Examination of Stopping Criteria for Differential Evolution based on a Power Allocation Problem

K. Zielinski*, P. Weitkemper†, R. Laur*, K.-D. Kammeyer†
*Institute for Electromagnetic Theory and Microelectronics (ITEM), University of Bremen, Germany
†Department of Communications Engineering, University of Bremen, Germany

***Abstract*— Usually the primary goal for the application of optimization algorithms is convergence to the global optimum, and the secondary goal is to use the least computational effort. By application of different stopping criteria the achievement of both objectives is influenced: If an optimization run is terminated too early, convergence may not be reached, but on the other hand computational resources may be wasted if the optimization run is stopped late. Because the two criteria that are applied mostly in evolutionary algorithms literature have some drawbacks, several stopping criteria are analyzed in this work, using the Differential Evolution algorithm. In contrast to a prior study a constrained optimization problem is used here. It consists of optimizing the power allocation for a CDMA (Code Division Multiple Access) system that includes a parallel interference cancellation technique.**

I. Introduction

Due to the growing complexity of technical systems optimization algorithms are integrated into design processes as for example in the optimization system MODOS for microelectronic circuits and micro systems [1]. The need for reliable stopping criteria arises because the optimization algorithms must be capable of terminating automatically without wasting computational resources but also without stopping prematurely considering the results accuracy. This is especially important if a lot of scenarios should be optimized that differ in the used parameters. An example is the here used problem that consists of the optimization of power allocation. It is interesting to have several results in dependance on the target bit error rate to meet the different quality requirements that result from diverse applications like voice or multimedia services. Furthermore, if a Monte Carlo simulation is done that e.g. regards different distributions of user locations, many optimization runs need to be done, so wasting of computational resources has to be prevented.

Stopping criteria can be implemented in several ways, but in literature mostly only two criteria are used. For problems with known optimum an error measure in dependance on the optimum is often applied. As the optimum is generally not known for real-world problems, this criterion is mainly applied for test functions. For real-world problems the stopping criterion consists mostly of a limitation of the number of objective function evaluations. The drawback of this method is that the number of function evaluations that is needed for convergence is highly dependent on the optimization problem. Usually trial-and-error methods are applied to determine a suitable setting. Another disadvantage is that the needed number of function evaluations is subject to fluctuations because evolutionary algorithms like Differential Evolution (DE) include randomness. Therefore, a safety margin for the number of function evaluations has to be added to ensure convergence. Especially for problems with computationally expensive objective functions this approach is unfavorable.

If the state of the optimization run is taken into account for the decision when to terminate, the problem with the fluctuations in the number of needed function evaluations can be avoided. In this work several stopping criteria are studied which possess this characteristic. It is reported which of the criteria are suitable for DE based on the examined optimization problem.

This paper is organized as follows: In Section II the Differential Evolution algorithm is introduced. The basic algorithm is described as well as the applied constraint-handling method and the approaching of boundaries. In Section III the stopping criteria are presented and Section IV gives background information about the optimization problem. In Section V the experimental settings are described, in Section VI results are presented and Section VII closes with conclusions.

II. Differential Evolution

Differential Evolution was developed in 1995 as a population-based stochastic evolutionary optimization algorithm [2]. The execution of the algorithm is similar to other evolutionary algorithms like Genetic Algorithms (GAs) or Evolution Strategies (ESs): The first generation is initialized randomly and further generations evolve by the application of certain evolutionary operators until a stopping criterion is satisfied. The evolutionary algorithms differ mainly in the representation of parameters (usually binary strings are used for GAs while parameters are real-valued for ESs and DE) and in the evolutionary operators. A characteristic of DE is that the evolutionary operators are dependent on members of the current generation, leading to an adaptive scaling of step sizes.

The size of the population is specified by the parameter NP that has to be set by the user. Usually it is kept fixed during an optimization run. The population members (also called individuals) are real-valued vectors with dimension D that equals the number of objective function parameters. The evolutionary operators mutation, recombination and selection are applied to every population member $\vec{x}_i$ with $i \in [0, NP\text{–}1]$ to generate a new generation. First, a mutated vector is built by adding the weighted difference

of two randomly chosen population members to a third randomly chosen individual:

$$\vec{v}_i = \vec{x}_{r_1} + F \cdot (\vec{x}_{r_2} - \vec{x}_{r_3}) \tag{1}$$

The amplification constant F is a control parameter of DE that has to be set by the user. The indices r_1, r_2, r_3 denote three mutually different individuals that are also different from the currently regarded individual $\vec{x}_i$. In Fig. 1 mutation is shown for the dimension $D = 2$.

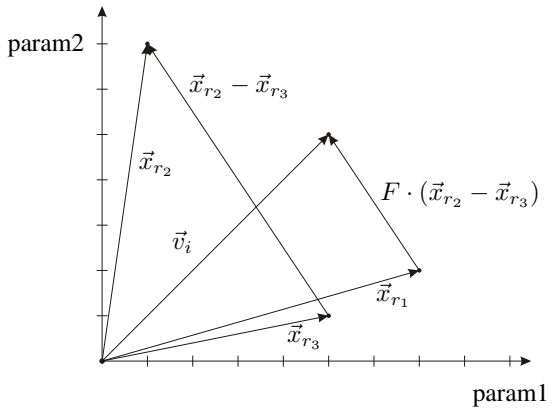

Fig. 1. Mutation for $D = 2$

The second evolutionary operator that is applied is recombination. The trial vector $\vec{u}_i$ is built by determining for every vector component $j \in [0, D-1]$ if the corresponding component should be copied from the target vector $\vec{x}_i$ or the mutated vector $\vec{v}_i$. The decision is based on a random variable $rand_j$ that is compared to the user-defined control parameter CR. However, for every individual it is ensured that at least one component of the trial vector $\vec{u}_i$ is copied from the mutated vector $\vec{v}_i$ by a random choice of a number $k \in [0, D-1]$.

$$u_{i,j} = \begin{cases} v_{i,j} & \text{if } rand_j \leq CR \text{ or } j = k \\ x_{i,j} & \text{otherwise} \end{cases} \tag{2}$$

In Fig. 2 recombination is shown for $D = 2$ dimensions. It can be seen that three potential trial vectors exist for every target vector in this case.

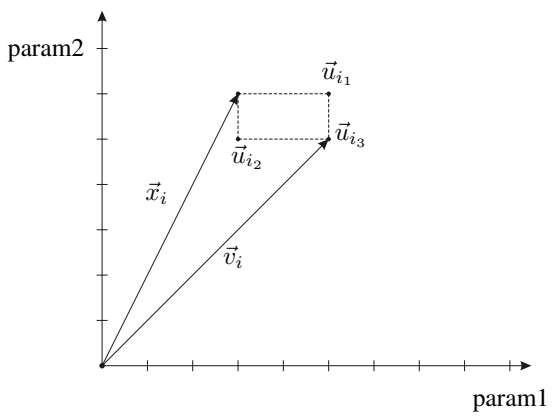

Fig. 2. Recombination for $D = 2$

By the use of the third evolutionary operator selection it is determined for every $i \in [0, NP-1]$ whether the target vector $\vec{x}_i$ or the corresponding trial vector $\vec{u}_i$ should be inserted into the next generation. As basis for the decision the objective function value is considered (for minimization problems a smaller objective function value is preferred). Because this selection scheme does not allow deterioration with regard to the objective function value, the DE selection scheme is called greedy.

Several variants of DE have been developed that differ in the mutation and recombination operators [3]. They are specified using the notation

$$DE/x/y/z \tag{3}$$

where x denotes the mutated vector, y is the number of difference vectors and z gives the crossover scheme [4]. The here used variant can be described as DE/rand/1/bin, meaning that $\vec{x}_{r_1}$ is a randomly chosen population vector, one vector difference $(\vec{x}_{r_2} - \vec{x}_{r_3})$ is used and a binomial crossover scheme is employed. This variant belongs to the first DE schemes that were developed [2] and it is used frequently in literature [5], [6], [7].

A. Handling of Constraints

The applied constraint-handling method is based on modifying the selection procedure. This technique is described in [8] for multi-objective optimization and a similar variant is used in [5] for single-objective DE. When a vector $\vec{a}$ is compared to a vector $\vec{b}$, $\vec{a}$ is considered better if:

- Both vectors are feasible, but $\vec{a}$ yields the smaller objective function value.
- $\vec{a}$ is feasible and $\vec{b}$ is not.
- Both vectors are infeasible, but $\vec{a}$ results in the lower amount of constraint violations.

An advantage over other constraint-handling techniques is that no additional parameters are needed. The only information analyzed are objective function values for feasible individuals and the amount of constraint violations for infeasible individuals. Solutions with smaller constraint violations are preferred, so the search is directed to feasible regions. For unconstrained problems the replacement procedure is the same as for the original DE.

B. Approaching of Boundaries

In the given optimization problem it has to be ensured that some boundary values are not violated. Several possibilities exist for this task :

- The positions that violate boundary constraints are newly generated until the constraints are satisfied.
- The boundary-exceeding values are replaced by random numbers in the feasible region.
- The boundary is approached asymptotically by setting the boundary-offending value to the middle between old position and boundary:

$$u_{j,i,G+1} = \begin{cases} \frac{1}{2} \cdot \left(x_{j,i,G} + x_j^U\right) & \text{if } u_{j,i,G+1} > x_j^U \\ \frac{1}{2} \cdot \left(x_{j,i,G} + x_j^L\right) & \text{if } u_{j,i,G+1} < x_j^L \\ u_{j,i,G+1} & \text{otherwise} \end{cases} \tag{4}$$

where x_j^U is the upper limit and x_j^L is the lower limit.

In this work the latter approach is used because of a recommendation in [4].

III. Stopping Criteria

Ten stopping criteria are examined which were mostly introduced in [9] for unconstrained single-objective optimization. In the following the criteria are described with the adjustments for constrained single-objective optimization problems. The structure of [9] is maintained, so the stopping criteria are sorted into six classes: Reference, exhaustion-based, improvement-based, movement-based, distribution-based and combined criteria.

1) Reference criteria: If the optimum is known, an error measure depending on the difference to the optimum can be employed for detecting convergence. As the optimum is generally not known for real-world problems, these criteria are usually only applicable for test functions. However, criteria can be derived that are adapted to real-world problems (see class 5). In this work only the adapted criteria are used.
2) Exhaustion-based criteria: Due to limited computational resources an optimization run might be terminated after a certain generation, number of objective function evaluations or CPU time. Although these criteria are commonly used in evolutionary algorithms literature, they are not investigated here because of the strong dependence on the objective function. However, a maximum number of generations G_{max} is used in combination with every stopping criterion to prevent the algorithm from running forever if a criterion is not able to stop the run. After preliminary tests $G_{max} = 2000$ is used.
3) Improvement-based criteria: If improvements are small over some time, an optimization run should be stopped. The following variants are examined:
   a) *ImpBest*: Improvement of the best objective function value is below a threshold t for a number of generations g [10]
   b) *ImpAv*: Improvement of the average objective function value is below a threshold t for a number of generations g [11]
   c) *NoAcc*: No acceptance occurred in a specified number of generations g [12] (note that acceptance equals improvement for DE)

   Criteria *ImpBest* and *ImpAv* have been adjusted for constrained optimization in the following way: If an individual is feasible in the current generation but it was infeasible in the previous generation, the improvement is assigned an arbitrary high number. If an individual is infeasible in both the current and the previous generation, the improvement is calculated based on the constraint violation.

   For criterion *NoAcc* no explicit adjustment had to be made because the constraint-handling is included in the selection scheme.
4) Movement-based criteria: The movement of individuals is used as basis for these criteria.
   a) *MovObj*: Movement in the population with respect to the average objective function value (objective space) is below a threshold t for a number of generations g

      For DE this criterion is the same as criterion *ImpAv* because of the greedy selection scheme that prohibits moves that deteriorate the solution. However, for another optimization algorithm (Particle Swarm Optimization, PSO) this criterion is beneficial [9], so it is mentioned here but not included in the examination.
   b) *MovPar*: Movement in the population with respect to positions (parameter space) is below a threshold t for a number of generations g

   For criterion *MovObj* the same adjustment for constrained optimization problems can be done as for *ImpBest* and *ImpAv*.

   Criterion *MovPar* does not need any adjustment because only the changes in positions of individuals are observed, regardless of their feasibility.
5) Distribution-based criteria: For DE usually all individuals converge to the optimum eventually. Therefore, it can be concluded that convergence is reached when the individuals are close to each other. Because it is assumed that the optimum is not known as for the reference criterion, the distances between the population members are examined. The first three criteria are applied in parameter space while the forth is used in objective space.
   a) *MaxDist*: Maximum distance from every vector to the best population vector is below threshold m [10]
   b) *MaxDistQuick*: The best $p\%$ of the population individuals are checked if the maximum distance to the best vector is below threshold m, respectively

      This criterion was newly presented in [9]. It is inspired by the observation of optimization algorithms: During an optimization run a state may occur where most population members have already converged to the optimum but some individuals are still searching. Using the *MaxDist* criterion the algorithm would not terminate until all population members are converged, so computational resources are wasted if the optimum is already found. Using the *MaxDistQuick* criterion the population is rearranged due to the objective function values of the individuals using a quicksort algorithm. Further only the best $p\%$ of the individuals are checked for their maximum distance to the best population member.

      For the functions used in [9] PSO benefits more from this method than DE. The reason is presumably the greedy selection scheme of DE that leads to a faster convergence of the individuals.
   c) *StdDev*: Standard deviation of the vectors is below threshold m

The standard deviation is calculated using

$$s = \sqrt{\frac{\sum_{i=0}^{NP-1} (x_{r,i} - \bar{x})^2}{NP-1}}$$

with the radius of the population members $x_{r,i} = \sqrt{\sum_{j=0}^{D-1} x_{i,j}^2}$ and the average radius $\bar{x} = \frac{1}{NP} \sum_{i=0}^{NP-1} x_{r,i}$. A similar criterion is used in [13].

d) *Diff*: Difference of best and worst objective function value is below threshold d and at least $p\%$ of the individuals are feasible [14]

For criteria *MaxDist* and *StdDev* no adjustment for constraint-handling is needed because only positions are examined, as for criterion *MovPar*.
However, for criterion *MaxDistQuick* it has to be assured that at least $p\%$ of the individuals are feasible because the population is sorted according to the objective function values of the individuals. Another possibility would be to sort the infeasible individuals based on their constraint violation.
Criterion *Diff* included only parameter d for unconstrained optimization problems in [9], but for constrained optimization parameter p has been added to account for the percentage of the population that is demanded to be feasible.

6) Combined criteria: Because objective functions possess different features, reactions to stopping rules may be different. Therefore, it is supposed to be advantageous to use several criteria in combination. Here the following combined criteria are applied:
   a) *ComCrit*: If the average improvement is below threshold t for a number of generations g, it is checked if the maximum distance is below threshold m
   In [9] only one combined criterion is described, but here another one is added. For clarity reasons the name of the already introduced criterion is maintained.
   b) *Diff_MaxDistQuick*: If the difference between best and worst objective function values is below threshold d, it is checked if the maximum distance of the best $p\%$ of the individuals to the best solution is below threshold m (at least $p\%$ of the individuals have to be feasible)
   In [9] it is assumed that a combination of distribution-based criteria in objective and parameter space may be beneficial, so this criterion is added here.

   The combined criteria are composed of several already presented stopping criteria, so the same adjustments for constrained optimization are done for them as for the individual criteria.

All criteria from classes 3 to 6 incorporate some sort of adaptiveness that enables them to evaluate information about the state of an optimization run. In the following it is examined if the respective mechanisms are actually effective for stopping the execution of Differential Evolution at an appropriate time.

IV. The Power Allocation Problem with Interference Cancellation

CDMA is a popular multiple access scheme that is currently used in UMTS (Universal Mobile Telecommunications System). CDMA allows several users to transmit data to one base station at the same time and in the same frequency band. It is possible to allocate the received power equally to all users by the application of a closed-loop power control but the system performance (or bit error rate) with interference cancellation can be improved significantly by assigning different receive powers to the users. The received power of one particular user depends on the transmit power which is limited especially when assuming mobile stations with battery supply. The properties of the channel including free space attenuation also limit the receive power.

For optimization the design variables are the received powers that are allocated to each individual user. For this work U=16 users are assumed. The objective function that is to be minimized consists of the sum of power for all users, resulting in a single-objective optimization problem.

The system performance is usually degraded by multi-user interference. By means of parallel interference cancellation (PIC) the interference can be estimated iteratively. If the applied interference cancellation method converges, the calculated interference can be subtracted from the received signal before detection and the single-user bound (SUB) is reached. In the following the constraint is presented that results from the applied parallel interference cancellation technique.

The convergence behavior of PIC can be visualized by using the multi-user efficiency (MUE), denoted as η [15]. For PIC the multi-user efficiency is given by

$$\eta = \frac{SINR}{SNR} = \frac{2\sigma_d^2/(\sigma_n^2 + \sigma_{\mathrm{MUI}}^2)}{2\sigma_d^2/\sigma_n^2} = \frac{1}{1 + \beta\mu E_s/N_0} \tag{5}$$

where SINR is the signal-to-interference-plus-noise-ratio, SNR is the signal-to-noise-ratio, σ_d^2 is the variance of the desired signal, σ_n^2 is the power of the noise, σ_{MUI}^2 is the variance of the remaining multi-user interference after cancellation, β is the system load, μ is the remaining mean squared error of the estimated symbols after decoding, E_s is the energy per symbol and $N_0/2$ is the power density of the noise (for more detailed information about CDMA see [16]). The multi-user efficiency at the iteration m is dependent on the multi-user efficiency at the previous iteration: $\eta^{(m)} = f\left(\eta^{(m-1)}\right)$. The transfer function f is displayed in Figure 3. Additionally, the so-called trajectory is given which describes the same facts as the transfer function but only considers discrete points of interest [17]. In the plot of the transfer function the point $\eta = 1$ characterizes perfect interference cancellation which equals the SUB. If the function crosses the bisecting line given by $\eta^{(m)} = \eta^{(m-1)}$, the detection will get stuck. Therefore, a constraint for the PIC optimization problem is that for each iteration the following condition

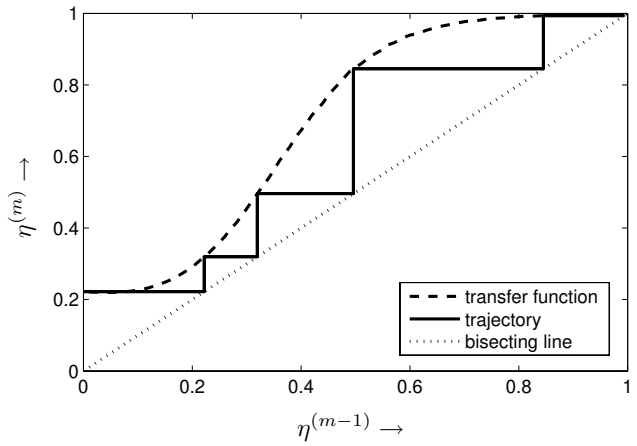

Fig. 3. Transfer function and trajectory for PIC

must hold:

$$f\left(\eta^{(m)}\right) > \eta^{(m)}\ ,\ \eta \in [0,1] \tag{6}$$

Apart from the constraint that results from the interference cancellation method, boundary constraints have to be regarded. As no negative power exists, the design variables have to be non-negative. However, it has been observed that setting the lower limit to zero leads to a trivial special case for which one user gets all the available power. Therefore, the lower boundary value is set to $X_{min} = 0.5$. An upper boundary value of $X_{max} = 4$ is also assigned to restrict the search space.

In preliminary tests it could be seen that permutations of solutions appeared for PIC. Because of the parallel processing the permutations have no influence on the results. Therefore, computational cost is saved by rearranging the users in ascending order.

V. Experimental Settings

The following experiments reflect which stopping criteria are suitable for DE when optimizing the described constrained single-objective optimization problem. Furthermore, it is analyzed which criteria show key problems that prevent their successful application. Additionally, the dependence on the settings of stopping criteria parameters is investigated.

For the control parameters of DE the following settings are used: $NP = 30$, $F = 0.7$ and $CR = 0.9$ [18].

The examined values of the stopping criteria parameters are displayed in Table I. Note that some of the start values are different from the ones used in [9]. For every parameter combination 100 independent optimization runs were carried out. In [18] the optimal value for the objective function was found to be 18.43, but as this exact value was found very infrequently (twice in 10000 runs), convergence was defined as reaching an objective function value of $\sum P_u \leq 18.5$.

There are two situations for which an optimization run is considered unsuccessful:

- If the objective function value of $\sum P_u \leq 18.5$ is not reached when terminating the algorithm.
- If the maximum number of generations is reached without termination of the algorithm, regardless of whether the optimum is found or not.

TABLE I

Examined parameter values of the stopping criteria

Criterion	Parameter	Start value	Stop value	Modifier
ImpBest, *ImpAv*	t	1e-2	1e-6	· 1e-1
	g	5	20	+ 5
NoAcc	g	1	5	+ 1
MovPar	t	1e-2	1e-6	· 1e-1
	g	5	20	+ 5
MaxDist, *StdDev*	m	1	1e-4	· 1e-1
MaxDistQuick	m	1	1e-4	· 1e-1
	p	0.1	1.0	+ 0.1
Diff	d	1e-1	1e-6	· 1e-1
	p	0.1	1.0	+ 0.1
ComCrit	t	1e-2	1e-6	· 1e-1
	g	5	20	+ 5
	m	1	1e-4	· 1e-1
Diff_MaxDistQuick	d	1e-1	1e-6	· 1e-1
	p	0.1	1.0	+ 0.1
	m	1	1e-4	· 1e-1

As performance measure the success performance is used that is defined in [19] in the following way: The success performance is calculated by multiplying the mean number of function evaluations for successful runs with the number of total runs and dividing this by the number of successful runs. A low success performance indicates a good result. A high success performance can occur due to two different reasons: Either the number of successful runs is low or the number of function evaluations for convergence is high.

In this work not the number of objective function evaluations is used for the calculation of the success performance but the number of constraint function evaluations because the objective function is not evaluated for infeasible individuals and the computation of the constraints can be equally computationally expensive.

Some parameter values are omitted in figures of the following section if the success performance could not be calculated because there was not a single successful run. Furthermore, all figures are scaled to a success performance of up to 20000 for comparability reasons.

VI. Results

For criterion *ImpBest* the success performance is displayed in Fig. 4. The same result is yielded as for unconstrained optimization in [9]: There is no parameter combination that results in a convergence rate of 100%. Obviously, improvement of the best objective function value occurs too irregularly, so criterion *ImpBest* provides no reliable stopping rule for DE.

Criterion *ImpAv* yields better results than *ImpBest* as $t \leq 10^{-5}$ results in a convergence rate of 100%. However, the success performance becomes rather high for these settings (see Fig. 5). Furthermore, choosing of parameter values is difficult as there is no obvious connection to the optimization problem. Therefore, criterion *ImpAv* cannot be recommended for use with DE, either.

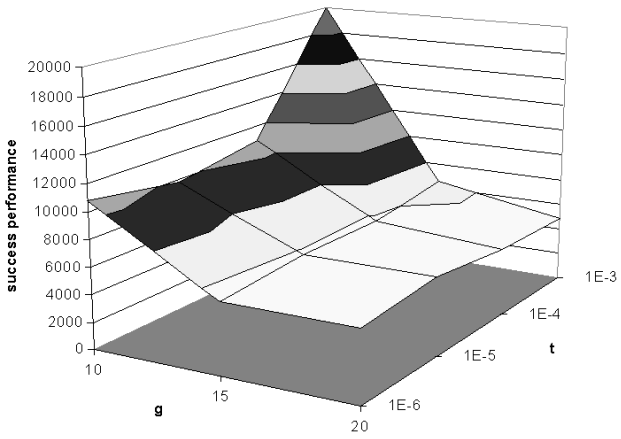

Fig. 4. Success performance for criterion *ImpBest*

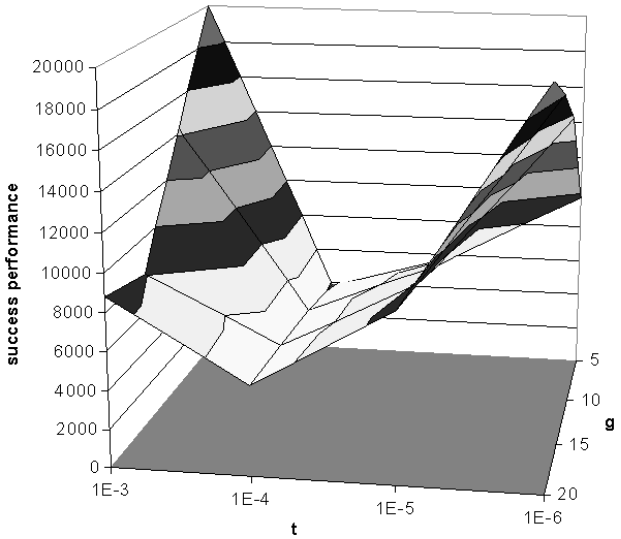

Fig. 5. Success performance for criterion *ImpAv*

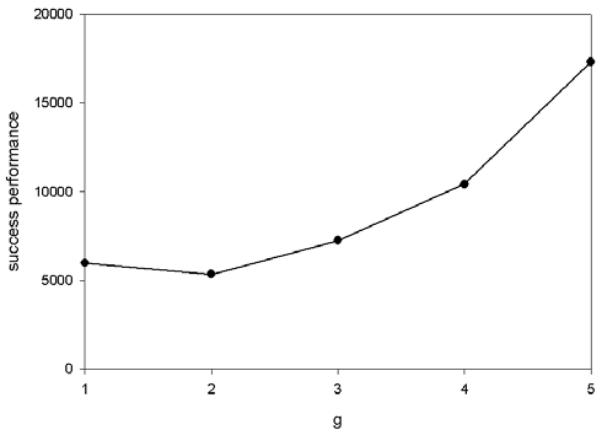

Fig. 6. Success performance for criterion *NoAcc*

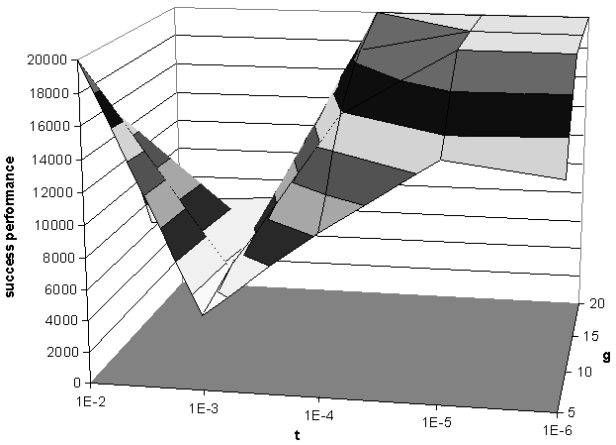

Fig. 7. Success performance for criterion *MovPar*

In Fig. 6 it can be seen that the success performance for criterion *NoAcc* increases quadratically with rising number of generations g, with the exception of $g = 1$ because of a convergence rate of only 57% (the other parameter settings yield convergence rates of approximately 100%). In [12] it is stated that long periods without acceptance of a vector are probably more common for DE than for other evolutionary algorithms, so g should not be set too low. However, for the optimization problem applied here a value of $g \geq 2$ is sufficient to induce reliable detection of convergence. The results of this work indicate that acceptance happens frequently as long as the population still improves, so *NoAcc* provides a suitable stopping criterion for DE. However, note that in [9] *NoAcc* does not show good results for a function with a flat surface.

The success performance for criterion *MovPar* is shown in Fig. 7. As for *ImpAv* and *ImpBest* choosing of suitable parameter settings is difficult because there is no observable connection to the optimization problem, so criterion *MovPar* cannot be recommended for use with DE.

The results for criteria *MaxDist* and *StdDev* are displayed in Fig. 8. In [9] both criteria yielded approximately the same results, but here the success performance of *MaxDist* is inferior to *StdDev*. This result can be partially explained by the relatively small G_{max} that is used here. Because a lot of optimization runs were not terminated before reaching G_{max} for *MaxDist*, the success performance is large. The remaining difference in performance can be explained by the fact that *StdDev* includes an average of positions while *MaxDist* regards every individual for itself. Why this behavior is more noticeable in this work than in [9] cannot be stated definitely, because there are several differences: The shapes of the objective functions may be significantly different, the dimension is considerably higher here ($D = 16$ in contrast to $D = 2$ in [9]) and furthermore the optimization problem applied here incorporates a constraint.

For both *MaxDist* and *StdDev* as well as *MaxDistQuick* (see Fig. 9) the success performance is strongly dependent on the maximum distance m. As the distribution-based criteria in parameter space generally produce good results if suitable parameter settings are used, it is analyzed in the following if there is a connection between parameter settings and the optimization problem.

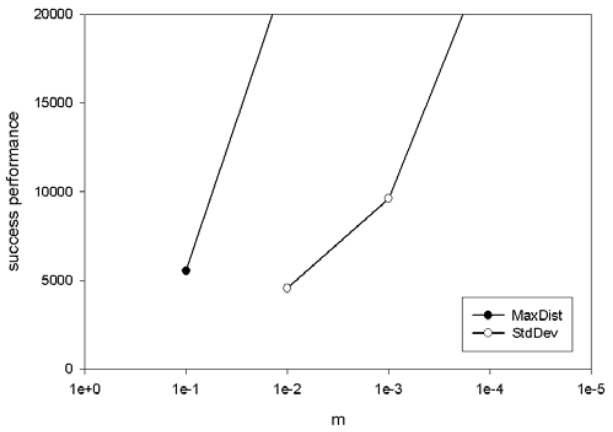

Fig. 8. Success performance for criteria *MaxDist* and *StdDev*

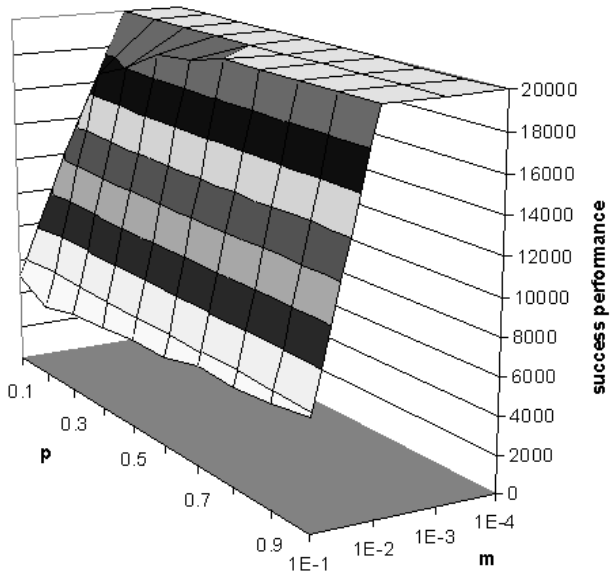

Fig. 9. Success performance for criterion *MaxDistQuick*

For the examined optimization problem best parameter settings are $m = \{10^{-1}, 10^{-2}\}$ for both *MaxDist* and *MaxDistQuick* while $m = 10^{-3}$ provided the best results for *StdDev* ($m = 10^{-2}$ has a lower success performance but a convergence rate of only 66%). In [12] it is recommended to use parameter settings that are several orders of magnitude lower than the desired accuracy if population statistics like *StdDev* are used as stopping criteria. As the best found setting for the examined problem is 18.43, but reaching a value of $\sum P_u \leq 18.5$ was defined as convergence, the demanded accuracy equals 0.07 here. Hence, the best parameter setting for *StdDev* is approximately one order of magnitude lower than the accuracy, and for the other two criteria values in the same order of magnitude as the accuracy resulted in good convergence behavior. Therefore, it can be concluded that these criteria are suitable for DE, but additional research should concentrate on analyzing the connection between the optimization problems and the stopping criteria parameters more precisely. Note that the computational complexity of *MaxDistQuick* is higher than for *MaxDist* [20], so *MaxDist* should be preferred for DE as there are no significant differences in the performance.

The success performance is approximately constant in dependance on p using criterion *Diff* (Fig. 10). In contrast large variations in the success performance are

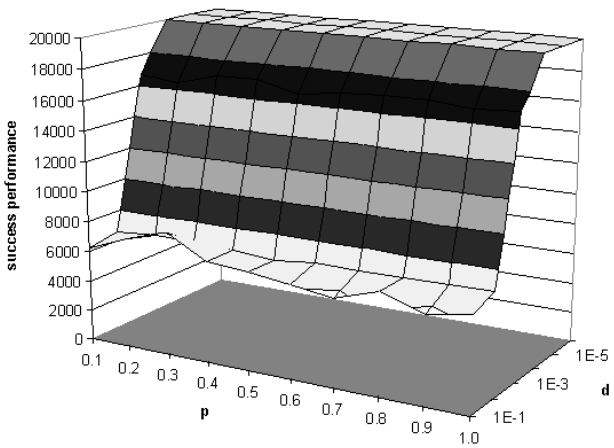

Fig. 10. Success performance for criterion *Diff*

observed when changing d. Reliable convergence behavior is yielded by parameter values of $10^{-2} \geq d \geq 10^{-4}$ for which all optimization runs converged. In [12] it is recommended to set d several orders of magnitudes lower than the allowed tolerance (distance) from the optimum. However, the success performance gets considerably larger with decreasing d (see Fig. 10), so here the best parameter setting $d = 10^{-2}$ is in the same order of magnitude as the desired accuracy. As for the distribution-based criteria in parameter space it can be concluded that criterion *Diff* is suitable for DE, but the relationship between the optimization problems and the parameter of the stopping criterion needs more thorough investigation.

For the two combined criteria the display of results is more difficult as three parameters are regarded, respectively. Reliable convergence behavior is yielded by $m = \{10^{-1}, 10^{-2}\}$ for *ComCrit* (as it was for *MaxDist*). Interestingly, for $m = 10^{-2}$ the success performance does not change with g and t and matches approximately the success performance of *MaxDist* for the same parameter setting. Therefore, only the results for $m = 10^{-1}$ are displayed in Fig. 11. The success performance is higher than for *MaxDist* with the same setting of m, so based on the examined optimization problem *MaxDist* should be preferred.

For criterion *Diff_MaxDistQuick* the success performance is relatively constant for varying p, so results are only given for a medium setting of $p = 0.5$ (see Fig. 12). In contrast to *MaxDist* and *Diff* larger values of m and d also lead to reliable convergence here but choosing of parameter values is less connected with the optimization problem and also more parameters have to be set, so it is easier to use the individual criteria.

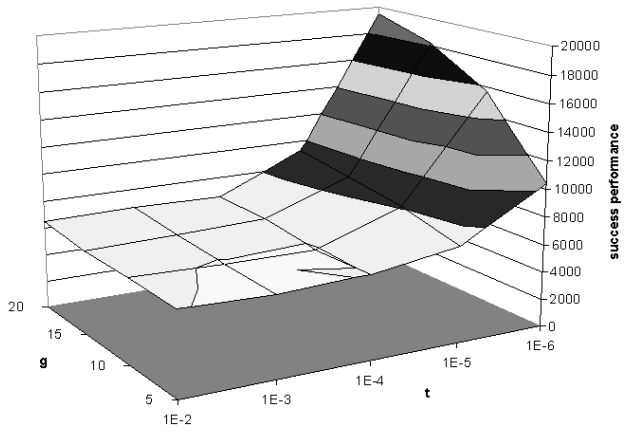

Fig. 11. Success performance for criterion *ComCrit* with $m = 10^{-1}$

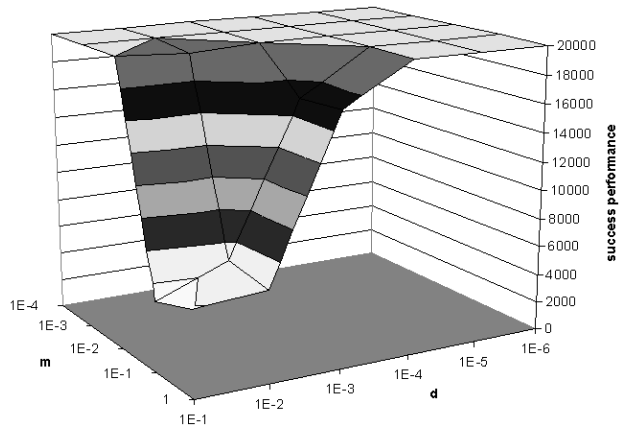

Fig. 12. Success performance for *Diff_MaxDistQuick* with $p = 0.5$

VII. Conclusions

Stopping criteria influence both the convergence probability and the convergence speed of optimization algorithms. In this work a constrained single-objective power allocation problem is used for the examination of stopping criteria that react adaptively to the state of an optimization run. By their application problems are avoided that are caused by fluctuations in the number of function evaluations that are needed for convergence. Therefore, the computational cost for optimization can be reduced. This is especially advantageous if a lot of optimization runs are necessary e.g. for a Monte Carlo simulation or an examination with varying parameters.

The best results are yielded by the following criteria: *NoAcc* (no acceptance occurred in a predefined number of generations), *MaxDist* (the maximum distance of every population member to the best solution is observed), *StdDev* (monitoring the standard deviation of positions) and *Diff* (the difference between best and worst objective function value in a generation is regarded). Especially for the latter three criteria, which consider the distribution of the individuals, the proper choice of parameter values is important but the results of this work and of previous examinations [9] indicate that suitable parameter settings are closely related to the desired accuracy.

It should be noted that results of stopping criteria may vary for different optimization algorithms [9].

For future work stopping criteria should be examined using functions with different characteristics to find out by which properties their performance is influenced.

References

[1] D. Peters and R. Laur, "An Efficient and Reliable Design Strategy for Micro Mechanical Sensors Using Automatic Optimization," in *Proceedings of the 8th International Conference on Optimization of Electric and Electronic Equipment*, 2002.

[2] R. Storn and K. Price, "Differential Evolution - A Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces," International Computer Science Institute, Berkeley, Tech. Rep. TR-95-012, 1995.

[3] G. C. Onwubolu, "Optimizing CNC Drilling Machine Operations: Traveling Salesman Problem-Differential Evolution Approach," in *New Optimization Techniques in Engineering*, G. C. Onwubolu and B. Babu, Eds. Berlin Heidelberg: Springer-Verlag, 2004, pp. 537–566.

[4] K. V. Price, "An Introduction to Differential Evolution," in *New Ideas in Optimization*, D. Corne, M. Dorigo, and F. Glover, Eds. London: McGraw-Hill, 1999, pp. 79–108.

[5] J. Lampinen and R. Storn, "Differential Evolution," in *New Optimization Techniques in Engineering*, G. C. Onwubolu and B. Babu, Eds. Berlin Heidelberg: Springer-Verlag, 2004, pp. 123–166.

[6] T. Robic and B. Filipic, "DEMO: Differential Evolution for Multiobjective Optimization," in *Proceedings of the Conference on Evolutionary Multiobjective Optimization*, 2005.

[7] R. L. Becerra and C. A. C. Coello, "Optimization with Constraints using a Cultured Differential Evolution Approach," in *Proceedings of the Conference on Genetic and Evolutionary Computation*, 2005.

[8] K. Deb, A. Pratap, S. Agrawal, and T. Meyarian, "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II," *IEEE Transaction on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[9] K. Zielinski, D. Peters, and R. Laur, "Stopping Criteria for Single-Objective Optimization," in *Proceedings of the Third International Conference on Computational Intelligence, Robotics and Autonomous Systems*, 2005.

[10] F. van den Bergh, "An Analysis of Particle Swarm Optimizers," Ph.D. dissertation, University of Pretoria, 2001.

[11] F. P. Espinoza, "A Self-Adaptive Hybrid Genetic Algorithm for Optimal Groundwater Remediation Design," Ph.D. dissertation, University of Illinois, 2003.

[12] K. V. Price, R. M. Storn, and J. A. Lampinen, *Differential Evolution - A Practical Approach to Global Optimization*. Springer-Verlag Berlin Heidelberg, 2005.

[13] D. Zaharie and D. Petcu, "Parallel Implementation of Multi-Population Differential Evolution," in *Proceedings of the 2nd Workshop on Concurrent Information Processing and Computing*, 2003.

[14] B. V. Babu and R. Angira, "New Strategies of Differential Evolution for Optimization of Extraction Process," in *Proceedings of International Symposium & 56th Annual Session of IIChE*, 2003.

[15] P. Weitkemper, V. Kühn, and K.-D. Kammeyer, "Analysis of Iterative Successive Interference Cancellation in SC-CDMA Systems," in *Fifth International Workshop on Multi-Carrier Spread Spectrum*, 2005.

[16] S. Verdú, *Multiuser Detection*. Cambridge University Press, 1998.

[17] A. Nordio, M. Hernandez, and G. Caire, "Low-Complexity Turbo Equalization and Multiuser Decoding for TD-CDMA," *IEEE Transactions on Wireless Communication*, vol. 3, no. 2, pp. 454–465, 2004.

[18] K. Zielinski, P. Weitkemper, R. Laur, and K.-D. Kammeyer, "Parameter Study for Differential Evolution Using a Power Allocation Problem Including Interference Cancellation," accepted for the IEEE Congress on Evolutionary Computation, 2006.

[19] J. Liang, T. P. Runarsson, E. Mezura-Montes, M. Clerc, P. Suganthan, C. A. C. Coello, and K. Deb, "Problem Definitions and Evaluation Criteria for the CEC 2006 Special Session on Constrained Real-Parameter Optimization," Tech. Rep., 2005.

[20] K. Zielinski, D. Peters, and R. Laur, "Run Time Analysis Regarding Stopping Criteria for Differential Evolution and Particle Swarm Optimization," in *Proceedings of the 1st International Conference on Experiments/Process/System Modelling/Simulation/Optimization*, 2005.