Honda Research Institute EU
Honda Research Institute Europe GmbH
https://www.honda-ri.de/

An Adversarial Optimization Approach for the Development of Robust Controllers

Baraq Mushtaq, Tobias Rodemann

2020

Preprint:

This is a post-peer-review, pre-copyedit version of an article published in Applications of Evolutionary Computation. EvoApplications 2020. The final authenticated version is available online at:
https://doi.org/10.1007/978-3-030-43722-0_25

An Adversarial Optimization Approach for the Development of Robust Controllers

Mohammed Baraq Mushtaq[1] and Tobias Rodemann[2]

[1] Technical University of Darmstadt
Karolinenplatz 5
64289 Darmstadt
Germany
mohammed_baraq.mushtaq@stud.tu-darmstadt.de
[2] Honda Research Institute Europe
Carl-Legien-Strasse 30
63073 Offenbach/Main
Germany
tobias.rodemann@honda-ri.de

Abstract. Due to increasing popularity of electric vehicles there is rising demand for smart controller solutions that optimize the flow of energy between buildings and electric vehicles. Simple rule-based controllers are (often manually) developed and tuned for specific use case scenarios, for example a specific building, mobility usage patterns and country-specific regulations. However, it is often very difficult to correctly anticipate the exact conditions the controller has to work on so that a high performance under worst-case conditions is a very important target. In this work we use an adversarial optimization approach in order to find both challenging scenarios and controller parameterizations that perform well in those scenarios. We can show that in comparison to a standard controller, our approach can find challenging scenarios for a standard controller and controllers that outperform the baseline on those worst-case scenarios.

Keywords: Adversarial Optimization, Controller, Robustness, Smart Grid

1 Introduction

Tuning controller parameters using evolutionary algorithms is a well-known approach [7]. However, over-fitting to specific settings is a constant danger, as with machine learning approaches in general. It is therefore essential to tune the controller on a representative set of probable scenarios. Unfortunately, this is not possible in many practical instances, when there is a large variety of possible scenarios. An example we investigate in this work is a smart charging controller that needs to deal with different situations at the customers' location, which are difficult to anticipate. Instead of performing a random sampling of potential scenarios we therefore propose an adversarial optimization approach, where

The final publication is available at Springer
http://dx.doi.org/10.1007/978-3-030-43722-0 25

two antagonistic optimization processes are working in parallel. One optimization process is trying to maximize the performance of a controller on a set of scenarios, while the other optimization process is aiming to find worst-case scenarios for the current controller. We find that through this approach we can tune controllers that can cope with worst-case scenarios and randomly sampled scenarios. After laying out an overview of recent research, the application domain is described. Subsequently, an iterative co-evolutionary optimization approach is introduced. Enhancing the previous approach an archive-based optimization method is proposed. Finally, the obtained results for the system model are discussed.

2 Related Work

To cope with uncertainties for the system's design, the concept of robust optimization is applied, where objective functions are optimized considering a confined set of feasible use-case scenarios [3]. Going further, there has been research on determining worst-case robust solutions. This problem can be described from a game theoretic perspective, where two optimization procedures for controller and scenario space are competing against each other [8]. Evolutionary optimization methods have been employed for such worst-case optimization problems. A nested minimax optimization is proposed in [16]. A neural net controller for a highly uncertain and safety critical plant is optimized with evolutionary algorithms using a bi-level optimization loop. In the lower level, for a controller configuration the worst possible plant is identified for every iteration in the upper loop. The goal of the upper level optimization loop is to identify a controller, which optimizes the fitness for the worst possible plant. This approach yields robust controllers with the best worst-case performance. In [2, 1], an iterative co-evolutionary optimization approach is presented. In this approach, in an iterative manner one population is frozen while the other population evolves for a given number of iterations. To avoid a cycling effect, where both populations follow local optima without yielding significantly better individuals, a $hall\ of\ fame$ is proposed in [11]. The authors suggest to store and utilize previously identified worst-case scenarios for the fitness assignment. The competitive co-evolutionary optimization approaches have mainly been evaluated on test problems, the application on real world problems is yet to be done. Fundamental limits to worst-case or adversarial robustness for classifiers have been introduced in [10]. Although the paper focuses on different types of classifiers, there might be similar difficulties in finding a single adversarial robust control strategy for an arbitrary scenario space.

3 Methods

3.1 Application Domain: Smart Charging

Smart charging strategies are a key factor to optimally control power flows in energy supply systems with renewable energy sources (RES). In this work a rule-

based energy management strategy (EMS) is implemented to enable sustainable and cost-efficient power flows within the energy supply system of a building complex and an attached charging station for electric vehicles (EV). The system model investigated is characterized by use of RES and utilization of an energy storage component. A predictive control strategy for efficient charging of EVs is introduced in [15]. Moreover, in [5] a predictive control strategy for a stationary battery is proposed. The realized EMS in this work combines and extends these previous approaches. In the following, the overall system configuration is introduced. The implemented EMS is subsequently described, including a scenario description for the control strategy. Finally, the objective or fitness function for the optimization of the EMS is defined.

3.2 System Configuration

The energy supply system can be divided into demand and supply side, additionally a storage component enables power flexibility for the system. The supply side comprises photovoltaic power modules (PV) and the external electricity grid, as the energy system is connected to the grid. Whereas, the demand side consists of the building power demand profile and the energy demand for EV charging. While the demand profile of the building complex can be estimated quite well based on measured data, the charging behavior of EVs and the respective power demand, however, holds higher uncertainty. The main sources of uncertainty stem from the state of charge (SoC) of EVs to be charged, charging time duration, arrival time of EVs at the charging station or the price of electricity.

The stationary battery within the system provides flexibility during a net surplus of PV power. The excess power is transferred into the battery storage. When there is net shortage of power, the battery can be discharged. The EMS controls the power flow of the EV charging station and the stationary battery. Both, energy supply system and EMS are parameterized. Fig. 1 gives an overview on the configuration of the system model. The model is set up in SimulationX and is composed of modules of the GreenCity Library [9, 17].

Control Strategy and Scenario Description The implemented EMS, as described before, is comprised of predictive EV and battery charging strategies. The control strategy tries to mitigate peak power demand for the charging process and to maximize self-consumption of the power generated by the PV module. The EMS can be illustrated using decision trees. Different operating modes are activated when certain threshold values are exceeded or not. For instance, at a certain value for the required power of the building complex, the charging power for electric vehicles is reduced. Accordingly, the thresholds parameterize and describe the control strategy. These parameters can then be optimized for specific objective functions. Fig. 2 shows the structure of an implemented EV charging strategy.

As mentioned before, the energy supply system is parameterized. The parameters define the system behavior and comprise, amongst others, the load

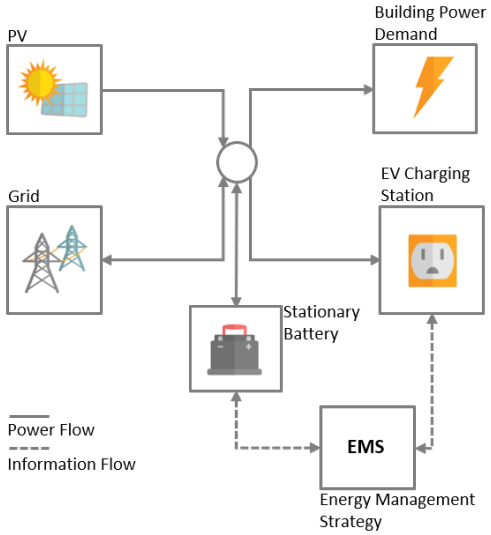


Fig. 1. Configuration of the analyzed energy supply system. The power supply within the system is provided by PV modules and the external grid. Building power demand and charging power demand from the EV charging station constitute the demand side. The energy system contains a stationary battery. The energy management strategy controls the power flow of the stationary battery and the EV charging station. Note the legend for the different connection types. (Figure has been designed using resources from Freepik.com)

profile for the building power demand, the averaged arrival time of EVs at the charging station and the standard deviation of the SoC of arriving EVs. These parameters are termed scenario parameters in the following.

3.3 Objective Function Formulation

The three objectives for the EMS are cost efficiency, resilience and customer satisfaction for EV charging. Hence, energy costs, time duration of grid independent operation and a customer satisfaction indicator (α) for EV charging are calculated. Equations 1 to 3 show these three objective functions. Energy costs (f_{cost}) are calculated by the sum of a base price (c_{bel}) and electric energy bought from (E_{supply}) and sold to (E_{feed}) the external grid over the simulation time. The price for bought energy c_{grid} holds a higher value than the remuneration c_{feed} for sold energy. Resilience for the energy system is measured by the minimum time duration (in minutes) in which the system can supply the power load P_{load} only using stored energy E_{bat} of the stationary battery. As indicated in Equation 2, the quotient of stored energy and power load is a function of time, the minimum of this function over the simulation time determines the resilience value

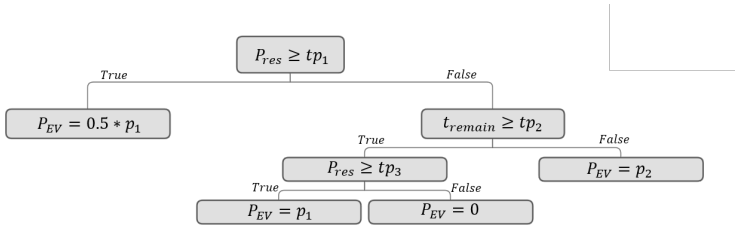


Fig. 2. Structure of the EV charging strategy within the EMS. P_{EV} denotes the EV charging power, which equals to p_i depending on the operating mode. Different operating modes are activated when the threshold parameters tp_i are exceeded for the remaining charge duration t_{remain} at the EV charging station and the residual power P_{res} between building power demand and PV power generation.

f_{res}. The function of the customer satisfaction index α is shown in Fig. 3. When the final SoC value after a charging process moves towards the desired SoC value ($SoC_{desired}$) for a charging process, α rapidly increases. Charging beyond $SoC_{desired}$ yields small increases of α. For a detailed description of the customer satisfaction index, it is referred to [15].

$$f_{\text{cost}} = c_{\text{grid}} * E_{\text{supply}} - c_{\text{feed}} * E_{\text{feed}} + c_{\text{bel}} \tag{1}$$

$$f_{\text{res}} = \min \frac{E_{\text{bat}}}{P_{\text{load}}}(t_{\text{sim}}) * 60 \tag{2}$$

$$f_{\text{EV}} = \alpha(SoC) \tag{3}$$

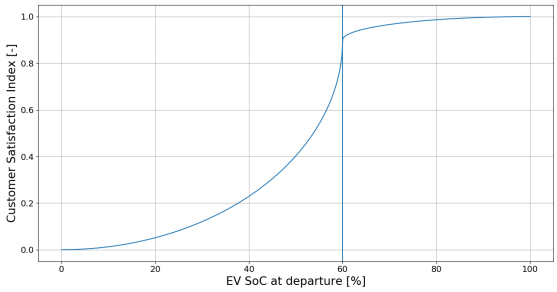


Fig. 3. Customer satisfaction index for the EV charging process [15]. Note the change in the function curve at the desired state of charge of 60 % for a charging process ($SoC_{desired}$).

Desirability Function The introduced objective functions have different value ranges. One approach to obtain an aggregated objective is to average over the objective functions [19]. However, varying scales of the objective functions impede the balanced optimization of the aggregated objective. Since, larger scale objective functions may have stronger weights on the overall objective. Therefore, desirability functions are employed [18, 14]. Desirability functions allow for uniform scaling of objective functions. This uniform scaling allows for proper aggregation of objective functions with various value ranges. The aggregated function can subsequently be utilized in an optimization framework. The desirability function $d(f)$ for a given objective function f is determined by a nonlinear transformation according to Equation 4 (with $b0, b1$ as manually tuned parameters).

$$d(f) = \exp\left(-\exp\left(-(b_0 + b_1 * f)\right)\right) \tag{4}$$

The desirability function is scaled to the interval $I = [0, 1]$. High desirability values correspond to objective values of high quality. For a detailed description of desirabilities, it is referred to [18]. The three resulting desirabilities are aggregated by calculating the arithmetic mean. The optimization is formulated as a minimization problem. Hence, the aggregate desirability function is internally defined as $1 - d(f)$.

4 Adversarial Optimization

4.1 Optimization Problem

For the development of robust control strategies, a certain performance level ought to be ensured even for failures in the system or unpredicted scenarios. Therefore, discrete scenarios can be sampled from the scenario space S. The aim here is to identify controller parameterizations, which exhibit low sensitivity to uncertainties in the scenario space. This approach establishes to some extent robustness against scenario variations. However, the robustness strongly depends on the sampled scenarios. To establish worst-case robustness of solutions, worst-case scenarios need to be specifically considered in the optimization process. For larger scenario spaces with many possible scenario configurations, the identification of worst-case scenarios is itself an optimization problem. Equation 4.1 shows the formulation of the adversarial optimization problem [13]. Formulating the problem gives conceptual insights into adversarial optimization. Essentially, adversarial optimization unites two optimization problems. Since the objective function for both problems is identical, a strongly coupled problem results. Moreover, considering a game theoretic perspective, both optimization problems act as antagonistic players. The goal of the first player is to identify a worst-case scenario $\boldsymbol{s}*$, for that the second player tries to find a respective worst-case robust solution $\boldsymbol{x}*$. Where the vector $\boldsymbol{s}$ comprises all scenario parameters for the energy supply system and $\boldsymbol{x}$ consists of all threshold parameters for the control strategy.

$$\begin{array}{cc} \min f_{\text{agg}}(\boldsymbol{x}, \boldsymbol{s}) & \max f_{\text{agg}}(\boldsymbol{x}, \boldsymbol{s}) \\ \boldsymbol{x}_{\text{l}} <= \boldsymbol{x} <= \boldsymbol{x}_{\text{u}} & \boldsymbol{s}_{\text{l}} <= \boldsymbol{s} <= \boldsymbol{s}_{\text{u}} \end{array} \tag{5}$$
$$g(\boldsymbol{x}, \boldsymbol{s}) = P_{\text{res}} - P_{\text{gmax}} <= 0$$

Here, f_{agg} denotes the aggregated objective function. The inequality restriction $g(\boldsymbol{x}, \boldsymbol{s})$ ensures that the power supply satisfies the power demand. Hence, $g(\boldsymbol{x}, \boldsymbol{s})$ restricts the residual power P_{res} to be less than or equal to the maximum grid power supply P_{gmax}.
Co-evolutionary optimization methods can be employed to solve such adversarial problem structures. In this approach, scenario and controller populations are optimized or evolved iteratively [2, 1]. In the following, the applied co-evolutionary optimization method is presented. Subsequently, enhancing the co-evolutionary approach, an archive based optimization method is proposed.

4.2 Archives

Optimization Archives are commonly used in dynamic optimization to store good solutions and retrieve them, when the objective space changes [6]. This can enable efficient optimization of a dynamically changing objective function. As mentioned in section 2, fundamental limits to adversarial robustness may exist. Based on the insight, that robustness of a single solution against all possible or even multiple worst-case scenarios might not be feasible, optimization archives for adversarial optimization are proposed in this work. It is assumed, that for enabling a level of robustness against different worst-case scenarios, a set of robust controllers is necessary. Such robust controllers ought to be stored in an archive during the optimization. Hence, in this work, optimization archives are adopted for adversarial optimization to obtain a set of optimal controllers at the end of an optimization run. The remaining question is, how individuals of the controller population are stored in the archive. It is important to maintain a diversity of good control strategies throughout the optimization. Therefore, a variance-based method is adopted from [6]. The individual k in the archive is removed, which maximizes the variance Var of the remaining control strategies (Equation 6). The current best control strategy from the controller population is subsequently stored in the archive.

$$\text{Var}(k) = \sum_{l=1}^{m} \sum_{a \in A/\{k\}} (x_{\text{la}} - \overline{x}_{\text{l}})^2 \tag{6}$$

Here, m denotes the length of the control strategy or decision variable $\boldsymbol{x}$, A the archive and k the currently dismissed controller in the archive. Var is determined by summing up the variance of all elements of $\boldsymbol{x}$. The mean value of element l over all variables in the archive is given by $\overline{x}_{\text{l}}$, x_{la} represents the value of element l for the a_{th} variable in the archive. In summary, the current best control strategy

replaces the strategy k in each iteration, maintaining a diverse archive of a given size.

4.3 Co-evolutionary Approach

First, the iterative coevolutionary optimization (CoevoI) is introduced. The optimization method for CoevoI is illustrated in Fig. 4. In this approach, iteratively one population is optimized for the previous best individual of the other population. Precisely, the controller population is optimized for the worst scenario identified in the previous scenario optimization. The scenario population is likewise optimized for the best previously identified controller. Utilizing the optimization archives mentioned in the previous section, a diverse set of robust control strategies is stored over the optimization run.

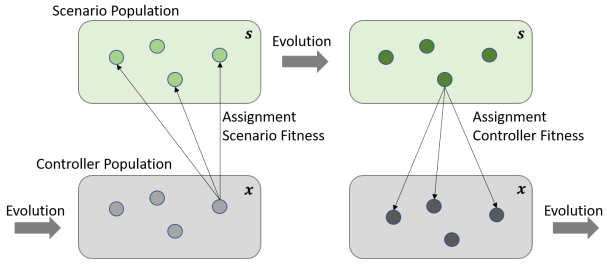


Fig. 4. Iterative co-evolutionary optimization (CoevoI). The controller population is optimized for the best scenario configuration $\boldsymbol{s}_{\mathrm{i}}$, identified in the previous scenario optimization loop. The scenario population is optimized vice versa. The current best control strategies $\boldsymbol{x}_{\mathrm{i}}$ of the controller population are stored in the controller archive using the variance-based storage scheme.

4.4 Nested, Archive based Optimization

Co-evolutionary optimization methods have to cope with intransitive behavior, where the optimization continuously moves towards different local optima [8]. As described in section 2, in [11] it is suggested to store identified worst-case scenarios and utilize them for the controller optimization. However, this goes along with rising computational costs, since in every iteration an additional worst-case scenario needs to be considered. In this work, to enhance the co-evolutionary optimization methods it is proposed to apply the concept of optimization archives for the scenario optimization. This ought to maintain a diverse set of worst-case scenarios, while limiting computational costs. It is assumed that a variety of identified worst-case scenarios can be represented by an archive of a given size n. The archive in each outer iteration is generated by determining the subset

of n scenarios, with the highest variance, drawn from the whole set of identified worst-case scenarios over the optimization run. Equivalent to the variance calculation for the controller archives, the variance of the scenario archive is determined using Equation 6. Hence, a nested, archive based optimization method (NAOpt) is proposed. Using a nested approach, the worst-case scenario for each current best control strategy can be determined. The nested approach allows more iterations for the scenario optimization, since a premature convergence of the scenario optimization is not an issue as opposed to the iterative approach. Convergence is rather favorable for each scenario optimization. The proposed optimization method is shown in Fig. 5. Similar to the iterative approach, a set of robust control strategies is also archived over the optimization run.

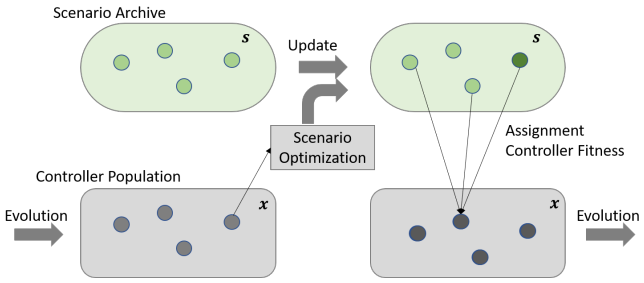


Fig. 5. Nested, Archive based optimization (NAOpt). In each outer iteration, the nested scenario optimization identifies the worst-case scenario for the current best control strategy. The subset of previously identified worst-case scenarios with maximum variance is stored in the scenario archive. For the controller population a minimax fitness assignment is utilized. Each controller individual is evaluated against all individuals in the scenario archive. The current best control strategies are stored in the controller archive using the variance-based storage scheme.

5 Results

5.1 Performance Evaluation

This section begins with a detailed description of the evaluation procedure for the analysis of the worst-case robustness for the optimized control strategies. Subsequently, the results for the iterative and archive based optimization methods are compared. The concept of Evolutionary Trajectories is then introduced for further analysis of the adversarial optimization methods.

Since the objective function $f(\boldsymbol{x}, \boldsymbol{s})$ is evaluated via an engagement of control strategy $\boldsymbol{x}$ and scenario configuration $\boldsymbol{s}$, the performance or fitness for either populations is relative. A good fitness function value for a controller can only be considered as such, when the corresponding scenario is a challenging one. The

fitness of each scenario itself depends on the controller population it is evaluated against. Hence, for a better evaluation of the results, a baseline control strategy is employed. The employed reference EMS (REMS) is similar to a reference control strategy described in [7]. The charging power of the EV charging station is set to a constant value. The battery is charged when the residual power P_{res}, which equals to the difference between building power demand and PV power generation, exceeds a threshold parameter tp_{charge}. The battery is discharged when P_{res} is below the threshold value tp_{charge}.
The baseline controller is evaluated against all worst-case scenarios identified during an optimization run and against randomly sampled scenario configurations. Similarly, the archive of optimized controllers from an optimization run is evaluated. The performance of the baseline controller can then be compared to the averaged performance of the corresponding archive. By evaluating the performance of the baseline controller against the identified worst-case scenarios, the severity of these scenarios can be analyzed. Comparing the performance of baseline to optimized control strategies gives insight into the robustness of the latter. For each controller in an optimization archive the worst performances for the worst-case scenarios and the randomly sampled scenarios are determined. The performance of a specific controller is determined by averaging these two performance values. Subsequently, the performances of the archives' controllers are averaged. The performance of the baseline is similarly calculated by averaging the performances for the worst-case scenarios and the randomly sampled scenarios. In the last step, the performance of the controller archives for each optimization run is then averaged. Similarly, the performance of the reference controller is averaged. Fig. 6 visualizes the evaluation procedure. The performance over the worst-case scenario $f_{\text{worst case}}$ is calculated as in Equation 7. The performance for the randomly chosen scenarios f_{random} is calculated according to Equation 8. As mentioned above, both calculated values for $f_{\text{worst case}}$ and f_{random} are averaged to determine overall performance.

$$f_{\text{worst case}} = \max_{\boldsymbol{s} \in S_{\text{worst case}}} f(\boldsymbol{x}_{\text{archive}}, \boldsymbol{s}) \tag{7}$$

$$f_{\text{random}} = \max_{\boldsymbol{s} \in S_{\text{random}}} f(\boldsymbol{x}_{\text{archive}}, \boldsymbol{s}) \tag{8}$$

5.2 Fitness Gap Analysis

In this section the optimization methods are evaluated and compared regarding the severity of the identified worst-case scenarios as well as the robustness of the optimized control strategies.
Both optimization methods are implemented utilizing the CMA-ES framework [12]. Considering computational costs, the population size for controller and scenario optimization is chosen to be 10. Initial value for mutation rate is specified to 0.2 for the scenario population and 0.25 for the controller population, these values correspond to mutation rates recommended for the used framework. The

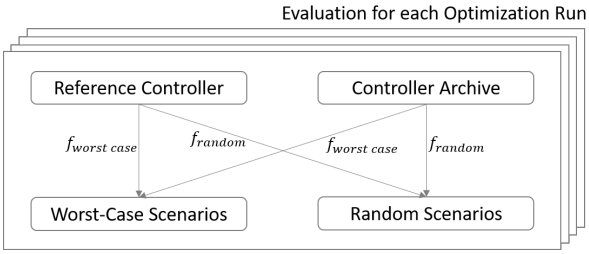


Fig. 6. Evaluation Procedure. For each optimization run, the reference controller and the identified controller archive are evaluated against the worst-case scenarios, collected over all iterations, and against randomly sampled scenario configurations. The performance of the controller archives over all optimization runs is subsequently averaged. Similarly, the average performance of the reference controller is determined.

robustness of control strategies for severe scenarios is determined by the desirability of the objective function value. The robustness of the control strategies is better for higher desirability values, worse robustness is indicated by lower desirability values. Likewise, analyzing the severity of identified worst-case scenarios, lower desirabilities for the baseline strategy indicate higher severity and vice versa.

Iterative Co-evolutionary Optimization For the iterative co-evolutionary optimization method (CoevoI), the number of inner iterations for the scenario optimization is set to 50 and for the controller optimization to 25. The outer iteration number is set to 20. More iteration steps are allocated to the scenario optimization loop. This ought to ensure, that scenarios with certain severity are identified. The controller population may slowly move towards better performance. Tab. 1 shows the averaged desirability values for baseline and optimized controllers. The baseline controller with a rather low mean desirability of 0.338 does not perform well for the identified worst-case scenarios. The optimized controller performs better with a mean value of 0.737. The results indicate, that the control strategies optimized by CoevoI show higher robustness compared to the baseline with a desirability difference (Δ Desirability) of approximately 0.4.

Nested, Archive based Optimization Tab. 2 shows averaged desirability values for baseline and optimized control strategies for the Nested, Archive based Optimization approach (NAOpt). The outer iteration number is set to 20. The number of inner iterations for controller optimization is set to 25. The number of inner nested iterations for scenario optimization is set to 70. The iteration steps for the scenario optimization in this nested approach are set higher compared to the iterative approach, since here for every iteration the worst-case scenario for the current best controller is searched. The mean desirability value of 0.148 for the baseline controller shows poor performance of the baseline for the worst-case

Table 1. Comparing averaged desirability values of baseline and optimized controller for the Iterative Co-evolutionary Optimization (CoevoI). Number of inner iterations for scenario optimization is set to 50 and for controller optimization to 25. The number of outer iterations is set to 20. A total of 20 optimization runs are analyzed.

Control Strategy	**Mean**	**Std. Deviation**
Baseline	0.338	0.170
CoevoI	0.737	0.162
Δ Desirability	0.399	

scenarios identified by NAOpt. The Δ Desirability value between baseline and optimized controllers amounts to approximately 0.6.

The severity of the identified scenarios is higher than for CoevoI by a difference in mean baseline desirability of 0.19. The difference in Δ Desirability between NAOpt and CoevoI amounts to 0.209. This indicates higher robustness of the controllers optimized by NAOpt. Higher severity of the identified scenarios is indicated by the lower mean desirability of the baseline control strategy.

Table 2. Comparing averaged desirability values of baseline and optimized controller for the Nested, Archive based Optimization (NAOpt). Number of iterations for the nested scenario optimization is set to 70 and for controller optimization to 25. The number of outer iterations is set to 20. A total of 20 optimization runs are analyzed.

Control Strategy	**Mean**	**Std. Deviation**
Baseline	0.148	0.106
NAOpt	0.756	0.170
Δ Desirability	0.608	

5.3 Evolutionary Trajectories

As indicated in section 2, adversarial optimization approaches can be described by predator-prey dynamics [8]. Fig. 7 shows a converging trajectory for a biologic predator-prey relation with competition within both populations [4]. Here, the variation in overall population size of predators and prey is reduced till there is almost no variation. Applied on adversarial optimization the convergence point bears similarities to a worst-case optimal solution, since from this point the population size or fitness value would not be reduced or worsened anymore. Evolutionary trajectories (ET) give insight into co-evolutionary dynamics during the optimization. Note that there is a key distinction between the displayed predator-prey dynamics and ET for adversarial optimization. The mean of the trajectory in Fig. 7 remains the same. However, for adversarial optimization this behavior is not necessary, rather a spiralling curve towards an optimal area in the objective space may be observed.

Fig. 8 to Fig. 9 show ETs for the implemented iterative and archive based optimization methods. The ETs show the relation between controller and scenario optimization. The trajectories consist of current best individuals for both optimization loops. For the controller optimization higher desirability or better performance and for the controller optimization lower desirability or worse performance is desired.

For the iterative approach (CoevoI) a rather straight trajectory towards higher controller and scenario desirability can be observed. CoevoI quickly start to converge similar to the predator-prey characteristic mentioned above. The iterative method reaches a local optimal area. The optimization then seems to follow local scenarios and controllers, this is indicated by the rather high variance of the convergence phase.

The nested, archive based optimization method (NAOpt) displays a different trajectory. In this approach worst-case scenarios, identified in previous iteration loops, are considered in the controller optimization. A key distinction of NAOpt compared to the previous method is the search path towards the convergence phase. Since a diversity of solutions is considered, utilizing scenario archives, the trajectory shows direction changes. This indicates proper searching for optimal areas. During the convergence phase there is low variance for the scenario fitness, however the variance of the controller fitness is a bit higher than for the iterative optimization approaches. In addition, the convergence phase shows decreasing variance. This is indicated by the sub-area with higher density within the convergence area. The final solution has worse controller desirability, but also better scenario desirability than for the iterative optimization method. Here, the desirability values for controller and scenario population seem to be balanced out more. This is opposed to the previous approaches, where for controller desirability or performance a rather direct increase is observed.

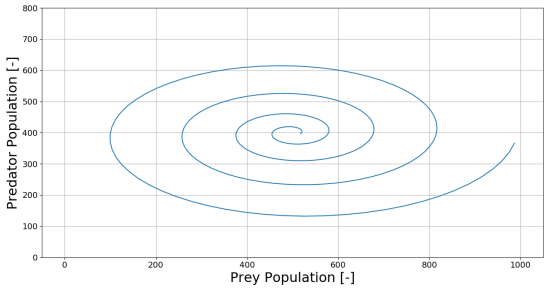


Fig. 7. Evolutionary trajectory of a biologic predator and prey population with intra-species competition, adapted from [4]

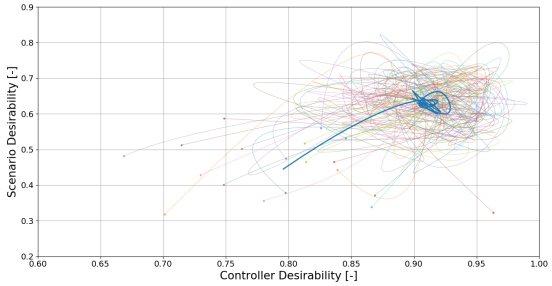


Fig. 8. Spline interpolated Evolutionary Trajectory of CoevoI. The trajectories consist of the current best individuals for both optimization loops. The mark at each trajectory indicates the starting point of the optimization run, the thicker line represents the averaged trajectory. A total of 20 optimization runs are analyzed. Higher variance of the convergence phase indicates low convergence for scenario and controller optimization.

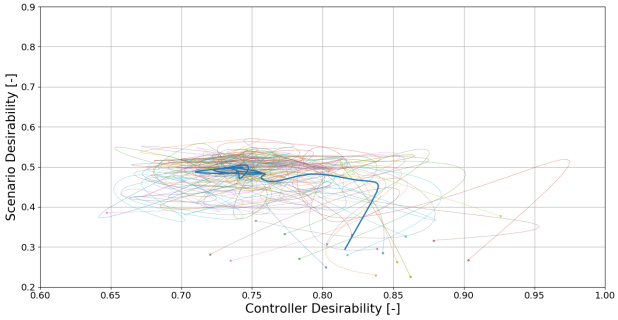


Fig. 9. Spline interpolated Evolution Trajectory of NAOpt. The trajectories consist of the current best individuals for both optimization loops. The mark at each trajectory indicates the starting point of the optimization run, the thicker line represents the averaged trajectory. A total of 20 optimization runs are analyzed. The trajectory shows a proper search path towards better performance areas in the objective space. Decreasing variance of the convergence phase can be observed.

6 Summary and Outlook

In this work we have implemented an adversarial optimization concept to identify robust controller parameters even for severe scenarios. Our analysis has shown that our approach can find scenarios more difficult than randomly sampled scenarios, and controllers that outperform baseline controllers on the evolved scenarios and also randomly sampled scenarios. The approach was tested on

the problem of smart charging of EVs in private homes under optimal self-consumption of PV power generation, energy costs, and customer satisfaction for EV charging.

We compared two different approaches for adversarial optimization, namely the Iterative Co-evolutionary Optimization (CoevoI) and the Nested, Archive based Optimization (NAOpt). The results showed that the utilization of scenario archives considerably improves the robustness of the optimized control strategies. This goes along with increased severity of the identified scenarios during the optimization path.

An analysis based on the concept of predator-prey dynamics showed that the adoption of evolutionary trajectories can give additional insights into adversarial optimization. Evolutionary trajectories show the relation between scenario and controller optimization, this allows to analyze the co-evolutionary dynamics of the adversarial optimization problem. The rather intuitive iterative optimization structure (CoevoI) exhibits negative co-evolutionary dynamics, where both optimization loops follow local optima. This manifests in the lack of finding severe scenarios and respective robust controllers. Whereas, NAOpt proves to identify more severe scenarios and respective robust controllers, considering an archive of worst-case scenarios identified over the course of optimization. The Evolutionary Trajectories for NAOpt accordingly display a different progress and convergence behavior compared to CoevoI. In particular, a proper search path towards optimal areas in the objective space can be observed. Additionally, the results show, that the chosen size of the scenario archive for NAOpt is able to represent worst-case scenarios identified during the optimization run, while reducing computational costs.

It is suggested to utilize the concept of Evolutionary Trajectories to further analyze and improve adversarial optimization methods. Improvement could be made by directly utilizing the Evolutionary Trajectory during the optimization, for instance in the fitness assignment. A more detailed analysis of the convergence behavior for controller and scenario optimization could give additional insights into the optimization methods. Better convergence might also narrow down the worst-case optimal area in the controller parameter space. For further research, the size of the controller archive might be analyzed. The assumption of needing a variety of optimal controller configurations to enable worst-case robustness could be further confirmed this way.

Acknowledgments

Baraq Mushtaq acknowledges the financial support from the Honda Research Institute Europe.

References

1. Al-Dujaili, A., Srikant, S., Hemberg, E., O'Reilly, U.M.: On the application of Danskin's theorem to derivative-free minimax optimization. arXiv preprint arXiv:1805.06322 (2018)

2. Barbosa, H.J.: A coevolutionary genetic algorithm for constrained optimization. In: Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406). vol. 3, pp. 1605–1611. IEEE (1999)
3. Ben-Tal, A., El Ghaoui, L., Nemirovski, A.: Robust optimization, vol. 28. Princeton University Press (2009)
4. Bischof, R., Zedrosser, A.: The educated prey: consequences for exploitation and control. Behavioral Ecology 20(6), 1228–1235 (2009)
5. Braam, F., Hollinger, R., Engesser, M.L., Müller, S., Kohrs, R., Wittwer, C.: Peak shaving with photovoltaic-battery systems. In: IEEE PES Innovative Smart Grid Technologies, Europe. pp. 1–5. IEEE (2014)
6. Branke, J.: Evolutionary optimization in dynamic environments, vol. 3. Springer Science & Business Media (2012)
7. Cheng, R., Rodemann, T., Fischer, M., Olhofer, M., Jin, Y.: Evolutionary many-objective optimization of hybrid electric vehicle control: from general optimization to preference articulation. IEEE Transactions on Emerging Topics in Computational Intelligence 1(2), 97–111 (February 2017)
8. Cramer, A.M., Sudhoff, S.D., Zivi, E.L.: Evolutionary algorithms for minimax problems in robust design. IEEE Transactions on Evolutionary Computation 13(2), 444–453 (2008)
9. ESI-ITI: SimulationX 4.0, http://www.simulationx.com/, https://www.simulationx.com/
10. Fawzi, A., Fawzi, O., Frossard, P.: Fundamental limits on adversarial robustness. In: Proc. ICML, Workshop on Deep Learning (2015)
11. Halck, O.M., Dahl, F.A.: Asymmetric co-evolution for imperfect-information zero-sum games. In: European Conference on Machine Learning. pp. 171–182. Springer (2000)
12. Hansen, N., Müller, S., Koumoutsakos, P.: Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation. Evolutionary Computation 11, 1–18 (2003), http://www.lri.fr/ hansen/publications.html
13. Herrmann, J.W.: A genetic algorithm for minimax optimization problems. In: Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406). vol. 2, pp. 1099–1103. IEEE (1999)
14. Ogino, Y., Iida, R., Rodemann, T.: Using desirability functions for many-objective optimization of a hybrid car controller. In: GECCO 2017 conference companion (2017)
15. Rodemann, T., Kitamura, K.: Simulation-based design and evaluation of a smart energy manager. In: Lecture Notes in Computer Science – Computer Aided Systems Theory. Springer (2019)
16. Sebald, A.V., Schlenzig, J.: Minimax design of neural net controllers for highly uncertain plants. IEEE Transactions on Neural Networks 5(1), 73–82 (1994)
17. Unger, R., Mikoleit, B., Schwan, T., Bäker, B., Kehrer, C., Rodemann, T.: Green building - modeling renewable building energy systems with emobility using Modelica. In: Proceedings of Modelica 2012 conference. Modelica Association, Munich, Germany (2012)
18. Wagner, T., Trautmann, H.: Integration of preferences in hypervolume-based multi-objective evolutionary algorithms by means of desirability functions. Evolutionary Computation, IEEE Transactions on 14(5), 688–701 (2010)
19. Zhu, L., Deb, K., Kulkarni, S.: Multi-scenario optimization using multi-criterion methods: A case study on byzantine agreement problem. In: 2014 IEEE Congress on Evolutionary Computation (CEC). pp. 2601–2608. IEEE (2014)