



HAL
open science

Sunflower phenotype optimization under climatic uncertainties using crop models

Victor Picheny, Ronan Trépos, Bastien Pouban, Pierre Casadebaig

► **To cite this version:**

Victor Picheny, Ronan Trépos, Bastien Pouban, Pierre Casadebaig. Sunflower phenotype optimization under climatic uncertainties using crop models. 2015. hal-01196643

HAL Id: hal-01196643

<https://hal.science/hal-01196643>

Preprint submitted on 14 Sep 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Sunflower phenotype optimization under climatic uncertainties using crop models

Victor Picheny[‡], Ronan Trépos^{*}, Bastien Pouban^{*}, Pierre Casadebaig[‡]

September 15, 2015

keyword Clustering; multi-objective optimization; subset sampling

Abstract

Accounting for the annual climatic variability is a well-known issue for simulation-based studies of environmental models. It often requires intensive sampling (e.g., averaging the simulation outputs over many climatic series), which hinders many sequential processes, in particular optimization algorithms. We propose here an approach based on a subset selection of a large basis of climatic series, using an ad-hoc similarity function and clustering. A non-parametric reconstruction technique is introduced to estimate accurately the distribution of the output of interest using only the subset sampling. The proposed strategy is non-intrusive and generic (i.e. transposable to most models with climatic data inputs), and can be combined to most “off-the-shelf” optimization solvers. We apply our approach to sunflower phenotype optimization using the crop model SUNFLO. The underlying optimization problem is formulated as multi-objective to account for risk-aversion. Our approach achieves good performances even for limited computational budgets, outperforming significantly more “naive” strategies.

1 Introduction

Using numerical models of complex dynamic systems has become a central process in sciences. In agronomy, it is now an essential tool for water resource management, adaptation of anthropic or natural systems to a changing climatic context or the conception of new production systems. In particular, in the past two decades crop models have received a growing attention (Boote et al., 1996; Brisson et al., 2003; Brun et al., 2006; Bergez et al., 2013; Brown et al., 2014; McNider et al., 2014), as they can be used to help improve the plant performances, either through cultural practices (Grechi et al., 2012; Wu et al., 2012) or model-assisted plant breeding (Semenov and Stratonovitch, 2013; Semenov et al., 2014; Quilot-Turion et al., 2012).

Many times, the objective pursued by model users amounts to solving an optimization problem, that is, find the set of input parameters of the model that maximize (or minimize) the output of interest (cost, production level, environmental impact, etc.). Examples of such problems abound with environmental models, including water distribution systems design (Tsoukalas and Makropoulos, 2014), agricultural watershed management (Cools et al., 2011) or the adaptation of cultural practices to climate change (Holzkämper et al., 2015). In phenotype optimization, (or *ideotype design*, Martre et al., 2015), plant performance (e.g., yield) is maximized with respect to its morphological and/or physiological traits.

Within the wide range of potential approaches to solve such optimization problems, *black-box optimization methods* have proved to be popular in this context (Maier et al., 2014; Martre et al., 2015; Quilot-Turion et al., 2012), as they only require limited expertise in optimization while being quite user-friendly, as they are in essence non-intrusive (i.e., they only require evaluations of the model at hand).

^{*}INRA, UR875 Mathématiques et Informatique Appliquées Toulouse, F-31326 Castanet-Tolosan, France

[†]Corresponding author, victor.picheny@toulouse.inra.fr

[‡]INRA, UMR AGIR, F-31326 Castanet-Tolosan, France

However, a well-known difficulty, shared by many models users, is to deal with climatic information. Many agricultural or ecological models require yearly series of day-to-day measures of precipitation, temperature, etc., as input variables. This is particularly crucial for agricultural or ecological models, for which the climate has a preponderant impact on the system. To avoid drawing conclusions biased by the choice of a particular set (i.e., year) of climatic data, one may either use scenarii approaches (duplicate the analysis for a small number of distinct climates), or average the model outputs over a (large) number of climatic datasets. Due to the complex plant-climate interaction, identifying scenarii may prove to be a very challenging task, and the alternative relies on intensive computation, which rapidly becomes computationally prohibitive if the analysis is embedded in a loop, even for moderately complex models.

A natural solution is to treat the climate as a random variable, which allows the use of the robust (or noisy) optimization framework. However, if readily available codes abound for continuous, box-constrained parameters and deterministic outputs, solutions become scarce for systems depending on stochastic phenomena. Besides, the problem formulation becomes more complex, as typically risk-aversion preferences need to be accounted for.

The methodological objective of this paper is two-fold. First, we wish to propose a clear optimization framework for optimization under climatic uncertainties, and in particular to account for risk-aversion concepts in a transparent manner. Second, as both optimization and uncertainty analysis are computationally intensive tasks, we need to provide an algorithmic solution to solve the problem in reasonable time. In addition, we wish to remain non-intrusive and generic (i.e. transposable to most models with climatic data inputs). Finally, in order to facilitate the use of parallel computing, we aim at limiting the complexity of the algorithm to its minimum.

In this work, we focus on the problem of sunflower ideotype design using the SUNFLO crop model. SUNFLO is a process-based model which was developed to simulate the grain yield and oil concentration as a function of time, environment (soil and climate), management practice and genetic diversity (Casadebaig et al., 2011). It allows to assess the performance of sunflower cultivars in agronomic conditions. A cultivar is represented by a combination of eight genetic coefficients (see Table 1), which are the variables to be optimized. The SUNFLO model computes the annual yield y (in tons per hectare) for a given climatic year.

The rest of this paper is organized as follow: Section 2 briefly reviews previous works on phenotype optimization, describes the SUNFLO model and the multi-objective optimization formulation to solve the problem at hand. Section 3 is dedicated to the optimization algorithm, which relies on a subset selection of the available climate data combined with a metaheuristic algorithm. Finally, Section 5 provides numerical results and compare our approach to classical solutions.

2 Problem definition

2.1 Brief review of phenotype optimization

Martre et al. (2015) provide a review of recent developments in this research domain named *model-assisted crop improvement* or *ideotype design*. A phenotype is defined as the expression in a particular environment of a specific genotype through its morphology, development, cellular, biochemical or physiological properties. An ideotype is defined as a combination of morphological and/or physiological traits optimizing crop performances to a particular biophysical environment and crop management. Letort et al. (2008) developed an approach to design plant ideotypes maximizing yield, using numerical optimization methods on coupled genetic and ecophysiological models. However, as most of the developed crop model do not include genetic-level inputs (Hammer et al., 2010), optimization mainly targets the phenotype level.

In the phenotype optimization setting, ideotype design can be formulated as a problem of optimizing model inputs related to cultivar practices (Grechi et al., 2012; Wu et al., 2012), or phenotypic parameters (Semenov and Stratonovitch, 2013; Semenov et al., 2014; Quilot-Turion et al., 2012). Different purposes are targeted such as the adaptation to climate change (Semenov and Stratonovitch, 2013; Semenov et al., 2014) or the multicriterion assessment of cultivar (Quilot-Turion et al., 2012; Qi et al., 2010). In most of these approaches (Letort et al., 2008; Qi et al., 2010; Quilot-Turion et al., 2012), the study has been performed on

a constant environment, in particular, using a single climatic year. Quilot-Turion et al. (2012) stated that further methodological developments are needed in the optimization side to reduce computational time in order to be able to consider multi-environments and large climatic series. In this work, the authors used the ‘Virtual Fruit’ model (Quilot et al., 2005) to design peach phenotypes for sustainable productions systems. Their aim is to optimize jointly three model outputs (fruit mass, sweetness and crack density) in four different scenarii using one climatic data serie in 2009. They first performed a sensitivity analysis in order to select six phenotypic model inputs amongst 60 and use the multi-objective optimization method NSGA-II (Deb et al., 2002) in order to solve the problem.

Semenov and Stratonovitch (2013) proposed to evaluate a phenotype by estimating an expected yield using 100 climatic series, by combining the use of the stochastic weather generator LARS-WG (Semenov and Stratonovitch, 2010) and the wheat crop model Sirius (Jamieson et al., 1998) in order to design high-yielding ideotypes for a changing climate in the case of two contrasting situations: Sevilla in Spain and Rothamsted in the United Kingdom. Inputs were nine cultivar-dependant parameters related to the photosynthesis, phenology, canopy, drought tolerance and root water uptake. The optimization problem was solved by using an evolutionary algorithm with self-adaptation (EA-SA, Beyer, 1995).

2.2 The SUNFLO model

In this work, we consider the SUNFLO crop model in order to assess the performance of sunflower cultivars in agronomic conditions. This model is based on a conceptual framework initially proposed by (Monteith, 1977) and now shared by a large family of crop models (Keating et al., 2003; Brisson et al., 2003; Stockle et al., 2003). In this framework, the daily crop dry biomass growth rate is calculated as an ordinary differential equation function of incident photosynthetically active radiation, light interception efficiency and radiation use efficiency. Broad scale processes of this framework, the dynamics of leaf area, photosynthesis and biomass allocation to grains were split into finer processes (e.g leaf expansion and senescence, response functions to environmental stresses) to reveal genotypic specificity and to allow the emergence of genotype \times environment interactions. Globally, the SUNFLO crop model has about 50 equations and 64 parameters (43 plant-related traits and 21 environment-related).

In this model, a cultivar is represented by a combination of eight genetic coefficients (see Table 1). These coefficients describe various aspects of crop structure or functioning: phenology, plant architecture, response curve of physiological processes to drought and biomass allocation. The consequence of genetic modifications can be emulated by changing the values of such parameters. We consider here the design of sunflower cultivars for a given set of cultural practices and a specific environment. The overall objective is to find a phenotype that maximizes the yield for the year to come, without knowing in advance the climate data. We assume that the coefficients can take continuous values between a lower and an upper bound, which are determined from a dataset of existing cultivars (see Table 1). We denote $\mathbf{x} \in \mathbb{X} \in \mathbb{R}^d$ a particular phenotype, where d is the number of input variables ($d = 8$).

The SUNFLO model computes the annual yield y (in tons per hectare) for a given climatic year. Hence, it requires as an additional input a climatic serie, which consists of daily measures over a year of five variables: minimal temperature (T_{\min} , °Cd), maximal temperature (T_{\max} , °Cd), global incident radiation (R , MJ/m^2), evapotranspiration (E , mm, Penman-Monteith) and precipitations (P , mm) We note: $c = \{T_{\min}, T_{\max}, R, E, P\}$. Figure 2 provides an example of such data.

We use historic climatic data from five french locations Avignon, Blagnac, Dijon, Poitiers and Reims (see Figure 1) from 1975 to 2012. The initial data is recorded over 365 days, but we consider only the cultural year (April to October, 180 days), as the yield computed by the model only depends on this period. We denote by Ω this set of climatic series, and we have $Card(\Omega) = N = 190$ and $c \in \mathbb{R}^{5 \times 180}$.

To summarize, the yield can be seen as a function of the phenotype and the climatic serie:

$$\begin{aligned} y : \mathbb{X} \times \Omega &\rightarrow \mathbb{R}^+ \\ \mathbf{x}, c &\mapsto y(\mathbf{x}, c). \end{aligned}$$

Table 1: Phenotypic coefficients and the bounds used for optimization.

<i>Symbol</i>	<i>Description</i>	<i>Min</i>	<i>Max</i>
TDF1	Temperature sum from emergence to the beginning of flowering ($^{\circ}\text{C}$)	765	907
TDM3	Temperature sum from emergence to seed physiological maturity ($^{\circ}\text{C}$)	1540	1830
TLN	Number of leaves at flowering	22.2	36.7
K	Light extinction coefficient during vegetative growth	0.780	0.950
LLH	Rank of the largest leave of leaf profile at flowering	13.5	20.6
LLS	Area of the largest leave of leaf profile at flowering (cm^2)	334	670
LE	Threshold for leaf expansion response to water stress	-15.6	-2.31
TR	Threshold for stomatal conductance response to water stress	-14.2	-5.81

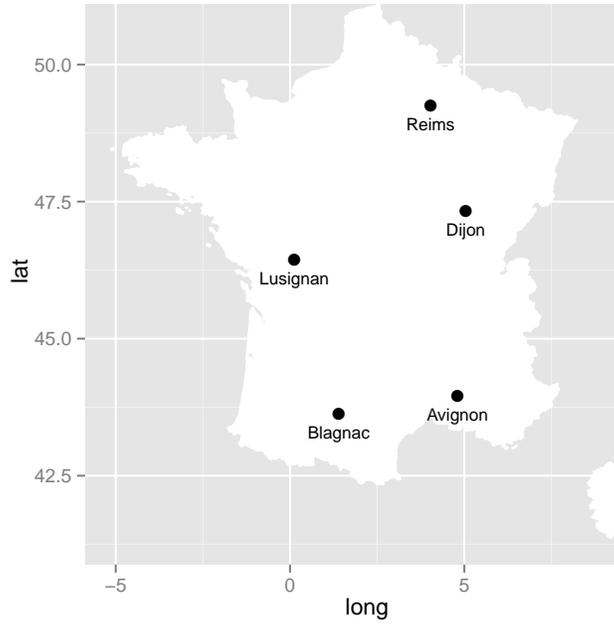


Figure 1: Location of the five French stations for the historic climatic data

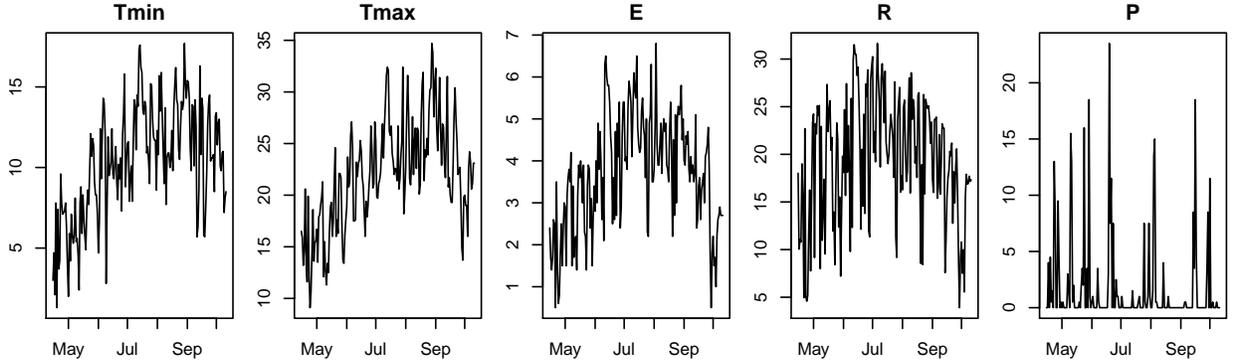


Figure 2: Dataset of the year 2009, Lusignan.

With a slight abuse of notations, we also define:

$$\begin{aligned}
 y(\mathbf{X}, c) &:= [y(\mathbf{x}_1, c), \dots, y(\mathbf{x}_I, c)]^T, \\
 y(\mathbf{x}, \mathbf{C}) &:= [y(\mathbf{x}, c_1), \dots, y(\mathbf{x}, c_J)]^T, \\
 y(\mathbf{X}, \mathbf{C}) &:= (y(\mathbf{x}_i, c_j))_{1 \leq i \leq I, 1 \leq j \leq J},
 \end{aligned}$$

that is, the yield function for a set of inputs, either for a set of phenotypes $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_I\}$ ($I \in \mathbb{N}^*$), a set of climatic series $\mathbf{C} = \{c_1, \dots, c_J\}$ ($1 \leq J \leq N$), or both.

2.3 A multi-objective optimization formulation for robust optimization

The objective is to find a phenotype that maximizes the yield for the year to come, without knowing in advance the climate data. Let C be the climatic serie of the upcoming year (the upper case denoting a random variable); we consider in the following that C is uniformly distributed over the discrete set Ω . Since C is random, the yield $y(\mathbf{x}, C)$ is also a random variable (which we denote in the following $Y(\mathbf{x})$), which makes its direct maximization with respect to \mathbf{x} meaningless.

A natural formulation is to maximize the yield expectation:

$$\max_{\mathbf{x} \in \mathbb{X}} \mathbb{E}[y(\mathbf{x}, C)] = \max_{\mathbf{x} \in \mathbb{X}} \mathbb{E}[Y(\mathbf{x})],$$

with here: $\mathbb{E}[Y(\mathbf{x})] = \frac{1}{N} \sum_{i=1}^N y(\mathbf{x}, c_i)$.

However, in general, a farmer also wishes to integrate some prevention against risk in its decision. Such a problem is often referred to as *robust optimization* in the engineering literature (see for instance Beyer and Sendhoff, 2007, for a review).

A popular solution is to replace the expectation by a performance indicator that provides a trade-off between average performance and risk aversion: typically, the expectation penalized by the variance or a so-called *utility function*. The drawback of such approaches is that the trade-off must be tuned beforehand by choosing penalization parameters specific to the method. Choosing the appropriate trade-off may not be straightforward, and modifying it requires to restart the entire optimization procedure.

We propose here an alternative, which is to consider this problem as multi-objective, by introducing a second criterion to maximize that accounts for the risk (as in Tsoukalas and Makropoulos, 2014, for instance). One may choose for instance to maximize a quantile:

$$\max_{\mathbf{x} \in \mathbb{X}} Q_\alpha [Y(\mathbf{x})],$$

with the usual definition of the quantile: $\mathbb{P}[Y \leq Q_\alpha(Y)] = \alpha$, and $\alpha \in (0, 0.5]$. Here, it amounts to maximizing the yield for the $(N \times \alpha)$ -th worst year. However, we consider here a close but numerically more stable criterion, called the conditional value-at-risk (CVaR, Rockafellar and Uryasev, 2000), defined as:

$$\text{CVaR}_\alpha [Y(\mathbf{x})] = \mathbb{E}[Y(\mathbf{x}) | Y(\mathbf{x}) \leq Q_\alpha [Y(\mathbf{x})]].$$

CVaR_α is the average yield over the $(N \times \alpha)$ -th worst years.

The multi-objective optimization problem is then:

$$\begin{cases} \max & \mathbb{E}[Y(\mathbf{x})] \\ \max & \text{CVaR}_\alpha [Y(\mathbf{x})] \\ \text{s.t.} & \mathbf{x} \in \mathbb{X}. \end{cases}$$

Such a formulation is relatively classical in robust optimization, although the second objective is often taken as the variance of the response: $\text{var}[Y(\mathbf{x})]$ (as for instance in Chen et al., 1999; Jin and Sendhoff, 2003). However, considering an expectation-variance trade-off does not make sense here, as a farmer will not want to minimize the variability of its income (i.e., minimizing the variance) but rather minimize the risk of low income.

3 Optimization with a representative subset

The two objective functions, $\mathbb{E}[Y(\mathbf{x})]$ and $\text{CVaR}_\alpha[Y(\mathbf{x})]$, require running the SUNFLO simulator N times everytime a new phenotype \mathbf{x} is evaluated. Embedded in an optimization loop, which typically requires thousands to millions calls to the objective functions, this evaluation step becomes prohibitive.

We propose to address this problem by replacing the large climatic data set Ω by a small representative set Ω_K . To do so, we first choose the set Ω_K prior to optimization using a clustering algorithm described in Section 3.1. Then, the optimization algorithm is run using Ω_K . Hence, $\mathbb{E}[Y(\mathbf{x})]$ and $\text{CVaR}_\alpha[Y(\mathbf{x})]$ are replaced by their estimates based on Ω_K , which are described in Section 3.2.

3.1 Choosing a representative subset of climatic data

3.1.1 Principle

To select our subset, we propose to define a distance (or, conversely, a similarity) between two climatic series, then choose series *far from each other* using clustering algorithms.

One can choose to consider only the dataset and define a distance that characterizes differences between the time series. However, the drawback of this method is that it is completely model-independent: two climatic series can be considered as far from each other but have a similar effect on the model, hence return a similar yield. Inversely, two climatic series can be generally close but return different yields because of small critical differences (say, a rainy week at an appropriate moment of the plant growth).

An alternative is to consider a model-based distance: two climatic series would be far from each other only if they return different yields for a given phenotype. This naturally implies that all the climatic series are run on a (small) phenotype learning set. Therefore, the distance will be very dependent on the choice of the set and may result in poor robustness.

Therefore, we propose here to combine both ideas, and define a hybrid distance that depends on intrinsic differences and on the effect on the model.

3.1.2 Dissimilarity between time series

As a climatic series is defined by five time series of different nature, we need first to define a metric to compare each series separately. Due to the nature of the data, Euclidian distance can be ruled out, as it makes little sense here. Indeed, all the series have important day-to-day variations (corresponding to good or bad weather), and similar events can be observed from one series to another shifted by one or several days.

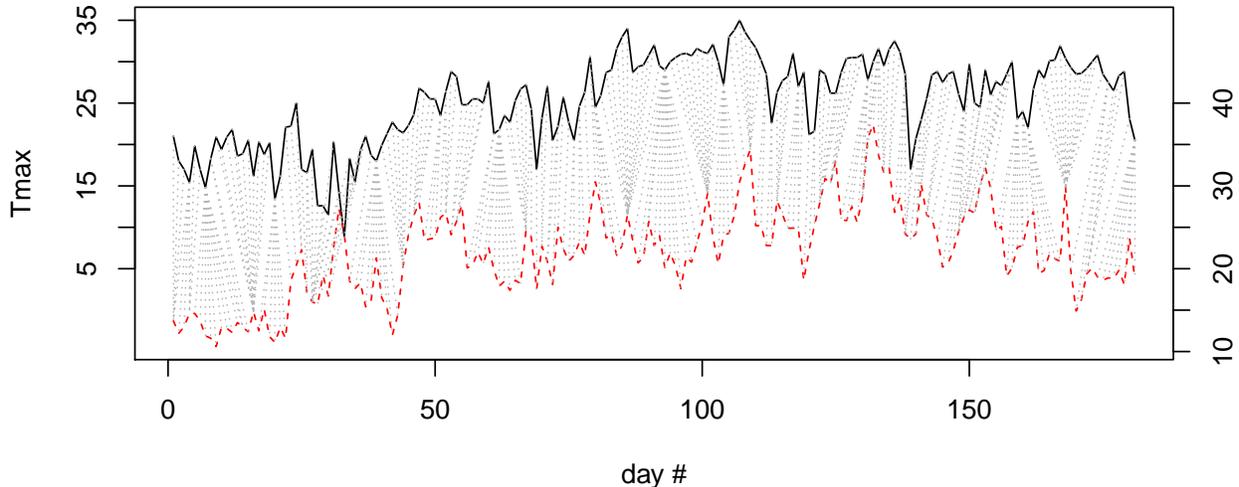


Figure 3: Computing of the dtw distance between two time series of maximal daily temperature ($Tmax$): Avignon in 1985 (upper curve, left scale) and Lusignan in 2012 (bottom curve, right scale). Dotted line represents the optimal matching of daily temperature computed by dtw, for a window size of 7 days.

This is particularly apparent for the precipitation series, which contain many zeros and several “peaks”: Euclidian distance would consider two series as far from each other, as long as the peaks do not coincide exactly.

A classical tool for time series analysis, sensible in our case, is an algorithm called dynamic time warping (DTW, Berndt and Clifford, 1994; Aach and Church, 2001; Kadous, 1999). In short, DTW allows two time series that are similar but locally out of phase to align in a non-linear manner, by matching events within a given window. Note that the DTW algorithm has a $\mathcal{O}(n^2)$ time complexity, which makes the dissimilarity computation non-trivial. However, this step should be performed only once. Given two weather series c_i and c_j , five distances can be computed, according to the weather variables: $d(c_i, c_j)^{Tmin}$, $d(c_i, c_j)^{Tmax}$, $d(c_i, c_j)^R$, $d(c_i, c_j)^E$ and $d(c_i, c_j)^P$.

3.1.3 Model-based dissimilarity

This dissimilarity measures a difference in the output of the model (the yield). To do so, we choose first a small set of l phenotypes: $\mathcal{B} = \{\mathbf{x}_1, \dots, \mathbf{x}_l\}$. Typically, \mathcal{B} can be chosen by Latin Hypercube Sampling (LHS, McKay et al., 1979) to “fill” the search space \mathbb{R}^d . For this basis, the yield is computed for all the climatic series: $y(\mathcal{B}, \Omega) \in \mathbb{R}^{l \times N}$. Then, the model-based distance is simply the Euclidian distance:

$$d(c_i, c_j)^{\mathcal{M}} = \sqrt{\frac{1}{l} \sum_{k=1}^l (y(\mathbf{x}_k, c_i) - y(\mathbf{x}_k, c_j))^2}$$

3.1.4 Combining dissimilarities

We want here to combine the six dissimilarities (one for each time series and the model-based one) into a single one, with equal weight to each variable. We propose to do so by normalizing the dissimilarities before summing them with uniform weights. As the variables are of different nature, the dissimilarities distributions are likely to be very different (uniform, heavy tailed, etc.), hence artificially weight the variables even if they are rescaled similarly.

Here, we follow a normalization procedure proposed in Olteanu and Villa-Vialaneix (2015) called “cosine preprocessing”, which works as follow: Let \mathbf{D} be a $N \times N$ matrix of dissimilarities (with values $d_{ij} = d(\mathbf{x}_i, \mathbf{x}_j)$,

$d_{ij} = d_{jj}$ and $d_{ii} = 0$). We first compute a corresponding similarity matrix \mathbf{S} , with values:

$$s_{ij} = -\frac{1}{2} \left[d_{ij} - \frac{1}{N} \sum_{k=1}^N (d_{ik} + d_{kj}) + \sum_{k=1}^N \sum_{k'=1}^N d_{kk'} \right].$$

Then, we normalize \mathbf{S} with:

$$\bar{s}_{ij} = \frac{s_{ij}}{\sqrt{s_{ii} + s_{jj}}},$$

and the normalized dissimilarity matrix $\bar{\mathbf{D}}$ has elements defined as:

$$\bar{d}_{ij} = \bar{s}_{ii} + \bar{s}_{jj} - 2\bar{s}_{ij} = 2 - 2\bar{s}_{ij}.$$

Now, we use a convex combination of the six normalized dissimilarities:

$$\delta_{ij} = \alpha_{T_{\min}} \bar{d}_{ij}^{T_{\min}} + \alpha_{T_{\max}} \bar{d}_{ij}^{T_{\max}} + \alpha_P \bar{d}_{ij}^P + \alpha_E \bar{d}_{ij}^E + \alpha_S \bar{d}_{ij}^S + \alpha_{\mathcal{M}} \bar{d}_{ij}^{\mathcal{M}}, \quad (1)$$

with $\alpha_{T_{\min}} + \dots + \alpha_{\mathcal{M}} = 1$. In the following, we use $\alpha_{\mathcal{M}} = 1/2$ and the other weights equal to $1/10$.

3.1.5 Choosing a representative subset using classification

Once the matrix of dissimilarities Δ is computed, most unsupervised clustering algorithms can be used to split the set of climatic series Ω into subsets. However, a difficulty here is that the centroids of the clusters cannot be computed. Hence, we use a variation of the k-means algorithm that only requires *dissimilarities* to the centroids. We follow the approach described in Olteanu and Villa-Vialaneix (2015); the corresponding pseudo-code is given in Algorithm 2.

The algorithm divides the set Ω into K classes $\mathcal{C}^{(1)}, \dots, \mathcal{C}^K$, not necessarily of equal sizes. A class \mathcal{C}^k contains N^k elements $\{c_1^k, \dots, c_{N^k}^k\}$. Any element $c \in \Omega$ is uniquely attributed to one class and we have: $\sum_{k=1}^K N^k = N$. For each class k , a representative element ω^k is chosen, which we use to define the representative set: $\Omega_K = \{\omega^1, \dots, \omega^K\}$.

3.2 Non-parametric reconstruction of distributions

The objective here is to obtain accurate estimations of the objective functions $\mathbb{E}[Y(\mathbf{x})]$ and $\text{CVaR}_\alpha[Y(\mathbf{x})]$ based on the yield computed for a new phenotype and the representative set: $y(\mathbf{x}, \Omega_K)$. Since this set is small, computing directly the objective functions would lead to large errors, in particular for $\text{CVaR}_\alpha[Y(\mathbf{x})]$, that requires an accurate representation of the tail distribution (see Figure 5). A natural alternative is to fit a parametric distribution the small data set, and infer the objectives on the distribution. However, the form of the empirical distribution (Figure 5) does not readily call for a given parametric model, and misspecifying the distribution shape may result with large bias.

Hence, we propose to reconstruct the distribution using a non-parametric method, by re-using the data computed for the classification step, that is, the yield computed for the phenotype learning basis and all the climatic series ($y(\mathcal{B}, \Omega)$).

The general idea is to consider a mixture model for the yield (each component corresponding to a class \mathcal{C}^k):

$$f_{Y(\mathbf{x})}(y) = \sum_{k=1}^K \frac{N^k}{N} f_{Y^k(\mathbf{x})}(y), \quad y \in \mathbb{R},$$

f standing for the probability density function (PDF), and $Y^k(\mathbf{x})$ being yield within the class k .

We decompose further $Y^k(\mathbf{x})$ as the sum of the value at the representative element and a residual:

$$Y^k(\mathbf{x}) = y(\mathbf{x}, \omega^k) + \varepsilon^k(\mathbf{x}).$$

The intra-class distribution is then characterized by the residuals $\varepsilon^k(\mathbf{x})$, which determine the form, spread (or amplitude), and bias (i.e., difference between the average value and the value of the representative element). All these elements vary from one class to another, which advocates the use of non-parametric approaches.

Method 1 (naive) From $y(\mathcal{B}, \Omega)$, we first compute the residuals $\varepsilon_j^k(\mathbf{x}_i) = y(\mathbf{x}_i, c_j^k) - y(\mathbf{x}_i, \omega^k)$ ($1 \leq i \leq l$; $1 \leq j \leq N^k$; $1 \leq k \leq K$). Then, we average the residuals over the phenotypes of \mathcal{B} :

$$\hat{\varepsilon}^k = [\hat{\varepsilon}_1^k, \dots, \hat{\varepsilon}_{N^k}^k], \text{ with } \hat{\varepsilon}_j^k = \frac{1}{l} \sum_{i=1}^l \varepsilon_j^k(\mathbf{x}_i).$$

The intra-class yield variety is re-created by adding the average residual vector to the yield computed for the representative value:

$$\hat{Y}^k(\mathbf{x}) = y(\mathbf{x}, \omega^k) + \hat{\varepsilon}_i^k,$$

with i uniformly taken from $\llbracket 1, N^k \rrbracket$. Thus, each component of the mixture has a fixed distribution (i.e. independent of \mathbf{x}), shifted according to its representative value, and the mixture shape and spread varies according to the distribution of the representative values (see Figure 4 for an illustration).

However, in practice, the values of the residuals can vary substantially from one phenotype to another, and averaging them over \mathcal{B} tends to destroy the shape information. To address this issue, we proposed the following modification:

Method 2 (rescaled) We introduce first the weighted variance of the yield over the representative set:

$$\sigma_K^2(\mathbf{x}) = \frac{1}{N} \sum_{k=1}^K N^k \left(y(\mathbf{x}, \omega_k) - \frac{1}{N} \sum_{j=1}^K N^j y(\mathbf{x}, \omega_j) \right)^2.$$

Note that for a new phenotype \mathbf{x} , the only data available is indeed $y(\mathbf{x}, \omega_j)$, so few alternatives are possible. We then define averages of *normalized* residuals:

$$\bar{\varepsilon}^k = [\bar{\varepsilon}_1^k, \dots, \bar{\varepsilon}_{N^k}^k], \text{ with } \bar{\varepsilon}_j^k = \frac{1}{l} \sum_{i=1}^l \frac{\varepsilon_j^k(\mathbf{x}_i)}{\sigma_K(\mathbf{x}_i)}.$$

and the yield vector is constructed with:

$$\hat{Y}^k(\mathbf{x}) = y(\mathbf{x}, \omega^k) + \sigma_K(\mathbf{x}) \times \bar{\varepsilon}_i^k,$$

with i uniformly taken from $\llbracket 1, N^k \rrbracket$.

Figures 4 and 5 illustrate the reconstruction technique for a given (randomly chosen) phenotype. On Figure 4, we see how the estimated distribution is built using the residuals corresponding to each class. We can see that the range and shape of the residuals vary considerably from one class to another. Also, their distribution around the representative element differs: as the residuals do not have a zero mean, the value of the representative element is not necessarily central for each class. Comparing the reconstructed (Figure 4, top) and actual (bottom) distributions, we see that the mixture is globally the same on both graphs.

Figure 5 shows the cumulative distribution function (CDF) of the actual yield and of three estimations: using the two methods described above and a simple parametric method, which consists in assuming a Gaussian distribution of the yield. The empirical CDF corresponding to the subset values only is also depicted, with unequal steps to account for the different number of elements in each class.

We first notice that the subset data only is obviously insufficient to evaluate accurately the mean or the CVaR. Then, we see that the actual distribution does not seem to belong to a known distribution, and using a normal distribution introduces a large bias. Inversely, using a non-parametric reconstruction allows us to match the shape of the actual distribution. The difference between the two methods is small for this example, yet the second approach is slightly better almost everywhere.

In our study, we found that this second method provided a satisfying trade-off between robustness, simplicity and accuracy. Yet, many refinements would be possible at this point, for instance by introducing intra-class rescaling (different normalization for each class), bias correction, or using the distance from the phenotype \mathbf{x} to the basis \mathcal{B} .

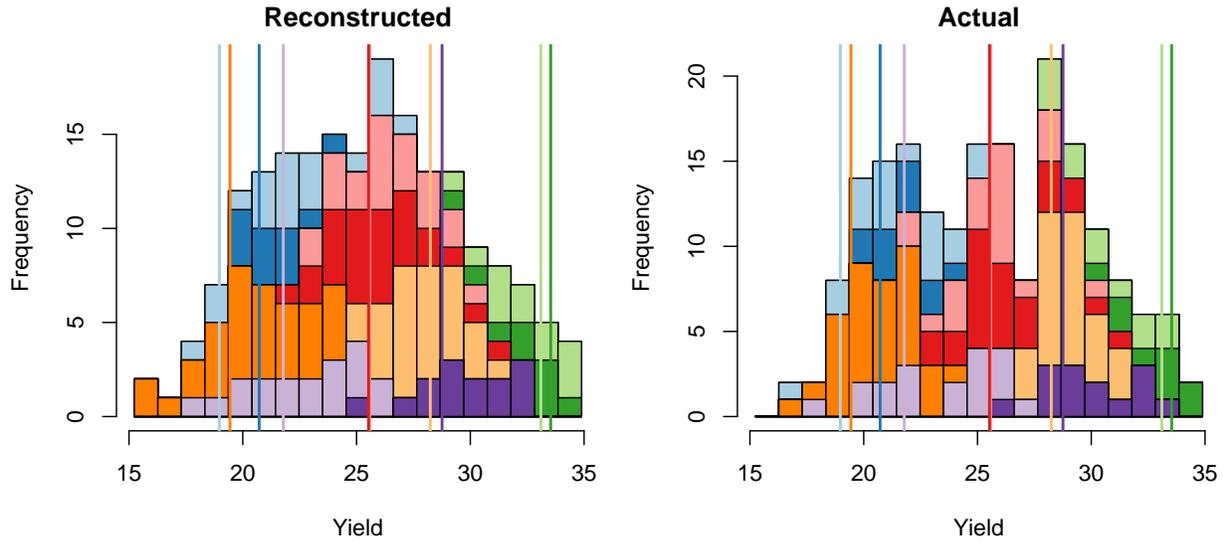


Figure 4: Estimated yield distribution of a given phenotype. The colors show how the reconstruction works: each color corresponds to a class k , and the vertical bars to the representative element of the class.

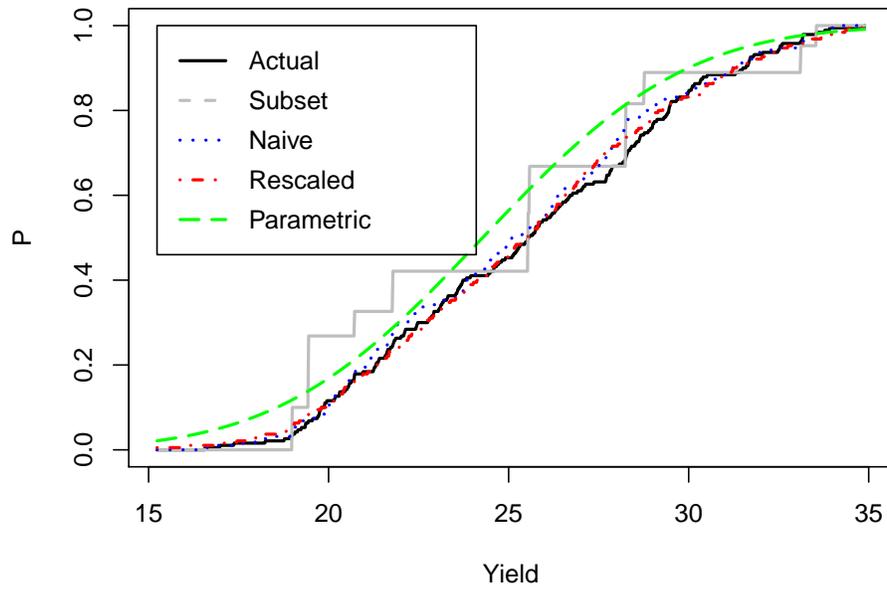


Figure 5: Actual and estimated distributions (CDF) of the yield of a given phenotype.

3.3 Optimization and reconstruction update

Finally, the multi-objective optimization problem solved is:

$$\begin{cases} \max & \mathbb{E} [\hat{Y}(\mathbf{x})] \\ \max & \text{CVaR}_\alpha [\hat{Y}(\mathbf{x})] \\ \text{s.t.} & \mathbf{x} \in \mathbb{X}, \end{cases}$$

with $\hat{Y}(\mathbf{x})$ a mixture of $\hat{Y}^1(\mathbf{x}), \dots, \hat{Y}^K(\mathbf{x})$.

One may note that $\mathbb{E} [\hat{Y}(\mathbf{x})]$ and $\text{CVaR}_\alpha [\hat{Y}(\mathbf{x})]$ serve as estimates of $\mathbb{E} [Y(\mathbf{x})]$ and $\text{CVaR}_\alpha [Y(\mathbf{x})]$, respectively. These estimates are based on the phenotype basis \mathcal{B} , which is sampled uniformly over \mathbb{X} to offer a general representation of the phenotype space. This feature is important at the beginning of the optimization to ensure that the optimizer does not get trapped into poorly represented regions. However, as the optimizer converges towards the solution, the search space becomes more narrow, and a substantial gain in performance can be achieved by modifying the estimates so that they are more accurate in the optimal region.

In theory, it is possible to re-run the entire clustering procedure after a couple of optimization iterations, by adding new phenotypes to the learning set. However, such strategy is likely to increase greatly the computational burden. We propose instead to modify only the reconstruction step, for which only very few additional calculations are required.

Indeed, the reconstructed yield distributions use the phenotype learning basis \mathcal{B} and their associated values $y(\mathcal{B}, \Omega)$. By replacing the initial \mathcal{B} with \mathcal{B}' formed by phenotypes chosen inside the optimal region, we obtain yield values $y(\mathcal{B}', \Omega)$ that are more likely to represent the actual distribution within this region. Such “specialization” may be to the detriment of the global accuracy of the estimates, but this is not critical as the optimizer concentrates on a narrow region.

Including a new phenotype \mathbf{x}' into the basis \mathcal{B} requires running the SUNFLO simulator N times to obtain $y(\mathbf{x}', \Omega)$. Therefore, an efficient trade-off must be found between pursuing the optimization and improving the estimates. Also, it may be beneficial to discard phenotypes in \mathcal{B} that are far from the optimal region. In summary, we need to: a) decide when to add phenotypes to the basis and b) when to discard them and c) choose which to add / discard.

A simple strategy is to perform only two steps: first, run the optimization with the initial basis \mathcal{B} . Then, select l new phenotypes from the obtained Pareto set and replace the entire basis \mathcal{B} after running the $N \times l$ simulations. Finally, restart the optimization with the new estimates. We have found (Section 5) that this two-step strategy was sufficient on our problem, while relatively easy to implement.

3.4 Optimization procedure overview

To summarize this section, Algorithm 1 describes the complete optimization procedure, including the initial clustering and the two-step strategy. Each step relies on the call to a metaheuristic algorithm such as NSGA-II (Deb et al., 2002) or MOPSO-CD (Raquel and Naval, 2005). Hence, two-step MOPSO-CD stands for the two-step algorithm using the MOPSO-CD metaheuristic.

Algorithm 1 Two-step optimization algorithm

Initialization

Choose phenotype database \mathcal{B} , and compute yield matrix $y(\mathcal{B}, \Omega)$
Compute matrix of dissimilarity Δ
Run clustering algorithm to obtain the classes $\mathcal{C}^{(1)}, \dots, \mathcal{C}^K$ and the representative set Ω_K
Get residuals from $y(\mathcal{B}, \Omega)$

Optimization: run 1

Choose population size q and number of iterations T
for $t = 1, \dots, T$ **do**
 Select new phenotypes $\{\mathbf{x}_{new}^1, \dots, \mathbf{x}_{new}^q\}$ according the metaheuristic.
 Calculate yield for the representative set for each new phenotype $y(\mathbf{x}_{new}^i, \Omega_K)$
 Reconstruct $\hat{Y}(\mathbf{x}_{new}^i)$ and evaluate $\mathbb{E}[\hat{Y}(\mathbf{x}_{new}^i)]$ and $\text{CVaR}_\alpha[\hat{Y}(\mathbf{x}_{new}^i)]$
 Post process $\{\mathbf{x}_{new}^1, \dots, \mathbf{x}_{new}^q\}$ according the metaheuristic.
end for
Get Pareto-optimal solutions \mathbf{X}^*

Optimization: run 2

Replace \mathcal{B} by \mathbf{X}^* , compute yield matrix $y(\mathcal{B}, \Omega)$
Get the new residuals from $y(\mathcal{B}, \Omega)$
for $t = 1, \dots, T$ **do**
 Select new phenotypes $\{\mathbf{x}_{new}^1, \dots, \mathbf{x}_{new}^q\}$ according the metaheuristic.
 Calculate yield for the representative set for each new phenotype $y(\mathbf{x}_{new}^i, \Omega_K)$
 Reconstruct $\hat{Y}(\mathbf{x}_{new}^i)$ and evaluate $\mathbb{E}[\hat{Y}(\mathbf{x}_{new}^i)]$ and $\text{CVaR}_\alpha[\hat{Y}(\mathbf{x}_{new}^i)]$
 Post process $\{\mathbf{x}_{new}^1, \dots, \mathbf{x}_{new}^q\}$ according the metaheuristic.
end for
Get Pareto-optimal solutions \mathbf{X}^*

4 Experimental setup

4.1 Climate subset selection

In this experiment, we used the R package `dtw` (Giorgino, 2009) to compute all the distances between climatic series. Note that the window size (that is, the maximum shift allowed) is a critical parameter of the method; we use here expert knowledge to choose it. For the precipitation, a window of ± 3 days is used; for the other variables, a window of ± 7 days is chosen. The phenotype basis \mathcal{B} is chosen as a 10-point LHS; hence, for this step the method required 1,900 calls to the SUNFLO model.

Once the dissimilarity matrix Δ is computed, the clustering algorithm (see Appendix 6) is run. Since this algorithm amounts to a gradient descent, it provides a local optimum only, so we need to restart it several times (by changing the initial values β_0) to ensure that a good optimum is found. We found in practice that 500 iterations and 10 restarts were sufficient to achieve a good robustness.

This algorithm does not choose automatically the number of classes K . We found empirically that $K = 10$ provided a satisfying trade-off between the representation capability of the subset and the computational cost during the optimization loop.

4.2 Optimization

To solve the multi-objective optimization problem, we chose to use the MOPSO-CD metaheuristic (Multi-Objective Particle Swarm Optimization with Crowding Distance Raquel and Naval, 2005). MOPSO-CD is a stochastic population-based algorithm inspired by the social behavior of bird flocking. In short, the algorithm

maintains over T generations a population P of individuals (candidate solutions). At each generation, each candidate is moved through the search space according to an individual direction (local improvement), a global direction (towards the best candidates of the population) and a crowding distance. This distance is used in order to build a set of solution that fills uniformly the Pareto front.

In the following experiments we used the R package `dtw` (Naval, 2013). The two main parameters of MOPSO-CD are the population size and number of generations (their product being equal to the number of function evaluations).

In order to assess the validity of our approach, we have conducted an empirical comparison to simpler approaches: random search and a “naive” optimizer, both using the full set of climatic series. In addition, we have conducted an intensive experiment to obtain an accurate representation of the actual Pareto set.

The intensive experiment consists in running two multi-objective algorithms (NSGA-II and MOPSO-CD) with a very large budget (number of calls to the simulator function) using the full set of climatic series. The two obtained Pareto fronts are merged to a single one, which we consider as “exact” in the following. We set the number of iterations to 300 and the population size to 200, hence computing the exact Pareto front requires $2 \times 200 \times 300 \times 190 = 22,800,000$ calls to SUNFLO.

Random search, or LHS search, is performed using a latin hypercube sampling approach to fill the search space \mathbb{X} . The naive optimization is performed using the original MOPSO-CD algorithm. Each sampled point is evaluated using the entire set of climatic series ($N = 190$) to estimate the expected yield and CVaR.

We compare the different approaches based on an equal number of calls to SUNFLO (that is, we do not consider the time costs related to each approach). We considered four budgets: large (380,000), medium (95,000), small (23,750) and very small (11,400).

For the naive and two-step approaches, we need to define the number of iterations and the population size. We set the number of iterations to approximately five times the population size, except for the very small budget where the population size would be too small. For the two-step algorithm, each evaluation of the expectation and CVaR requires 10 SUNFLO runs, which allows a larger population and number of iterations than the naive approach, but it is also necessary to compute two times $y(\mathcal{B}, \Omega)$ (the simulations of yields over all climatic series for the phenotype basis), which has a 10×190 cost.

The different setups are given in Table 2. Note that the budgets are only approximately equal (due to rounding issues). Nevertheless the budgets for the two-step approach are always equal or smaller than the naive one.

Since these three optimization approaches are stochastic, each experiment is replicated 10 times, to assess the robustness of the results.

The time cost of one call to the SUNFLO model is low ($\simeq 0.1$ sec), which makes it possible to perform such an extensive experiment. However, to limit the computational costs, these experiments are performed with either a symmetric multiprocessing (SMP) solution based on 30 cores or a message passing interface (MPI) implementation based on 40 cores, depending on memory requirements of experiments, which makes time costs comparisons meaningless.

The SUNFLO model has been implemented on the VLE software (Quesnel et al., 2009) in the RECORD project which is dedicated to agorecosystems study (Bergez et al., 2013). VLE is a multi-modeling and simulation platform coded in C++ that provides both a shared memory and a MPI based parallelisation for the simulation of multiple input combinations. A native port `rvle` to the software `R` is available in order to call simulations from this statistical tool. The other `R` packages used are `fExtremes` (computation of CVaR statistic), `lhs` (optimized LHS generation), `emoa` (dedicated tools for multiobjective problems) and `mco` (NSGA-II implementation). Finally, we are grateful to the genotoul bioinformatics platform Toulouse Midi-Pyrenees for providing help and/or computing and/or storage resources.

Table 2: Experiments performed for the two-step MOPSO-CD algorithm evaluation.

Optimization experiment	Budget	Nb of iterations	Pop size	Real nb of simulations
Intensive	Very large	300($\times 2$)	200	$\simeq 2 \times 10^7$
Random (or LHS)	very small	-	60	11,400
	small	-	125	23,750
	medium	-	500	95,000
	large	-	2,000	380,000
Naive MOPSO-CD	very small	12	5	12,350
	small	25	5	24,700
	medium	50	10	96,900
	large	100	20	383,000
Two-step MOPSO-CD	very small	42($\times 2$)	9	11,540
	small	71($\times 2$)	14	23,960
	medium	152($\times 2$)	30	95,600
	large	308($\times 2$)	61	380,780

5 Results and discussion

5.1 Climate subset selection

We analyze first the classification obtained with our approach. As the classification is based on non-trivial distances, it is difficult to characterize each class with integrated quantities (e.g. rainy / hot years, etc.). We provide in the following three tools for this analysis.

We first plot a 2D projection of the climatic series based on the matrix of distances Δ computed as in Section 3.1.4. To do this, we use the R package `cmdscale` (Classical Multidimensional Scaling) (Figure 6-a). Such a representation allows us to see whereas the classes are well-separated, if there are outliers, etc.

In Figure 6-b, the number of climatic series, grouped by their localization is given for each cluster. Finally, a decision tree has been learnt (with the R package `C50`) using the cluster index of climatic series as the variable to explain (Figure 6-c). We highlight here that this tree is solely for interpretation purpose and is not linked to the proposed classification strategy. We used temporal mean aggregation of climatic variables $\{T_{\min}, T_{\max}, R, E, P\}$ and the mean yield simulated on the 10 phenotypes in \mathcal{B} to build the decision tree.

Based on these three representations, one can conclude that some clusters correspond more or less to wheater types from the South of France (Avignon, Blagnac : 0, 5, 7, 9) rather warm (5, 7) or not (0, 9) and leading to high yields (5, 9) or not (0, 7). The three clusters 0, 5 and 7 seem indeed the most easy to characterize (Figure 6-a).

Cluster 1 represents climatic series leading to low yields from all locations. Clusters 3, 4, 6, 8 correpond rather to wheater types from the north of France leading to high yields (3, 4, 6) or not (8). Clusters 2, 4, 6, 9 can be characterized by a cold weather and high yields but there are difficult to distinguish from each other; there is indeed an important mixture of clusters in node 6 in Figure 6-c and one can make the same observation when studying the projection in Figure 6-a.

While a simple characterisation of clusters can be done, there are still differences between them that we do not achieve to characterize, which motivates the approach of using a distance between time series. Especially, there is a known high impact of rain episodes and their localization in time, however, the temporal mean aggregation of rain is not retained when building these decision trees.

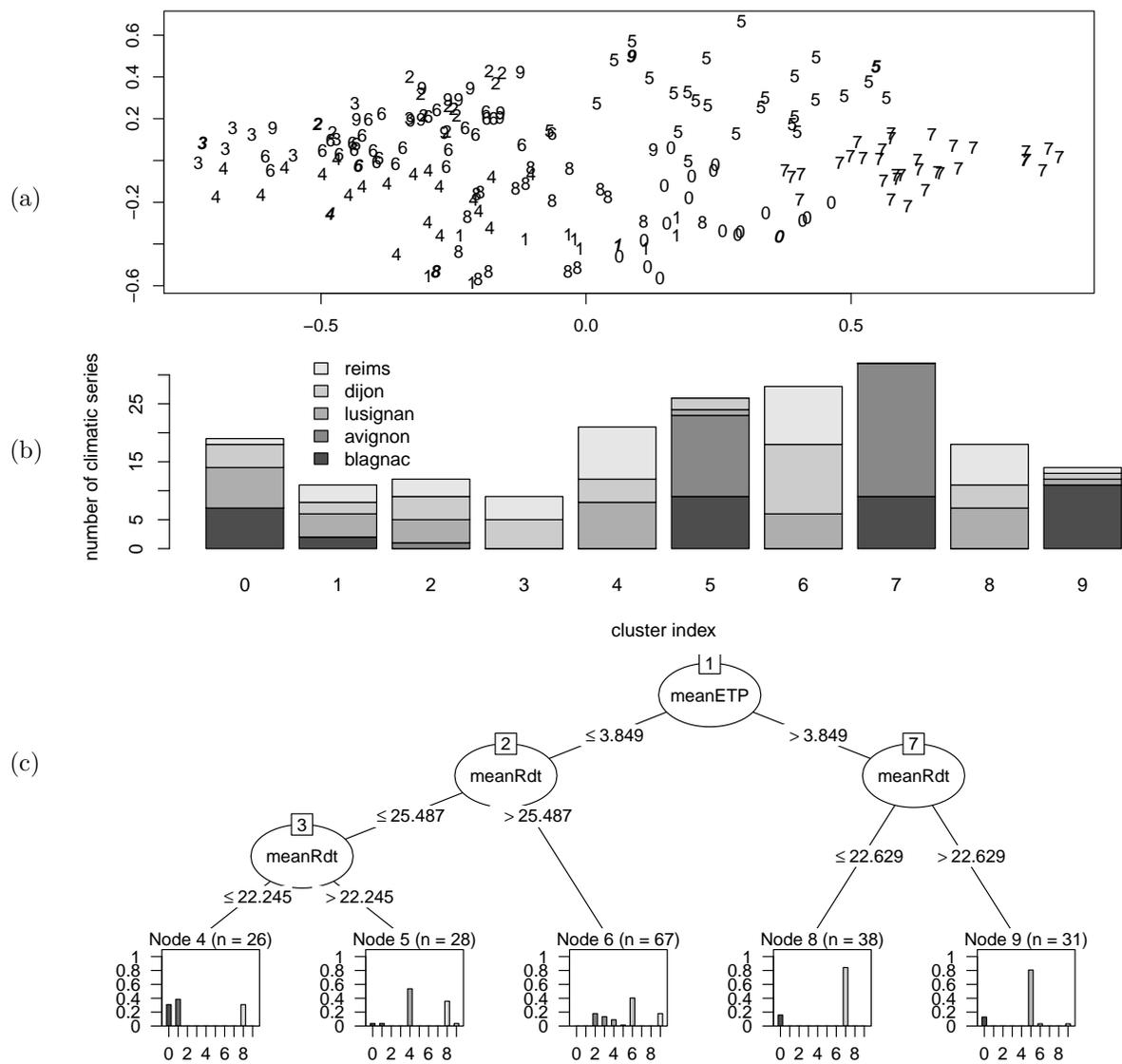


Figure 6: (a) Clusters and individuals (the 190 time series) are plotted in a 2D projection using Classical Multidimensional Scaling. Each digit represents a weather time series which value corresponds to its cluster. Climate series of representative set Ω_K are plotted in bold italic. (b) Number of climatic series by cluster splitted by localization in France. (c) A decision tree to explain clusters using, for each climatic series, the temporal mean values of climatic variables and the mean yield simulated on the 10 phenotypes in \mathcal{B} .

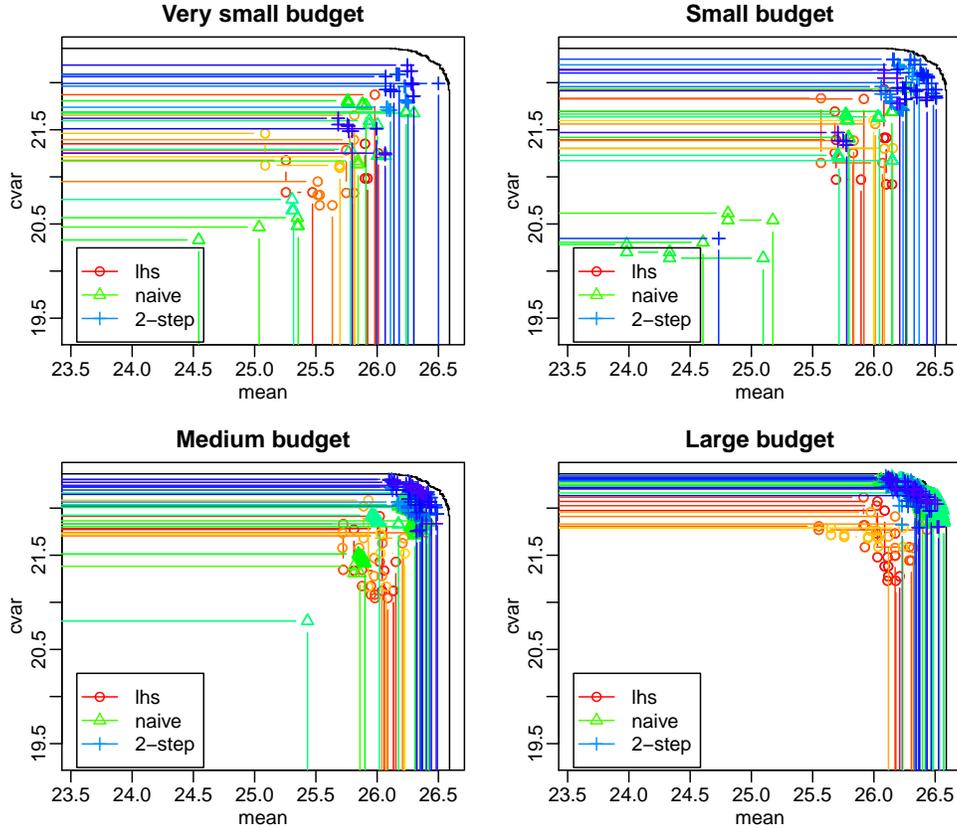


Figure 7: Pareto fronts obtained with the different methods for the four budgets considered.

5.2 Phenotype optimization

5.2.1 Algorithm performance

Next, we compare the performances of the three approaches. As measuring performance is non-trivial in multi-criteria optimization, we use three indicators: hypervolume, epsilon and R_2 indicators (as recommended in Zitzler et al., 2003; Hansen and Jaszkiwicz, 1998), all available in the R package `emoa` (Mersmann, 2012). They provide different measures of distance to the exact Pareto set and coverage of the objective space. In short, the hypervolume indicator is a measure of the volume contained between the Pareto front and a reference point (here, the worst value of each objective). The epsilon indicator is a maximum distance between two Pareto fronts (here, we use the exact Pareto front as reference), while the R_2 indicator can be seen as an average distance. Figure 7 shows all the Pareto fronts (of the different runs and methods) for the different budgets, and Figure 8 shows the corresponding performance indicators in the form of boxplots.

For the very small budget, we see that no method succeeds at finding the exact Pareto front. Besides, most of the Pareto fronts consists of a single point. However, the two-step approach still largely outperforms random search, while a naive use of MOPSO-CD performs worst, as it requires a certain number of iterations to find a descent direction.

For the small and medium budgets, the two-step approach consistently finds a good approximation of the Pareto front (with the exception of two outliers with the small budget). For the three indicators, it clearly outperforms the other approaches.

For the large budget, we see that the regular MOPSO-CD performs slightly better, which is expected.

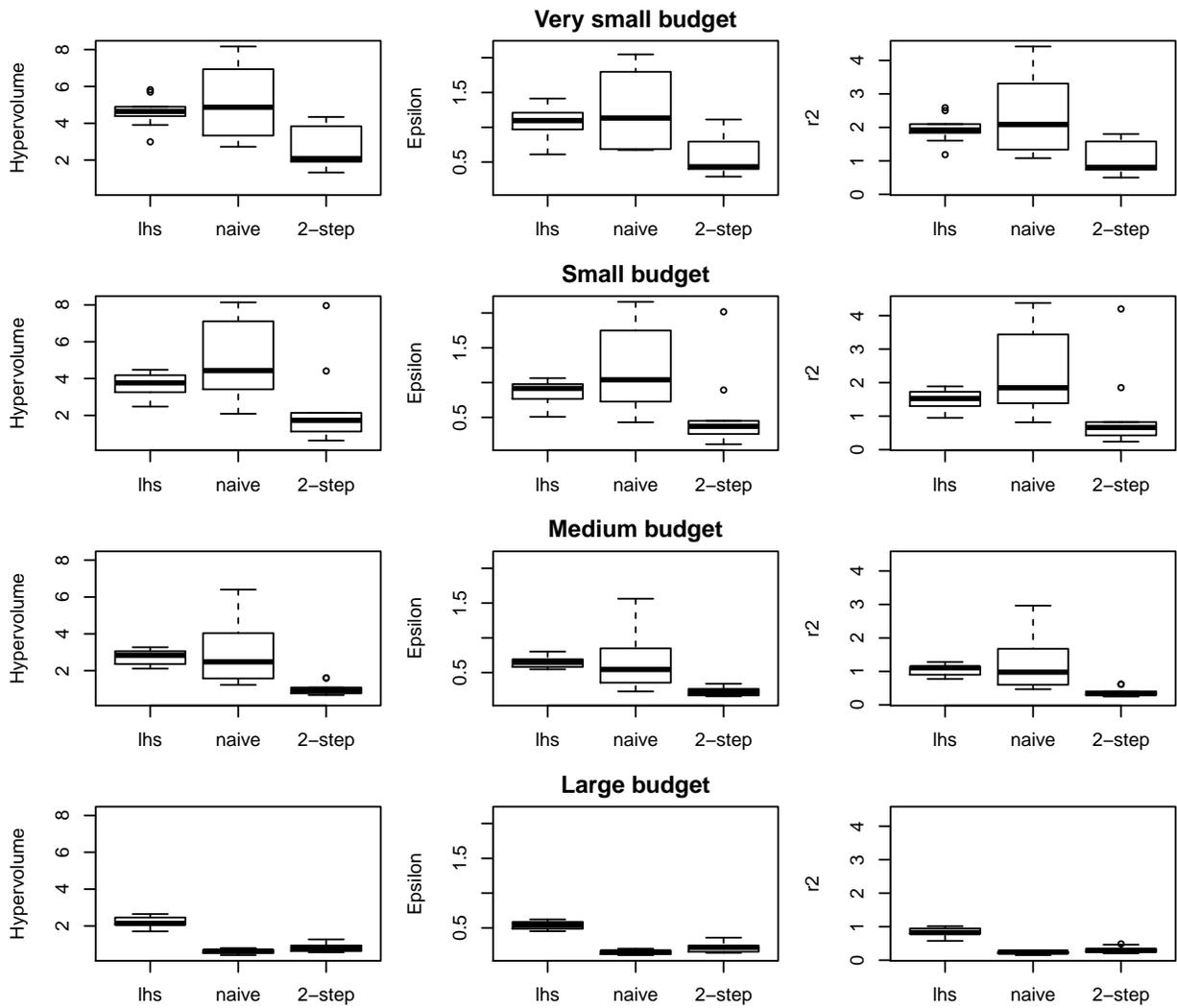


Figure 8: Performance indices of the different methods for the four budgets considered.

Indeed, as soon as there is no necessity of parcimony, using approximate objectives instead of actual ones tends to slower, rather than accelerate, convergence.

5.2.2 Results analysis

Finally, we characterize the results on the phenotype space. We compare here the exact Pareto set with one run of the two-step method; we chose the run on the medium budget with the median performance. For readability, we only consider a subset of the Pareto set of size five, equally spaced along the Pareto front. The Pareto fronts and sets are represented in Figure 9.

We can see first that considering both the expectation and CVaR for optimization leads to a large variety of optimal phenotypes. Looking back at the plant characteristics corresponding to those solutions, the optimum value for five traits had little variability, meaning that those traits were important plant characteristics for crop performance in the tested environments. Those five traits depicted plants adapted to water deficit: a late maturity (TDM3), a low leaf number (TLN), largest leaves at the bottom of the plant (LLH), a small plant area (LLS), and a conservative strategy for stomatal conductance regulation (TR). The three other traits (TDF1, K, LE) displayed variability in optimal values, which was identified as the basis of the performance/stability trade-off (expectation/CVaR). Here, the traits vary monotonically along the Pareto front.

Four distinct plant types could be identified in the phenotype space. For example, the *red* plant type had an early flowering (TDF1), a low light extinction efficiency (K) and a low plant leaf area (LLS); those characteristics correspond to a conservative resource management strategy. In an opposite manner, the *light-blue* type displays a late flowering, a high efficiency to intercept light and a larger plant leaf area, characteristics usually associated with a productive but risky crop type when facing strong water deficit (Connor and Hall, 1997). The strategy associated with plant types identified from the phenotype space matched their position in the Pareto front, i.e the *light-blue* plant type was more performant but less stable than the *red* one.

The Pareto set obtained with the two-step method reproduces part of these features: the fixed traits are similar (except TLN, which is fixed to approximately 0.5 instead of 0, (but this parameter is known to have little impact on the yield, see Casadebaig et al. (2011)) and the variation of TDF1 and LLS is well-captured. However, on this run the method failed at finding the variation of the K and LE traits: this probably explains why the largest mean values (left of the Pareto front) are missed.

Overall, the two-step method allowed to identify the few key traits were responsible for the cultivar global adaptation capacity whereas secondary traits supported alternative resource use strategies underlying the yield expectation/stability tradeoff.

6 Summary and perspectives

In this article, we proposed an algorithm for phenotype optimization under climatic uncertainties. Our approach does not require any *a priori* knowledge on the system besides parameter bounds, hence is usable with any simulator depending on similar climatic data. Using subset selection for the climates allowed us to reduce substantially the computational time without adding implementation issues. If bias correction seems inevitable during optimization, we showed that a two-step strategy was sufficient to achieve convergence: this point is critical as it allows our approach to be combined with any black-box multi-objective solver.

Nevertheless, we see many opportunities for further improvements. First, the distance used here between climate series does not account for the fact that agronomical systems are mostly sensitive to a few critical periods (e.g., during flowering, grain filling). Weighting the DTW distance using expert knowledge or the results of a sensitivity analysis may greatly improve the classification of the climates with respect to their impact on the model.

Second, the reconstruction step may benefit from additional study, in particular the effect of the subset size, which has been fixed to 10 in our study for practical reasons but could be chosen using preliminary experiments for instance. Another interesting topic would be to target the reconstruction to improve the

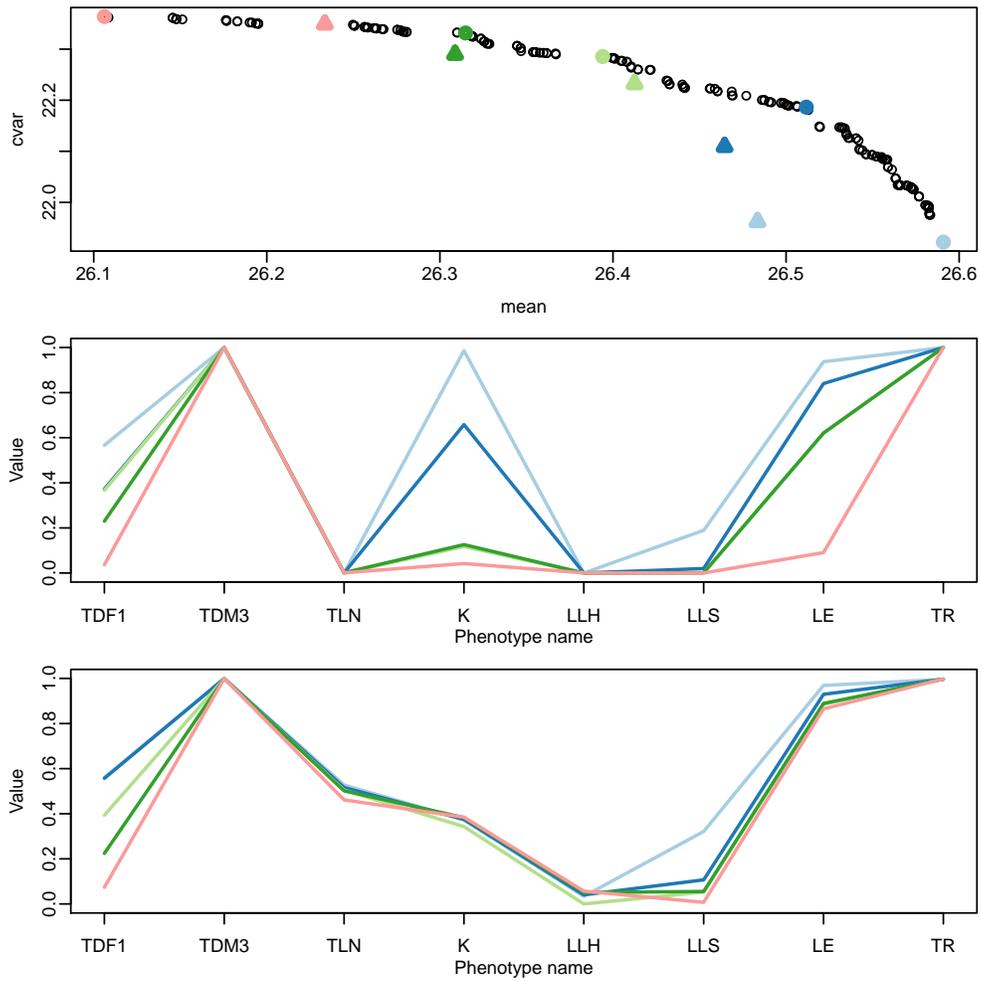


Figure 9: Top: exact Pareto front. The bold circles correspond to a subset of five optimal phenotypes; the triangles correspond to five phenotypes returned by the two-step method. Middle: optimal phenotype values (one curve corresponds to one phenotype). Bottom: phenotype values obtained with the two-step method.

quality of the objectives. Indeed, the proposed approach aims at reconstructing the entire output distribution, while it is only important to obtain good estimates of the expectation and the CVaR.

Third, a popular strategy to reduce the computational costs is to combine optimization with the use of surrogate modelling (see for instance Di Pierro et al., 2009; Tsoukalas and Makropoulos, 2014, for recent examples). Our approach straightforwardly extends to such approaches, and would result in very parsimonious algorithms that may be beneficial for expensive simulations.

Finally, we have chosen here to use a two-step strategy to allow the use of “off-the-shelf” optimization solvers. Interlinking optimization and learning may improve substantially the efficiency of the method, although requiring the development of an *ad hoc* algorithm.

Appendix: clustering algorithm

This section details the clustering algorithm used and the rule to choose the representative element of each class. The key of this particular approach is that, contrarily to a standard k-means algorithm, we cannot compute explicitly a central element (i.e., a “virtual” climatic series).

Algorithm 2 Clustering algorithm

Initialize β in $\mathbb{R}^{N \times K}$ randomly such that $\beta_{ij} \geq 0$, $\forall i, j$ and $\sum_{i=1}^N \beta_{ij} = 1$, $\forall j$. Each line β_k is the dissimilarity of the centroid $\tilde{\omega}_k$ to the climates.

for $t = 1, \dots, T$ **do**

 Pick i randomly in $1, \dots, N$ (one climate selected randomly)

Assignment step Find j (center closest to c_i) such that

$$j = \arg \min_{k=1, \dots, K} \left(\beta_k^T \Delta_i \right) - \frac{1}{2} \beta_k \Delta \beta_k^T,$$

with Δ_i the i -th line of Δ .

Representation step (update center)

$$\beta_j \leftarrow \beta_j + r(t) \times (\mathbb{1}_j - \beta_j),$$

where $\mathbb{1}_j$ is a vector of zeros except its j -th value equal to one and $r(t) = \frac{\epsilon_0}{1 + c_0 \frac{t}{K}}$.

end for

Once β has converged, each climate c_i is attributed to the class j , using:

$$j = \arg \min_{k=1, \dots, K} \left(\beta_k \Delta_i \right) - \frac{1}{2} \beta_k \Delta \beta_k^T.$$

For each class k , a representative element ω^k is chosen. We choose here the most central element in terms of dissimilarity. Let Δ^k be the submatrix of Δ corresponding to the elements of \mathcal{C}^k . We choose:

$$\omega^k = c_I^k \quad \text{with } I = \arg \min_{1 \leq i \leq N^k} \sum_{j=1}^{N^k} \delta_{ij}^k$$

References

References

Aach, J. and Church, G. M. (2001) Aligning gene expression time series with time warping algorithms. *Bioinformatics*, **17**, 495–508.

- Bergez, J.-E., Chabrier, P., Gary, C., Jeuffroy, M., Makowski, D., Quesnel, G., Ramat, E., Raynal, H., Rouse, N., Wallach, D., Debaeke, P., Durand, P., Duru, M., Dury, J., Faverdin, P., Gascuel-Oudou, C. and Garcia, F. (2013) An open platform to build, evaluate and simulate integrated models of farming and agro-ecosystems. *Environmental Modelling and Software*, **39**, 39–49.
- Berndt, D. J. and Clifford, J. (1994) Using dynamic time warping to find patterns in time series. In *KDD workshop*, vol. 10, 359–370. Seattle, WA.
- Beyer, H.-G. (1995) Toward a theory of evolution strategies: Self-adaptation. *Evolutionary Computation*, **3**, 311–347.
- Beyer, H.-G. and Sendhoff, B. (2007) Robust optimization—a comprehensive survey. *Computer methods in applied mechanics and engineering*, **196**, 3190–3218.
- Boote, K. J., Jones, J. W. and Pickering, N. B. (1996) Potential uses and limitations of crop models. *Agronomy Journal*, **88**, 704–716.
- Brisson, N., Gary, C., Justes, E., Roche, R., Mary, B., Ripoche, D., Zimmer, D., Sierra, J., Bertuzzi, P., Burger, P. et al. (2003) An overview of the crop model stics. *European Journal of agronomy*, **18**, 309–332.
- Brown, H. E., Huth, N. I., Holzworth, D. P., Teixeira, E. I., Zyskowski, R. F., Hargreaves, J. N. and Moot, D. J. (2014) Plant modelling framework: software for building and running crop models on the apsim platform. *Environmental Modelling & Software*, **62**, 385–398.
- Brun, F., Wallach, D., Makowski, D. and Jones, J. W. (2006) *Working with dynamic crop models: evaluation, analysis, parameterization, and applications*. Elsevier.
- Casadebaig, P., Guilioni, L., Lecoeur, J., Christophe, A., Champolivier, L. and Debaeke, P. (2011) Sunflo, a model to simulate genotype-specific performance of the sunflower crop in contrasting environments. *Agricultural and Forest Meteorology*, **151**, 163 – 178.
- Chen, W., Wiecek, M. M. and Zhang, J. (1999) Quality utility—a compromise programming approach to robust design. *Journal of mechanical design*, **121**, 179–187.
- Connor, D. and Hall, A. (1997) Sunflower physiology. *Sunflower Technology and Production. Agronomy Monograph*, **35**, 67–113.
- Cools, J., Broekx, S., Vandenberghe, V., Sels, H., Meynaerts, E., Vercaemst, P., Seuntjens, P., Van Hulle, S., Wustenberghs, H., Bauwens, W. et al. (2011) Coupling a hydrological water quality model and an economic optimization model to set up a cost-effective emission reduction scenario for nitrogen. *Environmental Modelling & Software*, **26**, 44–51.
- Deb, K., Pratap, A., Agarwal, S. and Meyarivan, T. (2002) A fast and elitist multiobjective genetic algorithm: Nsga-ii. *Transactions on Evolutionary Computation*, **6**, 182–197.
- Di Pierro, F., Khu, S.-T., Savić, D. and Berardi, L. (2009) Efficient multi-objective optimal design of water distribution networks on a budget of simulations using hybrid algorithms. *Environmental Modelling & Software*, **24**, 202–213.
- Giorgino, T. (2009) Computing and visualizing dynamic time warping alignments in R: The dtw package. *Journal of Statistical Software*, **31**, 1–24.
- Grechi, I., Ould-Sidi, M.-M., Hilgert, N., Senoussi, R., Sauphanor, B. and Lescouret, F. (2012) Designing integrated management scenarios using simulation-based and multi-objective optimization: Application to the peach tree–myzus persicae aphid system. *Ecological Modelling*, **246**, 47–59.

- Hammer, G. L., van Oosterom, E., McLean, G., Chapman, S. C., Broad, I., Harland, P. and Muchow, R. C. (2010) Adapting apsim to model the physiology and genetics of complex adaptive traits in field crops. *Journal of Experimental Botany*, erq095.
- Hansen, M. P. and Jaskiewicz, A. (1998) *Evaluating the quality of approximations to the non-dominated set*. IMM, Department of Mathematical Modelling, Technical University of Denmark.
- Holzkämper, A., Klein, T., Seppelt, R. and Fuhrer, J. (2015) Assessing the propagation of uncertainties in multi-objective optimization for agro-ecosystem adaptation to climate change. *Environmental Modelling & Software*, **66**, 27–35.
- Jamieson, P., Semenov, M., Brooking, I. and Francis, G. (1998) Sirius: a mechanistic model of wheat response to environmental variation. *European Journal Of Agronomy*, **8**, 161–179.
- Jin, Y. and Sendhoff, B. (2003) Trade-off between performance and robustness: an evolutionary multiobjective approach. In *Evolutionary Multi-Criterion Optimization*, 237–251. Springer.
- Kadous, M. W. (1999) Learning comprehensible descriptions of multivariate time series. In *ICML*, 454–463.
- Keating, B. A., Carberry, P. S., Hammer, G. L., Probert, M. E., Robertson, M. J., Holzworth, D., Huth, N. I., Hargreaves, J. N. G., Meinke, H. and Hochman, Z. (2003) An overview of APSIM, a model designed for farming systems simulation. *European Journal of Agronomy*, **18**, 267–288.
- Letort, V., Mahe, P., Cournède, P.-H., de Reffye, P. and B., C. (2008) Quantitative genetics and functional-structural plant growth models: Simulation of quantitative trait loci detection for model parameters and application to potential yield optimization. *Annals of Botany*, **101**, 1243–1254.
- Maier, H., Kapelan, Z., Kasprzyk, J., Kollat, J., Matott, L., Cunha, M., Dandy, G., Gibbs, M., Keedwell, E., Marchi, A. et al. (2014) Evolutionary algorithms and other metaheuristics in water resources: current status, research challenges and future directions. *Environmental Modelling & Software*, **62**, 271–299.
- Martre, P., Quilot-Turion, B., Luquet, D., Ould-Sidi Memmah, M., Chenu, K. and Debaeke, P. (2015) Model assisted phenotyping and ideotype design. In *Crop physiology: applications for genetic improvement and agronomy* (eds. V. Sadras and D. Calderini), chap. 14, 349–373. London, United Kingdom: Academic Press, 2nd edn.
- McKay, M. D., Beckman, R. J. and Conover, W. J. (1979) Comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, **21**, 239–245.
- McNider, R., Handyside, C., Doty, K., Ellenburg, W., Cruise, J., Christy, J., Moss, D., Sharda, V. and Hoogenboom, G. (2014) An integrated crop and hydrologic modeling system to estimate hydrologic impacts of crop irrigation demands. *Environmental Modelling & Software*.
- Mersmann, O. (2012) *emoa: Evolutionary Multiobjective Optimization Algorithms*. R package version 0.5-0.
- Monteith, J. L. (1977) Climate and the efficiency of crop production in Britain. *Royal Society of London Philosophical Transactions Series B*, **281**, 277–294.
- Naval, P. (2013) *mopsocd: MOPSOCD: Multi-objective Particle Swarm Optimization with Crowding Distance*. R package version 0.5.1.
- Olteanu, M. and Villa-Vialaneix, N. (2015) On-line relational and multiple relational som. *Neurocomputing*, **147**, 15–30.
- Qi, R., Ma, Y., Hu, B., De Reffye, P. and Cournède, P.-H. (2010) Optimization of source-sink dynamics in plant growth for ideotype breeding: a case study on maize. *Computers and Electronics in Agriculture*, **71**, 96–105.

- Quesnel, G., Duboz, R. and Ramat, E. (2009) The Virtual Laboratory Environment – An operational framework for multi-modelling, simulation and analysis of complex dynamical systems. *Simulation Modelling Practice and Theory*, **17**, 641–653.
- Quilot, B., Kervella, J., Génard, M. and Lescourret, F. (2005) Analysing the genetic control of peach fruit quality through an ecophysiological model combined with a qtl approach. *Journal of Experimental Botany*, **56**, 3083–3092.
- Quilot-Turion, B., Ould-Sidi, M.-M., Kadrani, A., Hilgert, N., Génard, M. and Lescourret, F. (2012) Optimization of parameters of the ‘virtual fruit’ model to design peach genotype for sustainable production systems. *European Journal of Agronomy*, **42**, 34 – 48. Designing Crops for new challenges.
- Raquel, C. R. and Naval, Jr., P. C. (2005) An effective use of crowding distance in multiobjective particle swarm optimization. In *Proceedings of the 7th Annual Conference on Genetic and Evolutionary Computation*, GECCO ’05, 257–264. New York, NY, USA: ACM.
- Rockafellar, R. T. and Uryasev, S. (2000) Optimization of conditional value-at-risk. *Journal of risk*, **2**, 21–42.
- Semenov, M. and Stratonovitch, P. (2010) Use of multi-model ensembles from global climate models for assessment of climate change impacts. *Climate Research*, **41**, 1–14.
- Semenov, M., Stratonovitch, P., Alghabari, F. and Gooding, M. (2014) Adapting wheat in europe for climate change. *Journal of Cereal Science*, **59**, 245–256. Cereal Science for Food Security, Nutrition and Sustainability.
- Semenov, M. A. and Stratonovitch, P. (2013) Designing high-yielding wheat ideotypes for a changing climate. *Food and Energy Security*, **2**, 185 – 196.
- Stockle, C. O., Donatelli, M. and Nelson, R. (2003) CropSyst, a cropping systems simulation model. *European Journal of Agronomy*, **18**, 289–307. URL <http://www.sciencedirect.com/science/article/B6T67-47FDS23-2/2/e747114524fb04bfe3ca1f30ddcca834>.
- Tsoukalas, I. and Makropoulos, C. (2014) Multiobjective optimisation on a budget: Exploring surrogate modelling for robust multi-reservoir rules generation under hydrological uncertainty. *Environmental Modelling & Software*.
- Wu, L., Le Dimet, F., De Reffye, P., Hu, B., Cournède, P. and Kang, M. (2012) An optimal control methodology for plant growth - case study of a water supply problem of sunflower. *Mathematics and computers in simulation*, **82**, 909–923.
- Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C. M. and Da Fonseca, V. G. (2003) Performance assessment of multiobjective optimizers: An analysis and review. *Evolutionary Computation, IEEE Transactions on*, **7**, 117–132.