Determining Crop-Production Functions using Multi-Objective Evolutionary Algorithms

Michael de Paly, Niels Schütze and Andreas Zell

Abstract—The determination of crop production functions which describe the relationship between irrigation water and crop yield under the assumption of optimal irrigation scheduling is a major building block for a more efficient and sustainable water management. In this paper we introduce a methodology to determine the entire crop production function for a given scenario within a single run of a multi-objective evolutionary algorithm. Further we compare the performance of four major algorithms (NSGA-II, NSDE, DEMO, and MO-CMA-ES), and a single-objective approach based on differential evolution on three different scenarios and two different population initialization methods on this problem. We show that the combination of a problem specific initialization with MO-CMA-ES is able to determine crop production functions which are extremely close to actual ones.

I. INTRODUCTION

One of the major challenges of this decade is dealing with the growing water scarcity while achieving and maintaining sufficient food supply. Irrigated agriculture is the largest contributor to water waste and has a large potential for water saving at the same time. Accordingly, the Food and Agriculture Organization of the United Nations (FAO) called for a revolution in water management and usage efficiency [1].

Constrained water supply during the growth period of the crop results in yield losses. The magnitude of these losses heavily depends on the used crop and on the timing and severity of the undersupply. While some crops react with catastrophic yield loss under such circumstances, the reaction of other crops like maize is much less dramatic. This leads to deficit irrigation as proposed in [2], which is focused on maximizing crop per water productivity and consciously accepts reduced yields while undersupplying the crop. Since the sensitivity of the crop to water stress depends on the time when the water stress occurs, careful planning of the irrigations timing and amount (depth) is required. Such planning involves decisions at various stages and levels of the overall water management process.

At the farm management level appropriate crops for the expected seasonal available irrigation water have to be chosen and allocated to available fields [3][4]. The basis for this is the relationship between the allocated irrigation water and the expected crop yield under the assumption of optimal irrigation scheduling. This relationship is described by the crop production function, which is exemplary shown in Fig. 1 for maize and potato for scenarios used in this paper (Scenario A+B). This relationship depends on an abundance of parameters like crop type, soil parameters, climate and in the case of arid areas foremost the irrigation scheduling [5]. At present this relationship is often incorporated into the decision process by extraordinary simple methods like experience of the farmer and rules of thumb. One major reason for this is the lack of suitable methods which are able to determine the crop production function within reasonable computational time for a given specific combination of field, climate and crop. The response of the crop yield to water stress has to be simulated for each point along the crop production function under the assumption of optimal irrigation scheduling. This means that for every point of this function a complex simulation based optimization problem has to be solved.

In combination with simplified and discretized water balance models with coarse grained irrigation decisions, dynamic programming (DP) approaches as proposed in [6], [7] are targeted to solve this problem. While these approaches have been improved over time to allow for daily irrigation decisions as in [9], they still rely on a coarse grained discretization to shrink the state space to a size that can be tackled by DP within reasonable computational time. This leads to a limited applicability of the results and prevents the use of more complex but more substantive models. A novel approach based on neuro-dynamic programming has just recently been proposed in [10], which makes use of machine learning techniques to approximate the cost function for DP, making it able to work in a continuous state space. This approach suffers from the high cost for the simulation runs needed to train the cost function approximator. For the determination of optimal irrigation schedules for a specific given amount of irrigation water, which marks a single point along the crop production function, much more efficient evolutionary algorithm approaches have been proposed in [11], [12], [13]. Using evolutionary algorithms for irrigation scheduling also circumvents almost all limitations of the underlying simulation model and has been shown to achieve excellent irrigation results when combined with more complex models like FIM [14].

An interesting aspect of evolutionary algorithms is their ability to solve multi-objective optimization problems, which means solving optimization problems with two or more objectives at the same time. For these problems we are interested in the pareto front consisting of solutions which
form a compromise between the often contradicting objectives, where no objective can be further improved without impairing an other one. In our context the crop production function can be seen as such a pareto front, where minimizing the irrigation depth is one objective and maximizing the yield or reducing the yield loss is the other one. Each point of this front is defined by an optimal irrigation schedule for a certain amount of irrigation water and the corresponding yield respectively yield loss.

In the first sections of this paper we introduce the methods needed to determine an entire pareto optimal crop production function for a given crop within a single run of an MO-EA. The second part of the paper is devoted to a systematic evaluation of the performance of four MO-EA (NSGA-II, NSDE, DEMO, and MO-CMA-ES) on this problem.

II. METHODS

A. Water Balance Model

To simulate the water balance in the field we used the water balance model described in [8]. The basic idea of water balance models is the consideration of the soil as a storage system, which is characterized by the relative soil moisture $\Theta_i$ and depth of the root zone $D_i$ for each day $i$. The upper limit of the storage capacity is defined by the field capacity $fc$. Excess water that would result in $\Theta_i > fc$ can not be held and is lost due to surface runoff or percolation. The lower limit is defined by the permanent wilting point $pwp$, which defines the minimal necessary soil moisture to prevent unrecoverable wilting of the crop resulting in a complete yield loss. Water transport processes in the field are modeled as sources and sinks as given in:

$$\Theta_i = \min(fc, \max(pwp, \frac{\Theta_{i-1}D_{i-1} + I_i + P_i - AET_i + \Theta_0(D_i - D_{i-1})}{D_i})),$$

(1)

where $P_i$ is the precipitation, $I_i$ is the irrigation water, and $AET_i$ is the actual evapotranspiration, which is the sum of evaporation and transpiration by the plants. $AET_i$, which depends on the type of the crop and the potential evapotranspiration $PET_i$, is calculated according to:

$$AET_i = \begin{cases} PET_i, & \Theta_iD_i \leq (1 - p(PET_i)) \\ \frac{\Theta_iD_i}{1 - p(PET_i)} & \Theta_iD_i < (1 - p(PET_i)) \\ (fc - pwp)D_i & \end{cases}$$

(2)

where $p(PET_i)$ is the crop dependent soil water depletion factor. This factor describes the relative amount of soil water the given crop can extract before it experiences water stress depending on $PET$. Equation (1) and (2) can be solved iteratively for each day of the growth period and allow a simulation of the development of $\Theta_i$ and $AET_i$ for the entire growth period. $AET_i$ can then be used as input for a subsequent yield prediction model as described in the next section.

Figure 2 shows an example of the model output for an irrigation schedule with 6 irrigation events (days 1, 10, 46, 56, 64, and 79), which are shown as bars in each subfigure. The development of $\Theta$ is shown in the upper subfigure and $PET$ and $AET$ as grey respectively black line in the middle. Gaps between $PET$ and $AET$ cause water stress and result in reduced yields. The yield loss for the given scenario which uses maize as crop is 0.38.

B. Yield Prediction

To obtain a yield estimate we used a multiplicative approach based on the yield prediction model published by the FAO in [15], which calculates the actual yield $Y_s$ for a given crop according to:

$$Y_s(I) = 1 - K_y(s) \frac{AET_s(I)}{PET_s}$$

(3)

where $PET_s$ is the integrated potential evapotranspiration for all days of the growth stage $s$. $AET_s(I)$ is the integrated actual evapotranspiration which is subject to the irrigation $I$ which is a vector of the irrigation volumes $I_i$ for each

---

**Fig. 1.** Sample crop production functions for maize and potato

**Fig. 2.** Sample output of the water balance model
day $i$ in $s$. $K$ is the yield response factor, which describes the sensitivity of a given crop to water stress. Since $K$ is estimated over time depending on the growth stage, we calculated the relative yield $Y$ for each growth stage $s = 1 \ldots 4$. The overall yield $Y$ can then be derived from multiplying the product of the relative yields for all stages with $Y_s$, which is the maximum obtainable yield under the assumption of optimal growing conditions:

$$Y(I) = Y_m \prod_{s} Y_s(I) .$$

### C. Objective Function and Constraints

1) Objective Function: Based on the model described above, we defined the two components of the actual objective function with

$$f_1(x) = Y_m - Y(I) : I_i = x_i ,$$

and

$$f_2(x) = \sum I_i : I_i = x_i .$$

$f_1$ is the yield loss for a given irrigation schedule $I$ described by the decision variables $x_1 \ldots x_n$, where each day of simulated growth period of $n$ days is assigned a $x_i$. $f_2$ is the cumulated irrigation depth. Hence the optimization problem has $n$ independent decision variables and a two-dimensional objective function, both of which have to be minimized. The pareto front defined by the objective corresponds to the crop production function, which can then be determined in extenso within a single optimization run.

2) Constraints: Since the daily irrigation depth is limited by the capacity of the irrigation system, and irrigation depths always have to be positive, the decision variables $x_i$ are subject to the box constraint

$$0 \leq x_i \leq q_{\text{max}}$$

where $q_{\text{max}}$ is the maximum daily irrigation depth that can be provided by the irrigation system.

### D. Algorithms

In the literature numerous MO-EAs have been proposed [16]. We have evaluated the performance of four of them on the crop production function determination problem. These are NSGA-II, which is one of the most common MO-EAs, two MO-EA (NSDE and DEMO) derived from differential evolution, which performed best on the single objective irrigation scheduling problem [12], and MO-CMA-ES, which has shown a strong performance on a wide range of multi-objective optimization problems.

1) NSGA-II: NSGA-II [17] is derived from genetic algorithms and shares the mutation and crossover operators, namely the simulated binary crossover (SBX) operator and the polynomial mutation with its single objective counterpart. With the help of these two operators, each generation, which consists of $\lambda_{\text{mo}}$ individuals, $\lambda_{\text{mo}}$ new offspring are generated. The parents are selected by tournament selection using the crowded comparison operator $\prec_n$, which first uses the pareto rank and than the crowding distance metric as sorting criterion. This means NSGA-II always selects the individuals with the lowest pareto rank from the tournament and only uses the crowding distance in the case of equal pareto ranks. The same principle is also applied in order to determine the parent population of the next generation, which is selected from the combination of the current parent population and the generated offspring.

2) NSDE: NSDE [18] is a rotation invariant multi-objective version of differential evolution. Every generation $g$ of each of the $\lambda_{\text{mo}}$ individuals $x_j^{(g)}$ generates a single offspring $x_j^{(g*)}$ based on the DE/current-to-rand/1 rule which reads,

$$x_j^{(g*)} = x_j^{(g)} + K \cdot (x_i^{(g)} - x_j^{(g)}) + F \cdot (x_{i_1}^{(g)} - x_{i_2}^{(g)}) ,$$

where $x_{i_1}^{(g)}, x_{i_2}^{(g)}, x_j^{(g)}$ are three randomly chosen individuals from the current population which have to be distinctive from each other and $x_j^{(g)}$. $F$ and $K$ are real valued parameters which govern the strength of variation. The population of the next generation $g + 1$ is selected from the individuals of the current generation $x_j^{(g)}$ and their offspring $x_j^{(g*)}$ similar to NSGA-II by non-dominated sorting and crowding distance as first resp. second sorting criterion.

3) DEMO: Like NSDE, DEMO [19] is a multi-objective version of differential evolution. Instead of a generational approach like NSDE, DEMO works as a steady state algorithm. At each iteration a parent $x_i$ is randomly chosen from the current population of $\lambda_{\text{mo}}$ individuals and is then combined with two additional distinct individuals $x_{i_1}^{(2,3)}$, using the DE/rand/1/bin update rule to generate the single offspring $x_i'$. $x_i'$ replaces $x_i$, if it dominates the parent and is discarded if it is dominated by the parent. In the other cases the offspring is added to the population, which is then pruned to its desired size using non-dominated sorted crowding as in NSGA-II and NSDE.

4) MO-CMA-ES: MO-CMA-ES [20] is a multi objective derivative of the well known CMA-ES and uses a population of $\lambda_{\text{mo}}$ individuals which each store their own covariance matrix. In the most common variant of MO-CMA-ES, which has also been used in this paper, each individual generates a single offspring according to the (1+1)-CMA-ES scheme by adding a zero mean gaussian, which is governed by the individual’s own covariance matrix. The covariance matrix of each individual is then adapted by a combination of an evolution path approach and an adapted success rule, which only takes the success of the parent and its direct offspring into account but is smoothed over time. Since MO-CMA-ES does not use any form of crossover, there is no transfer of information between individuals along the pareto front. Like the other algorithms, MO-CMA-ES uses the pareto rank as the first sorting criterion in order to select the population of the next generation from all individuals of the current generation and their offspring. As second level sorting criterion Igel et al. [20] proposed to use the contributing hypervolume metric as described by Emmerich et al. [21] instead of the crowding distance used by NSGA-II, NSDE and DEMO.
5) **DE**: For comparison we also tested single-objective differential evolution combined with a Lamarckian constrained handling strategy. This approach has shown the best performance of all tested algorithms for optimal irrigation scheduling for a specified amount of irrigation water $Q$ in [12]. In order to obtain an approximation of the crop production function the yield loss $f_1(x)$ was minimized for 10 ($10 \times$ so-DE) respectively 50 ($50 \times$ so-DE) different $Q$, which were evenly spread between 0 cm and $Q_{optimal}$. In this case the number of function evaluations is the sum of the function evaluations of all of the 10 or 50 optimization runs.

**E. Initialization**

An important aspect for the success of the optimization process is the initialization of the first generation. For this paper we used two different initialization strategies. The first strategy (uniform) is the commonly used uniform sampling of individuals with

$$x_{ji}^{(0)} \sim U(0, q_{max}^{init}).$$

(9)

The choice of $q_{max}^{init}$ plays a crucial role. Although the actual upper limit of the search space for each decision variable $x_i$, is $q_{max}$, choosing $q_{max}^{init}=q_{max}$ has an extremely high probability to result in a degenerated initial population. Because of the extremely high irrigation depth ($f_2$), most or even all individuals generated this way achieve a yield loss $f_1$ of zero while having unreasonably high values for $f_2$. Hence the initial pareto front consists only of the single individual $x_s$ with the smallest yet still high value for $f_2$ which achieves $f_1(x_s) = 0$. In the first generations all other individuals are pushed from the population by this individual and its offsprings resulting in a very low population diversity. For this reason we used $q_{optimal}/n$, where $Q_{optimal}$ is the irrigation depth needed to achieve zero yields loss under the assumption of optimal scheduling. This results in more reasonable values for $f_2(x_{ji}^{(0)})$ for each individual $x_{ji}^{(0)}$.

The simple structure of $f_2$ allows an assessment of the resulting distribution of objective function values. Since each decision variable $x_i$ is sampled from an uniform distribution and $f_2$ is the sum of all decision variables $x_1 \ldots x_n$, $f_2$ follows a scaled Irwin–Hall distribution. This means that after initialization $f_2$ has a mean of $nq_{max}/2$ and a variance of $nq_{max}^2/12$, which differs considerably from the desired close to uniform distribution of $f_2$ between 0 and $Q_{optimal}$. As second initialization strategy (scaled) we therefore propose a rescaling of the initial individuals $x_{ji}^{(0)}$ with scaling factor $z_j$ which is uniformly sampled between 0 and $Q_{optimal}$. This reads

$$x_{ji}^{(0)*} = x_{ji}^{(0)} \frac{z_j}{\sum_i x_{ji}^{(0)}} : z_j \sim U(0, Q_{optimal}).$$

(10)

**F. Performance Measure**

The performance of MO-EAs can not be directly compared by matching the objective function values of the best individuals of each optimization run. Instead the sets $A$ of individuals $a$ in the pareto front generated by the optimization algorithms have to be compared. For this various performance indicators have been proposed in the literature. In the given context of minimizing yield loss depending on the available irrigation water the area under the resulting crop yield curve is a suitable indicator of the overall performance of the optimization algorithm. This is akin to minimizing the hypervolume indicator $I_{S,A_{ref}}(A)$ proposed by Zitzler et al. [22]. $I_{S,A_{ref}}(A)$ is defined as

$$I_{S,A_{ref}}(A) = S_{a_{ref}}(A_{ref}) - S_{a_{ref}}(A)$$

(11)

where $S_{a_{ref}}(A)$ is the volume of the union of hypercubes in the objective space spanned by the points within the set $A$ and $a_{ref}$. $A_{ref}$ is the reference pareto front and $a_{ref}$ is a reference point in the objective space, which for all components is worse than all points in $A_{ref}$ and $A$. In order to compare all generated pareto fronts on a common basis $a_{ref}$ has to be chosen worse than any point in any observed $A$.

In the context of irrigation optimization as defined in section II-C the upper limit for $f_2$ is $nq_{max}$, which is several magnitudes larger than any reasonable irrigation depth necessary for achieving optimal yields even with bad scheduling. Especially in the first generations of the optimization process extremely high values for $f_2$ can be observed. Choosing extreme values for $a_{ref}$ in order to accommodate for these outliers severely impacts the informative value of $I_{S,A_{ref}}(A)$. In these cases the impact of the actual position of the outer edge $e$ of a given pareto front $A$ with $f_2(e) \geq f_2(a)$ $\forall a \in A$ completely overshadows all other points in the front as shown in Fig. 3. For this reason we introduced an artificial constraint for $f_2$ which defines an upper limit for the overall irrigation water, with:

$$f_2(x) \leq 1.5 \cdot Q_{optimal}.$$  

(12)

For the calculation of $I_{S,A_{ref}}(A)$ points which violate this constrain are pruned from $A$ and $a_{ref}$ has been set to $(f_1, f_2) = (1, 1.5 \cdot Q_{optimal})$, which means a complete yield loss under the assumption of an irrigation depth of 1.5 times $Q_{optimal}$.

![Fig. 3. Impact of the edges of the pareto fronts on the hypervolume indicator $I_{S,A_{ref}}(A)$ with unconstrained $a_{ref}$](image)

$A_{ref}$ for each trial scenario has been obtained determining the pareto front of the combination all sample points from
all tested algorithms and points generated with the help of a single objective approach for evolutionary irrigation scheduling described in [12], where differential evolution in conjunction with a Lamarckian constraint handling strategy was used to determine the yield loss for a specific irrigation depth, which we varied in small increments from 0 to \( Q_{optimal} \).

G. Scenarios

In order to compare the optimization algorithms mentioned above we used three different scenarios which differ in crop type and climate scenario. All scenarios are based on a field with a soil of silty loam texture with a field capacity of \( f_c = 0.4 \) and a permanent wilting point of \( pwp = 0.05 \). For all scenarios \( q_{max} \) has been set to 20 cm.

Scenario A uses smoothed PET data which has been taken from field experiments by J.C. Mailhol from CEMAGREF in Montpellier [23] and assumes a worst case scenario without precipitation. Figure 4 shows the PET values for each day of the considered growth period as used in the simulation. The crop was assumed to be maize which is one of the most grown crops worldwide. Maize has a root system of up to 2 m depth and its sensitivity to water stress changes considerably depending on the growth stage, which makes it an interesting case study for irrigation scheduling. Maize typically requires only a few irrigations with considerable gaps in between during its growth period. The values for \( p \) used in this scenario are shown in Fig. 5 and the values for \( K_y \) and the respective length of the growth stages are shown in Table I. \( Q_{optimal} \) for this scenario is 63.5 cm.

![Smoothed PET values used in scenario A and B](image)

**Fig. 4.** Smoothed PET values used in scenario A and B

![Soil water depletion factors \( p_i \) for maize and potato](image)

**Fig. 5.** Soil water depletion factors \( p_i \) for maize and potato

Scenario B is based on the same weather data as Scenario A but uses potato as crop, which is often grown in rotation with maize. Potato is relatively sensitive to soil water deficits over its entire growth period [15] and has a shallow root system. Almost 100\% of the water uptake stems from the upper soil (0.6 m) which seriously limits the available soil storage capacity. Therefore potato requires much more frequent irrigation than maize in order to achieve acceptable yields. For optimal yield water losses have to be replenished each or every second day. The values for \( p_i \) and \( K_y \) are shown in Fig. 5 and table I. For this scenario \( Q_{optimal} \) is 85 cm.

Scenario C is based on a more sophisticated weather scenario, which simulates the expected mean impact of changing rainfall and radiation conditions on the crop yield. For this, we used the weather generator LARS WG, which has been developed by M. Semenov [25], to generate 1000 equally likely time series for rainfall and PET. To compute the mean expected yield for a given irrigation schedule we determined the yield for each time series and averaged the resulting 1000 yield values. Like scenario A scenario C uses maize as simulated crop. Therefore, we used the same \( Q_{optimal} \) as in scenario A with \( Q_{optimal} = 63.5 \text{ cm} \).

H. Test methods and algorithm parameter settings

We investigated all four MO-EA described above on scenario A and B with uniform as well as scaled initialization. Because of the large runtime requirements scenario C was only tested in combination with scaled initialization. Each test was repeated 20 times and was stopped after 50000 function evaluations. The population size \( \lambda_{mo} \) was set to 50 for all algorithms.

For NSGA-II the crossover probability \( p_c \) was set to 0.9 and the mutation probability to \( p_m = 1/n \). The distribution indices have been set to \( \eta_f = \eta_m = 5 \). For NSDE and DEMO we set \( K = p_c = 0.4 \) and \( F = 0.8 \).

All algorithms have been tested using the EvA2 optimization framework [26], which is a comprehensive free optimization system written in Java developed in our group.

The constraints given in (7) for \( x_i \) have been handled by setting \( x_i \) to the closest feasible value \( x_i^{feasible} \) and adding a small penalty to the objective function values with,

\[
 f^i_j(x) = f_{1,2}(x_i^{feasible}) + \alpha ||x_i^{feasible} - x_i||, \tag{13}
\]

where \( \alpha \) is the penalty parameter and has been set to \( 10^{-4} \).

III. Results

Table II shows the mean and standard deviation of the hypervolume indicator achieved by the tested algorithms.

---

**Table I**

LENGTH OF GROWTH STAGES AND YIELD RESPONSE FACTORS FOR MAIZE AND POTATO [15], [24]

<table>
<thead>
<tr>
<th>Crop</th>
<th>Initial</th>
<th>Growth stage</th>
<th>Mid season</th>
<th>Late season</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maize</td>
<td>Length [days]</td>
<td>25</td>
<td>40</td>
<td>40</td>
</tr>
<tr>
<td></td>
<td>( K_y )</td>
<td>0.4</td>
<td>0.4</td>
<td>1.3</td>
</tr>
<tr>
<td>Potato</td>
<td>Length [days]</td>
<td>25.5</td>
<td>40</td>
<td>45</td>
</tr>
<tr>
<td></td>
<td>( K_y )</td>
<td>0.45</td>
<td>0.8</td>
<td>0.8</td>
</tr>
</tbody>
</table>
Fig. 6. Development of the mean of the hypervolume for each tested multi objective algorithm for scenarios A and B with scaled and uniform initialization

Fig. 7. Typical development of the pareto front for scenario A for MO-CMA-ES and DEMO for uniform and scaled initialization
TABLE II
Mean and standard deviation of the results ($I_{S,A,x}(A)$) of the tested algorithms and initialization methods on the three scenarios. The best result for each scenario is set in bold and the better initialization is set in italics.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Scenario A</th>
<th>Scenario B</th>
<th>Scenario C</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>uniform initialization</td>
<td>scaled initialization</td>
<td></td>
</tr>
<tr>
<td>NSGAII</td>
<td>3.51 ± 0.41</td>
<td>1.28 ± 0.11</td>
<td>1.11 ± 0.29 x</td>
</tr>
<tr>
<td>NSDE</td>
<td>6.99 ± 0.97</td>
<td>9.91 ± 1.11</td>
<td>1.43 ± 0.61 x</td>
</tr>
<tr>
<td>DEMO</td>
<td>0.24 ± 0.11</td>
<td>0.61 ± 0.13</td>
<td>0.34 ± 0.15</td>
</tr>
<tr>
<td>MO-CMA-ES</td>
<td>1.0 ± 1.29</td>
<td>0.10 ± 0.01</td>
<td>0.07 ± 0.05</td>
</tr>
<tr>
<td>10×so-DE</td>
<td>-</td>
<td>-</td>
<td>2.25 ± 0.04</td>
</tr>
<tr>
<td>50×so-DE</td>
<td>-</td>
<td>-</td>
<td>1.74 ± 0.07</td>
</tr>
</tbody>
</table>

Fig. 8. Typical irrigation schedules for maize and potato with varying irrigation depth $f_2(x)$

For scenario A DEMO provides the best result while having the second best result for all other cases. NSDE always performs considerably worse than the other algorithms, especially when combined with uniform initialization, where it fails to achieve comparable results at all. Regardless of the initialization method, MO-CMA-ES and DEMO significantly outperform the single objective 10×so-DE and 50×so-DE in all tested scenarios. 50×so-DE is performing better than 10×so-DE, which can be attributed to the low number of different $Q$ for 10×so-DE resulting in a very low density of sample points along the pareto front.

Figure 6 shows the development of the mean of the hypervolume indicator depending on the number of evaluations for each scenario and offers more detailed information about the convergence behaviour. In particular at the early stages of the optimization process scaled initialization results in faster convergence rates than uniform initialization. This can be attributed to the better distribution of the initial population along the range of $f_2$ as can be seen in Fig. 7(a), which shows the development of the pareto front for a typical instance of MO-CMA-ES with scaled initialization. Uniform initialization results in a much less diverse initial pareto front as shown in Figure 7(b). These fronts do not contain any individuals with extremely low or high irrigation depths $f_2(x)$. In several cases this results in optimization runs in which MO-CMA-ES failed to diversify the pareto front to areas with low values for $f_2$. This failure occurred in 7 of the 20 trials for scenario A. These runs also account for the large variance of the results for MO-CMA-ES for this scenario. For scenario B there is no such effect, which may be attributed to the seemingly low complexity of this scenario. The lower complexity also shows in the higher convergence rates than in scenario A for all tested algorithms except NSDE. The reason for the lower complexity of scenario B can be found in the almost constant high sensitivity of potato to water stress. Since $K^p_{potato}$ changes only slightly over time, the impact of actual irrigation timing on the yield loss is much less severe than for maize of scenario A. Combined with the shallow root system this results in irrigation schedules, which consist of small but almost daily irrigations. Examples for such schedules are provided in Fig. 8(b), which shows the schedules and their corresponding objective function values depending on the scenario and initialization method. The subscripts $\text{I,II,III}$ mark results that are significantly better than the results from NSGAII, NSDE, DEMO, respectively MO-CMA-ES, with a $p$-value $\leq 0.05$ according to the Wilcoxon rank sum test. The subscript $x$ marks significantly better results from scaled initialization compared to uniform initialization. MO-CMA-ES always delivers the best results with the exception of scenario A with uniform initialization. When using uniform initialization. MO-CMA-ES always delivers the best results with the exception of scenario A with uniform initialization.
of 6 equidistantly spaced individuals along the pareto front. Maize on the other hand has a deep root system and has a highly varying sensitivity to the timing of the irrigations, which results in schedules as shown in Fig. 8(a) with larger but much more infrequent irrigation events. Because of the structure of $x_i$ small changes of the timing of such irrigation events as moving an irrigation one day in time map to dislodged points within the search space making scenario A much harder to optimize than scenario B.

IV. CONCLUSION

In this paper we have presented a novel approach for determining crop production functions based on MO-EAs. We also introduced a problem specific initialization method. Furthermore we examined the performance of four different algorithms on three scenarios, which differed in crop type and climate scenario. In the experiments MO-CMA-ES and DEMO performed best, with MO-CMA-ES being slightly better in most cases. The solutions found by both algorithms are very close to the actual pareto front for all tested scenarios and are significantly better than the solutions found by the multi start single-objective DE. Especially when combined with the scaled initialization introduced in the paper both algorithms deliver excellent results within the first 50000 function evaluations.

The evolutionary algorithms used in this paper do not make assumptions about the underlying irrigation and yield prediction models. The methodology presented in this paper should therefore be applicable to a wide range of models. Further work will extend the approach to more complex models which are able to simulate the impact of fertilizer and salination on the crop yield allowing a more comprehensive evaluation of the ecological impact and sustainability of an irrigation strategy. A second aspect will be the incorporation of more sophisticated and more efficient methods for dealing with the stochastic aspects of climate and crop growth.

REFERENCES