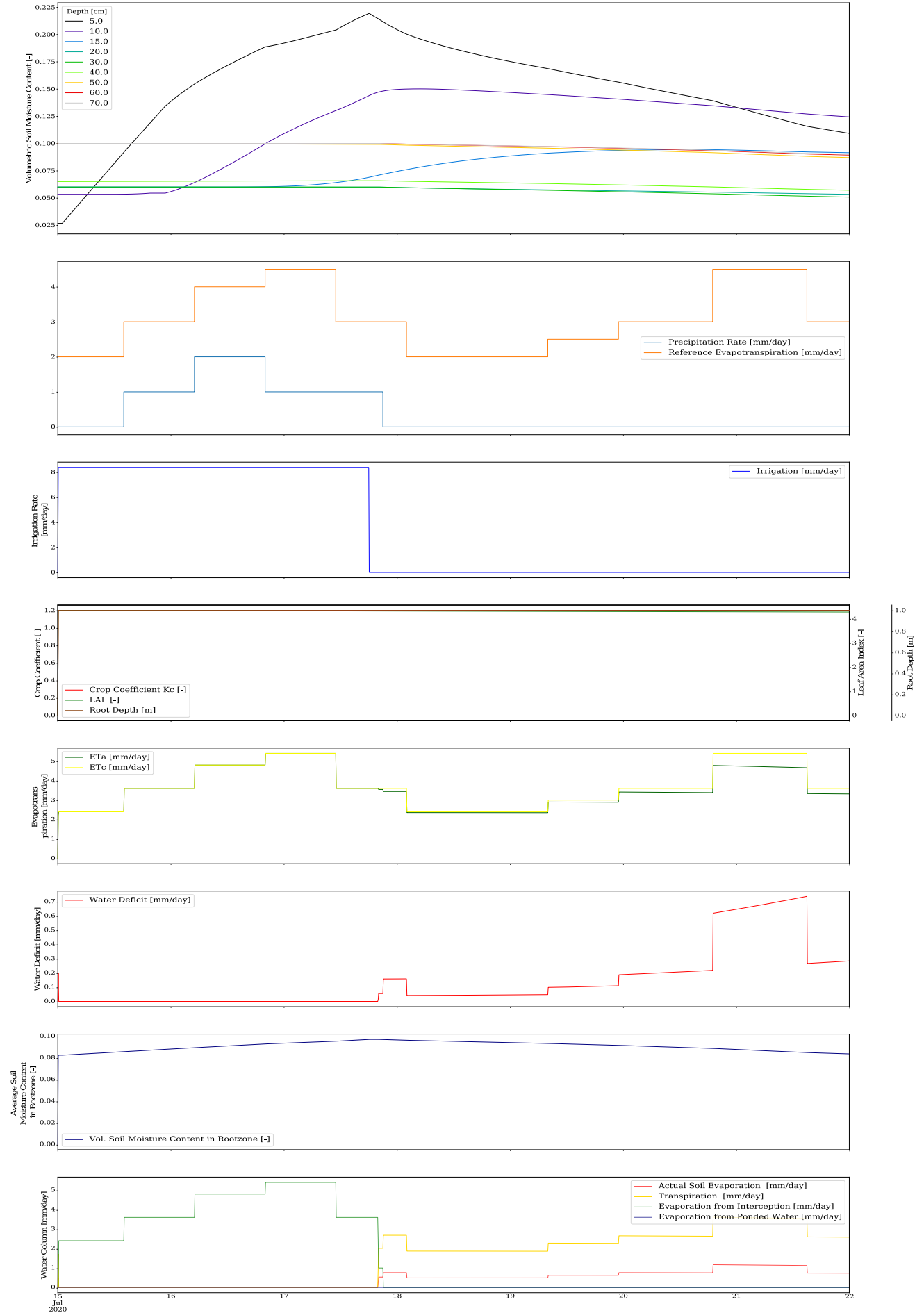
Running Metric:



Overview of resulting schedules:



# Result - best schedule for optimal conditions:



## A.6 Seasonal Optimization – Extremely Dry Year (2018)

### ***Model Setup***

---

Irrigation Method	Sprinkler
Crop	Maize
Vegetation Start Date	1st of May, 2018
Simulation Start Date	1st of May, 2018
Simulation End Date	18 <sup>th</sup> of Sep, 2018
Groundwater Table	4.1 m
Location	Peine, Lower Saxony

### ***Optimization Setup***

#### ***Constants***

---

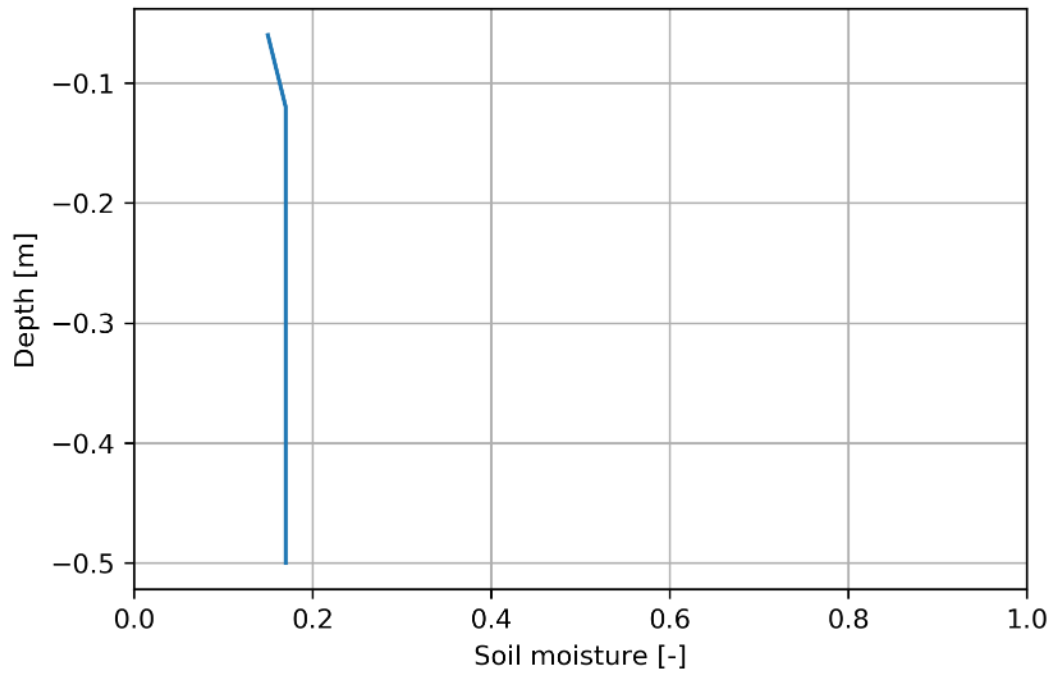
Optimal Soil Moisture Level	11.68%	<i>NSGA-II:</i>	
Optimization Interval	7 days	Population Size	10
Maximum Pumping Rate	2 mm/h	Offspring Size	10
Minimum Watering Depth	10 cm	SBX distribution index 1	
		PM distribution index 1	
<i>Control Variables Ranges:</i>			
Upper Limit Start Time	$\triangleq$ total timesteps - irrigation interval maximum		
		<i>Manual Sampling: (indicator based)</i>	
Lower Limit Start Time	0 ( $\triangleq$ current time)	Start Time Range [h]	0 - 12
		Range Irrigation Interval [h]	corresponding to seasonal value
Upper Limit Pumping Rate	$\triangleq$ max pumping rate	Pumping Rate Range	0.3 - 1.0 mm/h
Lower Limit Pumping Rate	0	Stress Indicator	0.2 mm/h

#### ***Seasonal Dependent Variables***

---

Parameter	Initial Stage	Crop Development	Flowering	Mid-season	Late season
Days after seeding	30	70	90	105	130
Depletion Factor	0.5	0.5	0.5	0.5	0.8
Lower Stress Level	8.27%	8.27%	8.27%	8.27%	3.31%
Upper Stress Level	13.76%	13.76%	13.76%	13.76%	13.76%
RAW [cm/m root depth]	8.27	8.27	8.27	8.27	13.23
Range Irrigation Interval [h]	4-36	48-96	48-96	48-96	48-96
No. of generations	4	7	8	7	4
Objective Weights f1/f2	0.5/0.5	0.3/0.7	0/1	0.3/0.7	0.4/0.6

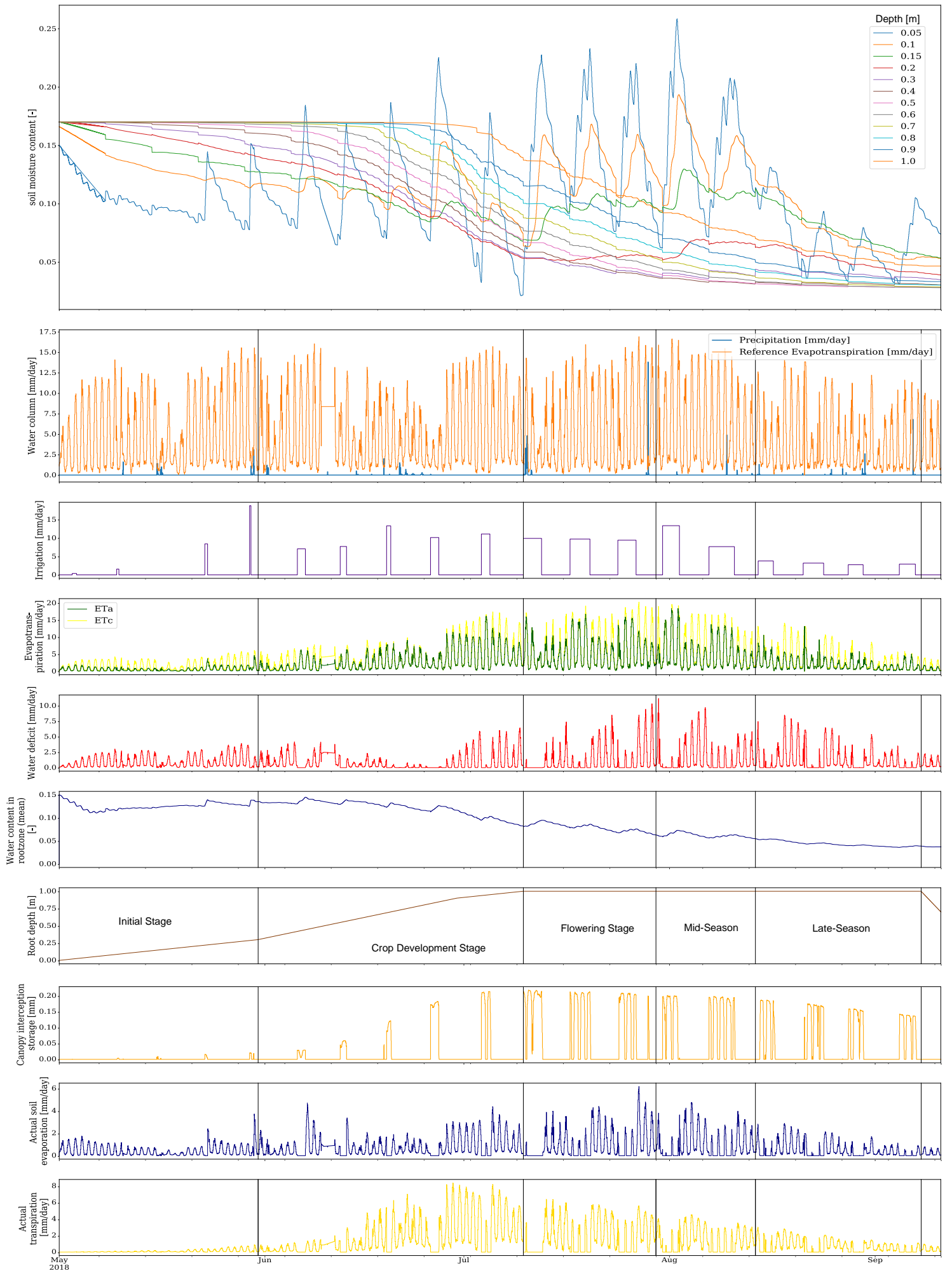
Initial soil moisture content:



**Results:**

Total water used: 247.6 mm

A result overview is given on the following page.



## A.7 Seasonal Optimization – Moderate Year (2017)

### ***Model Setup***

Irrigation Method	Sprinkler
Vegetation Start Date	1st of May, 2017
Simulation Start Date	1st of May, 2017
Simulation End Date	18 <sup>th</sup> of Sep, 2017
Groundwater Table	4.1 m
Location	Peine, Lower Saxony

### ***Optimization Setup***

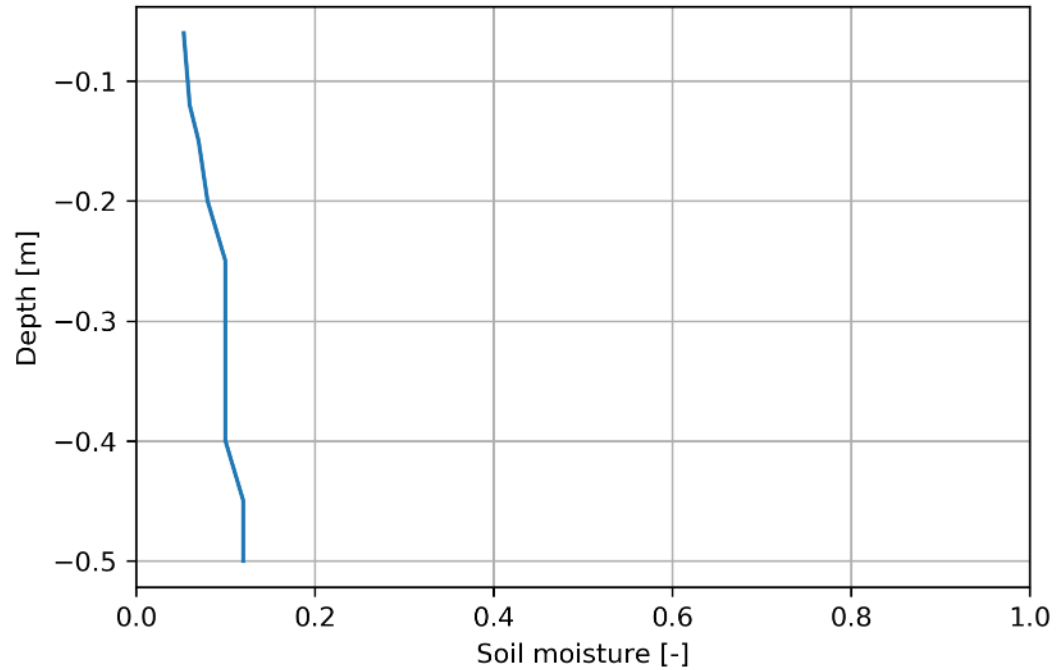
#### ***Constants***

Optimal Soil Moisture Level	11.68%	<i>NSGA-II:</i>	
Optimization Interval	7 days	Population Size	10
Maximum Pumping Rate	2 mm/h	Offspring Size	10
Minimum Watering Depth	10 cm	SBX distribution index 1	
<i>Control Variables Ranges:</i>		PM distribution index 1	
Upper Limit Start Time	$\triangleq$ total timesteps - irrigation interval maximum	<i>Manual Sampling: (indicator based)</i>	
Lower Limit Start Time	0 ( $\triangleq$ current time)	Start Time Range [h]	0 - 12
Upper Limit Pumping Rate	$\triangleq$ max pumping rate	Range Irrigation Interval [h]	corresponding to seasonal value
Lower Limit Pumping Rate	0	Pumping Rate Range	0.3 - 1.0 mm/h
		Stress Indicator	0.2 mm/h

#### ***Seasonal Dependent Variables***

<b>Parameter</b>	<b>Initial Stage</b>	<b>Crop Development</b>	<b>Flowering</b>	<b>Mid-season</b>	<b>Late season</b>
Days after seeding	30	70	90	105	130
Depletion Factor	0.5	0.5	0.5	0.5	0.8
Lower Stress Level	8.27%	8.27%	8.27%	8.27%	3.31%
Upper Stress Level	13.76%	13.76%	13.76%	13.76%	13.76%
RAW [cm/m root depth]	8.27	8.27	8.27	8.27	13.23
Range Irrigation Interval [h]	4-36	48-96	48-96	48-96	48-96
No. of generations	4	7	8	7	4
Objective Weights f1/f2	0.5/0.5	0.5/0.5	0/1	0.5/0.5	0.5/0.5

Critical initial soil moisture conditions:



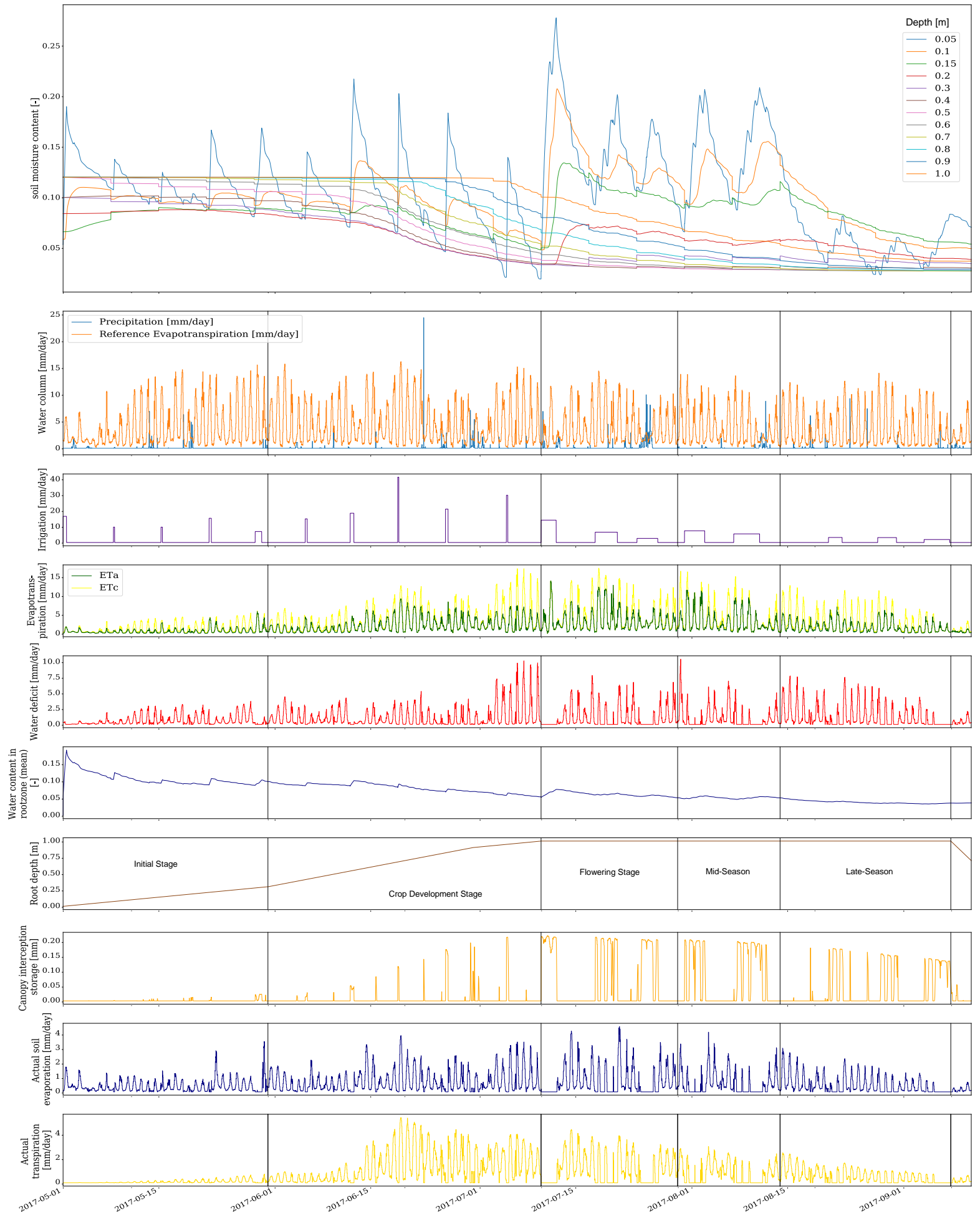
**Results:**

Total water used (from seeding until 25<sup>th</sup> of August): 168.2 mm.

Total water deficit (from seeding until 25<sup>th</sup> of August): 116.9 mm.

Total water deficit during flowering: 21.2 mm.

An overview is depicted on the following page.





## A.8 Seasonal Optimization – Pre-seasonal Watering

Parameter setting of model and optimization are identical to 2017 simulation.

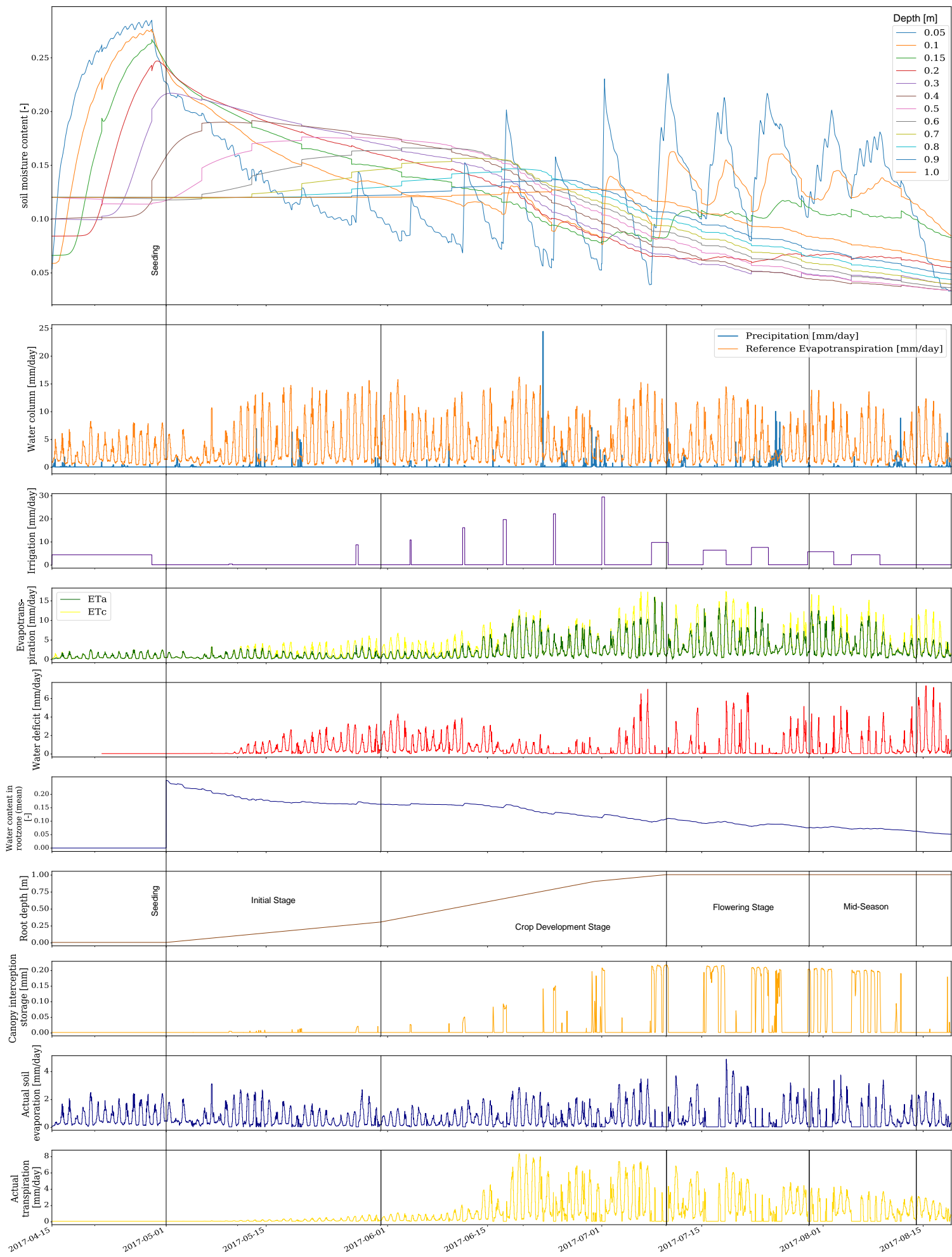
### **Results:**

Before seeding: 60.5 mm of water used.

Total water used (from seeding until 25<sup>th</sup> of August): 205.17 mm.

Total water deficit (from seeding until 25<sup>th</sup> of August): 77.4 mm.

Total water deficit during flowering: 13.2 mm.



## A.9 MIKE SHE Optimization – Integrated Scheduler

Conditions are the same as in the 2018 simulation.

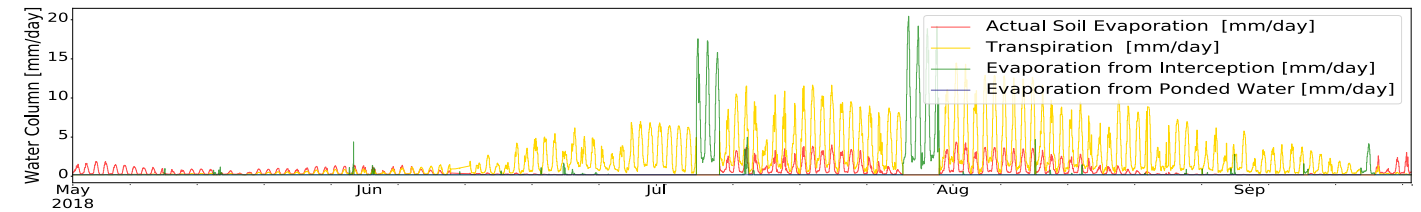
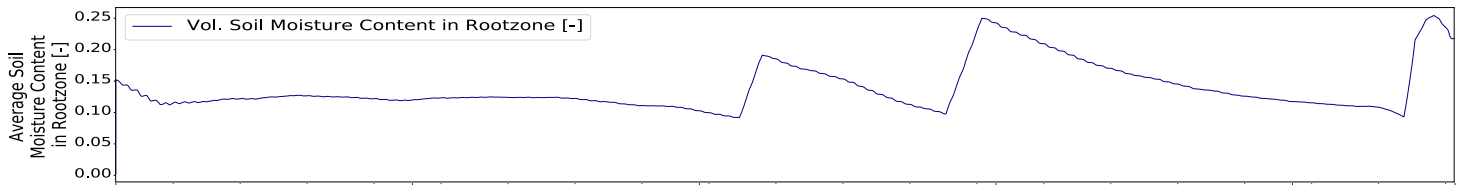
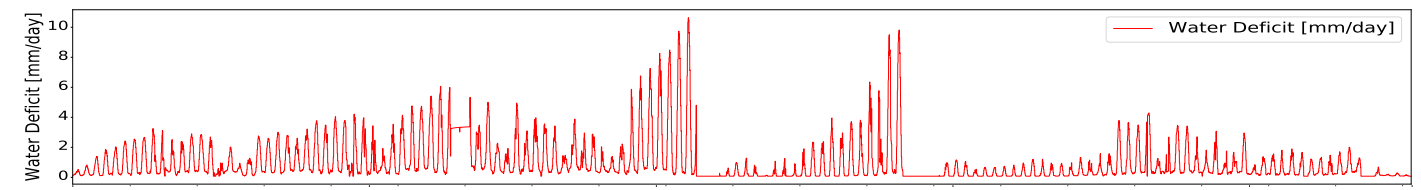
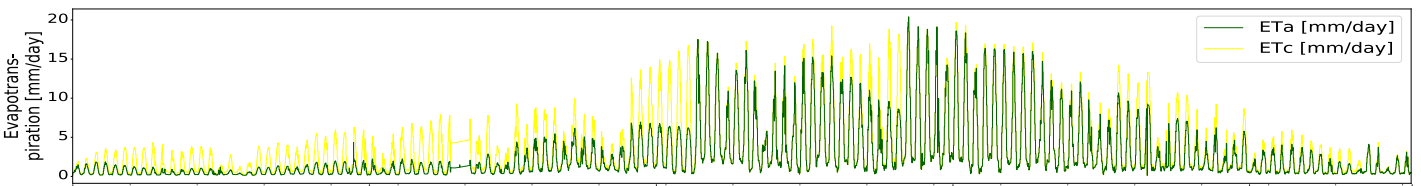
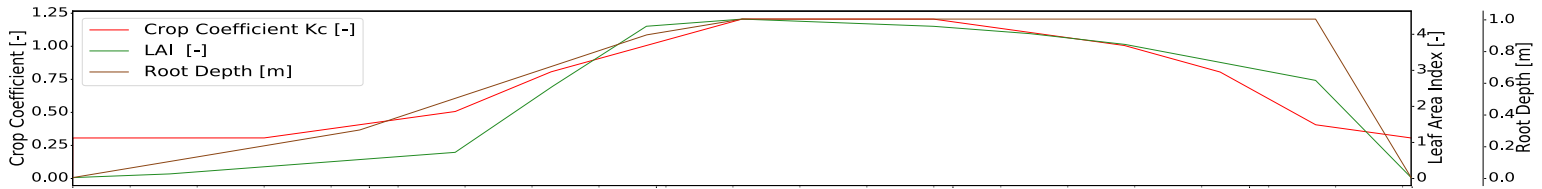
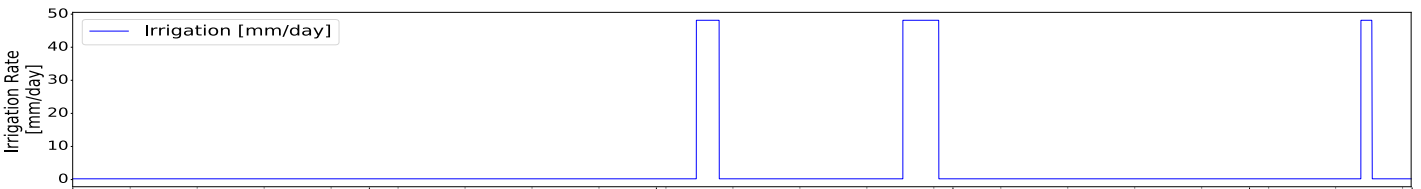
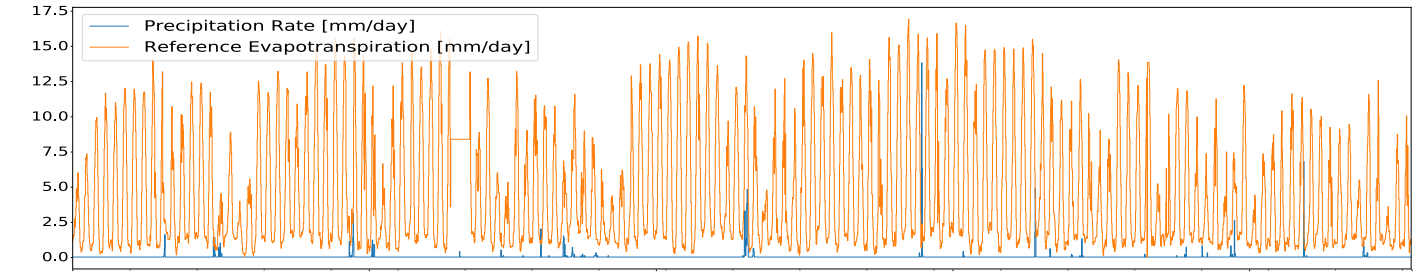
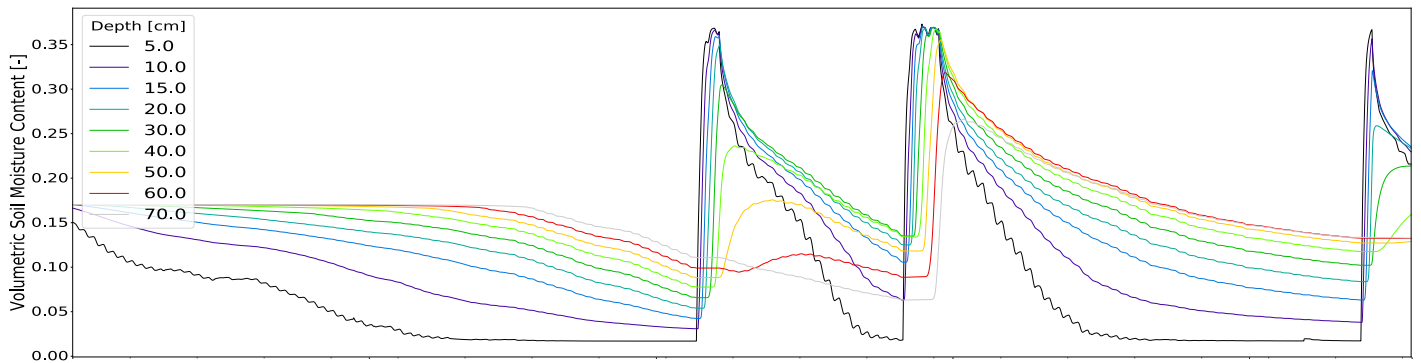
Lower water stress threshold: 50 % TAW.

Upper water stress threshold: 80 % TAW.

### **Results:**

Total water used: 323.7 mm.

Total water deficit: 131.0 mm.



## A.10 Multi-Field Optimization

<b>Model Parameter</b>	<b>Model 1</b>	<b>Model 2</b>
Soil Type	Brown Earth	Brown Earth
Irrigation Method	Sprinkler	Sprinkler
Vegetation Start Date	1st of May, 2018	1st of April, 2018
Simulation Start Date	1st of August, 2018	1st of August, 2018
Simulation End Date	14th of August, 2018	14th of August, 2018

### **Optimization Parameter**

Optimal Soil Moisture Level [% TAW]	11.68	11.68
Lower Stress Level [% TAW]	13.76	13.76
Upper Stress Level [% TAW]	6.82	6.82

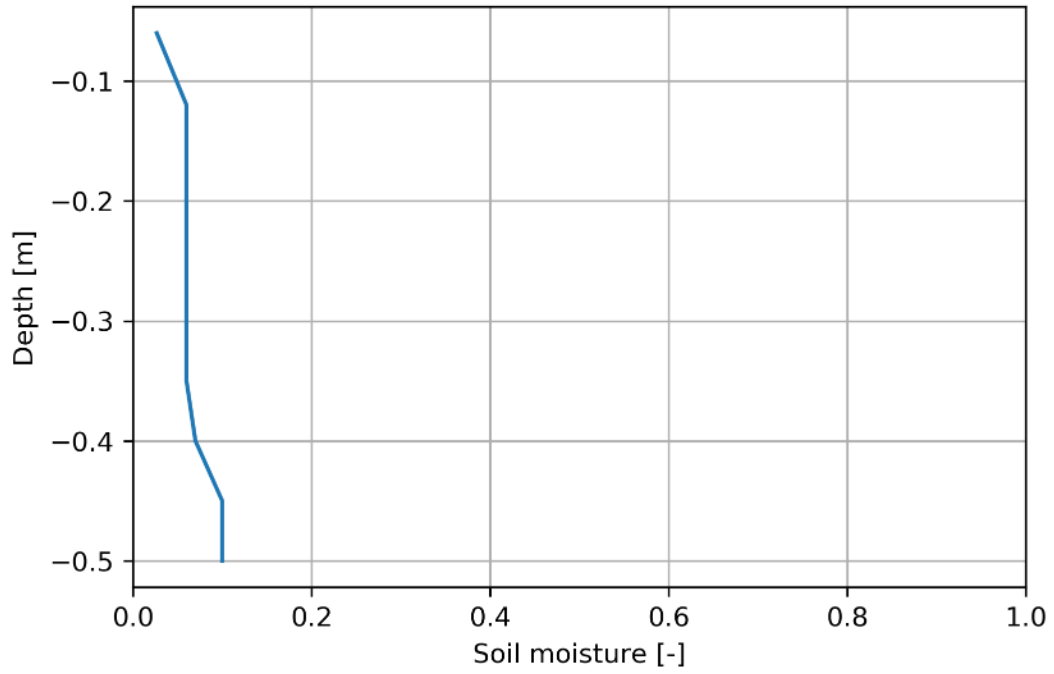
### NSGA-II:

Population Size	20
Offspring Size	20
SBX distribution index $\eta$	1
PM distribution index $\eta$	1
Termination Criterion	20 generations

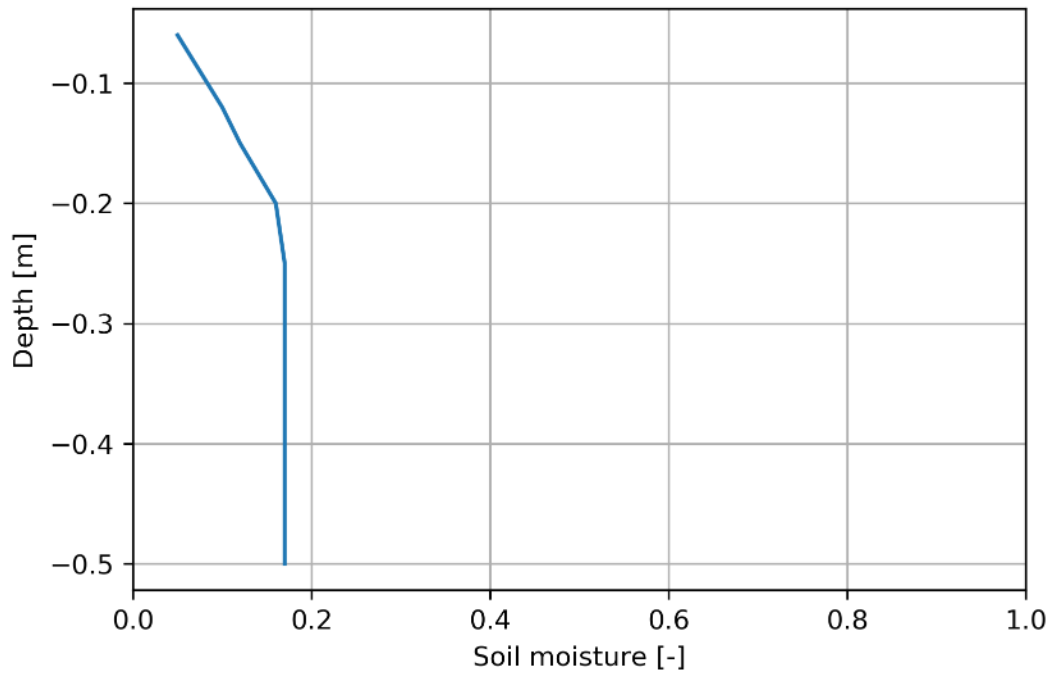
### Control Variables Ranges:

Pumping Rate (constant)	0.4 mm/hour
Upper Limit Start Time [h]	144 ( $\triangleq$ total timesteps - irrigation intervals maximum)
Lower Limit Start Time [h]	0
Range Irrigation Interval Field 1 [h]	4-96
Range Irrigation Interval Field 2 [h]	4-96
Weighting of objective functions	0.5/0.5

Initial Soil Moisture Conditions - Model 1:



Initial Soil Moisture Conditions - Model 2:



## Water Stress Indicator Result:

Field Priority:

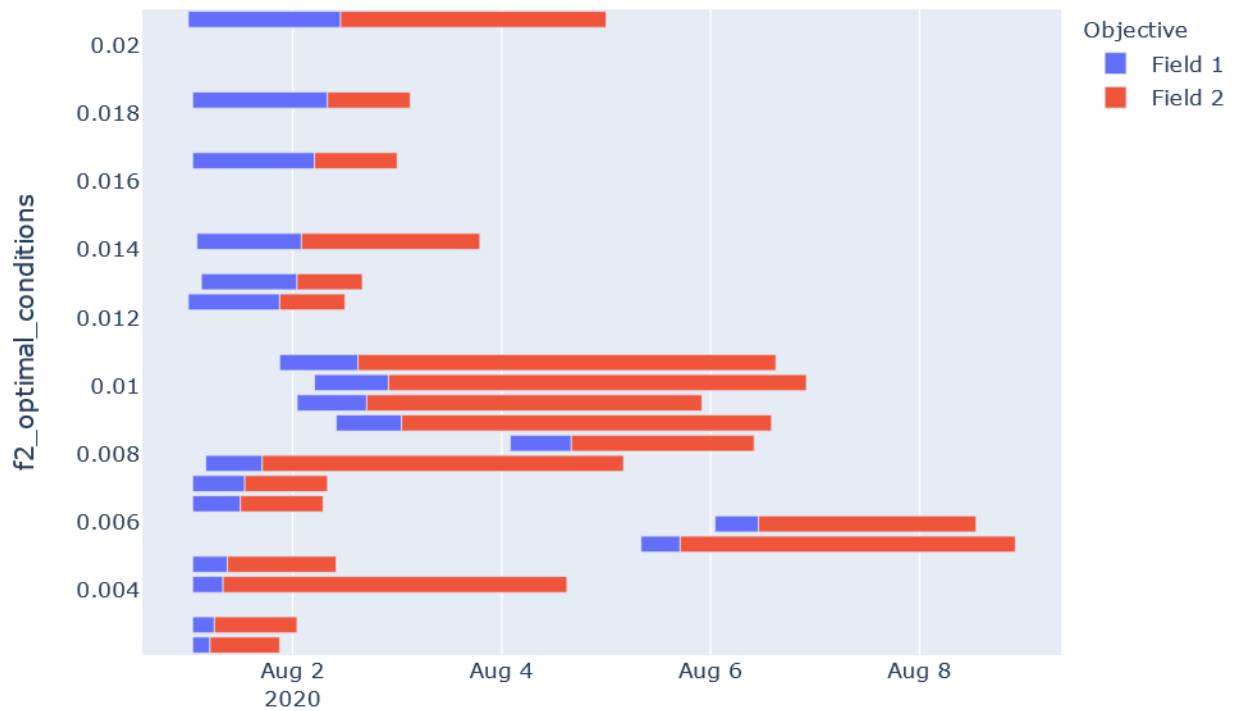
field 1 – water deficit average of 2.069 mm/day

field 2 – water deficit average of 0.3887 mm/day

➔ Field 1 is prioritized.

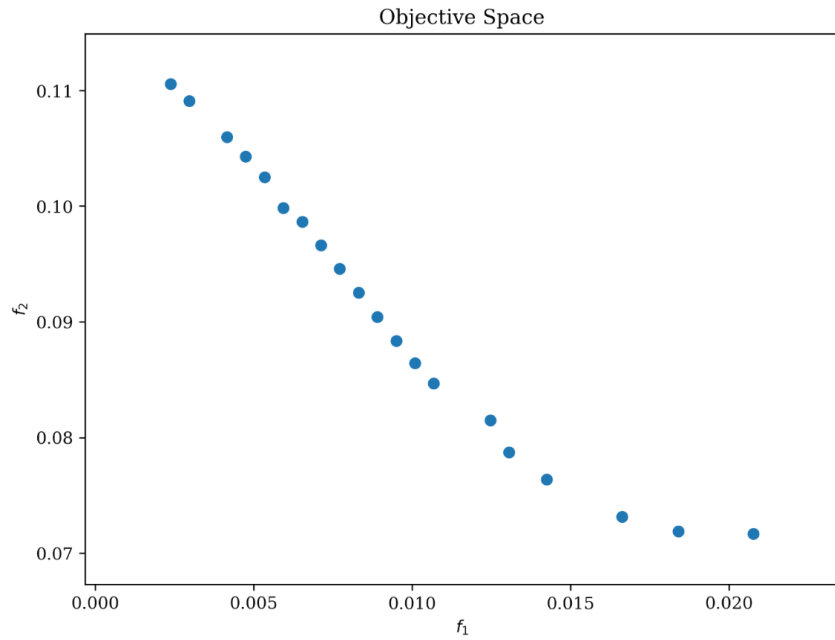
## Optimization Results

Schedule Overview:



### Pareto Optimal Front:

$f_1$  refers to water consumption objective function (Equation 32),  $f_2$  equals soil-moisture conditions objective function (Equation 33).



### Running Metric:

