arXiv:submit/4080421 [cs.CY] 20 Dec 2021

Evolutionary Hierarchical Harvest Schedule Optimization for Food Waste Prevention

Maurice Günder[1,2,*], **Nico Piatkowski**[2], **Laura von Rueden**[2], **Rafet Sifa**[2], **Christian Bauckhage**[1,2]

[1] Institute for Computer Science III, University of Bonn, Friedrich-Hirzebruch-Allee 5, 53115 Bonn, Germany
[2] Fraunhofer Institute for Intelligent Analysis and Information Systems IAIS, Schloss Birlinghoven, 53757 Sankt Augustin, Germany
{maurice.guender; nico.piatkowski; laura.von.rueden; rafet.sifa; christian.bauckhage}@iais.fraunhofer.de
[*] Corresponding author

Abstract

In order to avoid disadvantages of monocropping for soil and environment, it is advisable to practice intercropping of various plant species whenever possible. However, intercropping is challenging as it requires a balanced planting schedule due to individual cultivation time frames. Maintaining a continuous harvest reduces logistical costs and related greenhouse gas emissions, and contributes to food waste prevention. In this work, we address these issues and propose an optimization method for a full harvest season of large crop ensembles that complies with given constraints. By using an approach based on an evolutionary algorithm combined with a novel hierarchical loss function and adaptive mutation rate, we transfer the multi-objective into a pseudo-single-objective optimization problem and obtain faster convergence and better solutions than for conventional approaches.

1 Introduction

In a recent study, the Food and Agriculture Organization of the United Nations stated that almost 1.4 billion hectares or 30 % of the world's agricultural land are used to produce food that later goes to waste. They also stated that global food loss causes 3.3 gigatons of carbon dioxide emissions or 7 % of the world's total greenhouse gas emissions (FAO 2013, 2019). Efficient use of agricultural and logistical capacities is therefore crucial for sustainable global food supply.

A particular application of artificial intelligence in agriculture is the optimization of harvest schedules or agri-food supply chains (Taşkıner and Bilgen 2021). Even in other food-related disciplines, harvest schedules are of major importance, e.g. in aquaculture (Yu and Leung 2006). In this work, we will focus on optimizing a planting schedule with regard to a sustainable yield sketched in Figure 1. In particular, our goal is to find planting dates for many crop species such that the resulting yield is efficiently distributed. In this way, we facilitate prevention of food waste and surplus greenhouse gas emissions.

Scheduling problems are difficult to solve by involving numerous soft- and hard constraints and requiring to factor in environmental conditions. Common approaches thus consider stochastic models and commercial solvers which, however, are essentially "black box" systems whose decisions are opaque and difficult to comprehend. Alternatively, scheduling problems are commonly cast as multi- or many-objective optimization (Branke et al. 2008) problems. Besides many other optimization strategies like in (Ferrer et al. 2008) and (Varas et al. 2020), these problems are often optimized by evolutionary algorithms (EAs) like NSGA-II (Deb et al. 2002) or MOEA/D (Zhang and Li 2007) as in (Roghanian and Cheraghalipour 2019), or in combination with other models (Cheraghalipour, Paydar, and Hajiaghaei-Keshteli 2019).

Copyright © 2022, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

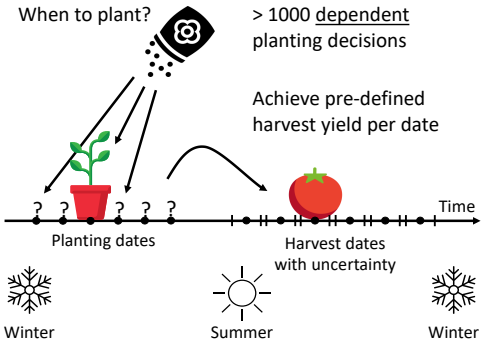

Figure 1: Problem setting: for given planting dates, corresponding harvest dates have to be predicted. Predictions are based on uncertain Growing Degree Unit (GDU) forecasts. The challenge is to schedule planting dates for more than 1000 individual crops and desired yields simultaneously, resulting in a continuous harvest yield distribution.

In the work at hand, we consider a basic evolution strategy (ES) often referred to as $(1+1)$-ES. It is simple to implement and can be generalized towards $(\mu + \lambda)$-ES which allow for distributed computation.

Our contributions can be summarized as follows:

- we propose a generic formalization of the harvest schedule optimization problem, which is agnostic with respect to time scales, crop types, and desired yield
- a novel hierarchical loss function is devised that converts naive multi-objective problems into a single-objective problem, enabling faster optimization
- uniform escape from local optima is facilitated via a new

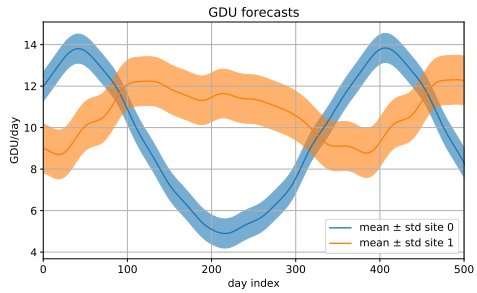

Figure 2: GDU forecast mean and $\pm 1\sigma$ standard deviation for site 0 (blue) and site 1 (orange).

dynamic oscillating mutation rate

- we provide an experimental evaluation on realistic data

This work provides a summary of results for the above points. A detailed discussion of methods and the full set of results will be available in an extended version of this manuscript.

2 Background

The first building block of our method is Gaussian process regression (GPR) for time series prediction (Rasmussen and Williams 2006). GPR is a kernel-based non-parametric method to inter- and extrapolate data points. It allows us to include available background knowledge about agricultural time series, e.g. the existence of annual periodicity, other seasonal effects, and trends, by choosing appropriate kernel functions.

The second building block are evolutionary algorithms (EAs) which are a class of meta-heuristic optimization algorithms inspired by biological mechanisms (Holland 1992; Bäck 1996). The common procedure is to generate one or more possible solutions in the form of state vectors, called *population*. During a *generation*, the populations, or *parents*, are *mutated*, resulting in *children*. An *environment* thereafter has to select the best population(s) which themselves are initial states for further mutations in the upcoming generation (Hassanat et al. 2019). This procedure within a generation cycle is referred to as *evolution strategy* (ES). We focus on $(\mu + \lambda)$-ES (Beyer and Schwefel 2002). μ is the number of parents that undergo mutations resulting in λ children. The "+" sign indicates that, when selecting the μ "fittest" mutants, parents *and* children are taken into account. Since EAs are meta-heuristic optimization methods, they typically yield only approximately optimal solutions to a problem. The major advantage, however, is that they can handle discrete high-dimensional and multi-objective inputs and that their objective- or *fitness* function does not have to be (continuously) differentiable as in the case of, say, gradient descent based methods.

3 Modeling the Problem

Next, we focus on our harvest scheduling problem by discussing constraints, the mathematical problem formalization, and adaptations to the optimization algorithm, i.e. oscillating mutation rate and the hierarchical loss.

3.1 Constraints

Seeding and growing of crop are fraught with several constraints and uncertainties. Two constraints are the windows or periods of possible planting and their planting as a whole at one point in time for each individual crop. Furthermore, we introduce the two variables, i.e. desired yield for each species and the "performance" of the field to accumulate crop growth. In phenology, the latter is often measured by *Growing Degree Units*. This is the accumulated time-dependent temperature $T(t)$ above a certain base temperature T_{base} during the interval of interest. Hence, $GDU = \int (T(t) - T_{\text{base}})dt$. Commonly, the interval is a day, which is why it is also referred to as Growing Degree Days (GDD) (McMaster and Wilhelm 1997). Seasonal (and geographical) variations in GDU accumulation have direct impact on the predicted harvest time. Thus, the harvest time is a probability density over time points, although further simplifications like mean harvest time and standard deviations or a simple time interval are conceivable as well. EAs work on discrete domains, so we will consider discrete time points and intervals. They do not necessarily have to be equidistant, but one usually considers days or weeks as units.

3.2 Formalization

Next, we introduce quantities that impact our problem. Let $\mathcal{D} = \{1, 2, \ldots, d_{\text{max}}\} \subset \mathbb{N}^+$ be the possible planting and harvesting dates. The dates themselves are encoded with an index $d \in \mathcal{D}$ as well as the earliest and latest planting dates for each species $s \in \mathcal{S}$, thus $d_{\text{early}}(s), d_{\text{late}}(s) \in \mathcal{D}$. We further introduce $\mathcal{S}$ as the set of species. Moreover, we have the required accumulated amount of GDU for each species to be harvested, i.e. $g_{\text{harvest}}(s)$. Thus, we calculate the harvest date by $d_{\text{harvest}}(s, d_{\text{plant}}) =$

$$\min_d \left\{ d \geq d_{\text{plant}} \;\middle|\; \sum_{x=d_{\text{plant}}}^{d} g_{\text{acc}}(x) \geq g_{\text{harvest}}(s) \right\}, \quad (1)$$

with GDU accumulation function $g_{\text{acc}}(x)$.

Given all this information, we define a harvest matrix $\mathbf{H} \in \mathcal{D}^{|\mathcal{S}| \times |\mathcal{D}|}$ that contains the harvest date for each species (as rows) and each planting date (as columns). We assign the value -1 impossible planting dates. Thus, $\mathbf{H} = \{H(i,j)\}$ where $H(i,j) =$

$$\begin{cases} d_{\text{harvest}}(s = i, d_{\text{plant}} = j) & \text{if } j \in [d_{\text{early}}(i), d_{\text{late}}(i)] \\ -1 & \text{otherwise} \end{cases}, \quad (2)$$

with harvest dates from Equation (1).

The precalculation of the harvest matrix reduces the schedule calculation time. With $\vec{q} = (q(s)) \in (\mathbb{R}_0^+)^{|\mathcal{S}|}$ as the vector of desired harvest quantity for each crop species, we can write the date harvest $\vec{h}(\vec{d}_{\text{plant}}) \in (\mathbb{R}_0^+)^{|\mathcal{D}|}$ as

$$\vec{h}(\vec{d}_{\text{plant}}) = \sum_{s \in \mathcal{S}} \left\{ \mathbf{I}_{d_{\text{max}}}(\mathbf{H}(s, d_{\text{plant}}(s)), d) q(s) \right\}_{d \in \mathcal{D}}, \quad (3)$$

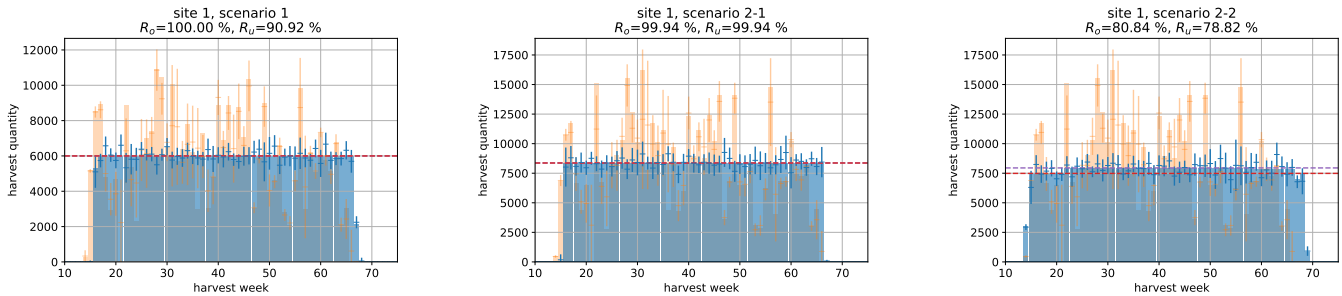

Figure 3: Weekly harvest quantity for all 3 scenarios on site 1. The bars show the harvest per week for an unoptimized schedule given in the dataset (orange) and our results (blue). The dashed lines mark target (red) and maximally needed capacity (purple). For uncertainty estimation (error bars), 100 harvest matrices are resampled from the GDU forecast. The over- and undershoot reduction ratios R_o and R_u refer to the unoptimized schedule. It is visible that our optimized schedule largely prevents over- and undershoot.

where $\mathbf{I}_{d_{\text{max}}}$ is the identity matrix with d_{max} dimensions mapping dates to themselves again. For instance, a date-to-week mapping matrix yields the weekly harvest.

3.3 Designing the Optimization Algorithm

In this work, we consider the $(1+1)$-ES. This turns out to be a good choice in our experiments, and there is no optimal choice due to the *No-free-Lunch-Theorem for Search* (Wolpert and Macready 1996). In the case that parent and child have the same loss value, we prefer the child to the parent, increasing the probability to overcome *plateaus* in the loss landscape.

Oscillating Mutation Rate. We draw the initial planting dates from a uniform distribution $\mathcal{U}$ by $d_{\text{plant}}(s) \sim \mathcal{U}[d_{\text{early}}(s), d_{\text{late}}(s)]$ for each species $s \in \mathcal{S}$ within its individual time window. Mutation is done by redrawing few elements. Further, we will use an initial mutation rate of $\frac{1}{|\mathcal{S}|}$, so we expect one planting to change by mutation. After this initial step, we exploit the advantages of an adaptive mutation rate (Blum, Puisa, and Wintermantel 2005; Agapie 2001) by introducing a counter $j = 0$ that is increased by 1 if the mutant is worse than the parent. This is done up to a maximum ratio ρ_{max} of elements. If the mutant is better than the parent, the counter is reset to 0. Moreover, there is the mentioned "plateau case" where the mutant is as good as the parent. In this case, the counter keeps its current value. Hence, we formulate a mutation rate that is oscillating between $\frac{1}{|\mathcal{S}|}$ and ρ_{max} with frequency ω.

$$\rho(j) = \frac{1}{|\mathcal{S}|}\left(1 + (\rho_{\text{max}}|\mathcal{S}| - 1)\sin^2(\omega j)\right), \tag{4}$$

which enables the algorithm to try a wide range of mutations and escape from local optima without turning into a random walk. This principle is connected with the concept of "temperature" in simulated annealing (Kirkpatrick, Gelatt, and Vecchi 1983) but with advantage to be adaptive to the landscape in vicinity of the current best solution.

Hierarchical Loss Function. First, we define the loss vector $\vec{L}(\vec{h}) = (L_+, L_-) = \left(\sum_{h\in\vec{h}_+} l_C(h), \sum_{h\in\vec{h}_-} l_C(h)\right)$ with

$$l_C(h) = \begin{cases} \frac{h}{C}\left(1 - \frac{h}{C}\right) & \frac{h}{C} < 1 \\ \exp\left(\frac{h}{C}\right) - \exp(1) & \frac{h}{C} \geq 1 \end{cases}, \tag{5}$$

which has to be minimized in the following. $\vec{h}_+$ only contains harvest quantities above and equal to the capacity limit, and $\vec{h}_-$ only contains those below. We discriminate between two cases with different optimization goals. For $h(d) < C_{\text{target}}$, we want either full capacity harvest or no harvest at all, whereas for $h(d) \geq C_{\text{target}}$, we want to minimize overshoot. The total harvest is constant, meaning that decrease of harvest at one date simultaneously increases harvest at another date by the same quantity. Our loss promotes high-harvest dates getting "full" in terms of the target capacity, while simultaneously eliminating low-harvest dates. Thus, in case of $h < C_{\text{target}}$, two harvest quantities *diverge*, provoked by a concave loss function. For $h \geq C_{\text{target}}$, the involved dates' quantities *converge*, provoked by a *convex* function.

Although this might sound like a multi-objective approach, we still formulate a single-objective problem by preferring the loss in the region above capacity limit (L_+) over the region below it (L_-). Thus, every net change from over into under capacity is promoted. This is what we call *hierarchical loss* where hierarchy decreases with increasing element index in $\vec{L}$. The flexibility to include further objectives or knowledge at this stage is obvious since one may add loss categories.

4 Application and Results

In the considered dataset, two harvesting sites 0 and 1 are available. We will perform the optimization along two scenarios S1 and S2, targeting different objectives. In S1, the sites have weekly harvest capacity limits, whereas in S2, the capacity is not given. Here, a reasonable capacity should be found during optimization. The GDU accumulation for harvesting season 2020/21 is forecasted by using a historical dataset of 11 years for both planting sites. The accumulation curves are plotted in Figure 2.

The GDU accumulation forecast comes along with uncertainties. Fortunately, GPR models have the property of not just extrapolating unseen regions, but also giving the confidence of their predictions. The actual predicted value is a mean value with a Gaussian standard deviation. We can exploit this to propagate uncertainties and estimate the sensitivity of our optimization to different GDU accumulation scenarios, i.e. weather conditions, by resample many harvest matrices by the bootstrap technique (Efron and Tibshirani 1993) and evaluate our result for all. We further substitute the identity matrix in Equation (3) for date-to-date mapping with a day-to-week mapping binary week matrix $\mathbf{W} \in \{0;1\}^{|\mathcal{D}|\times|\mathcal{W}|}$.

We further compared our method with two established EAs, namely NSGA-II and MOEA/D, by using two multi-objective optimization strategies each. For a short-time run of 10^6 generations, all those variants are outperformed by our method. Additionally, we formulated the problem as a mixed-integer linear program (MILP) and solved it with CPLEX®. Here, we observed as well that our method leads to significantly better results than the exact one of the solver. For the sake of brevity, a more detailed elaboration and plots can be found in the Appendix.

4.1 Harvest Schedule Optimization

For the optimization scenarios, our proposed EA applies the adaptive mutation rate shown in Equation (4). We use $\rho_{\max} = 1\,\%$ and $\omega = 5 \times 10^{-4}$. The number of species $|\mathcal{S}|$ is 1375 for site 0 and 1194 for site 1.

Our hierarchical loss needs the capacity limit for the separation between the two regions. For S1, the target capacity is given. We split S2 into two slightly different approaches, S2-1 and S2-2, since "lowest capacity needed" is always a trade-off with the number of total harvest weeks. The optimization is exactly the same as for S1 but with different inference of the capacity limit. In S2-1, we link the target capacity limit to the number of harvest weeks the current solution needs. For S2-2, we determine the maximum number of possible harvest weeks. Thus, we expect S2-1 to need a higher capacity in less weeks whereas S2-2 needs a lower capacity in more weeks if not all possible weeks are harvest weeks after all.

In Figure 3, the weekly harvest quantity is plotted against the harvest weeks for site 1. The light yellow bars are the weekly harvest quantity if the plants are planted by the original planting schedule given in the data. Blue bars represent our optimized result after 10^9 generations.

5 Discussion

In the following, we illuminate our optimization results from a technical as well as a social impact point of view.

5.1 Optimization Process

All scenarios show that the main optimization process needs orders of 10^6generations to saturate at a moderately well optimized state. The optimization process plot in Figure 4 shows that behavior exemplarily for two scenarios and three independent runs each. Beyond 10^7 generations, fine-tuning

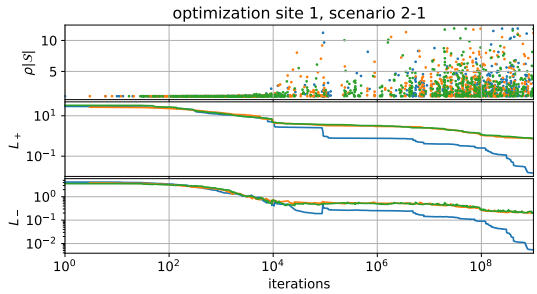

Figure 4: Optimization process with 3 independent runs. The two lower plots show the loss vector components L_+ and L_- against iterations. The upper plot shows the mutation rate for each successful improvement as a dot.

occurs, e.g., if a constellation is found that a complete harvest week can be canceled for. The optimization process directly shows the characteristics and the strength of our method. Firstly, the hierarchical character of the loss is observable by a monotonically decreasing L_+ and an L_- showing jumps to higher values. Secondly, fine-tuning preferably happens with higher mutation rates than the intermediate saturation. Using the adaptive mutation rate prevents the EA from getting stuck in local optima. More sophisticated optimization needs stronger mutations, i.e. higher mutation rates.

5.2 Social Impact

Our defined loss vector components have illustrative interpretations. For instance, L_+ minimizes food or crop waste by overshoot reduction, whereas L_- minimizes logistic costs accompanied by greenhouse gas emissions. By construction, our hierarchy prefers food waste prevention to emission reduction. Firstly, it reduces the food waste by accepting additional emission. Secondly, the emission is reduced by distributing the yield preferably efficient. Given the hierarchy, no new food waste is created thereby.

Furthermore, the bootstrap shows that the optimization just for one GDU forecast *overfits* on that single scenario. This is visible by the resulting error bars in Figure 3. Optimizing for many GDU forecast realizations is expensive due to many loss evaluations. Nevertheless, we observe that the uncertainty propagation from GDU forecast to harvest yield still stays within standard deviation. Thus, our optimization is robust against environmental forecast uncertainties and can schedule the harvest reliably. Additionally, the optimization can be adjusted on-the-fly, e.g. once a more precise forecast (harvest matrix) is available.

We introduce the overshoot (undershoot) reduction ratio R_o (R_u) that describes, to which extent we can reduce food waste and emissions with respect to the unoptimized case given in the data. As shown in Figure 3, we can reduce $62\,\%$ to $100\,\%$ of over- and undershoot for both sites and all scenarios which is a drastic improvement. Using our method, one could thus tackle global food waste on a supply chain as well as on a farming level, which additionally contributes to a more efficient agricultural land use.

References

Agapie, A. 2001. Theoretical analysis of mutation-adaptive evolutionary algorithms. *Evolutionary Computation*, 9(2): 127–146.

Bäck, T. 1996. *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms*. Oxford University Press.

Beyer, H.-G.; and Schwefel, H.-P. 2002. Evolution strategies – A comprehensive introduction. *Natural Computing*, 1(1): 3–52.

Blum, S.; Puisa, R.; and Wintermantel, M. 2005. Adaptive Mutation Strategies for Evolutionary Algorithms. *2nd Weimar Optimization and Stochastic Days*, 1–13.

Branke, J.; Deb, K.; Miettinen, K.; and Słowiński, R., eds. 2008. *Multiobjective Optimization: Interactive and Evolutionary Approaches*. Springer.

Cheraghalipour, A.; Paydar, M. M.; and Hajiaghaei-Keshteli, M. 2019. Designing and solving a bi-level model for rice supply chain using the evolutionary algorithms. *Computers and Electronics in Agriculture*, 162: 651–668.

Deb, K.; Pratap, A.; Agarwal, S.; and Meyarivan, T. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evolutionary Computation*, 6(2): 182–197.

Efron, B.; and Tibshirani, R. 1993. An Introduction to the Bootstrap.

FAO. 2013. *Food Wastage Footprint. Impact on Natural Resources*. Food and Agriculture organization of the United Nations.

FAO. 2019. *The State of Food and Agriculture 2019*. Food and Agriculture organization of the United Nations.

Ferrer, J.; Cawley, A.; Maturana, S.; Toloza, S.; and Vera, J. R. 2008. An optimization approach for scheduling wine grape harvest operations. *International Journal of Production Economics*, 112: 985–999.

Hassanat, A.; et al. 2019. Choosing Mutation and Crossover Ratios for Genetic Algorithms—A Review with a New Dynamic Approach. *Information*, 10(12): 390.

Holland, J. H. 1992. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*. MIT Press.

Kirkpatrick, S.; Gelatt, C. D.; and Vecchi, M. P. 1983. Optimization by Simulated Annealing. *Science*, 220(4598): 671–680.

McMaster, G.; and Wilhelm, W. 1997. Growing degree-days: one equation, two interpretations. *Agricultural and Forest Meteorology*, 87: 291–300.

Rasmussen, C. E.; and Williams, K. I. 2006. *Gaussian Processes for Machine Learning*. MIT Press.

Roghanian, E.; and Cheraghalipour, A. 2019. Addressing a set of meta-heuristics to solve a multi-objective model for closed-loop citrus supply chain considering CO2 emissions. *Journal of Cleaner Production*, 239: 118081.

Taşkıner, T.; and Bilgen, B. 2021. Optimization Models for Harvest and Production Planning in Agri-Food Supply Chain: A Systematic Review. *Logistics*, 5(3).

Varas, M.; Basso, F.; Maturana, S.; Osorio, D.; and Pezoa, R. 2020. A multi-objective approach for supporting wine grape harvest operations. *Computers & Industrial Engineering*, 145: 106497.

Wolpert, D.; and Macready, W. 1996. No Free Lunch Theorems for Search. Technical report, Santa Fe Institute.

Yu, R.; and Leung, P. 2006. Optimal Partial Harvesting Schedule for Aquaculture Operations. *Marine Resource Economics*, 21(3): 301–315.

Zhang, Q.; and Li, H. 2007. MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. *IEEE Trans. Evolutionary Computation*, 11(6): 712–731.

A Appendix

A.1 Comparison between Many-, Multi-, and Hierarchical Single-objective Optimization

In order to show that transferring the harvest scheduling into a single-objective optimization problem improves the convergence, we test our $(1+1)$-ES against the commonly known algorithms NSGA-II and MOEA/D as a baseline. To get a proxy for the speed of convergence, we feature how the methods perform in a short-run optimization stopped after 10^6 seen populations. We compare a many-objective approach with six efficiency and continuity criteria with our hierarchical single-objective approach. Additionally, a multi-objective approach by using the two components of the loss vector $\vec{L}$ as (coordinative) objectives is evaluated. The upper and middle plots of Figure 5 show the results for the three stated problem formulations. The goodness of optimization is shown by considering distance to the next optimum (capacity limit or zero harvest) for each harvest week. Thus, lower bars represent better optimization. As expected, our approach shows a faster convergence to a rather preferable interim solution. Certainly, the solutions can be further improved by a long-run optimization.

A.2 Comparison with Exact Methods

Indeed, it is possible to formulate our problem as a mixed-integer linear programming (MILP) problem. However, as stated in the Introduction chapter, exact methods like MILP need to have objective functions that are less generic than those of, for instance, (meta-)heuristic approaches like EAs. Thus, we have to deal with several limitations that can not be considered by the MILP method. For instance, the concave loss function cannot be minimized by the solver. Additionally, our scenarios with variable target capacity cannot be represented by an appropriate objective function either. The lower plot in Figure 5 shows the result, if we solve the S1 scenario (fixed target capacity) of our harvest scheduling problem by using the commercial solver CPLEX®. As an objective function, we are limited to use the summed squared error over all weeks between harvest yield and target capacity. As well as for the common two compared EA methods, our method clearly outperforms the MILP method. Note that the given MILP solution is already the best solution we can get at this point while our method was stopped in an early state. This vividly shows that complex problems like harvest scheduling can be solved better if the used optimization method allows for including sophisticated objectives and prior knowledge.

A.3 Code Availability

The dataset used in this work is restricted by copyright. Nevertheless, we generated a similar dataset with the same characteristics. Example code to reproduce optimization results and comparison with NSGA-II and MOEA/D is available at https://github.com/mrcgndr/harvest_schedule_optimization.

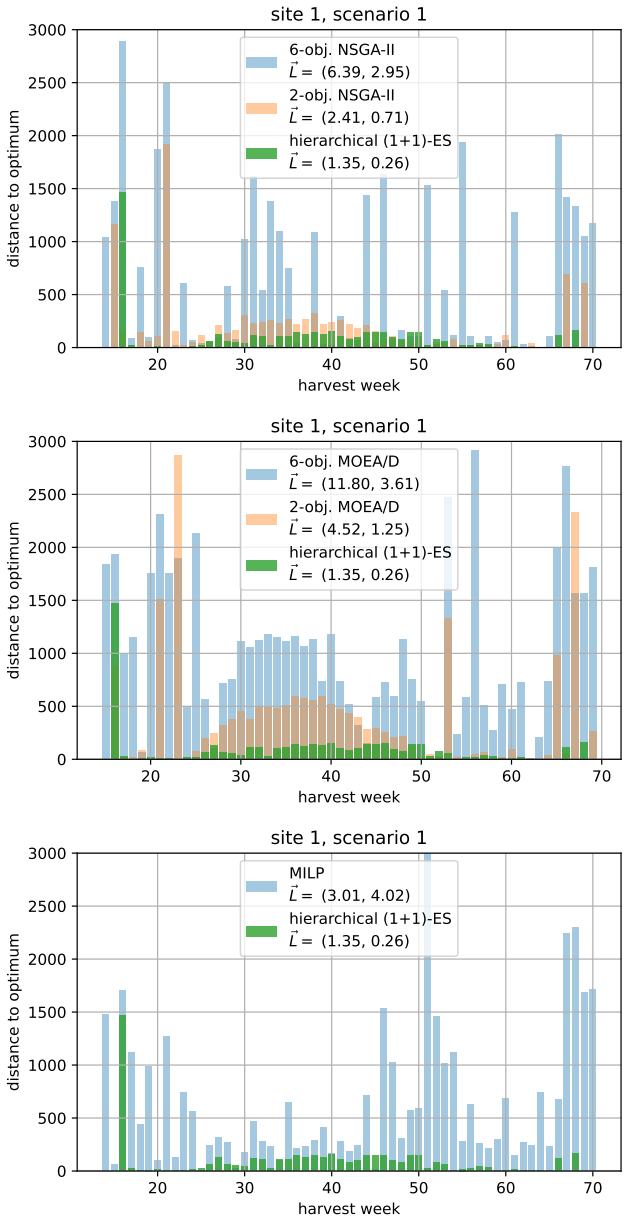

Figure 5: Comparison of different optimization strategies. For S1 and site 1, we compare our method with two established EAs, namely NSGA-II (top) and MOEA/D (middle). In addition, the bottom plot shows a comparison with an exact method represented by formulation as a mixed-integer linear program (MILP) solved with CPLEX®. Evolutionary approaches are stopped after 10^6 evaluations. The bars show the distance to the next optimum, i.e. capacity limit or zero harvest. Thus, the lower the bars, the better the optimization state. For the evolutionary approaches, we show the 6-objective (blue bars) and the 2-objective (orange bars) optimization with the respective algorithm. Our $(1+1)$-ES method, shown by the green bars in all plots, has the lowest overall deviations and, thus, outperforms the reference methods. The hierarchical loss, given in the legend, is only used for the 2-objective and our approaches.