Yugoslav Journal of Operations Research
24 (2014), Number 3, 307-319
DOI: 10.2298/YJOR140121035K

AN ADAPTIVE ES WITH A RANKING BASED CONSTRAINT HANDLING STRATEGY

ALI OSMAN KUSAKCI

Faculty of Engineering and Natural Sciences,
International University of Sarajevo, Bosnia and Herzegovina
akusakci@ius.edu.ba

MEHMET CAN

Faculty of Engineering and Natural Sciences,
International University of Sarajevo, Bosnia and Herzegovina
mcan@ius.edu.ba

Received: January 2014 / Accepted: October 2014

Abstract: To solve a constrained optimization problem, equality constraints can be used to eliminate a problem variable. If it is not feasible, the relations imposed implicitly by the constraints can still be exploited. Most conventional constraint handling methods in Evolutionary Algorithms (EAs) do not consider the correlations between problem variables imposed by the constraints. This paper relies on the idea that a proper search operator, which captures mentioned implicit correlations, can improve performance of evolutionary constrained optimization algorithms. To realize this, an Evolution Strategy (ES) along with a simplified Covariance Matrix Adaptation (CMA) based mutation operator is used with a ranking based constraint-handling method. The proposed algorithm is tested on 13 benchmark problems as well as on a real life design problem. The outperformance of the algorithm is significant when compared with conventional ES-based methods.

Keywords: Constrained Optimization, Evolution Strategies, Covariance Matrix Adaptation.

MSC: 65K10, 90C30, 90C59.

1. INTRODUCTION

Population based approaches inspired by nature have been widely applied in various scientific domains due to their global search ability and very precise approximation of the global solution [1].

Constrained optimization problems (COPs) of non-convex character requirescomplex optimization methods[2], [3]. While designing new algorithms, the focal point is the constraint handling strategy[4]–[6].The equality constraints in a COP impose a strict relation between problem variables. Mostly, they can be exploited to determine any unknown variable,in terms of other variables in the equation. However, sometimes,their direct use may be impossible or computationally expensive. Besides the fact that inequality constraints do not allow a direct elimination of the problem variables, they also establish relations between the unknowns in a less strict manner than the equality constraints [7].

This paper is an extension of the work in [7], which relies on the idea of exploiting the imposed relations between problem variables in an indirect way even if they do not mostly allow a direct use. This idea has been overlooked by most conventional constrained optimization methods. To obtain the desired behavior, a simplified version of Covariance Matrix Adaptation (CMA) based mutation strategy[8] is employed along with a ranking mechanism proposed by Ho and Shimizu[5]. By using a correlated mutation strategy, we expect that the algorithm elicits the distribution of promising solutions along the constraint boundaries and generates the next candidate solutions accordingly.

The remainder of the paper is organized as follows: in Section 2, the basics of the constrained optimization and related works are shortly discussed. Section 3 describes the proposed method in details while the results of the study will be demonstrated in Section 4. Lastly, Section 5 concludes the study.

2. CONSTRAINED OPTIMIZATION

2.1. Background

An *n* dimensional COP consists of two parts: (i) objective function, (ii) inequality and equality constraints. Without loss of generality, it is formulated as:

Minimize $$f(\vec{x}),\ \vec{x} = [x_1, \dots, x_n]^T\ \in F \subseteq S \subseteq \mathbb{R}^n, \tag{1}$$

subject to

$$g_i(\vec{x}) \le 0,\ i = 1, \dots, r, and\ h_j(\vec{x}) = 0, j = r + 1, \dots, m \tag{2}$$

where $S = \{\vec{x} \in \mathbb{R}^n | l_k \le x_k \le u_k, k = 1, \dots, n\}$,$F = \{\vec{x} \in S \big| g_i(\vec{x}) \le 0\ and\ h_j(\vec{x}) = 0\}$, $\vec{x}$ is solution vector $\vec{x} = [x_1, \dots, x_n]^T$, *r* is the number of inequality, and *m-r* is the number of equality constraints. The equality constraints are usually converted into inequalities by adding a small tolerance $\varepsilon > 0$, and an equality constraint *j* is rephrased as $|h_j(\vec{x})| - \varepsilon \le 0$. The same approach is used in this work, where $\varepsilon_{min} = 1.0e - 4$.

Accordingly, for any constraint *i*, the violation$v_i(\vec{x})$is defined as:

$$v_i(\vec{x}) = \begin{cases} max(0, g_i(\vec{x})), if\ 1 \leq i \leq r \\ max\ (0, |h_i(\vec{x})| - \varepsilon), \quad if\ r + 1 \leq i \leq m \end{cases}. \tag{3}$$

2.2. Related Work

Various techniques have been proposed to deal with COPs. Extended surveys are given in [9], [10]. The constrained optimization evolutionary algorithms (COEAs) can be classified in the following four categories: feasibility maintenance, penalty function, separation of constraint violation and objective value, multi-objective optimization evolutionary algorithms (MOEA)[11].

Approaches based on the maintenance of feasibility status of an individual aim to transfer the individual into the feasible domain. Repairing the infeasible individuals, and homomorphous mapping are two popular methods[12]. The methods based on penalty functions are the most popular approaches thanks to low complexity of implementation[4]. They rely on penalizing infeasible individuals. The third class separates the objective value and the constraint violation. Deb[13] suggested a constraint handling method with three rules, called as *three feasibility rules*, to determine the winner of the tournament among two individuals. Runarsson and Yao[6] developed stochastic ranking (SR) which is proved to be efficient and highly competitive with other methods[9].MOEA algorithms have been also widely applied in constrained optimization domain[14], where the constraints are handled as additional objectives to be optimized.

3. PROPOSED ALGORITHM

3.1. Ho and Schimizu's Ranking Approach

Among the constraint handling methods mentioned above, ranking based approaches are preferred thanks to the small number of control parameters required. Based on this fact, we employ a ranking based method by Ho and Shimizu [5]incorporated into an evolution strategy (ES).The method scales the objective function (1) and the constraint terms (2) to the same orders of magnitude so that the aggregation of both terms into one would be possible.

While ranking an individual, $\vec{x}$, a COP has three numerical properties to be considered:(i) $f(\vec{x})$, the objective function value; (ii) $viol(\vec{x}) = \sum_{i=1}^{m} v_i(\vec{x})$, sum of the constraint violations; (iii) $vcon(\vec{x})$, $0 \leq vcon(\vec{x}) \leq m$, number of violated constraints[5].

Since these three terms are of different order of magnitudes, a proper combination method must be employed. The issue is solved by [5] with a simple ranking method, where three independent ranking lists are created based on the three numerical properties, $f(\vec{x}), viol(\vec{x})$, and $vcon(\vec{x})$, respectively. For each numerical property, the individual having the smallest value is assigned a ranking of *one* while the ranking of the individual with the next smallest value is *two*, and so on. We refer Ho and Shimizu's algorithm as three ranking lists-ES (3RL-ES).

Kusakci and Can [7] propose the following ranking scheme:

$$R = \begin{cases} R_{viol}, & \textit{no feasible solution exists} \\ R_f + R_{viol}, & \textit{the solution is infeasible} \\ R_f + 1, & \textit{the solution is feasible.} \end{cases} \quad (4)$$

Equation (5) implies that the ranking of an individual is computed with R_{viol} if all individuals are infeasible. If there is at least one feasible individual, the ranking is done with $R_f + 1$, or $R_f + R_{viol}$, depending on the feasibility status of that particular individual. $R_f + 1$ implies that $R_{viol} = 1$ for a feasible solution [5].

3.2. Simplified Covariance Matrix Adaptation (CMA)

Hansen and Ostermeier[8] suggested a mutation operator, CMA, which exploits the correlations between variables instead of gradient information to navigate the population in the search space. Given a population of μ individuals, CMA relies on eigen decomposition of covariance matrix C of these μ individuals. A comprehensive tutorial can be found in [15].

A positive definite matrix $\boldsymbol{C} \in \mathbb{R}^{n\times n}$, has an orthonormal basis of eigenvectors, $\boldsymbol{B} = [\boldsymbol{b_1}, \ldots, \boldsymbol{b_n}]$, with corresponding eigenvalues, $d_1^2, \ldots, d_n^2 > 0$. Additionally, the eigendecomposition of $\boldsymbol{C}$ obeys $\boldsymbol{C} = \boldsymbol{BD^2B^T}$, where $\boldsymbol{B}$ is an orthogonal matrix, and the columns of $\boldsymbol{B}$ form an orthonormal basis of eigenvectors. $\boldsymbol{D} = diag(d_1, \ldots, d_n)$ is a diagonal matrix with square roots of the eigenvalues of $\boldsymbol{C}$ as its diagonal elements[15].

If denotes the covariance matrix of the population sampled over a multivariate normal distribution, $\mathcal{N}(\vec{x}_{mean}, C)$, a new search point $\vec{x}_i$ is generated as:

$$\vec{x}_i(g+1) = \vec{x}_{mean}(g) + \sigma(g)\, B\, D\, \mathcal{N}(0, I) \quad (5)$$

where$\mathcal{N}(0, I)$denotes multivariate standard normal distribution with mean zero and standard deviation one.$\sigma(g) > 0$ is ageneration specific step size. Based on equation (6), new search points (individuals) are generated while best performing individuals out of are selected.

The covariance matrix, $\boldsymbol{C(g)}$, is updated with a learning rate, $0 \le c_{cov} \le 1$ as:

$$\boldsymbol{C(g+1)} = (1 - c_{cov})\boldsymbol{C(g)} + c_{cov}\frac{1}{\sigma(g)^2}\sum_{i=1}^{\mu} \alpha_i\, \vec{y}_i \vec{y}_i^{\,T} \quad (6)$$

where$\vec{y}_i = \vec{x}_i - \vec{x}_{mean}(g)$ and $\alpha_i,\ i = 1, \ldots, \mu$, are positive recombination weights. For α_i, Hansen [15] suggests that $\alpha_i > \alpha_{i+1}$ and $\sum_{i=1}^{\mu}\alpha_i = 1$, where fitter individuals are given more weight. The parameter c_{cov} is called learning rate and defines the extent of the information retention between two generations. A reasonable choice forc_{cov} is $2/(n^2 + \sqrt{2})$[15].

The population mean $\vec{x}_{mean}(g+1)$ in next generation is updated as:

$$\vec{x}_{mean}(g+1) = \sum_{i=1}^{\mu}\alpha_i \vec{x}_i. \quad (7)$$

An updating scheme similar to (8) for step size is written as [7];

$$\sigma(g+1) = (1 - c_{cov})\,\sigma(g) \; + \; c_{cov}\frac{1}{\sigma(g)}\left\|\vec{x}_{mean}(g+1) - \vec{x}_{mean}(g)\right\| \quad (8)$$

where $\left\|\vec{x}_{mean}(g+1) - \vec{x}_{mean}(g)\right\|$ stands for Euclidean norm of the difference of the population means in adjacent generations. Namely, the new step size is estimated based on the ratio between the changeof population means over the previous step size.

3.3. Why CMA can be a promising Mutation Operator for COPs?

In CMA, the contour lines of the density function of $\mathcal{N}(\bar{\boldsymbol{x}}, \boldsymbol{C})$ form an (hyper-) ellipsoid while the principal axes of the ellipsoid correspond to the eigenvectors of $\boldsymbol{C}$, and the squared axes lengths correspond to the eigenvalues. In this section, the merit of the described mutation strategy is discussed.

In general, the purpose of employing a mutation operator is to introduce some random information. However, CMA relies exactly on the opposite idea of derandomizing the mutations in a way that the search is guided to more promising directions. Thus, an improvement is more likely based on an individual generated by CMA than the one generated by an uncorrelated simple mutation operator.

Yu and Gen [11] demonstrate this idea by a 2-dimensional example where CMA is compared with an uncorrelated mutation operator. Given the standard deviations in each direction,$\sigma_{x_1} = 1$, and $\sigma_{x_2} = 1/3$, the main part of the possible sampling region is the ellipse with the solid line in Figure 1. If the global optimal solution lies at the point shown by the square in the figure, then the simple mutation method has little chance to catch it due to the limitations of sampling region. To achieve this goal, we can enlarge the standard deviations, e.g., $\sigma_{x_1} = 3$ and $\sigma_{x_2} = 1$. However, now the ellipse with dashed line is rather large so that the chance of sampling the square is still quite small. On the other hand, if we can generate the mutation ellipsoid based on the covariance matrix, the probability of sampling the optimal point is quite high with the correlated normal distribution oriented toward the square in Figure 1.

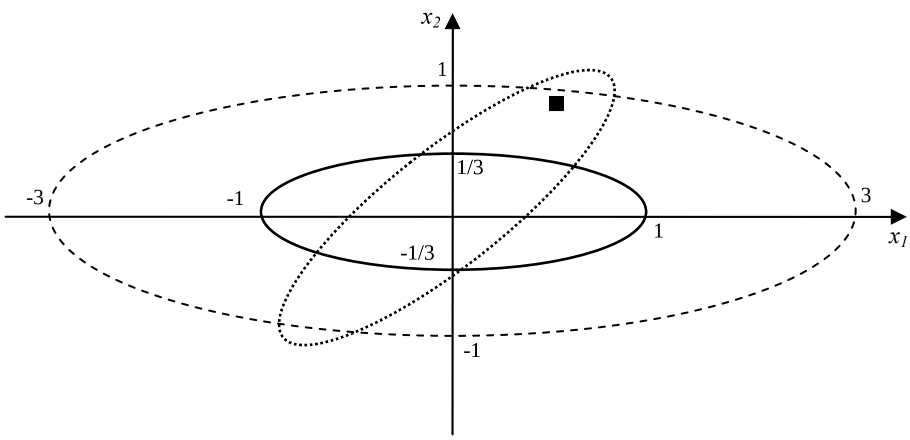

Figure 1: A demonstration of benefit of CMA in a 2-dimensional problem[11]

Similarly, the covariance matrix can capture the relation between variables imposed by a constraint and simulate the constraint boundary. Having this in mind, this property can be exploited while generating new individuals if CMA is combined with a proper constraint handling method.

A short literature survey reveals that CMA has been extensively employed in different subdomains of EC[16]–[18]

3.3. Main Steps of the proposed Algorithm

The proposed algorithm is a (μ, λ)-ES with a ranking based constraint handling strategy formulated in (5). Adaptive tolerance adjustment scheme for equality constraints proposed by[4] is incorporated into the ranking method. The algorithm employs a simplified CMA strategy formulated in (6)-(9). We name the proposed algorithm as Adaptive 2 Ranking Lists-ES (A2RL-ES).

The algorithm has two main stages: (i) initialization, and (ii) main loop. The steps of the algorithm are given below.

Step 1: Tolerance Initialization: Randomly generated μ individuals is used to determine initial tolerance values, $\varepsilon_i(0)$, with

$$\varepsilon_i(0) = \overline{v_i(\vec{x})} \quad i = r+1, \dots, m \tag{9}$$

where $\overline{v_i(\vec{x})}$ stands for average violation of μ individuals for constraint *i*.

Step 2: Covariance Matrix, Step Size, and Learning Rate Initialization: $\boldsymbol{C}(\mathbf{0})$ will be initialized as an $n \times n$-identity matrix. Based on the experiments conducted, a good choice for $\sigma(0)$is $\sigma(0) \approx 1/n$. The proper learning rate c_{cov} for benchmark problems[6] is estimated by [7] as $c_{cov} \approx 2/(n^2 + \sqrt{n})$[8].

Step 3: CMA based Mutation: CMA-like mutation is applied to μ parents to generate λ offspring, as given in (6).

Step 4: Repairing Individuals out of Definition Domain: If an offspring is outside the definition domain S, the offspring is simply set on the respective boundary of S.

Step 5: Updating Tolerances: Based on the comparison of the feasible percentage of the population, $\tau_i(g) \in [0,1]$, for constraint *i*, and the desired ratio, $0 \geq \tau_{tar} \geq 1$, the tolerance value, $\varepsilon_i(g) \in [1.0e-4, \infty)$, is updated with given factors, $\varphi_{eq} \in [1, \infty)$, and $\varphi_{eq2} \in [1, \infty)$, with the following formula[4]:

$$\begin{aligned} \varepsilon_i(g+1) &= \varepsilon_i(g)/\varphi_{eq}, if\ \tau_i\,(g) > \tau_{tar} \\ \varepsilon_i(g+1) &= \varphi_{eq2}\varepsilon_i(g)\,, \quad if\ \tau_i(g) \leq\ \tau_{tar}\,. \end{aligned} \tag{10}$$

Step 6: Computing overall Ranking and Replacement: The ranking lists are created and combined with (5) and the best μ individuals out of λ are selected as new population. We use elitism, and the best feasible individual found so far replaces the worst one in the current population.

Step 7: Estimation of Distribution: The new distribution parameters $\boldsymbol{C}(\boldsymbol{g}+\mathbf{1})$, and $\sigma(g+1)$ are updated with (7) and (9) based on the distribution of the new population.

Step 7: Stopping Criteria: The algorithm stops if maximum number of iterations has been reached or a solution $\vec{x}$ has been found with $|f(\vec{x}) - f(\vec{x}^*)| < 1.0e - 4$, where $\vec{x}^*$ denotes the optimal solution.

4. RESULTS

4.1. First Benchmark Problems

The algorithm is tested on 13 problems given in [6]. For all benchmark problems, a (20, 100)-ES has been adopted except g2 (for g2, (40,200)-ES is used), for which this strategy leads to premature convergence. This problem has been regarded as the hardest problem in the benchmark[4], [19]. The algorithm is implemented in MATLAB and experiments are repeated 30 times. The parameter settings adopted are given in Table 1.

The results of experiments conducted with A2RL-ES are summarized Table 2, and Table 2 shows the best, mean, and worst objective function values obtained with A2RL-ES and 3RL-ES. When compared the mean values, A2RL-ES performs better than 3RL-ES in 8 problems, g2, g5, g6, g7, g9, g10, g11 and g13, while the same results are obtained for the other problems. 3RL-ES could not find the optimum point for problems g2, g7, g10 and g13 whereas our method finds the optimum with high reliability. The deviations in median, mean and worst objective function values reveal that 3RL-ES is not as reliable as A2RL-ES. Thus, the proposed algorithm is superior to the underlying method by [5]. Namely, the correlated mutation strategy has improved significantly the performance of the ranking based constraint handling strategy. In both tables, the bold-faced values indicate that a better or at least equal result has been achieved by the corresponding algorithm.

Table 1: Design parameters used in the algorithm

Name	**Value**	**Notes**
(μ, λ)	(20, 100)	
τ_{tar}	60%	
φ_{eq}	1.01	first update factor for tolerance values
φ_{eq2}	1.00001	second update factor for tolerance values
c_{cov}	$2/(n^2 + \sqrt{n})$	
Maximum generations	5000	

Table 2 shows the best, mean, and worst objective function values obtained with A2RL-ES and 3RL-ES. When compared the mean values, A2RL-ES performs better than 3RL-ES in 8 problems, g2, g5, g6, g7, g9, g10, g11 and g13, while the same results are obtained for the other problems. 3RL-ES could not find the optimum point for problems g2, g7, g10 and g13 whereas our method finds the optimum with high reliability. The deviations in median, mean and worst objective function values reveal that 3RL-ES is not as reliable as A2RL-ES. Thus, the proposed algorithm is superior to the underlying method by [5]. Namely, the correlated mutation strategy has improved significantly the performance of the ranking based constraint handling strategy.

Table 2: Comparison of A2RL-ES with 3RL-ES in terms of the best, mean, and worst objective function values

		Best		***Mean***		***Worst***	
Fun.	***Opt. value***	***A2RL-ES***	***3RL-ES***	***A2RL-ES***	***3RL-ES***	***A2RL-ES***	***3RL-ES***
g1	-15	**-15**	**-15**	**-15**	**-15**	**-15**	**-15**
g2	-0.803619	**-0.803619**	-0.803602	**-0.79812**	-0.780828	**-0.79261**	-0.712177
g3	1	**1**	**1**	**1**	**1**	**1**	**1**
g4	-30665.53	**-30665.53**	**-30665.53**	**-30665.53**	**-30665.53**	**-30665.53**	-30659.01
g5	5126.498	**5126.498**	5126.499	**5126.498**	5433.689	**5126.498**	6080.091
g6	-6961.81	**-6961.81**	**-6961.81**	**-6961.81**	-6952.210	**-6961.81**	-6566.977
g7	24.3062	**24.3062**	24.307	**24.3062**	24.384	**24.3062**	25.004
g8	-0.095825	**-0.095825**	**-0.095825**	**-0.095825**	**-0.095825**	**-0.095825**	**-0.095825**
g9	680.63	**680.63**	**680.63**	**680.63**	680.649	**680.63**	680.863
g10	7049.248	**7049.248**	7063.312	**7049.248**	7289.269	**7049.248**	8220.442
g11	0.75	**0.75**	**0.75**	**0.75**	0.826	**0.75**	0.957
g12	1	**1**	**1**	**1**	**1**	**1**	**1**
g13	0.053949	**0.053949**	0.443019	**0.053949**	0.997364	**0.053949**	0.998250

Table 3: Comparison of A2RL-ES with 3RL-ES in terms of median and standard deviations of objective function values, and average number of generations to reach the optimum point

	Median		***Std***		***G_{mean}***	
Fun.	***A2RL-ES***	***3RL-ES***	***A2RL-ES***	***3RL-ES***	***A2RL-ES***	***3RL-ES***
g1	**-15**	**-15**	4.68e-04	**0.0e+00**	850	**691**
g2	**-0.79261**	-0.780828	**9.17e-04**	2.0e-02	4492	**1225**
g3	**1**	**1**	**7.09e-10**	2.5e-05	**299**	1682
g4	**-30665.53**	**-30665.53**	**9.66e-04**	9.9e-01	**268**	679
g5	**5126.498**	5433.689	**2.54e-05**	2.7e+02	1577	**247**
g6	**-6961.81**	-6952.210	**7.17e-05**	9.5e+01	42	**34**
g7	**24.3062**	24.384	**1.12e-05**	1.2e-01	1035	**533**
g8	**-0.095825**	**-0.095825**	1.20e-08	**3.0e-17**	**13**	383
g9	**680.63**	680.649	**5.07e-06**	3.9e-02	391	**387**
g10	**7049.248**	7289.269	**3.80e-04**	2.5e+02	1556	**678**
g11	**0.75**	0.826	**4.66e-06**	8.5e-02	**222**	493
g12	**1**	**1**	**2.48e-07**	0.0e+00	**29**	63
g13	**0.053949**	0.997364	**1.32e-04**	8.5e-02	**1151**	1750

The median and standard deviations of objective function values obtained over 30 runs are given in Table 3. The last two columns in Table 3 show the average number of generations needed to find the best solution. A closer look at the table reveals that A2RL-ES was able to converge to the optimum in less than 40,000 objective function evaluations (FES) for g3, g4, g6, g8, g9, g11, and g12. The problems g1, g2, g7, and g13 required high number of function evaluations due to the high complexity they possess. Furthermore, the algorithm was very reliable as it was able to find the global optimum in all 30 runs for all problems, but g2. This is highly desired property for real life applications with limited computational time. Considering that 3RL-ES used (30,200) strategy,A2RL-ES converges to the optimum in less function evaluations (FES) for g1, g3, g4, g6,g7, g8, g9, g11, g12, and g13. However, the reliable convergence behavior of

A2RL-ES to the optimum for complex problems, g2, g5, and g7 is paid with high computational costs. Thus, the proposed algorithm is in general faster than 3RL-ES.

Table 4 shows the comparison of performance indicators of A2RL-ES, SR[6], simple multimembered evolution strategy (SMES)[20] methods on the same benchmark. When compared with other ES-based methods, A2RL-ES delivers better or similar results in terms of best, median, mean, and worst objective function values for all problems. The reliability of our method is higher than the reliability of the other approaches, as it is indicated by mean and median objective function values.

Table 4: Comparison of performances of A2RL-ES with SR[8], SMES **[30]**

Fun.	***Opt. Value***	***Method***	***Best***	***Median***	***Mean***	***Worst***	G_{mean}
g1	-15	A2RL-ES	-15	-15	-15	-15	850
		SR	-15	-15	-15	-15	741
		SMES	-15	-15	-15	-15	671
g2	0.803619	A2RL-ES	0.803619	0.79261	0.79293	0.76409	3078
		SR	0.803515	0.7858	0.781975	0.726288	1086
		SMES	0.803601	0.7912549	0.785238	0.751322	NA
g3	1	A2RL-ES	1	1	1	1	299
		SR	1	1	1	1	1146
		SMES	1	1	1	1	184
g4	-30665.53	A2RL-ES	-30665.53	-30665.53	-30665.53	-30665.53	268
		SR	-30665.53	-30665.53	-30665.53	-30665.53	441
		SMES	-30665.53	-30665.53	-30665.53	-30665.53	129
g5	5126.498	A2RL-ES	5126.498	5126.498	5126.498	5126.498	1577
		SR	5126.49	5127.372	5128.881	5142.472	258
		SMES	5126.599	5160.198	5174.492	5304.167	NA
g6	-6961.81	A2RL-ES	-6961.81	-6961.81	-6961.81	-6961.81	42
		SR	-6961.81	-6961.81	-6875.940	-6350.262	590
		SMES	-6961.81	-6961.81	-6961.284	-6952.482	249
g7	24.3062	A2RL-ES	24.3062	24.3062	24.3062	24.3062	1035
		SR	24.307	24.357	24.374	24.642	715
		SMES	24.327	24.426	24.475	24.843	NA
g8	0.095825	A2RL-ES	0.095825	0.095825	0.095825	0.095825	13
		SR	0.095825	0.095825	0.095825	0.095825	381
		SMES	0.095825	0.095825	0.095825	0.095825	18
g9	680.63	A2RL-ES	680.63	680.63	680.63	680.63	391
		SR	680.63	680.641	680.656	680.763	557
		SMES	680.632	680.642	680.643	680.719	NA
g10	7049.248	A2RL-ES	7049.248	7049.248	7049.248	7049.248	1556
		SR	7049.316	7372.613	7559.192	8835.655	642
		SMES	7051.903	7253.603	7253.047	7638.366	NA
g11	0.75	A2RL-ES	0.75	0.75	0.75	0.75	222
		SR	0.75	0.75	0.75	0.75	57
		SMES	0.75	0.75	0.75	0.75	88
g12	1	A2RL-ES	1	1	1	1	29
		SR	1	1	1	1	82
		SMES	1	1	1	1	77
g13	0.053949	A2RL-ES	0.053949	0.053949	0.053949	0.053949	1151
		SR	0.053957	0.057006	0.067543	0.216915	349
		SMES	0.053986	0.061873	0.166385	0.468294	NA

To check whether the differences reported above are statistically significant, we employed Holm procedure [21]. The four algorithms are evaluated based on the obtained mean objective function values. According to[22], the algorithm demonstrating the best performance is assigned a score of *four* while the algorithm showing the worst performance is scored as *one*. For each algorithm, the scores are summed up and averaged over the number of problems. The resulting number denotes the overall score, sc_j, of the j^{th} algorithm. Next, A2RL-ES is isolated from the set of algorithms and is labeled with zero as the reference algorithm. Then, the remaining algorithms (three algorithms in our case) are sorted in ascending order based on sc_j. For each of the remaining algorithms, a z_j value is computed with:

$$z_j = \frac{sc_j - sc_0}{\sqrt{N_a(N_a+1)/6N_p}} \tag{11}$$

where N_a and N_p stand for the number of the algorithms considered (four in our case) and the number of the test problems, respectively. sc_0corresponds to the overall score of A2RL-ESas the reference algorithm. Based on z_j values, the corresponding cumulative normal distribution probabilities p_j have been calculated. Then, the p_j values have been compared with $/(N_a - j)$, where δ indicates level of confidence, set to 5%. The Holm procedure tests the hypotheses whether the performances of the compared two algorithms are statistically indistinguishable. Thus, in our case, the rejection of the null-hypotheses indicates that A2RL-ES outperforms the compared algorithm. On the other hand, the acceptance of it implies that there is no significant difference between the two algorithms.

Table 5: Results of Holm procedure comparing the algorithms in terms of mean objective function values over 30 runs

Algorithm	J	sc_j	z_j	p_j	$\delta/(N_a - j)$	Hypotheses
3RL-ES	1	1.46153	-3.038e+00	1.190e-03	1.667e-02	rejected
SR	2	1.92307	-2.127e+00	1.672e-02	2.500e-02	rejected
SMES	3	2.15384	-1.671e+00	4.736e-02	5.000e-02	rejected

Table 6: Results of Holm procedure comparing the algorithms in terms of mean number of fitness function evaluations

Algorithm	J	sc_j	z_j	p_j	$\delta/(N_a - j)$	Hypotheses
SMES	1	0.61538	-4.709e+00	1.243e-06	1.667e-02	rejected
3RL-ES	2	1.38461	-3.190e+00	7.110e-04	2.500e-02	rejected
SR	3	1.61538	-2.734e+00	3.125e-03	5.000e-02	rejected

The first test compares the four algorithms in terms of mean objective function values achieved over 30 runs. The results are given in **Table 5** along with the decision whether the null-hypotheses is rejected. The table shows that A2RL-ES outperforms, on average, other four algorithms over 13 test problems, and the outperformance of A2RL-ES is significant.

A second test is conducted to compare the convergence speeds of the algorithm (FES needed to reach the optimum), see **Table 6**. The results indicate that A2RL-ES is again significantly better than the other algorithms.

4.2. A real Life Application: Car Side Impact Design

Besides the above benchmark problems, the merit of the algorithm is demonstrated by a real life application. The problem deals with the designing of a car-side, which is exposed to a certain impact according to European Enhanced Vehicle Safety Committee (EEVC)[23]. The model consists of eleven variables: thicknesses of B-Pillar inner, B-Pillar reinforcement, floor side inner, cross members, door beam, door beltline reinforcement and roof rail, materials of B-Pillar inner, floor side inner, barrier height and hitting position ($x_1, \dots, x_{11}$), respectively. The problem is approximated by [23]with global response surface methodology and aims to minimize the weight of the components:

Minimize:

$$f(\vec{x}) = 1.98 + 4.90x_1 + 6.67x_2 + 6.98x_3 + 4.01x_4 + 1.78x_5 + 2.73x_7$$

subject to:

$$g_1(\vec{x}) = 1.16 - 0.3717x_2x_4 - 0.00931x_2x_{10} - 0.484x_3x_9 + 0.01343x_6x_{10} - 1 \leq 0$$

$$g_2(\vec{x}) = 0.261 - 0.0159x_1x_2 - 0.188x_1x_8 - 0.019x_2x_7 + 0.0144x_3x_5 + 0.0008757x_5x_{10} + 0.08045x_6x_9 + 0.00139x_8x_{11} + 0.00001575x_{10}x_{11} - 0.32 \leq 0$$

$$g_3(\vec{x}) = 0.214 + 0.00817x_5 - 0.131x_1x_8 - 0.0704x_1x_9 + 0.03099x_2x_6 - 0.018x_2x_7 + 0.0208x_3x_8 + 0.121x_3x_9 - 0.00364x_5x_6 + 0.0007715x_5x_{10} - 0.0005354x_6x_{10} + 0.00121x_8x_{11} - 0.32 \leq 0$$

$$g_4(\vec{x}) = 0.74 - 0.061x_2 - 0.163x_3x_8 + 0.001232x_3x_{10} - 0.166x_7x_9 + 0.227x_2^2 - 0.32 \leq 0$$

$$g_5(\vec{x}) = 28.98 + 3.818x_3 - 4.2x_1x_2 + 0.0207x_5x_{10} + 6.63x_6x_9 - 7.7x_7x_8 + 0.32x_9x_{10} - 32 \leq 0$$

$$g_6(\vec{x}) = 33.86 + 2.95x_3 + 0.1792x_{10} - 5.057x_1x_2 - 11.0x_2x_8 - 0.0215x_5x_{10} - 9.98x_7x_8 + 22.0x_8x_9 - 32 \leq 0$$

$$g_7(\vec{x}) = 46.36 - 9.9x_2 - 12.9x_1x_8 + 0.1107x_3x_{10} - 32 \leq 0$$

$$g_8(\vec{x}) = 4.72 - 0.5x_4 - 0.19x_2x_3 - 0.0122x_4x_{10} + 0.009325x_6x_{10} + 0.000191x_{11}^2 - 4 \leq 0$$

$$g_9(\vec{x}) = 10.58 - 0.674x_1x_2 - 1.95x_2x_8 + 0.02054x_3x_{10} - 0.0198x_4x_{10} + 0.028x_6x_{10} - 9.9 \leq 0$$

$$g_{10}(\vec{x}) = 16.45 - 0.489x_3x_7 - 0.843x_5x_6 + 0.0432x_9x_{10} - 0.0556x_9x_{11} - 0.000786x_{11}^2 - 15.7 \leq 0$$

where the bounds of the problem are $0.5 \leq x_1, x_3, x_4 \leq 1.5$, $0.45 \leq x_2 \leq 1.35$, $0.875 \leq x_5 \leq 2.625$, $0.4 \leq x_6, x_7 \leq 1.2$, and $0.5 \leq x_{10}, x_{11} \leq 1.5$. $x_8$and x_9 can attain only discrete values from the set $\{0.192, 0.345\}$.

Gandomi et al. [23] used a newly emerging algorithm, firefly algorithm (FA) which mimics mate finding procedure of fire flies. A static penalty approach was incorporated to the algorithm as a constraint handling mechanism. We refer this algorithm as penalty based Firefly Algorithm (PFA).

Table 7: Comparison of results obtained by A2RL-ES and PFA

Opt. Value	***Method***	***Best***	***Mean***	***Worst***	***Std***	***FESav***
22.569	A2RL-ES	**22.569**	**22.6331**	**23.3585**	**2.46e-1**	54840
	PFA	22.8429	22.8937	24.0662	1.66e-1	**20000**

The reported results in **Table 7** show that the algorithm is highly reliable and competitive when compared with PFA. A2RL-ES found a better solution, which is paid by more fitness function evaluation (FES_{av}) on average[24].

5. CONCLUSION

Constrained optimization problems (COPs) require precisely tailored algorithms due to their complexity. If looked at from a different angle, inequality and equality constraints of a COP establish so called weak and strong relations among the problem variables, which may be exploited indirectly with the simplified (CMA)-like mutation operator. In this paper, we proposed an ES for constrained optimization which combines the correlated mutation operator and the ranking-based constraint handling strategy used by[5]. While the ranking-based constraint handling establishes a balance between feasible and infeasible population members, the algorithm learns the covariance matrix of the promising solutions in the vicinity of constraint boundaries, and increases the chance of generating individuals on the boundary. When compared with other algorithms, the algorithm delivers very promising results. The results on car-side design problem also indicate that A2RL-ES can deliver better results with more reliability. The robustness of the method is remarkable. The main drawback of the algorithm is its slightly lower convergence speed than the speed of the other methods for high dimensional problems.

REFERENCES

[1] Yang,Y. W., Xu, J. F., and Soh, C. K., “An evolutionary programming algorithm for continuous global optimization”, *European Journal of Operational Research*, 168 (2) (2006) 354–369.

[2] Ben Hamida, S. and Petrowski, A., “The need for improving the exploration operators for constrained optimization problems”, in: *Proceedings of the 2000 Congress on Evolutionary Computation, CEC00 (Cat. No.00TH8512)* (2000) 1176–1183.

[3] Michalewicz, Z. and Schoenauer, M., “Evolutionary Algorithms for Constrained Parameter Optimization Problems”, *Evolutionary Computation*, 4 (1996) 1–32.

[4] Ben Hamida, S. and Schoenauer, M., “ASCHEA: New Results Using Adaptive Segregational Constriant Handling”, in: *2002 Congress On Evolutionary Computation IEEE*, (2002) 884–889.

[5] Ho, P. Y. and Shimizu, K., “Evolutionary constrained optimization using an addition of ranking method and a percentage-based tolerance value adjustment scheme”, *Information Sciences*, 177 (14) (2007) 2985–3004.

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

[7] Kusakci, A. O. and Can, M., "An adaptive penalty based covariance matrix adaptation–evolution strategy", *Computers & Operations Research*, 40 (10) (2013) 2398–2417.

[8] Hansen, N. and Ostermeier, A., "Completely derandomized self-adaptation in evolution strategies", *Evolutionary computation*, 9 (2) (2001) 159–95.

[9] Mezura-Montes, E. and Coello Coello, C. A., "Constraint-handling in nature-inspired numerical optimization: Past, present and future", *Swarm and Evolutionary Computation*, 1 (4) (2011) 173–194.

[10] Kusakci,A. O. and CAn, M., "Constrained Optimization with Evolutionary Algorithms : A Comprehensive Review", *Southeast Europe Journal of Soft Computing*, 1 (2) (2012) 16–24.

[11] Yu,X. and Gen, M., *Introduction to Evolutionary Algorithms*, Springer, London, 2010.

[12] Koziel, S. and Michalewicz, Z., "Evolutionary Algorithms, Homomorphous Mappings, and Constrained Parameter Optimization", *Evolutionary Computation*, 7 (1) (1999) 19–44.

[13] Deb, K., "An efficient constraint handling method for genetic algorithms", *Applied Mechanics and Engineering*, 186 (2000) 311–338.

[14] Singh, H. K., Isaacs, A., Nguyen, T. T., and Ray, T., "Performance of infeasibility driven evolutionary algorithm (IDEA) on constrained dynamic single objective optimization problems", in: *2009 IEEE Congress on Evolutionary Computation*, (2009) 3127–3134.

[15] Hansen, N., "The CMA Evolution Strategy : A Tutorial", (2008) [Online]. Available: URL www. bionik. tu-berlin. de/user/niko/cmatutorial. pdf. [Accessed: 20-Feb-2013].

[16] Hansen, N. and Kern, S., "Evaluating the CMA evolution strategy on multimodal test functions", *Parallel Probl. Solving from Nature-PPSN VIII*, 10 (2004) 1–10.

[17] Igel, C., Hansen, N., and Roth, S., "Covariance matrix adaptation for multi-objective optimization", *Evolutionary computation*, 15 (1) (2007) 1–28.

[18] Arnold, D. V and Hansen, N., "A (1+1)-CMA-ES for constrained optimisation", in: *Proceedings of the fourteenth international conference on Genetic and evolutionary computation conference - GECCO '12* (2012) 297–304.

[19] Farmani, R. and Wright, J. A., "Self-adaptive fitness formulation for constrained optimization", *Evolutionary Computation IEEE*, 7 (5) (2003) 445–455.

[20] Mezura-Montes, E. and Coello, C. A. C., "A Simple Multimembered Evolution Strategy to Solve Constrained Optimization Problems", *IEEE Transactions on Evolutionary Computation*, 9 (1) (2005) 1–17.

[21] Holm, S., "A simple sequentially rejective multiple test procedure", *Scandinavian Journal of Statistics*, 6 (2) (1979) 65–70.

[22] Garcia, S., Fernandez, A., Luengo, J., Herrera, F., García, S., and Fernández, A., "A study of statistical techniques and performance measures for genetics-based machine learning: accuracy and interpretability", *Soft Computing*, 13 (10) (2008) 959–977.

[23] Gandomi, A. H., Yang, X. S., and Alavi, A. H., "Mixed variable structural optimization using Firefly Algorithm", *Computers & Structures*, 89 (23–24) (2011) 2325–2336.

[24] Kusakci, A. O. and Can, M., "An adaptive evolution strategy for constrained optimisation problems in engineering design", *Southeast Europe Journal of Soft Computing*, 6 (3) (2014) 175-191.