Evolutionary parameter optimization of a fuzzy controller which is used to control a sewage treatment plant

Marc Ebner
Eberhard Karls Universität Tübingen, Wilhelm-Schickard-Institut für Informatik, Abt. Rechnerarchitektur, Sand 1, 72076 Tübingen, Germany

Patrick Stalph
Julius-Maximilians-Universität Würzburg, Lehrstuhl für Informatik II, Am Hubland, 97074 Würzburg, Germany

Martin Michel
PASSAVANT-INTECH GmbH, Kettelerstr. 5-11, 97222 Rimpar, Germany

Roland Benz
Jacobs-University Bremen, School of Science and Technology, Campus-Ring 1, 28759 Bremen, Germany

Abstract
In order to meet new environmental standards, sewage treatment plants may need to be redesigned or extended. Instead of reconstructing large parts of a sewage treatment plant, which can be very costly, it is in many cases sufficient to install relatively inexpensive equipment, which controls parts of the plant in a new way. Fuzzy controllers are often used for this task. Use of these controllers often leads to an improved water quality. Such fuzzy controllers contain a number of parameters which are determined by a human expert. With this contribution, a dedicated multi-objective evolutionary algorithm is developed to optimize these parameters. The evolutionary algorithm is based on the successful strength pareto evolutionary algorithm 2 (SPEA2). The fuzzy control parameters, which are optimized are continuous parameters. Therefore, an evolution strategy was employed which uses the multi-objective ranking as used by the SPEA2 algorithm. Optimal parameters were first evolved on simulated sewage treatment plants. One set of parameters was also tested on an actual plant. Due to the enormous computational demands of simulating a sewage treatment plant, it is only possible to work with small population sizes. Nevertheless, it was possible to evolve parameters which were equally well as those found by a human expert indicating that the parameter tuning can be automized.

Keywords
Evolutionary Algorithm, Fuzzy Controller, Sewage Treatment, Intermittent Nitrification/Denitrification

Introduction

Sewage treatment plants play an important role in our sewage system. They clean communal and industrially created waste water up to a point where it can be safely released into the surrounding environment, e.g. a river. In recent years, the standards for the outgoing (purified) water have been increased. For instance, in Germany in 2002 the maximum allowed concentration of nitrogen has been reduced from 18 mg/l to 13 mg/l for plants serving populations of over 100.000 people (Karner et al. 2003). As the number of households or industrial plants feeding their sewage into the sewage plant grows, the sewage plant has to be redesigned or extended to meet the new standards. This is obviously very costly. However, it has become apparent that intelligent control methods can provide a solution to this problem. Instead of reconstructing large parts of the sewage treatment plant, it is in many cases sufficient to install relatively inexpensive equipment which controls parts of the plant in a new way and thereby leads to an improved water quality.

Until the introduction of intelligent control methods, plants were often controlled manually, e.g. by providing additional oxygen to the plant whenever this was thought to improve the purity of the outgoing water. PID controllers (Craig 1989) are usually not used for this task. The expert knowledge how to operate such a plant is best caught by a set of fuzzy rules which depend in the current state of the plant. That's why fuzzy controllers have been developed to operate sewage treatment plants (Manesis et al. 1998, Jordy et al. 1998). Such controllers have been used very successfully over the last 10 years in a number of sewage treatment plants in Germany. With this contribution, use of evolutionary methods (Eiben and Smith 2007) for optimizing these fuzzy controllers is explored. Evolutionary methods use the main operations of natural evolution (reproduction, selection and variation) to solve an optimization problem. Here, the goal is to maximize water quality for a given plant. Since the fuzzy controller uses several parameters to operate the plant, and the fuzzy controller is based on a considerable amount of expert knowledge, an evolution strategy (Rechenberg 1994) is used to find optimal parameters for this controller. So far, these parameters are set manually. Due to the fact that water quality is based on different criteria, a multi-objective evolution strategy is used to find parameters which are optimal for a given plant.

Computer simulations of a sewage treatment plant are used to evaluate different parameter settings. These simulations require an enormous amount of computational resources. Hence, it is only possible to work with very small population sizes. A (4/2,20) evolution strategy was used which was extended with an archive similar to the successful strength pareto evolutionary algorithm 2 (SPEA2) multi-objective optimization method (Zitzler et al. 2001). The (4/2,20) evolution strategy works with 4 parent individuals. For each generation, 20 offspring are created by adding vectors whose components are Gaussian distributed. The crossover operation is also used. The Darwinian principle of survival of the fittest (Darwin 1859) is used to select candidate parents for the next generation from the archive. With this approach it was possible to evolve parameters that could be used to drive an actual sewage treatment plant. When tested on the real plant, water quality improved compared to previously used manual settings.

This article is structured as follows. Background information about related work is given in the next section. Next, the fuzzy controller and the simulation environment is described. After that, the strength pareto evolution strategy, a multi-objective evolutionary algorithm, is introduced. This algorithm is used to find optimal parameters for the fuzzy controller. Finally, results from four different simulated sewage treatment plants are shown. A set of evolved parameters is also tested on a real plant.

Background

Manesis et al. (1998) propose the use of fuzzy logic to control a wastewater treatment plant. The idea is that the knowledge of the experts which previously manually operated the sewage treatment plant can be represented best as a set of fuzzy rules. The set of rules can be extended or adapted easily. Jordy et al. (1998) have developed a fuzzy controller for sewage treatment plants which operate in intermittent mode, i.e. which only have a single tank for nitrification and denitrification. The controller uses the current concentration of oxygen within the tank as well as the oxidation reduction potential to operate an oxygen blower which pumps oxygen into the tank whenever this is required.

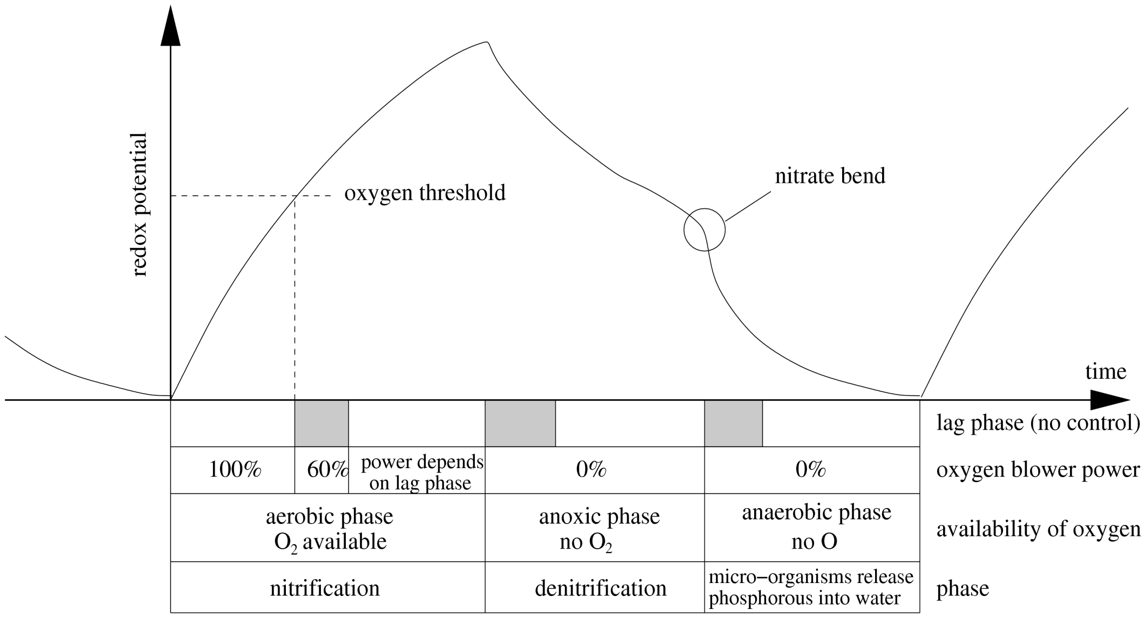


Figure 1: Idealized plot of the redox potential.

Figure 1 shows a typical plot of the oxidation reduction potential (see also Jordy et al. (1998) and Michel (2002)) within the aeration tank of a sewage treatment plant operating in intermittent mode. The oxygen reduction potential measures the ability of a physical or chemical system to oxidize materials. During the first phase (nitrification), oxygen is being pumped into the tank. During this phase, substances contained inside the water are oxidized. The number of free electrons inside the water rises which is measured by the oxidation reduction potential. During the second phase (denitrification), no oxygen is pumped into the tank. The free electrons are used up during denitrification. During this phase, bacteria contained inside the water reduce nitrate NO_3^- to N_2 and use up the remaining oxygen. The remaining electrons are used up during third phase when the phosphorus that has previously been taken up by the microorganisms is returned to the water. The controllers task is to keep this process going. It needs to choose the right times when additional oxygen is required and it also needs to stop pumping oxygen into the tank whenever a sufficient amount of oxygen has been pumped into the tank.

The controller developed by Jordy et al. (1998) uses an oxygen electrode and an oxygen reduction potential electrode as input. Based on the fuzzyfied measurements of the two electrodes, a compressor is operated which pumps oxygen into the tank at the appropriate times. The fuzzy controller is able to remove more nitrogen from the waste water compared to a conventional operation of the activated sludge plant. The amount of ammonia and nitrate was reduced considerably compared to a normal operation of the activated sludge plant.

Karner et al. (2003) improved upon the original fuzzy controller. They use two fuzzy controllers to operate the oxygen blower of the aeration tank. The first controller determines the time when the oxygen pump is switched off. It uses 273 if-then-rules set up by an expert. The second controller determines the time when the oxygen pump is switched on again. It uses 81 if-then-rules. Both controllers use data from two sensors which measure the NH_4-N concentration and the NO_3-N concentration. The first controller also uses input data from an additional sensor which measures the NH_4-N concentration in the outgoing sludge. Karner et al. (2003) found that the amount of flocculants needed to remove the phosphorus was reduced considerably by their controller. Other than that, they were also able to reduce nitrogen compared to a normal operation of the plant. The plant also needed less energy to operate (approx. 12% less) compared to a normal operation of the plant.

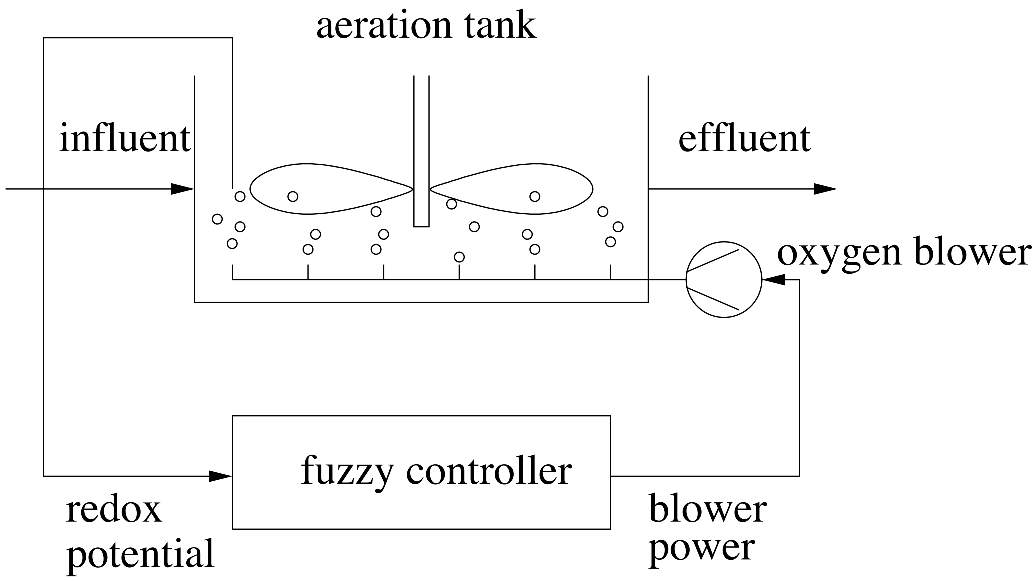


Figure 2:
A fuzzy controller is used to operate the power of the oxygen blower based on the measurements of the oxygen concentration and redox potential inside the aeration tank.

Over the last 10 years, this fuzzy controller has been continuously improved. Today, this fuzzy controller is known under the name Aqualogic (Karner and Benz 2001). It consists of several fuzzy controllers, e.g. to control recirculation of nitrate in the sewage, to apply flocculants to the plant or to add additional carbon during denitrification. For this work, the focus is on the controller which operates the oxygen blower (Figure 2). This controller determines when the aeration phase has been completed, when the anoxic phase is completed, when the anaerobic phase is completed and is also used to maintain the oxygen level within the activated sludge tank at a constant level during the aerobic phase. The entire Aqualogic controller consists of a rule base of 845 rules. The Aqualogic controller has been installed in almost 280 sewage treatment plants in Germany. It is maintained by the company PASSAVANT-INTECH, a spin-off from the chair for biotechnology at the University of Würzburg.

Table 1: Parameters used by the Aqualogic fuzzy controller

	Parameter Name	Minimum	Maximum
x_1	oxygen threshold	0.3 mg/l	3 mg/l
x_2	minimum nitrification duration	15 min	90 min
x_3	maximum nitrification duration	60 min	360 min
x_4	minimum denitrification duration	15 min	60 min
x_5	maximum denitrification duration	30 min	180 min
x_6	minimum biological phosphate-elimination duration	0 min	15 min
x_7	maximum biological phosphate-elimination duration	0 min	45 min
x_8	blower activation duration	40 %	140 %
x_9	maximum aeration duration	60 min	210 min
x_{10}	redox zero point	-0.1 mV/s^2	0 mV/s^2

Methods

A. The Aqualogic fuzzy controller

The Aqualogic fuzzy controller uses a number of parameters for its operation. The full list of parameters is shown in Table 1. See also Michel (2002). Each parameter has an associated reasonable range. These parameters had previously been set by a human expert. The oxygen blowers are operated at maximum speed during the start of the nitrification phase until the oxygen threshold has been reached. Once this has happened, the power of the oxygen blower is reduced to 60% during a lag phase. Data is gathered during the lag phase and the fuzzy controller then decides to what speed the oxygen blower should be set. The fuzzy controller also decides when the nitrification phase is over. The denitrification phase starts with a lag phase. The oxygen blower is completely turned off during this time. After that, the redox potential is continually monitored to determine the exact time when the nitrate bend has been reached. Once this has been detected, another lag phase is initiated. The duration of the biological phosphate-elimination phase depends on the slope of the redox potential. It ends when the slope is approximately zero. The length of this phase also depends on the minimum and maximum duration set for this phase. Such minimum and maximum duration parameters exist for all three phases.

The first seven parameters are used by the controller to decide when a fuzzy rule becomes active. Parameter x_1 determines when a sufficiently high oxygen concentration (as shown in Figure 1) has been reached. Parameters x_2 and x_3 determine the minimum and maximum duration of the nitrification phase (aerobic phase). Parameters x_4 and x_5 determine the minimum and maximum duration of the denitrification phase (anoxic phase). Parameters x_6 and x_7 determine the minimum and maximum duration of the phase when phosphate is eliminated (anaerobic phase). The final three parameters are used implicitly by the fuzzy rules. Parameter x_8 specifies the duration the blower is activated (in percent). A duration of 100% corresponds to the time in which ammonium is completely removed from the sewage assuming a temperature of 17°C. Parameter x_9 specifies the maximum duration the oxygen blower is turned on. Parameter x_{10} specifies the redox zero point that is used in determining when the nitrate bend has occurred.

Depending on how these parameters are set, the plant may or may not perform optimally. The goal is to reduce the ammonium NH_4-N, nitrate NO_3-N and phosphate PO_4-P concentrations in the outgoing water as much as possible. In other words, this is a multi-objective optimization problem (Deb 2001) and, generally, it is desirable to obtain all of the pareto-optimal settings for this controller. A variant of the highly successful SPEA2, adapted to work with continuous parameters, was used for this multi-objective optimization problem.

B. Simulating a sewage treatment plant

Experiments were carried out on a simulator of several accurately modeled sewage treatment plants. The experimental setup is shown in Figure 3. An evolution strategy is used to find optimal parameters for this fuzzy controller. Each individual of the population represents a set of parameters. The Aqualogic fuzzy controller receives the parameter set from the evolution strategy and uses these parameters to control the sewage treatment plant. Based on the parameters, the fuzzy controller decides when to activate the blower in the aeration tank. For each parameter setting the behavior of the plant is simulated for approximately four days. Fitness is computed based on the quality of the outgoing water. Therefore, the fitness of a particular parameter set is obtained by simulating the virtual sewage treatment plant using these parameters. Since ammonium NH_4-N, nitrate NO_3-N and phosphate PO_4-P concentrations should be reduced, fitness is a three component vector. For each component a maximum concentration is given which must not be exceeded at any time. These limits depend upon local standards and specifications for the plant. The goal is to minimize all three components. The fitness components together with the maximum allowed concentrations are shown in Table 2.

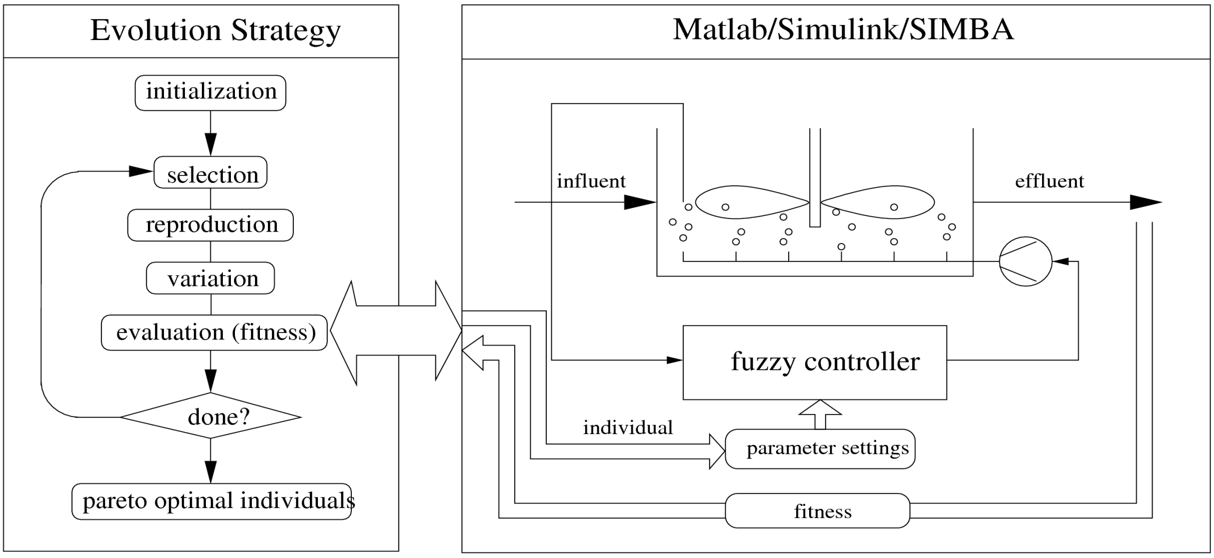


Figure 3
Experimental setup. An evolution strategy is used to find optimal parameters for a fuzzy controller.

Table 2:
Fitness components measured from the outgoing water.

	Component	Units	Constraint
y_1	NH_4-N	mg/l	< 6 mg/l
y_2	NO_3-N	mg/l	< 10 mg/l
y_3	PO_4-N	mg/l	< 3 mg/l

Four different sewage treatment plants were simulated in Matlab using Simulink and SIMBA. Simulink is a Matlab environment for simulation and design of dynamic systems. It can be used to simulate time-varying systems and is used in many different domains such as signal processing, communications or video processing. It is required by SIMBA, a software specifically developed for the dynamic simulation of wastewater systems. The SIMBA software, available from the company ifak system (www.ifak-system.com), can be used to compute the concentration of various compounds in the effluent of the plant. Using this software, four different sewage treatment plants were modeled: Bergrheinfeld, Hammelburg, Herbstein and Michelstadt. An extensive description of the simulation environment and the models used is given by (Pfaff 2007). ASM 2d was used to model the waste water treatment plant in SIMBA. The process parameters have been adapted to model the actual waste water treatment plant as closely as possible. All models have been adapted and calibrated.

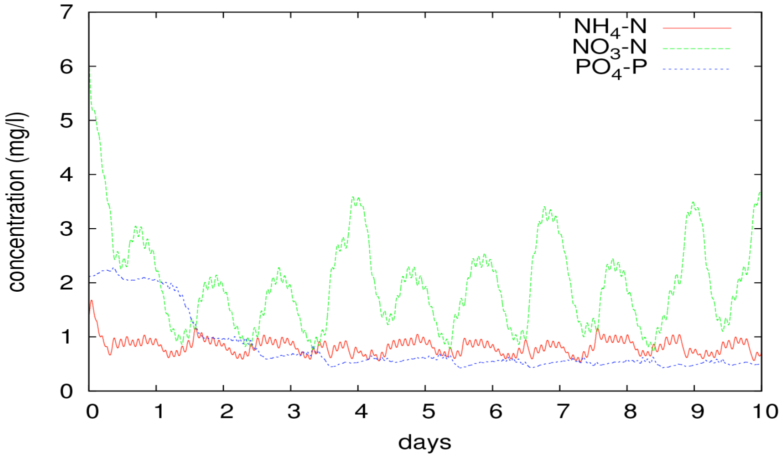


Figure 4:
Concentration of NH_4-N, NO_3-N, and PO_4-N for a sample run of the simulation.

It takes approximately three days for a plant to converge to normal operating mode when started from scratch. Figure 4 shows the output of a sample simulation. After half a day, the NO_3-N concentration cycles between 1 mg/l and approx 3.5 mg/l. After 3 days, the PO_4-P concentration is reduced to a normal level. Hence, each plant simulation is run for 3 days before computing fitness based on the performance of the plant for the fourth day. On the fourth day, NH_4-N, NO_3-N, and PO_4-N concentrations are read every minute and then averaged. In other words, each fitness component y_i is the average of 1440 measurements.

Running the plant in real time would mean that it would have taken 4 full days in order to evaluate a single individual of the population. This is clearly infeasible. Hence, the simulation was accelerated 100 times. In other words, the virtual plant is not simulated in real time, it is accelerated. With the accelerated simulation 1 minute of computer time corresponds to 100 minutes which have passed in the virtual simulation. Using the accelerated simulation, each individual still takes 0.96 hours of real time to evaluate. This is because the model of the sewage treatment plant consists of several differential equations that have to evaluated over and over again. Unfortunately, it was not possible to accelerate the simulation beyond the factor of 100 as this reduced the accuracy of the simulation. The Aqualogic controller was never meant to operate in an accelerated simulation environment. The results is that almost an hour of computation time is required to evaluate a single individual. Understandably, it is only possible to work with very small population sizes.

In order to evaluate a given controller setting, the Aqualogic controller has to be informed of the new parameter settings. Then the controller has to be started and we have to wait for a given period (as described above) after which the concentrations of NH_4-N, NO_3-N, and PO_4-N is measured for another period. The concentrations are averaged for this period and the controller is stopped. These averaged concentrations of NH_4-N, NO_3-N, and PO_4-N in the effluent which are obtained from the simulation are basically noisy measurements because the simulation and the Aqualogic fuzzy controller could not be perfectly synchronized. The controller and the simulation run in parallel processes and communication is limited to the controller specific interface. Synchronization is a well known programming problem. Synchronizing the two perfectly would have meant rewriting the entire Aqualogic controller which is outside the scope of this project. Consequently one has to deal with noisy measurements, restricting the acceleration to a reasonable factor of 100.

An option to get rid of the noise would be to resample individuals (Pietro et al. 2004, Beyer 2000). However, evaluating each individual k times would increase the time required for the entire simulation by a factor of k. This is not feasible here, as the time required to evaluate a single individual is already one hour. As a solution, each individual was equipped with an age, i.e. the number of generations for which this individual has remained in the population. Individuals who have exceeded the age of five are re-evaluated. The fitness of this individual then becomes the mean of all fitness values which were obtained for this individual leading to a very reliable fitness measure for older individuals.

The constraints for the maximum allowed concentrations of the three molecules NH_4-N, NO_3-N, and PO_4-N are enforced by adding a high penalty value [100,100,100] to all fitness components if one of the components exceed the maximum allowed value. For clarity, such individuals are removed from all plots showing the fitness components of the individuals. Since the goal is to obtain all pareto-optimal parameter settings, multi-objective optimization was used.

C. The Strength Pareto Evolutionary Strategy

The strength pareto evolutionary algorithm (SPEA) and its successor SPEA2 (Zitzler et al. 2001). have been shown to perform very well on a suite of multi-objective test functions (Zitzler and Thiele 1998, 1999). In its original formulation the SPEA as well as the SPEA2 algorithm is a genetic algorithm working with a bit string representation. For the problem domain presented here, a representation of real valued vectors is more natural. Therefore, the main ingredients of the SPEA2 algorithm were integrated into an evolution strategy, called strength pareto evolution strategy (SPES).

A number of researchers have used evolution strategies (Rechenberg 1994, Schwefel 1995, Beyer 1995) for multi-objective optimization, e.g. (Kursawe 1991, 1992, Binh and Korn 1996, Costa and Oliveira 2002). Knowles and Corne have developed the pareto archived evolution strategy (Knowles and Corne 1999, 2000). However, this evolutionary algorithm is actually a genetic algorithm. It is not an evolution strategy which uses normally distributed mutations.

Elitism appears to be very important for multi-objective optimization as noted by Zitzler et al. (2000). This was also confirmed by Costa and Oliveira (2002). Costa and Oliveira developed an evolution strategy based upon the non-dominated sorting technique developed by Srinivas and Deb (1995) see also Deb et al. (2000). Given the success of the SPEA2 algorithm, the elitist selection technique of SPEA2 was adapted for use with an evolution strategy. Due to the enormous computational requirements of simulating the plants, it is only possible to work with very small population sizes. Given the difficulty of the problem, a single step size for all of the parameters was used. Individual step sizes for all parameters would have made the problem more difficult because twice as many parameters would have to be optimized. This was confirmed on the ZDF benchmark problems. It is currently unknown whether a multi-objective variant of the covariance matrix adaptation evolution strategy would provide an advantage for this problem domain (Igel et al. 2007).

The Strength Pareto Evolution Strategy (SPES) works as follows. Let $P(0)$ be the initial population. All pareto optimal individuals which have been found so far are stored in an archive $A(t)$ for each generation t. The next generation of individuals $P(t+1)$ is generated from the current population $P(t)$ by

- creating λ individuals by uniform crossover from the individuals found in $P(t)$ which gives us a population of offspring $O(t)$ (for each crossover ρ parents are selected),
- mutating the offspring $O(t)$ using the normally distributed mutation with automatic step size adaptation,
- evaluating all offspring $O(t)$,
- computing the union $U(t)=O(t) \cup A(t-1)$,
- computing the SPEA2 ranking for $U(t)$ (the SPEA2 ranking method is fully described in Zitzler et al. (2001)),
- copying non-dominated individuals from $U(t)$ to $A(t)$,

- if the number of individuals inside the archive A(t) exceeds the archive size S_A, the size of the archive is reduced using the truncation technique of SPEA2,
- if the number of individuals inside the archive A(t) is below the desired size, A(t) is filled with the best individuals (as determined according to the SPEA2 rank) from U(t),
- create P(t+1) by selecting μ parents from A(t) using the SPEA2 ranking and binary tournament selection with replacement.

The algorithm is halted after a fixed number of generations. The pareto optimal individuals found so far, are available from the archive.

The parameters used by this algorithm together with the corresponding values for the experiments are shown in Table 3. The size of the archive S_A was set to 20 which results in a reasonable approximation of the true pareto front. In other words the algorithm is basically a (4/2,20) evolution strategy with an archive as used by the SPEA2 algorithm.

Table 3:
Parameters used for the experiments.

Parameter	Description	Value
μ	total number of parents	4
ρ	number of parents selected for each crossover	2
λ	number of offspring	20
δ_{start}	initial step size	0.1
τ	step size adaptation factor	ln(1.3)
S_A	size of the archive	20

The above algorithm also uses crossover. Depending on the type of problem, crossover may or may not be useful (Fogel and Atmas 1990, Fogel and Stayton 1994). Crossover is particularly useful for a given problem, if the search parameters are independent. This is the case for the majority of the ZDT benchmark problems (Zitzler et al. 2000). In preliminary experiments it was confirmed that the Strength Pareto Evolution Strategy performs better on the ZDT benchmark problems, if crossover is used. It is likely, that for the optimization problem presented here, different good parameter settings for the sewage treatment plant can be recombined to form an even better parameter setting. In other words, it is assumed that independent subspaces can be recombined. Costa and Oliveira (2002) investigated, whether a discrete or intermediate crossover operator results in better performance on the ZDT problems. They have shown that a discrete recombination on decision variables and a discrete recombination on step sizes performed best. That's why the same technique was used here.

Note that the parameters x_i are defined over different ranges as shown in Table 1. For this reason, a different constant scaling factor s_i was used when mutating parameter x_i, normalizing the mutation with respect to the range of the parameter. The scaling factor s_i is given as $s_i = max_i - min_i$ where max_i and min_i denote the minimum and maximum parameter value for parameter x_i as shown in Table 1. The mutation operation is defined as follows. First an updated step size δ for the mutation is computed

$$\delta := \delta \cdot \exp(N(0,\tau))$$

where N(m,σ) is a normally distributed random number with mean m and standard deviation σ. Using the new step size δ and the scaling factor s_i the mutated parameters x_i are computed.

$$x_i := x_i + s_i\,\delta \cdot \exp(N(0,1/\mathrm{sqrt}(n)))$$

The step size adaptation factor τ is set such that the step size adaptation is distributed around a mean of 0 by approximately 30%. Parameters outside of the range $[max_i,min_i]$ are not allowed for our problem domain. If a parameter x_i leaves this range after a mutation, one could set this parameter to the corresponding boundary value. However, this would result in a search bias towards the boundaries. One could also keep the old parameters if one of the parameters becomes invalid or one could retry a different mutation on these parameters. However, this would create a bias away from the boundaries. To avoid this, the mutation is retried in the same direction using a different normally distributed random number. This approach seems to be a reasonable approach to deal with invalid parameters (Huband et al. 2003).

Table 4:
Data for the sewage treatment plants used in the experiments.

	Units	Bergrheinfeld	Hammelburg	Herbstein	Michelstadt
Plant built for	persons	17.000	28.000	3.500	37.600
Amount of waste from	persons	12.000	22.000	3.500	34.000
Direct sludge feed	yes/no	yes	no	yes	no
Depth of aeration system	m	3	4.5	3,2	4
Maximum oxygen volume	m^3/d	76.320	54.000	46.000	144.000
Return activation sludge volume	m^3/d	3.496	5.500	1.250	5.424
Excess sludge removal	m^3/d	86	108	14	240

Results and Discussion

Preliminary results on two sewage treatment plants were reported by Stalph et al. (2008). Here, the above method is tested on a much larger scale. The SPES (4/2,20) algorithm was used on four different sewage treatment plants in simulation in order to obtain parameter settings which are optimal for this particular plant. Table 4 shows the key data for each plant. The parameters of SPES were set as follows: $\delta_{start}=0.1$, $\tau=\ln(1.3)$ and $S_A=20$. Figure 5 shows the results for all four plants. Only non-dominated individuals are shown. For the plant Bergrheinfeld 30 generations were run. This experiment lasted for over 4 weeks. For the three other plants (Hammelburg, Herbstein and Michelstadt), the experiment was run for 15 generations each. Each of these runs took approximately 12 days to complete. Note that the time required for the experiments does not scale linearly due to repeated sampling. The top view of the non-dominated individuals which were found in those runs shows that PO_4-P appears to be relatively independent of the other two fitness components NH_4-N and NO_3-N. The dependency between NH_4-N and NO_3-N is clearly visible in plots shown in Figure 5(b), (d), (f), and (h). This is also clear from the fact that long aeration times decrease the concentration of ammonia. Because ammonia is converted to nitrate through biological oxidation. Figure 6 shows a box and whisker plot for the parameters of the non-dominated individuals found for all four plants. The values are scaled to the minimum and maximum values for each parameter. Each box contains 50% of the values. The median of each parameter is also shown inside each box. The box and whisker plots show that for each plant several of the parameters of the non-dominated individuals are highly clustered within a small parameter range.

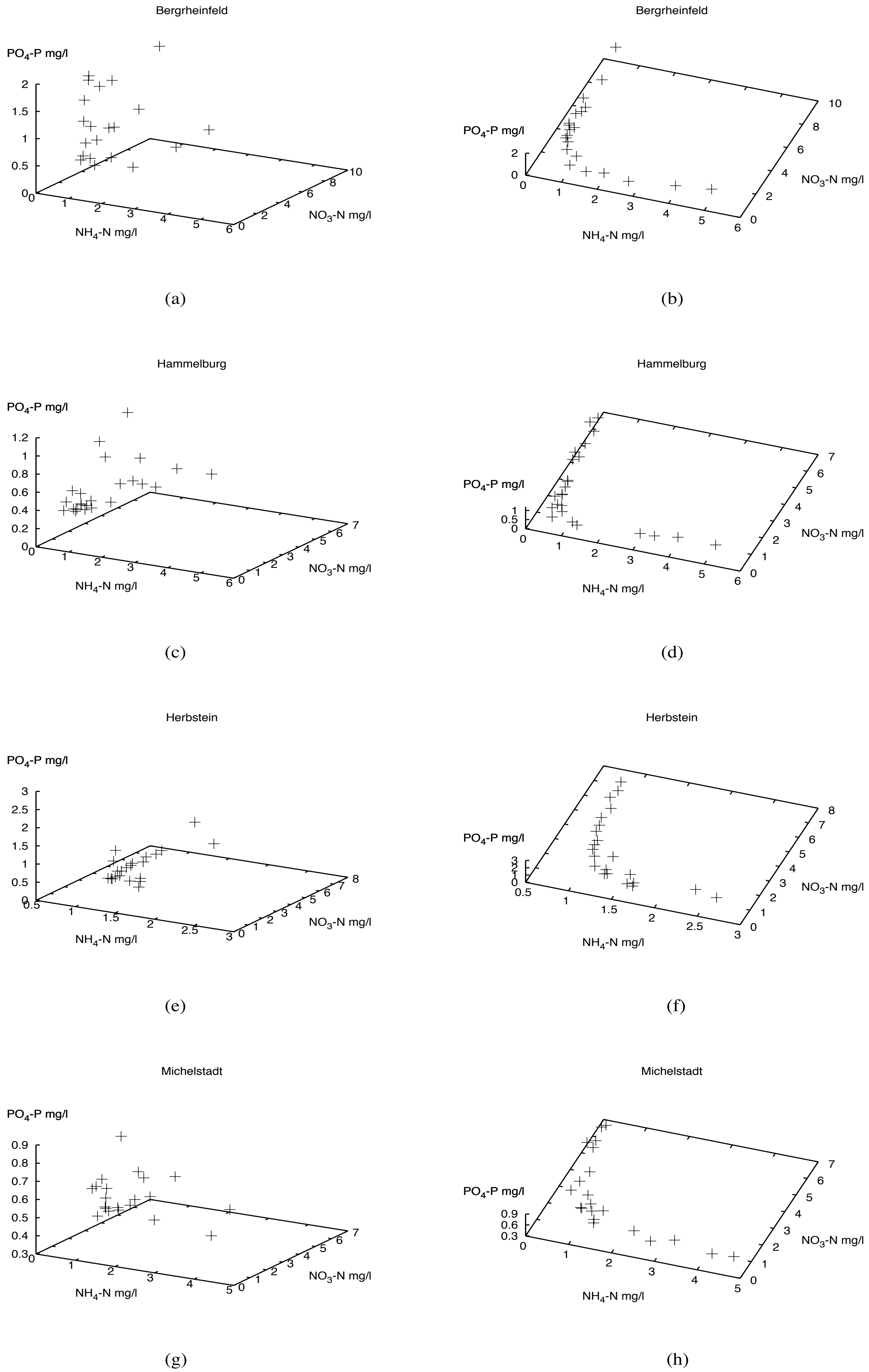
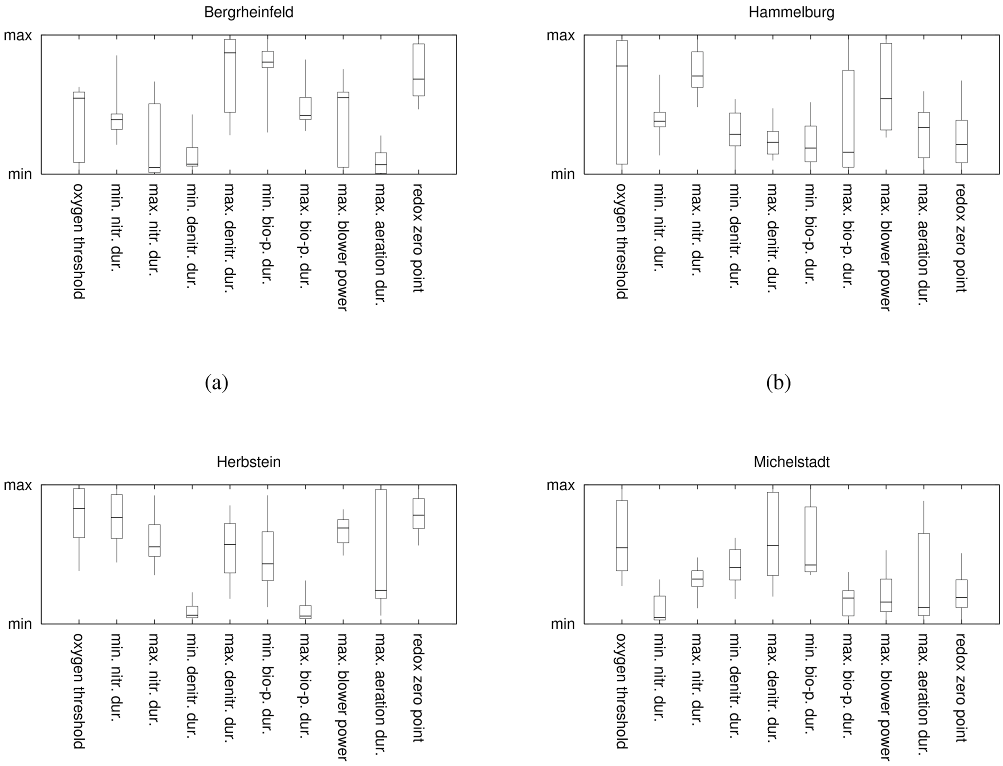


Figure 5: Fitness of non-dominated individuals for four different plants. (a) and (b): Bergrheinfeld. (c) and (d): Hammelburg. (e) and (f): Herbstein. (g) and (h): Michelstadt.



(a) (b) (c) (d)

Figure 6:
Box and whisker plots for the parameters of the non-dominated individuals found for plants (a) Bergrheinfeld (b) Hammelburg (c) Herbstein and (d) Michelstadt. Note that the scale for each parameter is given in Table 1.

Table 5:
Parameters used for testing.

	Parameter Name	Units	2006/ 2007	Evolved value	Value used for testing
x_1	oxygen threshold	mg/l	1.9	0.591	0.9
x_2	minimum nitrification duration	min	60	54.472	54
x_3	maximum nitrification duration	min	140	238.477	238
x_4	minimum denitrification duration	min	60	16.613	17
x_5	maximum denitrification duration	min	150	122.370	90
x_6	minimum bio. phosphate-elimination duration	min	10	9.642	10
x_7	maximum bio. phosphate-elimination duration	min	15	17.611	18
x_8	maximum blower power	%	50	93.116	93
x_9	maximum aeration duration	min	150	98.414	98
x_{10}	redox zero point	mV/s^2	-0.07	-0.042	-0.04

The SPES clearly is able to automate the search for optimal parameter setting. The solutions found by SPES during simulation contains the parameter settings which are set for the particular plant by a human specialist. Table 5 shows the evolved parameter set. This parameter set was tested on the real plant Bergrheinfeld. For testing, the oxygen threshold was increased by 0.3 mg/l because the measured data from the sensor was also offset by this amount. The maximum denitrification duration was reduced to 90 min because of safety reasons. The SPES algorithm found a highly interesting point in the search space: a low oxygen threshold in combination with a long aerobic (nitrification) phase and a short anaerobic phase. Such an operating mode could be called simultaneous intermittent denitrification. A particular advantage of having a low oxygen threshold is that less power is required to operate the oxygen blower. Hence, the entire plant is able to run with a considerable energy saving. In the above experiments, the concentrations of ammonium, nitrate and phosphate were very low for this point. When tried on the actual plant, unfortunately, the sedimentation behavior of the sludge was reduced.

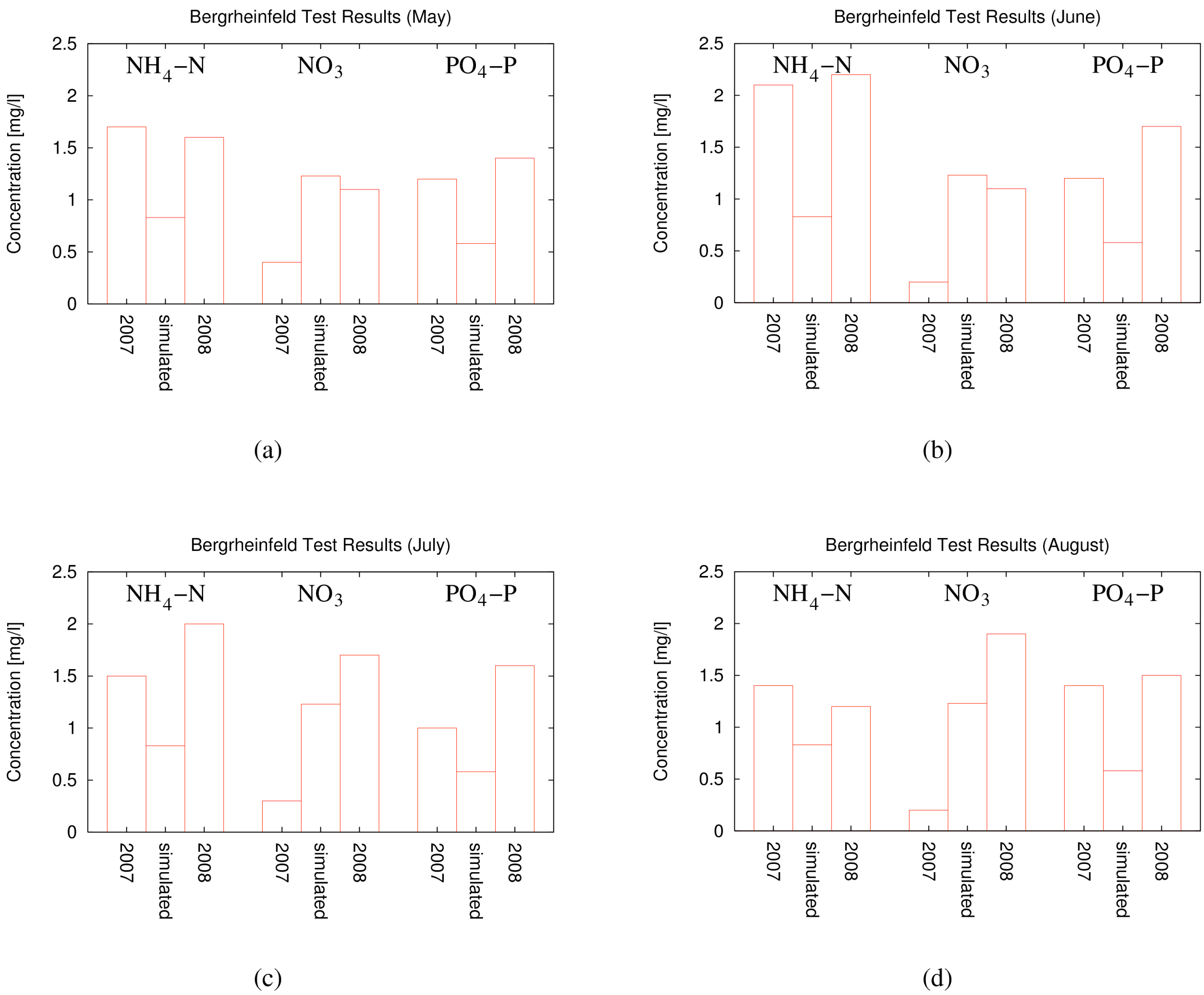


Figure 7:
Test results for the evolved controller on the actual sewage treatment plant Bergrheinfeld for the months (a) May, (b) June, (c) July and (d) August.

Figure 7 shows how the controller with these evolved parameters performed when tested on the actual sewage treatment plant Bergrheinfeld. In the effluent, NH_4-N, NO_3-N as well as PO_4-P was measured. For comparison, the data from 2007 which uses a manual parameter setting is also shown. In 2007 a parameter setting which was manually tuned by a human expert was used. In 2008, the evolved parameter settings were used. Note that the evolved parameter setting was implemented in a slightly modified form. Table 5 lists the parameter values which were actually used. The tested parameter values differ from the evolved values because of safety concerns. The high values of PO_4-P in 2008 are probably a result of a smaller amount of flocculants being used to remove phosphorus. In short, the evolved parameter settings are comparable to those set by a human expert.

Table 6:
Degradation performance of the chemical oxygen demand (COD), NH_4-N, N and P of the plant Bergrheinfeld increases in 2008 compared to the years 2006 and 2007 during the test period.

		Average 2006	Average 2007	Average 2008
COD influent	kg/month	52377	53330	38782
COD effluent	kg/month	2084	2068	1491
COD degradation performance	%	96.02	96.12	96.16
NH_4-N influent	kg/month	3846	3470	2709
NH_4-N effluent	kg/month	158	157	77
NH_4-N degradation performance	%	95.89	95.48	97.16
N influent	kg/month	5186	5298	3702
N effluent	kg/month	445	488	216
N degradation performance	%	91.42	90.79	94.17
P influent	kg/month	837	862	597
P effluent	kg/month	119	124	78
P degradation performance	%	85.78	85.61	86.93
dry matter	mg/l	6.59	7.25	6.69
sludge volume index	mg/l	70.86	78.75	73.29

Table 6 shows how the degradation performance improved when the fuzzy controller was used with the evolved parameters (see also Michel (2009)). Compared to 2006 the average COD, NH_4-N, N, and P degradation performance improved by 0.14, 1.27, 2.75, 1.15 percentage points respectively. Compared to 2007 the average COD, NH_4-N, N, and P degradation performance improved by 0.04, 1.68, 3.38, 1.32. percentage points respectively. The dry matter of the sludge and the sludge volume index remained at a value which was in between the values of 2006 and 2007.

The SPES approach described here, is able to find the optimal parameters for a given plant. Thus, eliminating the need for a human specialist to set those parameters through a time consuming process. The expert may choose one of the solutions which are still within the allowed range for the plant having the lowest energy consumption.

Conclusions

Sewage treatment plants have to be redesigned or extended in order to meet new environmental requirements. Installing sophisticated control mechanisms, which are either able to improve the quality of outgoing water or are able to reduce the energy consumption of the plant, is usually a good alternative to a complete redesign of the plant. Fuzzy controllers are often used for this task. However, these controllers have to be tuned to the given plant. Seasonal changes may require additional changes to the controller settings.

A multi-objective evolutionary algorithm was used to automatically find suitable parameters for the Aqualogic controller that is used in many different sewage treatment plants in Germany. The goal was to maximize the quality of the outgoing water, i.e. to reduce the concentrations of ammonium, nitrate, and phosphate. Since evaluating a single set of parameters would take several days in real time on the actual plant, Simulink and SIMBA were used to simulate the behavior of four different sewage treatment plants. Evolving optimal parameters in simulation is computationally very expensive due to the complexity of the problem. It was therefore only possible to work with very small population sizes. A multi-objective evolution strategy was developed which can find parameter settings that are comparable to those found by a human expert.

References

Beyer, H.-G. (1995). Towards a theory of evolution strategies: The (μ,λ)-theory. Evolutionary Computation, 2(4):381-407.

Beyer, H.-G. (2000). Evolutionary algorithms in noisy environments: theoretical issues and guidelines for practice. Computer Methods in Applied Mechanics and Engineering, 186(2-4):239-267.

Binh, T. T. and Korn, U. (1996). An evolution strategy for the multiobjective optimization. The Second International Conference on Genetic Algorithms, Brno, Czech Republic, pages 23-28.

Costa, L. and Oliveira, P. (2002). An evolution strategy for multiobjective optimization. In Congress on Evolutionary Computation (CEC 2002), Volume 1, pages 97-102, Piscataway, New Jersey. IEEE.

Craig, J. J. (1989). Introduction to Robotics: Mechanics and Control}. Addison-Wesley Publishing Company, Reading, Massachusetts, second edition.

Darwin, C. (1859) The Origin of Species. Oxford University Press, Oxford, England.

Deb, K. (2001). Multi-Objective Optimization using Evolutionary Algorithms. John Wiley & Sons, Chichester, England.

Deb, K., Agrawal, S., Pratap, A., and Meyarivan, T. (2000). A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In Proceedings of the Parallel Problem Solving from Nature VI Conference, pages 849-858. Springer-Verlag.

Eiben, A. E. and Smith, J. E. (2007). Introduction to Evolutionary Computing. Springer-Verlag, Berlin.

Fogel, D. B. and Atmar, J. W. (1990). Comparing genetic operators with gaussian mutations in simulated evolutionary processes using linear systems. Biological Cybernetics, 63:111-114.

Fogel, D. B. and Stayton, L. C. (1994). On the effectiveness of crossover in simulated evolutionary optimization. BioSystems, 32:171-182.

Huband, S., Hingston, P., While, L., and Barone, L. (2003). An evolution strategy with probabilistic mutation for multi-objective optimization. In Proceedings of the 2003 Congress on Evolutionary Computation, volume 3, pages 2284-2291, Canberra, Australia. IEEE Press.

Igel, C., Hansen, N., and Roth, S. (2007). Covariance matrix adaptation for multi-objective optimization. Evolutionary Computation, 15(1):1-28.

Jordy, M., Karner, A., Wimmer, M., and Benz, R. (1998). Fuzzy Logic zur Regelung intermittierend betriebener Kläranlagen. Wasser Abwasser Praxis, 5:37-42.

Karner, A. and Benz, R. (2001). An intelligent control system for wastewater treatment plants. Biotechnology in Bavaria, pages 40-44.

Karner, A., Birlinger, M., Benz, R., and Jordy, M. (2003). Leistungssteigerung auf Großkläranlagen durch intermittierende Betriebsweise. KA - Abwasser, Abfall, 50(7):901--907.

Knowles, J. and Corne, D. (1999). The pareto archived evolution strategy: A new baseline algorithm for pareto multiobjective optimisation. In Proceedings of the 1999 Congress on Evolutionary Computation, pages 98-1105, Mayflower Hotel, Washington D.C., USA. IEEE Press.

Knowles, J. D. and Corne, D. W. (2000). Approximating the nondominated front using the pareto archived evolution strategy. Evolutionary Computation, 8(2):149-172.

Kursawe, F. (1991a). Evolution strategies for vector optimization. In Preliminary Proceedings of the Tenth International Conference on Multiple Criteria Decision Making, Taipei, China, pages 187-193. National Chiao Tung University.

Kursawe, F. (1991b). A variant of evolution strategies for vector optimization. In Parallel Problem Solving from Nature}, pages 193-197, Berlin. Springer-Verlag.

Manesis, S. A., Sapidis, D. J., and King, R. E. (1998). Intelligent control of wastewater treatment plants. Artificial Intelligence in Engineering, 12:275-281.

Michel, M. (2002). Untersuchung zum Betrieb des Fuzzy-Control-Reglers AQUALOGIC in der Kläranlage ``Schwarzacher Becken''. Zulassungsarbeit, Lehrstuhl für Biotechnologie, Theodor-Boveri-Institut für Biowissenschaften, Universität Würzburg.

Michel, M. (2009). Einsatz von evolutionären Algorithmen zur Einstellung von Fuzzy-Logic Reglersystemen. Dissertation, Fakultät für Biologie, Bayerische Julius-Maximilians-Universität Würzburg (submitted).

Pfaff, B. M. (2007). Optimierung von Belebtschlammanlagen. Dissertation, Fakultät für Biologie, Bayerische Julius-Maximilians-Universität Würzburg.

Pietro, A. D., While, L., and Barone, L. (2004). Applying evolutionary algorithms to problems with noisy, time-consuming fitness functions. In Proceedings of the 2004 Congress on Evolutionary Computation, volume 2, pages 1254-1261. IEEE.

Rechenberg, I. (1994). Evolutionsstrategie '94. frommann-holzboog, Stuttgart.

Schwefel, H.-P. (1995). Evolution and Optimum Seeking. John Wiley & Sons, New York.

Srinivas, N. and Deb, K. (1995). Multiobjective optimization using nondominated sorting in genetic algorithms. Evolutionary Computation, 2(3):221-248.

Stalph, P., Ebner, M., Michel, M., Pfaff, B., and Benz, R. (2008). Multiobjective evolution of a fuzzy controller in a sewage treatment plant. In Proceedings of the Genetic and Evolutionary Computation Conference, 2008, Atlanta, GA, pages 535-536, New York. ACM.

Zitzler, E., Deb, K., and Thiele, L. (2000). Comparison of multiobjective evolutionary algorithms: Empirical results. Evolutionary Computation, 8(2):173-195.

Zitzler, E., Laumanns, M., and Thiele, L. (2001). SPEA2: Improving the Strength Pareto Evolutionary Algorithm. Technical Report 103, ETH Zürich, Department of Electrical Engineering, Gloriastrasse 35, CH-8092 Zürich, Switzerland.

Zitzler, E. and Thiele, L. (1998). SPEA2: Improving the Strength Pareto Evolutionary Algorithm. Technical Report 43, ETH Zürich, Gloriastrasse 35, CH-8092 Zürich, Switzerland.

Zitzler, E. and Thiele, L. (1999). Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach.
IEEE Transactions on Evolutionary Computation, 3(4):257-271.