2006 IEEE Congress on Evolutionary Computation
Sheraton Vancouver Wall Centre Hotel, Vancouver, BC, Canada
July 16-21, 2006

Approximate Evolution Strategy using Stochastic Ranking

Thomas Philip Runarsson, *Member, IEEE*

***Abstract*— The paper describes the approximation of an evolution strategy using stochastic ranking for nonlinear programming. The aim of the approximation is to reduce the number of function evaluations needed during search. This is achieved using two surrogate models, one for the objective function and another for a penalty function based on the constraint violations. The proposed method uses a sequential technique for updating these models. At each generation the surrogate models are updated and at least one expensive model evaluation is performed. The technique is evaluated for some twenty-four benchmark problems.**

I. Introduction

An approximation of the evolution strategy using stochastic ranking (SRES) [11] for nonlinear programming is presented. The optimization technique SRES has been shown to be effective for a number difficult real world optimization tasks such as climate control [2] and parameter estimation in biochemical pathways [9], [3]. For these tasks computing the objective value and constraint violations typically involves computationally expensive models. It is therefore of significant importance to reduce the total number of function evaluations (FEs) needed during search. The approach taken here is to use surrogate models. Surrogate models are essentially computationally inexpensive models based on scarce samples of the expensive model. Surrogate models have been used in the past in optimization [15], [1], [13] with the aim of reducing the number of FEs. In the evolutionary computation literature surrogate models are thought of as fitness approximations. A recent survey of fitness approximation in evolutionary computation is presented in [5] and a framework for evolutionary optimization established in [6].

The handling of general nonlinear constraints in evolutionary optimization using surrogate models has not received much attention. However, initial results for the approach taken here using surrogate models were presented by the author in [10]. In this work a different view is taken when applying surrogate models in evolutionary optimization. The quality of the surrogate is assessed by its population ranking performance rather than how well it fits the underlying fitness landscape. The approach is called approximate ranking [10]. Two surrogate models are applied: one for the objective function and another for a penalty function based on the constraint violations. The quality of these surrogate models are determined by their consistency in ranking the population. The proposed method uses a sequential technique for updating these models. At each generation the surrogate models are updated and at least one expensive model evaluation is performed. The surrogate models are said to be sufficiently accurate if any improvement in the surrogate does not change the parent set currently selected. The key idea here is to evaluate the accuracy of the model by observing how it changes the behavior of the evolution strategy. An extension to this method, presented here, is that an upper limit of expensive function evaluations per generation is set to a fraction of the population size. The evolution strategy used is furthermore a hybrid of the original version described in [11] and the improved version described more recently in [12].

The paper is organized as follows. In section II the evolution strategy ES for the general nonlinear programming problem is described. In section III stochastic ranking, the surrogate model and approximate ranking is described. This is followed by an experimental study of the search method and its approximation on 24 benchmark functions in section IV. The paper concludes with a discussion and summary.

Thomas Philip Runarsson is with the Science Institute, University of Iceland, Iceland. (email: tpr@hi.is).

II. Constrained evolutionary optimization

Consider the general nonlinear programming problem formulated as

$$\text{minimize} \quad f(\boldsymbol{x}), \quad \boldsymbol{x} = (x_1, \ldots, x_n) \in \mathcal{R}^n, \tag{1}$$

where $f(\boldsymbol{x})$ is the objective function, $\boldsymbol{x} \in \mathcal{S} \cap \mathcal{F}$, $\mathcal{S} \subseteq \mathcal{R}^n$ defines the search space bounded by the parametric constraints

$$\underline{x}_i \leq x_i \leq \overline{x}_i, \tag{2}$$

and the feasible region $\mathcal{F}$ is defined by

$$\mathcal{F} = \{\boldsymbol{x} \in \mathcal{R}^n \mid g_j(\boldsymbol{x}) \leq 0, h_k(\boldsymbol{x}) = 0 \quad \forall\ j, k\}, \tag{3}$$

where $g_j(\boldsymbol{x})$, $j = 1, \ldots, n_g$, are inequality constraints and $h_k(\boldsymbol{x})$, $k = 1, \ldots, n_h$ are equality constraints. Using a penalty function approach the constraint violations are treated as a single function,

$$\phi(\boldsymbol{x}) = \sum_{j=1}^{n_g} [g_j^+(\boldsymbol{x})]^2 + \sum_{k=1}^{n_h} [h_k^+(\boldsymbol{x})]^2 \tag{4}$$

where $g_j^+(\boldsymbol{x}) = \max\{0, g_j(\boldsymbol{x})\}$ and $h_k^+(\boldsymbol{x}) = |h_k(\boldsymbol{x})|$ if $|h_k(\boldsymbol{x})| \leq \delta$, otherwise $h_k^+(\boldsymbol{x}) = 0$. In [12] an effective algorithm for solving nonlinear programming problems was introduced. The algorithm is essentially an improved version of the (μ, λ) evolution strategy (ES) using stochastic ranking presented in [11]. A variation of these versions is described by the pseudocode in fig. 1. The algorithm in fig. 1 also uses stochastic ranking, which balances the influence of the penalty and objective function in determining the overall ranking of the population. In particular the population of individuals, of size λ, are ranked from best to worst, denoted

```
1  Initialize: σ'_k = (x̄_k − x̲_k)/√n,
              x'_k = x̲_k + (x̄_k − x̲_k)U_k(0,1), k = 1,...,λ
2  while termination criteria not satisfied do
3     evaluate: f(x'_k), φ(x'_k), k = 1...,λ
4     rank the λ points and copy the best μ in their ranked order:
5     (x_i, σ_i) ← (x'_{i;λ}, σ'_{i;λ}), i = 1,...,μ
6     for k := 1 to λ do (replication)
7        i ← mod (k − 1, μ) + 1 (cycle through the best μ points)
8        if (k < μ) do (differential variation)
9           σ'_k ← σ_i
10          x'_k ← x_i + γ(x_1 − x_{i+1}) (retry using standard mutation)
11       else (standard mutation using global intermediate recombination)
12          σ'_k ← (σ_i + σ_p)/2 (randomly sampled p ∈ {1,...,μ})
13          σ'_{k,j} ← σ'_{k,j} exp(τ'N(0,1) + τN_j(0,1)), j = 1,...,n
14          x'_k ← x_i + σ'_k N(0,1) (if out of parametric bounds then retry)
15       od
16    od
17 od
```

Fig. 1. A (μ, λ) ES variant using the differential variation (line 10) performed once for each of the best $\mu - 1$ points ($\gamma = 0.85$). $U(0,1)$ is a uniform random number in $[0,1]$ and $N(0,1)$ is normally distributed with zero mean and variance one. $\tau' \propto 1/\sqrt{2n}$, $\tau \propto 1/\sqrt{2\sqrt{n}}$, see also [12].

$(\boldsymbol{x}_{1;\lambda}, \ldots, \boldsymbol{x}_{\mu;\lambda}, \ldots, \boldsymbol{x}_{\lambda;\lambda})$, and only the best μ are selected. A further modification, presented here, is simply the attempt to replace both $\phi(\boldsymbol{x})$ and $f(\boldsymbol{x})$ by surrogate models with the hope of reducing the number of function evaluations required. The surrogate models only influence the ranking, the remainder of the algorithm is unchanged.

III. Approximate Stochastic Ranking

This section describes the stochastic ranking procedure, followed by nearest neighbor regression as the surrogate and finally the approximate ranking procedure.

A. Stochastic Ranking

Stochastic ranking [11] is achieved by a bubble-sort-like procedure. In this approach a probability P_f of using only the objective function for comparing individuals in the infeasible region of the search space is introduced. That is, given any pair of two adjacent individuals, the probability of comparing them (in order to determine which one is fitter) according to the objective function is 1 if both individuals are feasible, otherwise it is P_f. The procedure provides a convenient way of balancing the dominance in a ranked set. In the bubble-sort-like procedure, λ individuals are ranked by comparing adjacent individuals in at least λ sweeps. The procedure is halted when no change in the rank ordering occurs within a complete sweep. Fig. 2 shows the stochastic bubble sort procedure used to rank individuals in a population [11].

Because one is interested at the end in feasible solutions, P_f should be less than 0.5 such that there is a pressure against infeasible solutions. The strength of the pressure can be adjusted easily by adjusting only P_f. In previous studies it has been found that a value of $P_f = 0.45$ is often sufficient. When $P_f = 0$ the ranking is equivalent to an over-penalization, where feasible points are ranked highest according to their objective value, followed by infeasible points w.r.t. the penalty function.

```
I_j = j ∀ j ∈ {1,...,λ}
for i = 1 to λ do
    for j = 1 to λ − 1 do
        sample u ∈ U(0,1) (uniform random number generator)
        if (φ(x_{I_j}) = φ(x_{I_{j+1}}) = 0) or (u < P_f) then
            if f(x_{I_j}) > f(x_{I_{j+1}}) then
                swap(I_j, I_{j+1})
            fi
        else
            if φ(x_{I_j}) > φ(x_{I_{j+1}}) then
                swap(I_j, I_{j+1})
            fi
        fi
    od
    if no swap done break fi
od
```

Fig. 2. Stochastic ranking procedure, $P_f = 0.45$.

B. Nearest Neighborhood Regression

Deciding on a general surrogate model for optimization, especially when taking the no-free lunch theorems [14] into consideration, is difficult. Commonly used surrogate models include, among others, Kriging, polynomial regression and radial basis function [4]. Perhaps the most simple and transparent surrogate model is the nearest neighbor (NN) regression model, that is

$$\hat{h}(\boldsymbol{x}_i) = h(\boldsymbol{y}_j) \text{ where } j = \underset{k=1,\ldots,\ell}{\operatorname{argmin}} \|\boldsymbol{x}_i - \boldsymbol{y}_k\| \tag{5}$$

where $\boldsymbol{Y} \equiv \{\boldsymbol{y}\}_{j=1}^{\ell}$ are the set of points which have been evaluated using the expensive model $h(\boldsymbol{y})$. The approximate model passes exactly through the function values at the given points $\boldsymbol{Y}$. For this reason points evaluated using either the surrogate or expensive model may be compared directly. There exist also a number of fast implementations of the NN algorithm, see for example [8]. An obvious refinement of the nearest neighbor regression model is to weight the contribution of κ neighbors according to their distance to the query point $\boldsymbol{x}_i$. However, in a previous study [10] the simple NN regression model was found to be superior when a wider range of test functions were considered.

C. Approximate Ranking

The individuals evaluated using the expensive model in previous generations are stored in the set $\boldsymbol{Y}$. The size of the set is limited to $\ell = \lambda$, where a new point replaces the oldest point in $\boldsymbol{Y}$. Given a completely new set of λ search points (offspring) at some generation, which should be evaluated using the expensive model? A number of different sampling strategies were investigated in [10] and it was concluded that the best approach is to evaluate the highest ranked individual according to the surrogate. Once evaluated this new point is added to the set $\boldsymbol{Y}$ and hence the surrogate improved. If now the offspring are re-ranked and the new parent set remains the same as before the improvement, then no further offspring need to be evaluated using the expensive model. Often only a few expensive function evaluation are needed at any generation. However, there are cases where the surrogate

approximate: $\hat{f}(\boldsymbol{x}'_k)$, $\hat{\phi}(\boldsymbol{x}'_k), k = 1 \ldots, \lambda$
rank and determine the parent set $\boldsymbol{X}_1 \equiv \{\boldsymbol{x}'_{i:\lambda}\}_{i=1}^{\mu}$
$\boldsymbol{y}_j \leftarrow \text{argmin}_{\boldsymbol{x}'_{i:\lambda}}\ i$ for $\boldsymbol{x}'_{i:\lambda} \notin \boldsymbol{Y}$, $j \leftarrow j+1$ *(approximated best point)*
evaluate: $f(\boldsymbol{y}_j)$, $\phi(\boldsymbol{y}_j)$ *(expensive model evaluation)*
for $i := 2$ to $\lambda/4$ **do** *(attempt for at most a quarter the population)*
  re-approximate: $\hat{f}(\boldsymbol{x}'_k)$, $\hat{\phi}(\boldsymbol{x}'_k), k = 1 \ldots, \lambda$
  determine new parent set $\boldsymbol{X}_i \equiv \{\boldsymbol{x}'_{i:\lambda}\}_{i=1}^{\mu}$
  if $\boldsymbol{X}_{i-1} \neq \boldsymbol{X}_i$ **do** *(the parent set has changed)*
    $\boldsymbol{y}_j \leftarrow \text{argmin}_{\boldsymbol{x}'_{i:\lambda}}\ i$ for $\boldsymbol{x}'_{i:\lambda} \notin \boldsymbol{Y}$, $j \leftarrow j+1$
    evaluate: $f(\boldsymbol{y}_j)$, $\phi(\boldsymbol{y}_j)$
  else *(parent set remains unchanged)*
    break *(exit for loop)*
  od
od

Fig. 3. The *approximate ranking* procedure where initially $Y \neq \emptyset$.

always results in a changed parent set and as a consequence all offspring must be evaluated using the expensive function call. In [10] this occurred mostly at the start of a run and in some rare cases towards the end. The more rugged the local fitness landscape the greater the number of fitness evaluations needed. In the version presented here, however, it is decided to set an upper limit on the number of expensive function evaluations per generation. This limit is set to a quarter the population size $\lambda/4$. In this case the ranking will results in a less accurate approximation of the parent set than that described in [10].

From the evolutionary algorithm's perspective as long as a good approximate estimate of the parent set is found there is no need to call the expensive model. Therefore, the simple heuristic proposed is that the surrogate model is approximately correct as long as the parent set does not change when the surrogate model is improved. As a result the approximate ranking method, shown in fig. 3, is used to determine indirectly the number expensive fitness evaluations needed at any given generation. The maximum number of FEs per generation is $\lambda/4$ and the minimum number is 1.

IV. Experimental Study

In the following experiment 24 nonlinear programming problems from the literature, as described in [7], are studied. The experimental setup is as requested by the WCCI'06 special session on *Constrained Real-Parameter Optimization*, see [7]. All experimental runs are terminated after 5×10^5 function evaluations (FEs). The complete search algorithm is described, along with its default parameter settings, in figs. 1, 2 and 3 (for the approximation). Algorithmic complexity [7] for the $(30, 200)$ SRES is $T1 = 0.023$, $T2 = 0.288$, and $(T1 - T1)/T1 = 11.370$ and for the $(60, 400)$ SRES, $T2 = 0.454$ and $(T1 - T1)/T1 = 18.467$. The approximate SRES (ASRES) has a higher complexity which varies during search.

The search is successful in locating the best known feasible solution, $f(\boldsymbol{x}^*)$, when

$$f(\boldsymbol{x}_{best}) - f(\boldsymbol{x}^*) < 0.0001.$$

Figures 4, 5, 6 and 7 show how this error decreases in terms of the number of function evaluations (FES) for the median run and all 24 test problems using the $(60, 400)$ SRES. The figures also include the constraint violation measure $\bar{v}$, defined by

$$\bar{v} = \Big(\sum_{j=1}^{n_g} g_j^+(\boldsymbol{x}) + \sum_{k=1}^{n_h} h_k^+(\boldsymbol{x}) \Big) / (n_g + n_h) \qquad (6)$$

The success rate and performance for the SRES runs is given in table I. The success performance measure is computed as follows,

mean(FES for a successful run) $\times$ 25 / # of successful runs.

The feasibility rate is also presented in this table along with summary statistics for the number of FES needed for successful runs. The SRES fails to find feasible solutions to problems g20 and g22. A feasible solution to test problem g20 is still lacking in the literature and the penalty function

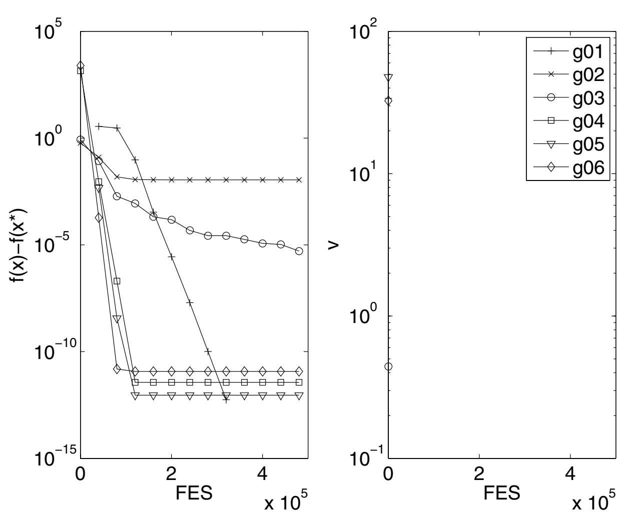

Fig. 4. Convergence graph for problems 1–6.

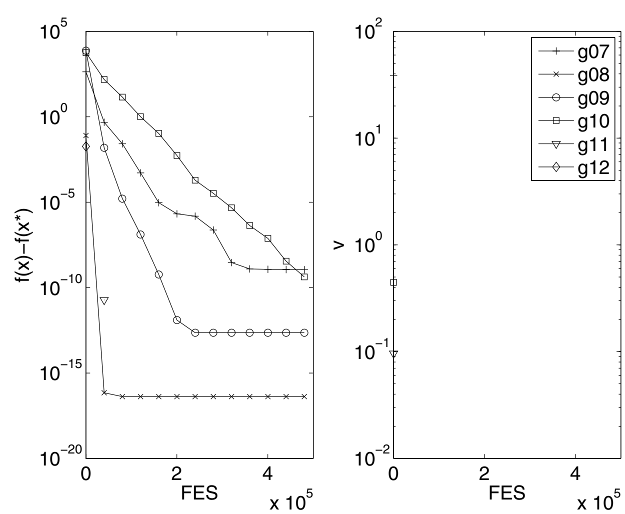

Fig. 5. Convergence graphs for problems 7–12.

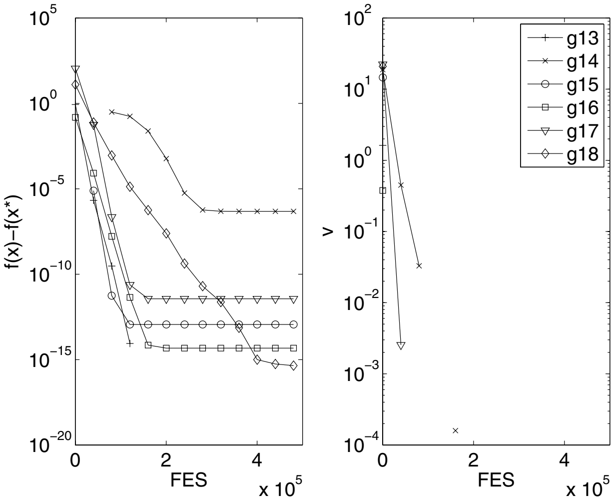

Fig. 6. Convergence graphs for problems 13–18.

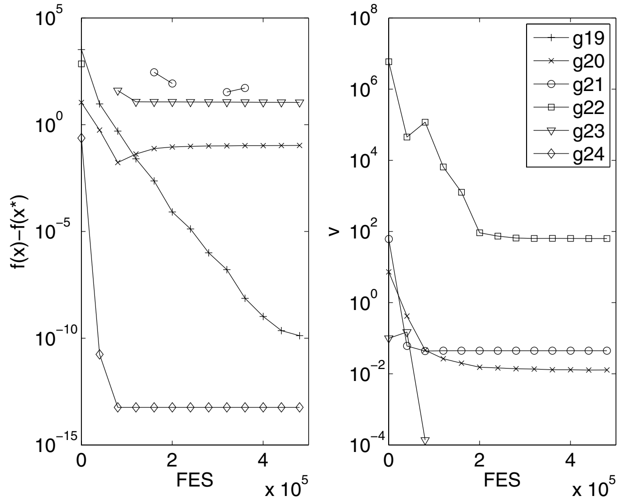

Fig. 7. Convergence graphs for problems 19–24.

approach used here seems to have problems with $g22$. Test function $g22$ has many equality constraints. A larger pressure on the constraint violations (i.e. $P_f < 0.45$) is needed for test function $g21$ in order to obtain consistently feasible solutions.

When the experiment is repeated using the $(60, 400)$ ASRES the number of FES needed is reduced significantly. The results for ASRES are given in table II. However, the success rate is also decreased for a number of problems. Nevertheless, the quality of solutions found are some. This can be seen by comparing the summary statistics for the best feasible solutions given in tables III and IV for the SRES and ASRES respectively.

Intermediate summary statistics for the errors achieved for the $(60, 400)$ SRES are presented at 5×10^3, 5×10^4, and 5×10^5 FES in tables V, VI, VII and VIII. The values in parenthesis are the number of constraints violated for the corresponding solution. Also shown are the number of constraints violated by more than 1.0, 0.01 and 0.0001 respectively (denoted by c). Furthermore, the mean constraint violation $\bar{v}$ for the median solution is given.

V. Summary and Conclusion

An approximate evolution strategy using stochastic ranking has been presented. Its aim has been to reduce the number of function evaluations needed without sacrificing the performance of the original SRES search method. In most cases the number of function evaluations were reduced significantly and in few cases the method is unsuccessful.

The results indicate that the SRES still has room for improvement. However, in general an improvement made in efficiency and effectiveness for some problems, whether due to the constraint handling method or search operators, comes at the cost of making some assumptions about the functions being optimized. As a consequence, the likelihood of being trapped in local minima for some other functions may be greater. This is in agreement with the no-free-lunch theorem [14].

References

[1] J.-F. M. Barthelemy and R.T. Haftka. Approximation concepts for optimum structural design – A review. *Stuctural Optimization*, 5:129–144, 1993.

[2] J.R. Banga C.G. Moles, P. Mendes. Parameter estimation in biochemical pathways: A comparison of global optimization methods. *Genome Research*, 13(11):2467–2474, November 2003.

[3] X. Ji and Y. Xu. libSRES: a C library for stochastic ranking evolution strategy for parameter estimation. *Bioinformatics*, 22(1), January 2006.

[4] Y. Jin. A comprehensive survey of fitness approximation in evolutionary computation. *Soft Computing*, 2003.

[5] Y. Jin. A comprehensive survey of fitness approximation in evolutionary computation. *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, 9(1):3–12, January 2005.

[6] Y. Jin, M. Olhofer, and B. Sendhoff. A framework for evolutionary optimization with approximate fitness functions. *IEEE Transactions on Evolutionary Computation*, 6(5), October 2002.

[7] J. J. Liang, T. P. Runarsson, E. Mezura-Montes, M. Clerc, P. N. Suganthan, C. A. Coello Coello, and K. Deb. Problem definitions and evaluation criteria for the CEC 2006 special session on constrained real-parameter optimization. Technical report, Nanyang Technological University, 2006.

[8] J. McNames. A fast neartest neighbor algorithm based on a principle axis search tree. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(9), September 2001.

[9] C.G. Moles, J.R. Banga, and K. Keller. Solving nonconvex climate control problems: pitfalls and algorithm performances. *Applied Soft Computing*, 5(1):35–44, December 2004.

[10] T. P. Runarsson. Constrained evolutionary optimization by approximate ranking and surrogate models. In *Parallel Problem Solving from Nature VII (PPSN-2004)*, volume 3242 of *LNCS*, pages 401–410. Springer Verlag, 2004.

[11] T. P. Runarsson and X. Yao. Stochastic ranking for constrained evolutionary optimization. *IEEE Transactions on Evolutionary Computation*, 4(3):284–294, September 2000.

[12] T. P. Runarsson and X. Yao. Search biases in constrained evolutionary optimization. *IEEE Transactions on System, Man, and Cybernetics: Part C*, 35(2):233–243, 2005.

[13] T. W. Simpson, J. Peplinski, P. N. Koch, and J. K. Allen. On the use of statistics in design and the implications for deterministic computer experiments. In *Design Theory and Methodology - DTM'97*, number DETC97/DTM-3881, Sacramento, CA, 1997. ASME.

[14] D. H. Wolpert and W. G. Macready. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82, 1997.

[15] S. Yesilyurt and A. T. Patera. Surrogates for numerical simulations; optimization of eddy-promoter heat exchangers. *Comp. Methods Appl. Mech. Engr.*, 121:231–257, 1995.

Prob.	Best	Median	Worst	Mean	Std	**Feasible Rate**	**Success Rate**	**Success Performance**
g01	141200	162800	182800	162976	10927	100%	100%	162976
g02	288800	321200	357200	322400	34215	100%	12%	2686666
g03	196000	235200	312400	238736	30641	100%	100%	238736
g04	54000	57600	60800	57552	1898	100%	100%	57552
g05	46000	50000	53600	50464	1853	100%	100%	50464
g06	41200	48400	53600	47840	3061	100%	100%	47840
g07	122400	135600	273600	141104	28916	100%	100%	141104
g08	2800	4800	6800	4720	901	100%	100%	4720
g09	64400	72000	82000	72672	4166	100%	100%	72672
g10	236800	276000	304400	273296	16432	100%	100%	273296
g11	8000	13200	18800	12736	2647	100%	100%	12736
g12	6800	15600	23600	14912	3632	100%	100%	14912
g13	29200	34000	38000	34720	2206	100%	100%	34720
g14	201200	216800	267600	220417	15402	100%	92%	239548
g15	28800	33600	38400	33520	2218	100%	100%	33520
g16	34800	39200	46800	40336	2999	100%	100%	40336
g17	56800	59600	63600	59856	1928	100%	100%	59856
g18	104000	119600	142400	120383	9888	100%	96%	125399
g19	192400	212000	441600	229600	56007	100%	92%	249565
g20	–	–	–	–	–	0%	0%	–
g21	158400	232600	306800	232600	104934	16%	8%	2907500
g22	–	–	–	–	–	0%	0%	–
g23	265600	317600	478000	344700	95728	100%	16%	2154375
g24	10800	14000	16400	13616	1639	100%	100%	13616

TABLE I

SUCCESS RATE, FEASIBLE RATE AND SUCCESS PERFORMANCE FOR THE EVOLUTION STRATEGY USING STOCHASTIC RANKING.

Prob.	Best	Median	Worst	Mean	Std	**Feasible Rate**	**Success Rate**	**Success Performance**
g01	31698	34911	42313	35406	2581	100%	100%	35406
g02	–	–	–	–	–	100%	0%	–
g03	–	–	–	–	–	100%	0%	–
g04	13616	15041	16541	15104	706	100%	100%	15104
g05	17526	19188	21087	19281	885	100%	100%	19281
g06	7875	9542	11765	9603	918	100%	100%	9603
g07	75890	76782	77674	76782	918	100%	8%	959775
g08	603	1007	1527	1027	229	100%	100%	1027
g09	26972	30241	36534	30618	2535	100%	100%	30618
g10	–	–	–	–	–	100%	0%	–
g11	1701	2800	4012	2792	550	100%	100%	2792
g12	1476	3101	3932	2996	584	100%	100%	2996
g13	9936	11459	12402	11292	624	100%	84%	13422
g14	84631	92820	101010	92820	11581	100%	8%	1160256
g15	7015	8473	9984	8519	620	100%	100%	8519
g16	13638	16341	19416	16179	1470	100%	100%	16179
g17	17817	19296	61561	21491	9743	100%	76%	28277
g18	36502	41015	49801	40840	3214	100%	92%	44391
g19	–	–	–	–	–	100%	0%	–
g20	–	–	–	–	–	0%	0%	–
g21	–	–	–	–	–	0%	0%	–
g22	–	–	–	–	–	0%	0%	–
g23	–	–	–	–	–	64%	0%	–
g24	2909	3648	4420	3638	391	100%	100%	3638

TABLE II

SUCCESS RATE, FEASIBLE RATE AND SUCCESS PERFORMANCE FOR THE APPROXIMATE EVOLUTION STRATEGY USING STOCHASTIC RANKING.

TABLE III

SUMMARY STATISTICS FOR THE BEST SOLUTIONS FOUND FOR THE 25 INDEPENDENT RUNS FOR THE EVOLUTION STRATEGY USING STOCHASTIC RANKING.

fcn	optimal	best	median	mean	st. dev.	worst
g01	-15.000	-15.000	-15.000	-15.000	$0.0E+00$	-15.000
g02	-0.804	-0.804	-0.793	-0.788	$1.3E-02$	-0.746
g03	-1.001	-1.000	-1.000	-1.000	$6.4E-06$	-1.000
g04	-30665.539	-30665.539	-30665.539	-30665.539	$0.0E+00$	-30665.539
g05	5126.497	5126.497	5126.497	5126.497	$2.0E-12$	5126.497
g06	-6961.814	-6961.814	-6961.814	-6961.814	$1.9E-12$	-6961.814
g07	24.306	24.306	24.306	24.306	$1.1E-07$	24.306
g08	-0.096	-0.096	-0.096	-0.096	$0.0E+00$	-0.096
g09	680.630	680.630	680.630	680.630	$4.6E-13$	680.630
g10	7049.248	7049.248	7049.248	7049.248	$1.2E-10$	7049.248
g11	0.750	0.750	0.750	0.750	$1.1E-16$	0.750
g12	-1.000	-1.000	-1.000	-1.000	$0.0E+00$	-1.000
g13	0.054	0.054	0.054	0.054	$2.8E-17$	0.054
g14	-47.765	-47.765	-47.765	-47.765	$1.3E-04$	-47.764
g15	961.715	961.715	961.715	961.715	$4.6E-13$	961.715
g16	-1.905	-1.905	-1.905	-1.905	$6.8E-16$	-1.905
g17	8853.540	8853.540	8853.540	8853.540	$5.6E-12$	8853.540
g18	-0.866	-0.866	-0.866	-0.851	$7.4E-02$	-0.495
g19	32.656	32.656	32.656	32.656	$2.9E-04$	32.657
g20	0.097	–	–	–	–	–
g21	193.725	193.725	324.703	291.958	$6.5E+01$	324.703
g22	236.431	–	–	–	–	–
g23	-400.055	-400.055	-388.963	-313.066	$1.3E+02$	-12.420
g24	-5.508	-5.508	-5.508	-5.508	$0.0E+00$	-5.508

TABLE IV

SUMMARY STATISTICS FOR THE BEST SOLUTIONS FOUND FOR THE 25 INDEPENDENT RUNS FOR THE APPROXIMATE EVOLUTION STRATEGY USING STOCHASTIC RANKING.

fcn	optimal	best	median	mean	st. dev.	worst
g01	-15.000	-15.000	-15.000	-15.000	$0.0E+00$	-15.000
g02	-0.804	-0.739	-0.700	-0.697	$2.1E-02$	-0.657
g03	-1.001	-0.998	-0.995	-0.995	$1.6E-03$	-0.992
g04	-30665.539	-30665.539	-30665.539	-30665.539	$0.0E+00$	-30665.539
g05	5126.497	5126.497	5126.497	5126.497	$2.3E-12$	5126.497
g06	-6961.814	-6961.814	-6961.814	-6961.814	$1.9E-12$	-6961.814
g07	24.306	24.306	24.307	24.307	$1.6E-03$	24.312
g08	-0.096	-0.096	-0.096	-0.096	$5.1E-17$	-0.096
g09	680.630	680.630	680.630	680.630	$2.2E-06$	680.630
g10	7049.248	7049.408	7053.856	7061.087	$1.9E+01$	7126.958
g11	0.750	0.750	0.750	0.750	$1.1E-16$	0.750
g12	-1.000	-1.000	-1.000	-1.000	$0.0E+00$	-1.000
g13	0.054	0.054	0.054	0.116	$1.4E-01$	0.439
g14	-47.765	-47.765	-47.763	-47.762	$3.3E-03$	-47.753
g15	961.715	961.715	961.715	961.715	$4.6E-13$	961.715
g16	-1.905	-1.905	-1.905	-1.905	$1.1E-15$	-1.905
g17	8853.540	8853.540	8853.540	8853.717	$8.5E-01$	8857.816
g18	-0.866	-0.866	-0.866	-0.851	$5.3E-02$	-0.675
g19	32.656	32.665	32.790	32.829	$1.6E-01$	33.242
g20	0.097	–	–	–	–	–
g21	193.725	–	–	–	–	–
g22	236.431	–	–	–	–	–
g23	-400.055	-348.816	-106.585	-106.937	$1.4E+02$	203.455
g24	-5.508	-5.508	-5.508	-5.508	$0.0E+00$	-5.508

TABLE V

SRES PERFORMANCE VALUES FOR PROBLEMS 1-6.

FEs		g01	g02	g03	g04	g05	g06
5×10^3	Best	7.18E+000(0)	3.78E-001(0)	8.25E-001(0)	1.00E+002(0)	4.36E+002(1)	7.70E+001(0)
	Median	1.03E+001(0)	4.82E-001(0)	9.80E-001(1)	2.18E+002(0)	-4.29E+002(2)	2.84E+002(0)
	Worst	1.12E+001(1)	4.99E-001(0)	8.50E-001(1)	1.97E+002(0)	-4.98E+000(3)	3.48E+002(0)
	c	0,0,0	0,0,0	0,0,1	0,0,0	2,2,2	0,0,0
	$\overline{v}$	0.00E+000	0.00E+000	3.50E-004	0.00E+000	2.19E+001	0.00E+000
	Mean	1.05E+001	4.76E-001	8.83E-001	2.24E+002	-2.18E+002	3.70E+002
	Std	1.6E+000	3.1E-002	1.3E-001	6.0E+001	4.5E+002	2.9E+002
5×10^4	Best	2.97E+000(0)	3.68E-002(0)	1.24E-002(0)	3.56E-004(0)	4.40E-005(0)	1.10E-006(0)
	Median	4.49E+000(0)	7.34E-002(0)	1.95E-002(0)	7.23E-004(0)	8.43E-005(0)	3.29E-005(0)
	Worst	5.54E+000(0)	8.05E-002(0)	2.31E-002(0)	6.43E-004(0)	4.47E-005(0)	3.24E-004(0)
	c	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0
	$\overline{v}$	0.00E+000	0.00E+000	0.00E+000	0.00E+000	0.00E+000	0.00E+000
	Mean	4.45E+000	8.03E-002	2.59E-002	9.76E-004	1.33E-004	9.69E-005
	Std	6.5E-001	3.3E-002	1.4E-002	6.0E-004	1.0E-004	1.9E-004
5×10^5	Best	0.00E+000(0)	5.41E-005(0)	4.41E-007(0)	3.64E-012(0)	0.00E+000(0)	1.18E-011(0)
	Median	0.00E+000(0)	1.10E-002(0)	5.14E-006(0)	3.64E-012(0)	9.09E-013(0)	1.18E-011(0)
	Worst	3.55E-015(0)	1.10E-002(0)	2.87E-006(0)	3.64E-012(0)	0.00E+000(0)	1.18E-011(0)
	c	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0
	$\overline{v}$	0.00E+000	0.00E+000	0.00E+000	0.00E+000	0.00E+000	0.00E+000
	Mean	1.42E-016	1.53E-002	6.80E-006	3.64E-012	7.64E-013	1.18E-011
	Std	7.1E-016	1.3E-002	6.4E-006	0.0E+000	7.3E-013	0.0E+000

TABLE VI

SRES PERFORMANCE VALUES FOR PROBLEMS 7-12.

FEs		g07	g08	g09	g10	g11	g12
5×10^3	Best	1.54E+002(0)	2.46E-006(0)	5.70E+001(0)	6.73E+003(0)	3.61E-004(0)	1.43E-004(0)
	Median	5.70E+002(0)	5.55E-005(0)	1.89E+002(0)	7.31E+003(2)	5.22E-002(0)	6.17E-003(0)
	Worst	2.43E+002(2)	9.00E-006(0)	5.70E+001(0)	1.28E+004(2)	1.34E-002(1)	8.04E-003(0)
	c	0,0,0	0,0,0	0,0,0	0,2,2	0,0,0	0,0,0
	$\overline{v}$	0.00E+000	0.00E+000	0.00E+000	8.60E-002	0.00E+000	0.00E+000
	Mean	3.93E+002	6.75E-005	1.95E+002	6.29E+003	5.37E-002	4.20E-003
	Std	2.3E+002	6.0E-005	9.0E+001	4.3E+003	7.4E-002	3.9E-003
5×10^4	Best	7.82E-002(0)	4.16E-017(0)	1.02E-003(0)	5.91E+001(0)	1.09E-014(0)	0.00E+000(0)
	Median	1.79E-001(0)	5.55E-017(0)	2.54E-003(0)	1.17E+002(0)	6.15E-013(0)	0.00E+000(0)
	Worst	2.78E-001(0)	5.55E-017(0)	2.51E-003(0)	3.22E+002(0)	3.61E-013(0)	0.00E+000(0)
	c	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0
	$\overline{v}$	0.00E+000	0.00E+000	0.00E+000	0.00E+000	0.00E+000	0.00E+000
	Mean	1.83E-001	5.61E-017	2.90E-003	1.30E+002	1.89E-012	0.00E+000
	Std	5.0E-002	9.4E-018	1.1E-003	6.0E+001	3.9E-012	0.0E+000
5×10^5	Best	-1.07E-014(0)	4.16E-017(0)	0.00E+000(0)	0.00E+000(0)	0.00E+000(0)	0.00E+000(0)
	Median	1.01E-009(0)	4.16E-017(0)	2.27E-013(0)	5.00E-011(0)	0.00E+000(0)	0.00E+000(0)
	Worst	4.08E-007(0)	5.55E-017(0)	3.41E-013(0)	5.00E-011(0)	0.00E+000(0)	0.00E+000(0)
	c	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0
	$\overline{v}$	0.00E+000	0.00E+000	0.00E+000	0.00E+000	0.00E+000	0.00E+000
	Mean	5.27E-008	4.94E-017	2.14E-013	8.32E-011	4.44E-018	0.00E+000
	Std	1.1E-007	9.0E-018	1.2E-013	1.2E-010	2.2E-017	0.0E+000

TABLE VII

SRES PERFORMANCE VALUES FOR PROBLEMS 13-18.

FEs		g13	g14	g15	g16	g17	g18
5×10^3	Best	5.95E-001(1)	-1.59E+002(3)	3.41E-001(0)	7.63E-002(0)	7.94E+001(1)	-2.57E+000(9)
	Median	3.66E-001(1)	-3.73E+002(3)	-1.09E+000(1)	1.97E-001(0)	1.83E+001(4)	-1.07E+001(10)
	Worst	-5.39E-002(2)	-4.42E+002(3)	-6.81E-001(2)	1.28E-001(0)	-5.49E+000(3)	-1.21E+001(11)
	c	1,1,1	3,3,3	1,1,1	0,0,0	4,4,4	10,10,10
	$\overline{v}$	5.33E-001	1.04E+001	5.95E-001	0.00E+000	1.08E+001	1.03E+001
	Mean	3.86E-001	-3.68E+002	-6.06E-001	2.07E-001	5.35E+001	-9.10E+000
	Std	3.0E-001	7.5E+001	1.8E+000	6.4E-002	4.2E+001	6.2E+000
5×10^4	Best	1.03E-007(0)	2.19E+000(3)	2.70E-008(0)	3.31E-006(0)	5.77E-004(0)	2.38E-002(0)
	Median	2.32E-007(0)	-5.14E+000(3)	1.66E-007(0)	1.46E-005(0)	4.47E-003(1)	5.63E-002(0)
	Worst	5.66E-007(0)	4.03E+000(3)	1.85E-007(0)	1.10E-005(0)	2.43E-003(3)	6.50E-002(0)
	c	0,0,0	0,3,3	0,0,0	0,0,0	0,0,1	0,0,0
	$\overline{v}$	0.00E+000	2.38E-001	0.00E+000	0.00E+000	1.17E-004	0.00E+000
	Mean	2.55E-007	-5.77E+000	2.86E-007	1.71E-005	1.60E+001	7.24E-002
	Std	1.2E-007	8.5E+000	4.4E-007	9.1E-006	3.7E+001	7.9E-002
5×10^5	Best	-9.71E-017(0)	2.13E-013(0)	-1.14E-013(0)	3.77E-015(0)	3.64E-012(0)	2.22E-016(0)
	Median	-7.63E-017(0)	4.79E-007(0)	1.14E-013(0)	4.66E-015(0)	3.64E-012(0)	4.44E-016(0)
	Worst	-7.63E-017(0)	7.91E-006(0)	0.00E+000(0)	3.77E-015(0)	3.64E-012(0)	4.44E-016(0)
	c	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0
	$\overline{v}$	0.00E+000	0.00E+000	0.00E+000	0.00E+000	0.00E+000	0.00E+000
	Mean	-8.55E-017	3.76E-005	5.91E-014	4.59E-015	3.64E-012	1.49E-002
	Std	1.1E-017	1.3E-004	9.4E-014	2.5E-016	0.0E+000	7.4E-002

TABLE VIII

SRES PERFORMANCE VALUES FOR PROBLEMS 19-24.

FEs		g19	g20	g21	g22	g23	g24
	Best	1.95E+002(0)	4.91E+000(16)	-5.14E+000(1)	-2.28E+002(11)	-7.07E+002(3)	1.64E-003(0)
	Median	4.00E+002(0)	5.21E+000(17)	2.39E+002(2)	1.87E+002(10)	-2.12E+002(2)	9.62E-003(0)
5×10^3	Worst	8.02E+002(0)	6.20E+000(13)	-5.14E+000(5)	3.38E+003(12)	-6.93E+002(5)	1.59E-002(0)
	c	0,0,0	2,17,17	1,2,2	9,10,10	2,2,2	0,0,0
	$\overline{v}$	0.00E+000	3.91E+000	7.11E+000	2.92E+006	1.48E+000	0.00E+000
	Mean	4.30E+002	6.21E+000	3.74E+001	1.09E+003	-8.41E+002	1.10E-002
	Std	1.2E+002	1.0E+000	1.6E+002	2.6E+003	3.4E+002	6.7E-003
	Best	2.05E+000(0)	1.68E-001(15)	1.47E+002(0)	-2.36E+002(10)	8.14E+001(2)	5.86E-014(0)
	Median	4.43E+000(0)	2.57E-001(15)	-1.84E+002(1)	-2.33E+002(10)	1.64E+002(6)	9.15E-014(0)
5×10^4	Worst	7.44E+000(0)	1.68E-001(16)	1.54E+002(3)	-2.18E+002(16)	3.28E+001(4)	2.05E-013(0)
	c	0,0,0	1,14,15	0,1,1	9,10,10	0,6,6	0,0,0
	$\overline{v}$	0.00E+000	8.59E-002	9.56E-004	8.12E+005	3.01E-001	0.00E+000
	Mean	4.50E+000	2.72E-001	-8.69E+001	-2.08E+002	-1.38E+002	1.17E-013
	Std	1.2E+000	6.7E-002	1.5E+002	6.5E+001	3.7E+002	1.2E-013
	Best	4.90E-013(0)	1.07E-001(16)	-2.84E-014(0)	-2.36E+002(7)	8.62E-009(0)	5.77E-014(0)
	Median	1.28E-010(0)	1.07E-001(18)	-1.26E+002(3)	-2.36E+002(11)	1.11E+001(0)	5.77E-014(0)
5×10^5	Worst	1.47E-003(0)	1.08E-001(18)	1.26E+002(1)	-2.36E+002(5)	6.67E-001(0)	5.77E-014(0)
	c	0,0,0	0,9,18	0,1,3	3,11,11	0,0,0	0,0,0
	$\overline{v}$	0.00E+000	1.25E-002	1.29E-004	5.96E+001	0.00E+000	0.00E+000
	Mean	7.08E-005	1.07E-001	-5.75E+000	-2.36E+002	8.70E+001	5.77E-014
	Std	2.9E-004	3.4E-004	1.6E+002	0.0E+000	1.3E+002	0.0E+000