Sex and Death: Towards Biologically Inspired Heuristics for Constraint Handling

Oliver Kramer
University of Dortmund
FB Informatik, LS XI
44221 Dortmund, Germany
oliver.kramer@udo.edu

Stephan Brügger
University of Paderborn
Fakultät EIM-I, FG WBS
33098 Paderborn, Germany
stebr@upb.de

Dejan Lazovic
University of Paderborn
Fakultät EIM-I, FG WBS
D-33098 Paderborn, Germany
dejan@upb.de

ABSTRACT

Constrained continuous optimization is still an interesting field of research. Many heuristics have been proposed in the last decade. Most of them are based on penalty functions. Here, we experimentally investigate the two constraint handling heuristics proposed by Kramer and Schwefel [15]. The two sexes evolution strategy (TSES) is inspired by the biological concept of sexual selection and pairing. The death penalty step control evolution strategy (DSES) is based on the controlled reduction of a minimum step size depending on the distance to the infeasible search space. These two methods are able to overcome the problem of premature mutation strength reduction, a result of the self-adaptation mechanism of evolution strategies in constrained environments. All methods are experimentally evaluated on a couple of typical constrained test problems. These experiments offer recommendations for the TSES population ratios and the speed of the ϵ-reduction process of the DSES.

Categories and Subject Descriptors

G.1.6 [**Numerical Analysis**]: Optimization—*Constrained optimization*

General Terms

Algorithms, Experimentation

Keywords

Evolution Strategies, Constrained Real Parameter Optimization

1. INTRODUCTION

In this paper we concentrate on the experimental evaluation of two of the three constraint handling methods for evolution strategies (ES) proposed by Kramer and Schwefel [15]. After the definition of the NLP problem and a short introduction to ES, we summarize preliminary work in the field of constraint handling and describe the problem of premature mutation strength reduction. In section 2 we present the two sexes ES (TSES) with a short description, its pseudo code and an experimental parameter analysis on two test problems. The TSES allows a part of the population to discover the infeasible part of the search space. Afterwards, we present the death penalty step control evolution strategy (DSES) in section 3. The DSES makes use of a minimum mutation strength in order to prevent premature step size reduction. A heuristic is introduced to reduce this minimum value. Finally, we summarize the experimental analysis on all constrained test functions and draw our attention to an outlook to future work.

1.1 The NLP Problem

First of all, we repeat the definition of the constrained continuous nonlinear programming problem: In the N-dimensional search space $\mathbb{R}^N$ find a solution $\mathbf{x} = (x_1, x_2, ..., x_N)^T$, which minimizes $f(\mathbf{x})$:

$$\begin{array}{lll} f(\mathbf{x}) \to \min., & \mathbf{x} \in \mathbb{R} & \text{with subject to} \\ \text{inequalities} & g_i(\mathbf{x}) \leq 0, & i = 1, ..., n_1 \\ \text{equalities} & h_j(\mathbf{x}) = 0, & j = 1, ..., n_2 \end{array} \tag{1}$$

A feasible solution $\mathbf{x}$ satisfies all n_1 inequality and n_2 equality constraints. Many constraint-handling techniques like penalty functions make use of a constraints violation measurement G:

$$G(\mathbf{x}) = \sum_{i=1}^{n_1} \max[0, g_i(\mathbf{x})]^\beta + \sum_{j=1}^{n_2} |h_j(\mathbf{x})|^\gamma \tag{2}$$

The parameters β and γ are usually chosen as one or two. In the following, only inequality constraints are taken into account.

1.2 Evolution strategies

We focus our research about constraint handling on ES. For a comprehensive introduction to ES see Beyer and Schwefel [3]. A $(\mu/\rho \overset{+}{,} \lambda)$-ES for continuous search spaces uses a parent population with cardinality μ and an offspring population with cardinality λ. In the standard $(\mu \overset{+}{,} \lambda)$-ES a vector of $n_\sigma = N$ step sizes is used, which results in mutation ellipsoids:

$$\vec{z} := (\sigma_1 \mathcal{N}_1(0,1), ..., \sigma_N \mathcal{N}_N(0,1)) \tag{3}$$

The corresponding strategy parameter vector is mutated with the extended log-normal rule:

$$\vec{\sigma}' := e^{(\tau_0 \mathcal{N}_0(0,1))} \cdot (\sigma_1 e^{(\tau_1 \mathcal{N}_1(0,1)}, ..., \sigma_N e^{(\tau_1 \mathcal{N}_N(0,1)}) \tag{4}$$

GECCO'07, July 7–11, 2007, London, England, United Kingdom.

There exist two main variants of recombination within evolution strategies. The intermediate crossover produces new offspring $\vec{o} = (o_1, ..., o_N)$ by calculating the arithmetic mean of ρ parents p^i with $1 \leq i \leq \rho$ with $p^i = (p^i_1, ..., p^i_N)$:

$$o_k := \frac{1}{\rho} \sum_{i=1}^{\rho} p^i_k \tag{5}$$

For discrete representations rounding procedures have to be used.

The dominant crossover chooses each component from one of the m parents randomly with uniform distribution.

$$o_k := p^i_k \text{ with } i := \text{Random } \{1, ..., \rho\} \tag{6}$$

As an extension of the comma selection scheme (selection of the best μ individuals exclusively out of the current offspring generation) the parameter κ specifies the number of reproduction cycles individuals are allowed to survive in the parental population if they cannot be replaced by fitter offspring solutions.

1.3 Preliminary work

Most of the constraint handling methods for evolutionary algorithms are based on penalty functions. An early, rather general penalty approach is the SUMT (sequential unconstrained minimization technique) by Fiacco and McCormick [9]. The constrained problem is solved by a sequence of unconstrained optimizations in which the penalty factors are stepwise intensified. In other penalty approaches penalty factors can be defined statically [11] or depending on the number of satisfied constraints [16]. They can dynamically depend on the number of generations of the EA as Joines and Houck propose [13]. Annealing penalties can be adapted according to an external cooling scheme [13] or by adaptive heuristics [1]. In the death penalty (DP) approach infeasible solutions are rejected and new solutions are created until enough feasible ones exist. There exist other constraint handling methods as well, e.g. the segregated genetic algorithm by Le Riche et al. [23] where two penalty functions, a weak and an intense one, are calculated in order to surround the optimum. In the coevolutionary penalty-function approach by Coello Coello [6] the penalty factors of an inner evolutionary algorithm are adapted by an outer EA. Some methods are based on the assumption that any feasible solution is better than any infeasible one [22], [8]. An example are the metric penalty functions by Hoffmeister and Sprave [10]. Decoders build up a relationship between the constrained search space and an artificial search space easier to handle [18], [14]. Repair algorithms either replace infeasible solutions or only use the repaired solutions for evaluation of their infeasible pendants [7], [2]. Multiobjective optimization techniques are based on the idea of handling each constraint as an objective and are used by Parmee and Purchase [21], Jimenez and Verdegay [12], Coello Coello [5], and Surry et al. [26]. In the behavioral memory-method by Schoenauer and Xanthakis [24] the EA concentrates on minimizing the constraint violation of each constraint in a certain order and optimizing the objective function in the last step. A predator-prey approach to handle constraints is proposed by Paredis [20] using two separate populations. A comprehensive overview to constraint handling techniques is given by Coello Coello [7] and also by Michalewicz [18]. Recently, Coello Coello [19] introduced a technique based on a multimembered evolution strategy combining a feasibility comparison mechanism with several modifications of the standard ES. Other approaches point at using differential evolution for constraint handling [4]. Most state-of-the-art methods and their experimental validation can be found in the CEC 2006 Special Session on Constrained Real Parameter Optimization.

1.4 Premature step size reduction

Often, ES suffer from premature step size reduction in case of active inequality constraints. This problem is described in detail in [15]. Here, we shortly revise the reason for premature stagnation. When the distance of individuals of a population to the boundary of the feasible search space is smaller than the mutation step size, a huge part of the region where better individuals are produced is cut off. Also in the case of penalty functions this region is punished with worse fitness. The success region is not cut off or punished in the case of smaller step sizes. Hence, the self-adaptation favors smaller step sizes before reaching the optimum. This is called premature mutation strength reduction and results in premature fitness stagnation. To overcome this problem, in the following two heuristics are proposed. The TSES aims at getting rid of constraint boundaries by letting a part of the population discover the infeasible region. The DSES hinders the self-adaptation to reduce the mutation strength beyond a minimum value, but reduces this minimum value with a control heuristic.

2. CONSTRAINT HANDLING WITH TWO SEXES

At first, we introduce the TSES and describe its biological motivation. Afterwards, we analyze its behavior under different parameter settings experimentally on two test functions.

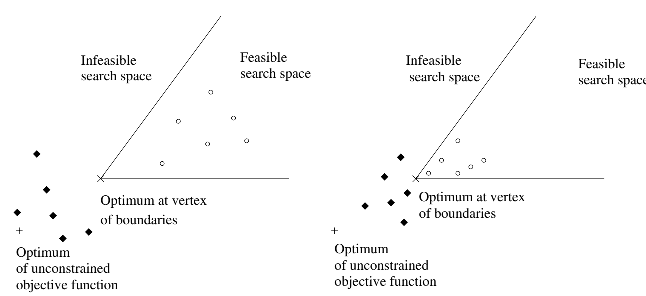

Figure 1: Idea of pairing in the vertex of feasibility: the intermediary recombination allows the TSES to approach the optimum from two sides.

2.1 Biologically inspired constraint-handling

The idea of the TSES is to handle the objective function and the constraint functions as separate objectives [15]. Every individual of the TSES is assigned to a new feature called its sex. Similar to nature, individuals with different sexes are selected according to different objectives. Individuals with sex o are selected by the objective function. Individuals with sex c are selected by the fulfillment of constraints. The intermediary recombination operator allows only pairing between parents of different sex. Consider the situation

TSES	κ	best	avg	worst	dev	cv	σ	ffc/cfc
g01								
$(8+8, 13+87)$	200	-15.0	-14.800866854295	-12.461685434305	$5.45 \cdot 10^{-2}$	0.0	$1.47 \cdot 10^{-11}$	360599
$(20+20, 25+200)$	50	-15.0	-14.999999999999	-14.999999999999	$9.11 \cdot 10^{-15}$	0.0	$2.72 \cdot 10^{-14}$	243920
$(20+20, 25+200)^*$	50	-15.0	-14.895808401022	-12.440215102243	$4.06 \cdot 10^{-2}$	0.0	$2.94 \cdot 10^{-14}$	277823
$(40+40, 50+400)$	50	-15.0	-14.999999999999	-14.999999999999	$5.36 \cdot 10^{-15}$	0.0	$1.86 \cdot 10^{-14}$	411 732
$(40+40, 50+400)^*$	50	-15.0	-14.999999999999	-14.999999999999	$5.58 \cdot 10^{-15}$	0.0	$1.74 \cdot 10^{-14}$	468 580
DP	0	-14.999999999966	-14.173473498602	-11.5825566209	1.106	0.0	$5.164 \cdot 10^{-11}$	143187
g12								
$(8+8, 15+85)$	50	-1.0	-1.0	-1.0	0.0	0.0	$5.56 \cdot 10^{-8}$	12305
$(8+8, 15+85)^*$	50	-1.0	-1.0	-1.0	0.0	0.0	$5.92 \cdot 10^{-8}$	11722
$(8+8, 13+87)$	50	-1.0	-0.9999437499999	-0.9943749999921	$5.62 \cdot 10^{-5}$	0.0	$1.71 \cdot 10^{-7}$	14534
$(8+8, 13+87)^*$	50	-1.0	-1.0	-1.0	0.0	0.0	$6.92 \cdot 10^{-8}$	12720
$(8+8, 10+90)$	50	-1.0	-0.9999999999999	-0.9999999999999	$7.46 \cdot 10^{-16}$	0.0	$1.57 \cdot 10^{-6}$	18938
$(8+8, 10+90)^*$	50	-1.0	-0.9999999999999	-0.9999999999999	$1.64 \cdot 10^{-16}$	0.0	$1.49 \cdot 10^{-6}$	17237
$(15, 100)$ DP	0	-1.0	-0.9999999999982	-0.9999999999802	$2.94 \cdot 10^{-13}$	0.0	$5.76 \cdot 10^{-5}$	20318

Table 1: Above: Experimental analysis of the TSES on problem g01. The standard DP method and the (8+8,13+87)-TSES show bad results, an increase of population sizes is necessary. All other experiments were more successful, the optimum could be reached in almost every run. Below: the TSES on problem g12. The TSES approximates the optimum -1.0 in most of the experiments with sufficient accuracy. In some runs of the $(8+8, 13+87)$-TSES with fixed starting points a premature termination was observed.

presented in figure 1. The optimum lies at the boundaries of the feasible search space. The optimum of the unconstrained objective function lies in the infeasible search space. In the $(\mu_o+\mu_c, \lambda_o+\lambda_c)$-TSES μ_o parents are selected out of λ_o individuals with sex o, whereas μ_c parents are selected out of λ_c offspring individuals of the previous generation with sex c. As the individuals with sex o are selected according to the objective function, they tend to lie finally in the infeasible search space (black squares) whereas the c-individuals are selected by the fulfillment of all constraints and mostly lie in the feasible search space (white circles). The measurement G for the fulfillment of constraints has already been defined in equation 2. By means of intermediary recombination, all individuals get closer to the optimum of the problem, but still are found on opposite sides of the boundaries between the feasible and the infeasible search space. The TSES does not demand feasible starting points during initialization.
Figure 2 shows the pseudo-code of the TSES. The usual self-adaptation process effects an explosion of the mean step sizes, as the invasion of c-individuals into the unconstrained search space is beneficial. To prevent this we introduce a two-step selection operator for the sex c, according to the metric penalty function by Hoffmeister and Sprave [10]. At first, we select these individuals by fulfillment of constraints, secondly, if enough feasible solutions exist by the objective function. Another modification is the introduction of a finite life span $1 < \kappa < \infty$ for individuals with the sex c. These modifications lead to promising results on test functions. Experiments show that the population of c-individuals should be much higher than the population of o-individuals which furthermore emphasizes the importance of c-individuals.

```
Start
  t:=0;
  Initialize partental population P_o with sex o;
  Initialize partental population P_c with sex c;
  Repeat
    For k=1 To λ_1 + λ_2 Do
      Choose one parent from P_o^t and one from P_c^t
      recombination_step_sizes;
      recombination_objective_variables;
      mutation_step_sizes;
      mutation_objective_variables;
      fitness of a_k := f(x_k);
      If k < λ_2 Then
        sex(a_k) = o;
      Else
        sex(a_k) = c;
      Add a_k to offspring population O zu;
    Next;
    Select parental population P_o
    from O considering the fitness function;
    Select parental population P_c
    from O considering the constraint violation;
    t := t + 1;
  Until termination condition
End
```

Figure 2: Pseudo-code of the TSES.

2.2 Parameter analysis of the TSES

We now accomplish some experimental runs of the TSES under various parameter settings for $\mu_o, \mu_c, \lambda_o, \lambda_c$ and κ. Table 1, upper part, shows the outcome of the experiments on problem g01. Problem g01 exhibits a quadratic objective function and nine linear inequality constraints. The experiments marked with a star make use of randomized starting individuals. The other tests use a fixed initialization. A (8+8,13+87)-TSES was not at all able to approximate the optimum. But a modification of the sex ratios could change the situation. Both, a (20+20,25+200)- and a (40+40,50+400)-TSES, were able to find the optimum with arbitrary accuracy in almost every run. Only the (20+20,25+200)-TSES with randomized starting point suffered from premature step size reduction before reaching the optimum. In comparison to DP a significant improvement could be observed.

Let us now have a look onto the behavior of the TSES on problem g12. What we can observe here is a high quality of the experimental results. The (8+8,15+85)-, the (8+8,13+87)- as well as the (3+3,10+90)-TSES find the optimum with satisfying accuracy. The only exception is the (8+8, 13+87)-TSES with a fixed initial starting point. These results can be explained with outliers as the neighboring sex ratios are successful. Again, we observe a saving of constraint function calls in comparison to DP.

3. CONSTRAINT HANDLING WITH DEATH PENALTY AND STEP CONTROL

As mentioned in section 1.4 the DP method suffers from premature step size reduction because of insufficient birth surplus. The death penalty step control evolution strategy (DSES) is based on DP, i.e. rejection of infeasible solutions. For the initialization feasible starting points are required. The key principle of the approach is a minimum step size ϵ, a lower bound on the step sizes σ, that prevents the self-adaptation from premature step size reduction. But it also prevents the optimization process from unlimited approximation of the optimum when reaching the range of ϵ. Consequently, a control mechanism is introduced with the task of reducing ϵ when approximating the optimum. Intuitively, the reduction process depends on the number of infeasible mutations produced when reaching the area of the optimum at the boundary of the feasible search space.

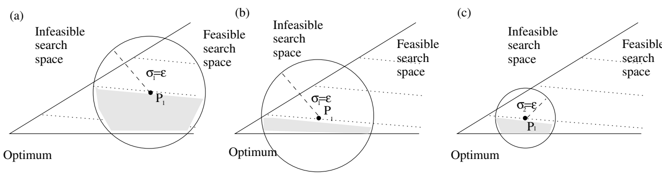

Figure 3: Working principle of the DSES. In the vicinity of the infeasible search space (a) the population approximates the optimum until the minimum mutation strength ϵ decreases the probability for successful offspring (b). It is time for reduction of ϵ for further approximation (c). When the number of infeasible trials exceeds the parameter mod the minimum step size ϵ is reduced and a further approximation of the optimum becomes possible.

Consider the situation presented in figure 3. Again, for the sake of better understanding we assume that all mutations fall within a σ-circle around the parental individual instead of using a normal distribution with standard deviation σ. On the left (a), the parent P has come close to the optimum at a vertex of the feasible search space. Further approximation (b) with the same minimum step size means an increase of infeasible mutations which are counted with the parameter z. The reduction process of ϵ depends on the number z of rejected infeasible solutions: Every ϖ infeasible trials ϵ is reduced by a factor $0 < \vartheta < 1$ according to the equation:

$$\epsilon' := \epsilon \cdot \vartheta \tag{7}$$

The DSES is denoted by [ϖ; ϑ]-DSES. Figure 4 shows the pseudo-code of the DSES.

3.1 Parameter analysis of the DSES

The analysis of the DSES parameter settings shows table 2. Above, the analysis of the DSES on problem g01 reveals a significant improvement in comparison to DP. The [400;0.5]-, the [400;0.3]- and the [400;0.1]-DSES were able to approximate the optimum with arbitrary accuracy. The ϵ-reduction was performed too fast for the [100,ϑ]-DSES with $\vartheta = 0.1, 0.3, 0.5$.

We analyze the behavior of the DSES on problem g07. In comparison to the method DP a significant improvement of

```
Start
  t:=0;
  Initialize parental population P
  Repeat
    For k=1 To λ Do
      z = 0;
      Repeat
        Choose ρ parents from P
        recombination_step_sizes;
        recombination_objective_variables;
        mutation_step_sizes;
        If (z mod mod) == 0 Then
          ε = ε· melt;
        For j=1 To N Do
          If σ_j^k < ε Then
            σ_j^k = ε
          Next
        mutation_objective_variables;
        fitness of a_k := f(x);
        z:=z+1;
      Until a_k feasible
      Add a_k to offspring population O;
    Next
    Select parental population P from O;
    t := t + 1;
  Until termination condition
End
```

Figure 4: Pseudocode of the DSES.

accuracy can be observed. The slower ϵ is decreased the more the quality of the results can be improved. But we have to admit that the number of fitness function calls *ffc* and in particular the number of constrained function calls *cfc* explode. Hence, we can only recommend the DSES on such problems when *ffc* and cfc are not too expensive. The experiments show that DP completely fails concerning the quality of the results.

4. EXPERIMENTAL ANALYSIS

We summarize the results of our experimental analysis on the various constrained problems. Note that we translate equality constraints $h_j(\mathbf{x}) = 0, \quad j = 1, ..., n_2$ into inequality constraints $g_i(\mathbf{x}) \leq \epsilon, \quad i = 1, ..., n_2$ and $g_j(\mathbf{x}) \leq -\epsilon, \quad j = 1, ..., n_2$ and $\epsilon = 0.0001$. For all ES we used $n_{sigma} = N$ step sizes (N ist the dimension of the problem) and in most cases the mutation parameter recommendation $\tau_0 = (\sqrt{2N})^{-1}$ and $\tau = (\sqrt{2\sqrt{N}})^{-1}$. We chose the initial step sizes $\sigma_i = \frac{|x^{(0)} - x^*|}{N}$ with $1 \leq i \leq N$, starting point $x^{(0)}$ and optimum x^*. As termination condition we chose premature mutation strength reduction, i.e. the ES is terminated if the difference between the best individuals of two successive generations is smaller than $\iota = 10^{-12}$. All DP and DSES tests are based on a (15/2,100)-ES with intermediary recombination. Population sizes of the TSES are marked out explicitly.

The measured parameters are the usual ones concerning the fitness of the best individual in the last generation of the various runs (*best*, *avg*, *worst* and *dev*). Parameter cv measures the constraint violation of this best individual. Parameter σ is the mean of the mutation strength in the last generation and therefore and indicator for premature stagnation. *Ffc* counts the number of fitness function calls, cfc the number of constraint function calls. For the DSES *tries* counts the average number of tries the ES needs to produce a feasible individual.

DSES	best	avg	worst	dev	cv	σ	ffc	cfc	tries
g01									
[400; 0.5]	−14.999999999889	−14.999999999128	−14.9999999908	$1.781 \cdot 10^{-9}$	0.0	$2.062 \cdot 10^{-10}$	93224	3174166	34.0
[400; 0.3]	−14.999999999925	−14.999999998887	−14.9999999888	$2.379 \cdot 10^{-9}$	0.0	$2.139 \cdot 10^{-10}$	75484	2035043	26.9
[400; 0.1]	−14.999999999953	−14.999999997637	−14.9999999550	$8.908 \cdot 10^{-9}$	0.0	$3.712 \cdot 10^{-10}$	55992	1136101	20.2
[100; 0.5]	−14.999999999940	−14.919999999578	−12.9999999998	0.399	0.0	$6.875 \cdot 10^{-11}$	51764	548682	10.6
[100; 0.3]	−14.999999999979	−14.818124750686	−12.4531187811	0.634	0.0	$1.329 \cdot 10^{-10}$	40372	312837	7.7
[100; 0.1]	−14.999999999978	−14.753748891277	−12.8437223436	0.681	0.0	$2.617 \cdot 10^{-10}$	37716	260461	6.9
DP	−14.999999999966	−14.173473498602	−11.5825566209	1.106	0.0	$5.164 \cdot 10^{-11}$	33067	110120	3.3
g07									
[70; 0.7]	24.306237739205	24.306723093725	24.3081991031	$4.574 \cdot 10^{-4}$	0.0	$3.828 \cdot 10^{-9}$	1 655 956	11 159 837	6.7
[70; 0.5]	24.306330953891	24.307474634131	24.3103445834	$9.562 \cdot 10^{-4}$	0.0	$9.199 \cdot 10^{-10}$	936 436	6 287 404	6.7
[70; 0.3]	24.306433546473	24.309502933630	24.3276119592	0.004	0.0	$1.792 \cdot 10^{-9}$	577 736	3 850 091	6.6
[40; 0.7]	24.308209926624	24.335015185002	24.3771310990	0.019	0.0	$6.385 \cdot 10^{-11}$	68 436	401 509	5.8
[40; 0.5]	24.315814462518	24.357010892756	24.4785842220	0.039	0.0	$8.378 \cdot 10^{-11}$	47 996	263 890	5.5
[40; 0.3]	24.337966344507	24.400529764694	24.5778543888	0.055	0.0	$2.158 \cdot 10^{-10}$	37 084	189 393	5.1
DP	24.449127539670	26.337577676591	30.9348325535	1.472	0.0	$1.208 \cdot 10^{-11}$	30 835	87 884	2.8

Table 2: Above: Experimental analysis of a (15/2, 100)-DSES with intermediary recombination and an initial step size of $\sigma_i = 0.1$ on problem g01. The DSES is able to approximate the optimum with arbitrary accuracy while DP fails on g01. Below: A (15/2, 100)-DSES with intermediary recombination on problem g07. The initial step size was chosen as $\sigma = 10.93$ for the DSES and $\sigma = 0.1$ We observe significant better results than with DP and achieve the best results with the slowest ϵ-reduction. On the other hand the slow reduction causes an explosion of constraint function calls.

4.1 The TSES on all functions

The upper part of table 3 shows the experimental results of the TSES on all considered problems. The behavior of the TSES on g01 has already been described. On problem g04 a (8+8,15-85)-TSES failed, but the TSES with an offspring sex ratio of 13+87 and 10+90 found the optimum in every run. Again, the improvement in comparison to DP is significant. On problem g06, a highly constrained problem with a ratio of only 0.0066% of feasible search space, all algorithms, including DP were successful. The TSES could improve the results on problem g08 in comparison to DP, but demands a higher number of *ffc*/*cfc*. The results of the TSES on g09 were not satisfying, but slightly better than DP. Again, this improvement has to be paid with approximately 5 to 10 times higher number of *ffc*/*cfc*. On problem g11 various experiments around the offspring sex ratio (8+8,13+87) like recommended by Kramer and Schwefel [15] could not achieve promising results. Tests around the ratio (8+8,10+200) were more successful. All experiments with the (8+8,10+200)-TSES and $\kappa = 200$ or $\kappa = 300$ showed sufficient results. A higher κ causes a loss of efficiency. We have to emphasize that the standard DP method shows comparable results concerning the quality of the results. But the number of constraint function calls (cfc) is about ten times higher than the cfc values of the TSES. So, we can observe a efficiency advantage of the TSES.

The experiments on problem g12 have already been commented in section 2.2. On problem g16 the (8+8,13+87)-TSES is not more successful, but less efficient than DP with an average accuracy of 3 decimal places. But the (40+40,50+400)-TSES reaches the optimum with arbitrary accuracy demanding 3-4 times more *ffc*/*cfc*. The (8+8,10+90)-TSES is the best compromise between quality of the results and efficiency on problem g24. But we recommend to use DP as it demands much less *ffc*/*cfc*.

While DP completely fails on Schwefel's constrained problem 2.40 the (8+8,13+87)-TSES reaches the optimum in every run. The same behavior can be observed on Schwefel's problem 2.41. In contrast to these problems the TSES is not able to approximate the optimum of problem TR2 arbitrarily. But at least a significant improvement in comparison to the results with DP can be observed.

4.2 The DSES on all functions

The experimental results of the DSES on all considered problems are summarized in table 3, lower part. The behavior of various [ϖ; ϑ]-settings on problem g01 have already been described in section 3.1. On g02 the experiments show that also a slow reduction of ϵ can only improve the quality of the results slightly, but has to be paid with a high number of fitness and constraint functions calls. Obviously, we cannot recommend to use the DSES on problem g02, because no significant improvement can be achieved although we pay with inefficiency.

On problem g06 all tested DSES variants achieved promising results. The standard DP has to be recommended as the cfc values are lower in comparison. The behavior of the DSES on g07 has already been described. Like on g06 the DSES and DP are able to approximate the optimum of g08 very well. But here, the method DP exhibits no significant performance advantages. Only the slowest decrease of ϵ on problem g09 enables the DSES to approximate the optimum better than DP and *faster* DSES variants. Again, this improvement has to be paid with higher *ffc* and *cfc*. As already stated, the method DP performs well on problem g24. The DSES performs similarly on this problem.

On Schwefel's problem 2.40 and 2.41 the DSES is able to approximate the optimum with most of the tested settings. DP fails on these problems and suffers from premature mutation strength reduction. Problem TR2 is hard to tackle. But the results of a [15; 0.5]-DSES on TR2 are slightly better than standard DP. As expected the number of *ffc* and *cfc* explode, i.e. they are 100 times higher.

5. CONCLUSION AND FUTURE WORK

Constraint handling for evolutionary computation offers a huge potential for heuristics. We analyzed the two heuristics TSES and DSES experimentally on known constrained prob-

pb	TSES	κ	best	avg	worst	dev
TSES						
TR2	(8+8,13+87)	300	2.0000000000	2.0000000095	2.0000000633	$1.19 \cdot 10^{-8}$
2.40	(8+8,13+87)	50	-5000.0000000000	-4999.9999999999	-4999.9999999997	$3.31 \cdot 10^{-7}$
2.41	(8+8,13+87)	50	-17857.1428571482	-17857.1428571426	-17857.1428571374	$8.56 \cdot 10^{-10}$
g01	(40+40,50+400)	50	-14.9999999999	-14.9999999999	-14.9999999999	$5.36 \cdot 10^{-15}$
g02	(40+40,50+400)	50	-0.7926079090	-0.6583469345	-0.4352731883	0.009732
g04	(8+8,13+87)	50	-30665.5386717833	-30665.5386717833	-30665.5386717832	$1.78 \cdot 10^{-11}$
g06	(8+8,10+90)	50	-6961.8138755801	-6961.8138755801	-6961.8138755801	$8.69 \cdot 10^{-13}$
g07	(40+40,50+400	50	24.3171014481	24.4613633236	24.8935876076	0.0101172
g08	(8+8,15+85)	50	-0.0958250414	-0.0958250414	-0.0958250414	$5.64 \cdot 10^{-18}$
g09	(8+8,13+87)	50	680.6303038768	680.6345393750	680.6530448662	$4.65 \cdot 10^{-4}$
g11	(8+8,10+200)	300	0.7499900000	0.74999001691	0.7499903953	$4.43 \cdot 10^{-9}$
g12	(8+8,10+90)	50	-1.0000000000	-0.9999999999	-0.9999999999	$7.46 \cdot 10^{-16}$
g16	(40+40,50+400)	50	-1.9051552585	-1.9051552585	-1.9051552585	$1.65 \cdot 10^{-15}$
g24	(8+8,10+90)	50	-5.5080132715	-5.5080132715	-5.5080132715	$4.36 \cdot 10^{-13}$
DSES						
TR2	$[15; 0.3]$		2.0000000008	2.0000042774	2.0000539455	$8.45 \cdot 10{-}6$
2.40	$[100; 0.7]$		-4999.9999999999	-4999.9999999995	-4999.9999999960	$7.99 \cdot 10^{-10}$
2.41	$[75; 0.7]$		-17857.1428571428	-17857.1428571425	-17857.1428571404	$4.71 \cdot 10^{-10}$
g01	$[400; 0.5]$		-14.9999999998	-14.9999999991	-14.9999999908	$1.78 \cdot 10^{-9}$
g02	$[15; 0.3]$		-0.8036187549	-0.7658619287	-0.6999587065	0.029
g04	$[25; 0.7]$		-30665.5386717833	-30665.5386717831	-30665.5386717826	$1.60 \cdot 10^{-10}$
g06	$[10; 0.3]$		-6961.8138755801	-6961.8138755801	-6961.8138755800	$1.90 \cdot 10^{-11}$
g07	$[70; 0.7]$		24.3062377392	24.3067230937	24.3081991031	$4.57 \cdot 10^{-4}$
g08	$[2; 0.9]$		-0.0958250414	-0.0958250414	-0.0958250414	$9.06 \cdot 10^{-17}$
g09	$[18; 0.7]$		680.6301304921	680.6308434198	680.6322597787	$6.59 \cdot 10^{-4}$
g11	$[10; 0.5]$		0.7499000007	0.7499008144	0.7499035419	$1.02 \cdot 10^{-6}$
g12	$[200; 0.5]$		-0.9999999999	-0.9999999999	-0.9999999999	$2.05 \cdot 10^{-12}$
g16	$[100; 0.5]$		-1.9051552585	-1.9051552584	-1.9051552580	$9.12 \cdot 10^{-11}$
g24	$[15; 0.3]$		-5.5080132715	-5.5080132715	-5.5080132714	$2.36 \cdot 10^{-11}$

Table 3: Comparison of the experimental results of the TSES (upper part) and the DSES (lower part) on the considered problems. The best, the average, and the worst fitness as well as the standard deviation of the best solution of the last generations of 50 runs are shown.

lems from literature. Our experiments revealed the shortcomings of the standard method DP. We showed that the introduction of a least mutation strength ϵ together with an adaptation technique to reduce ϵ offers the possibility to approximate the optimum in many cases. ϖ and ϑ define the speed of the ϵ-reduction process. Before applying the DSES it has to be considered whether it is worth to invest the higher number of fitness and constraint function calls.

The TSES is inspired by the concepts of sex and pairing. The experiments revealed that the TSES is able to approximate the optimum in most of the cases. Sometimes a performance win in comparison to DP or the DSES was observed. An advantage of the TSES is that no infeasible starting points have to be available at the beginning of the search.

The main disadvantage of the proposed methods is the dependence on new parameters. But this argument can be weakened. Population sizes have to be defined for almost every EA. The success of the TSES depends on the sex ratios, but here we offer examples for successful population ratios. The DSES only depends on two new parameters, ϖ and ϑ, which can be treated as only one parameter: speed. In the future we will focus our attention on a further experimental and also theoretical analysis of the proposed methods. We will also try to answer the question how we can identify features of the constrained problems which determine the applicability of the proposed methods.

APPENDIX

A. PROBLEMS

Tangent problem.

Minimize

$$F(\mathbf{x}) = \sum_{i=1}^{n} x_i^2 \qquad \text{(n-dim. sphere model)} \tag{8}$$

Constraints

$$g(\mathbf{x}) = \sum_{i=1}^{n} x_i + t > 0, \qquad t \in \mathbb{R} \qquad \text{(tangent)} \tag{9}$$

For n=k and t=k the minimum is

$$x^* = (1, ..., 1)^T, \text{ mit } \quad F(x^*) = k. \tag{10}$$

Problem 2.40.

Schwefel's problem 2.40 [25]

Minimize:

$$f(x) = -\sum_{i=1}^{5} x_i$$

Constraints:

$$g_j(x) = \begin{cases} x_j \geq 0, & \text{for } j = 1, ..., 5 \\ -\sum_{i=1}^{5} (9+i)x_i + 50000 \geq 0, & \text{for } j = 6 \end{cases}$$

Minimum:

$$x^* = (5000, 0, 0, 0, 0)^T$$

$$f(x^*) = -5000$$

g_2 to g_6 active.

Problem 2.41.

Schwefel's problem 2.41 [25]
Minimize:

$$F(\vec{x}) = -\sum_{i=1}^{5}(ix_i)$$

Constraints like problem 2.40.
Minimum:

$$x^* = (0, 0, 0, 0, \frac{50000}{14})^T$$

$$f(x^*) = -\frac{250000}{14}$$

g_j active for $j = 1, 2, 3, 4, 6$.

*problems g**.*

For the definition of the constrained g**-problems, see [17] for example.

B. REFERENCES

[1] J. C. Bean and A. B. Hadj-Alouane. A Dual Genetic Algorithm for Bounded Integer Programs. Technical report, Department of Industrial and Operations Engineering, The University of Michigan, 1992.

[2] S. V. Belur. CORE: Constrained Optimization by Random Evolution. In J. R. Koza, editor, *Late Breaking Papers at the Genetic Programming 1997 Conference*, pages 280–286, Stanford University, California, Juli 1997. Stanford Bookstore.

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

[4] J. Brest, V. Žumer, and M. S. Maučec. Self-adaptive Differential Evolution Algorithm in Constrained Real-Parameter Optimization. In *The 2006 IEEE Congress on Evolutionary Computation CEC2006*, pages 919–926. IEEE Press, 2006.

[5] C. A. Coello Coello. Treating Constraints as Objectives for Single-Objective Evolutionary Optimization. *Engineering Optimization*, 32(3):275–308, 2000.

[6] C. A. Coello Coello. Use of a Self-Adaptive Penalty Approach for Engineering Optimization Problems. *Computers in Industry*, 41(2):113–127, Januar 2000.

[7] C. A. Coello Coello. Theoretical and Numerical Constraint Handling Techniques used with Evolutionary Algorithms: A Survey of the State of the Art. *Computer Methods in Applied Mechanics and Engineering*, 191(11-12):1245–1287, Januar 2002.

[8] K. Deb. An Efficient Constraint Handling Method for Genetic Algorithms. *Computer Methods in Applied Mechanics and Engineering*, 186(2/4):311–338, 2001.

[9] A. Fiacco and G. McCormick. The sequential unconstrained minimization technique for nonlinear programming - a primal-dual method. *Mgmt. Sci.*, 10:360–366, 1964.

[10] F. Hoffmeister and J. Sprave. Problem-independent handling of constraints by use of metric penalty functions. In L. J. Fogel, P. J. Angeline, and T. Bäck, editors, *Proceedings of the Fifth Annual Conference on Evolutionary Programming (EP'96)*, pages 289–294, San Diego, California, Februar 1996. The MIT Press.

[11] A. Homaifar, S. H. Y. Lai, and X. Qi. Constrained Optimization via Genetic Algorithms. *Simulation*, 62(4):242–254, 1994.

[12] F. Jiménez and J. L. Verdegay. Evolutionary techniques for constrained optimization problems. In H.-J. Zimmermann, editor, *7th European Congress on Intelligent Techniques and Soft Computing (EUFIT'99)*, Aachen, 1999. Verlag Mainz.

[13] J. Joines and C. Houck. On the use of non-stationary penalty functions to solve nonlinear constrained optimization problems with GAs. In D. Fogel, editor, *Proceedings of the first IEEE Conference on Evolutionary Computation*, pages 579–584, Orlando, Florida, 1994. IEEE Press.

[14] S. Koziel and Z. Michalewicz. Evolutionary Algorithms, Homomorphous Mappings, and Constrained Parameter Optimization. *Evolutionary Computation*, 7:19–44, 1999.

[15] O. Kramer and H. P. Schwefel. On three new approaches to handle constraints within evolution strategies. *Natural Computing: an international journal*, 5(4):363–385, 2006.

[16] A. Kuri-Morales and C. V. Quezada. A Universal Eclectic Genetic Algorithm for Constrained Optimization. In *Proceedings 6th European Congress on Intelligent Techniques & Soft Computing, EUFIT'98*, pages 518–522, Aachen, September 1998. Verlag Mainz.

[17] 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. In *Technical Report*, Singapore, 2006. Nanyang Technological University.

[18] Michalewicz and D. B. Fogel. *How to Solve It: Modern Heuristics*. Springer, 2000.

[19] E. M. Montes and C. A. Coello Coello. A simple multi-membered evolution strategy to solve constrained optimization problems. *IEEE Transactions on Evolutionary Computation*, 9:1–17, February 2005.

[20] J. Paredis. Co-evolutionary Constraint Satisfaction. In *Proceedings of the 3rd Conference on Parallel Problem Solving from Nature*, pages 46–55, Berlin, 1994. Springer.

[21] I. C. Parmee and G. Purchase. The development of a directed genetic search technique for heavily constrained design spaces. In I. C. Parmee, editor, *Adaptive Computing in Engineering Design and Control*, pages 97–102, Plymouth, UK, 1994. University of Plymouth.

[22] D. Powell and M. M. Skolnick. Using genetic

algorithms in engineering design optimization with non-linear constraints. In S. Forrest, editor, *Proceedings of the Fifth International Conference on Genetic Algorithms (ICGA-93)*, pages 424–431, San Mateo, California, Juli 1993. University of Illinois at Urbana-Champaign, Morgan Kaufmann.

[23] R. G. L. Riche, C. Knopf-Lenoir, and R. T. Haftka. A Segregated Genetic Algorithm for Constrained Structural Optimization. In L. J. Eshelman, editor, *Proceedings of the Sixth International Conference on Genetic Algorithms (ICGA-95)*, pages 558–565, San Mateo, California, Juli 1995. University of Pittsburgh, Morgan Kaufmann.

[24] M. Schoenauer and S. Xanthakis. Constrained GA Optimization. In S. Forrest, editor, *Proceedings of the Fifth International Conference on Genetic Algorithms (ICGA-93)*, pages 573–580, San Mateo, California, Juli 1993. University of Illinois at Urbana-Champaign, Morgan Kaufmann.

[25] H.-P. Schwefel. *Evolution and Optimum Seeking.* Sixth-Generation Computer Technology. Wiley Interscience, New York, 1995.

[26] P. D. Surry, N. J. Radcliffe, and I. D. Boyd. A Multi-Objective Approach to Constrained Optimisation of Gas Supply Networks : The COMOGA Method. In T. C. Fogarty, editor, *Evolutionary Computing. AISB Workshop. Selected Papers*, Lecture Notes in Computer Science, pages 166–180. Springer, Sheffield, U.K., 1995.