ISSN 1870-4069

Elements of C-LSHADE Algorithm: An Empirical Study on Mechatronic Design Problems

Sebastián José de la Cruz Martínez[1], Efrén Mezura Montes[2]

[1] Laboratorio Nacional de Informática Avanzada,
Mexico

[2] Centro de Investigación en Inteligencia Artificial,
Mexico

emezura@uv.mx, sdelacruz.mca17@lania.edu.mx

Abstract. This paper presents an experimental study of the components of the Constrained Success History-Based Adaptive Differential Evolution with Linear Population Size Reduction (C-LSHADE) algorithm, to clarify its importance in generating good results by solving two instances of mechatronic optimal design. C-LSHADE has four main components: (1) a historical memory to adapt CR and F parameters, (2) a mutation strategy called *current-to-pbest*, (3) a constraint handling technique based on feasible rules; and (4) a function that linearly reduces the population size over generations. Based on the final results, the linear population size decreasing is the only component that, if omitted, affects the performance of the algorithm.

Keywords: evolutionary algorithms, differential evolution, dimensional synthesis, four-bar mechanism.

1 Introduction

A particular problem when designing mechatronic systems is finding the optimal dimensional synthesis of mechanisms to perform a prescribed task in the best possible way. The dimensional synthesis is responsible for specifying angular positions and lengths of each component to find solutions to problems of trajectory, function or movement generation to established specifications [4]. Such problem is solved by treating it as a numerical optimization problem. There are different optimization techniques, which could be classified as follows: traditional, stochastic, statistical and modern or nontraditional techniques [6]. According to the specialized literature, there is evidence of their usage to solve optimal design problems: prebil performed a study to find the optimal dimensional synthesis of a mechanism used as hydraulic support in the mining industry with the help of a gradient method generalization called Adaptive Grid Refinement algorithm

(AGR), where the distance between an arbitrary coupler point and a prescribed path is minimized.

saravanan employed Multi-objective Genetic Algorithm (MOGA), Elitist Non-dominated Sorting Genetic Algorithm (NSGA-II) and Multi-objective Differential Evolution (MODE) to find geometric dimensions of three end effectors, optimal Pareto front and decrease the computational time involved in solving the problem. They also make a comparison between the algorithms through multi-objective performance measures and propose a software package for users who wish to solve a design problem in any field of study. In [2], the authors solved the synthesis of an Ackermann Steering Mechanism considering linkage lengths and distribution of precision points as optimization parameters using an algorithm inspired on the biological immune system of vertebrates. Zapata in [14] added a constraint-handling mechanism to the algorithm LSHADE, originally designed to solve unconstrained optimization problems, obtaining very competitive results when solving mechanical design problems. However, as C-LSHADE has different mechanisms within, it is unknown which ones are responsible of such good performance.

Motivated by the above, this paper proposes an experimental study of the C-LSHADE algorithm to clarify the importance of its components in obtaining good results when solving two optimal design problems.

The document is organized as follows: Section 2 presents the dimensional synthesis of a four-bar mechanism as well as case studies to be solved. Section 3 provides a description of the C-LSHADE algorithm. Section 4 shows the experimental results achieved as their discussion. Finally, Section 5 presents conclusions and future work lines.

2 Synthesis of Four-bar Linkage Mechanisms

Let be a four-bar mechanism type crank-rod-rocker shown in Figure 1, built by a reference bar (r_1), an input bar or crank (r_2), connecting rod or coupler (r_3) and an output bar or rocker (r_4). Two coordinate systems are established, the first fixed to the real world (O_1) and the second one for reference (O_2), where (x_0, y_0) is the distance between both systems, θ_0 corresponds to the mechanism's angle movement according to the horizontal axis, angles $\theta_1, \theta_2, \theta_3$ and θ_4 corresponding to the four bars angles and C(r_{cx}, r_{cy}) point that defines the coupler position [8].

In this work, it is desired to obtain the optimal design of a four-bar mechanism with the least possible error, that is, the coupler's point C must proceed as accurately as possible between the precision points C_d^i and the lowest distance of calculated points C_i. The suggested objective function is as follows (Eq. 1):

$$error = \sum_{i=1}^{n} \left[\left(C_{xd}^i - C_x^i \right)^2 + \left(C_{yd}^i - C_y^i \right)^2 \right]. \quad (1)$$

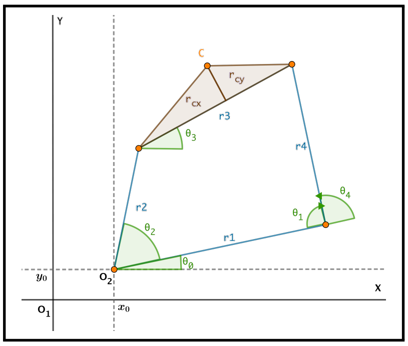


Fig. 1: Four-bar mechanism.

Subject to :

$$
\begin{aligned}
g_1\left(\overrightarrow{p}\right) &= p_1 + p_2 - p_3 - p_4 \leq 0,\\
g_2\left(\overrightarrow{p}\right) &= p_2 - p_3 \leq 0,\\
g_3\left(\overrightarrow{p}\right) &= p_3 - p_4 \leq 0,\\
g_4\left(\overrightarrow{p}\right) &= p_4 - p_1 \leq 0,
\end{aligned}
\tag{2}
$$

where $C_d^i = \left[C_{xd}^i, C_{yd}^i\right]^T$ is a precision point that defines the trajectory, a set of them as $\Omega = \{C_d^i | i \in N\}$ where N is the total number of points and $C^i = \left[C_x^i, C_y^i\right]$, each generated point expressed in accordance with the input bar and the set of bar lengths and their parameters x_0, y_0 and θ_0. For all case studies, 200 points C_i were considered. The kinematics of the mechanism can be found in [14,**?**].

Eq. 3 is a representation of the design variables vector established to four-bar mechanisms in this work:

$$
\begin{aligned}
\overrightarrow{p} &= \left[p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8, p_9\right],\\
&= \left[r_1, r_2, r_3, r_4, r_{cx}, r_{cy}, \theta_0, x_0, y_0\right],
\end{aligned}
\tag{3}
$$

where variables r_1, r_2, r_3, r_4 correspond to bar lengths, r_{cx}, r_{cy} correspond to coupler position, θ_0 movement angle of the mechanism concerning the horizontal axis of the second system and $O_2(x_0, y_0)$ starting point of the latter.

2.1 Numerical Optimization Problems

This section presents the optimization problems to be solved. To identify them, each problem was labeled with the capital letter M, associated with the word "mechanism"; and an integer, problem's index in the problem set enumeration.

(M01) Mechanism that follows a vertical linear path. Study case taken from [10], the dimensional synthesis of a mechanism that follows a vertical linear path defined by six points of precision with the least possible error is sought. The set of precision points is:

$$\Omega = \{(20,20), (20,25), (20,30), (20,35), (20,40), (20,45)\}. \tag{4}$$

Design variables vector has nine dimensions (Eq. 3). The boundaries defined for each one of them are:

$$\begin{aligned} &r_1, r_2, r_3, r_4 \in [0, 60], \\ &r_{cx}, r_{cy}, x_0, y_0 \in [-60, 60], \\ &\theta_0 \in [0, 2\pi]. \end{aligned} \tag{5}$$

The objective function to this single-objective problem is presented in Eq. 1, subject to constraints shown in Eq. 2.

(M02) Mechanism that follows a path defined by five precision points. Problem recovered from [14], the coupler must crosses through five points of precision that form a curve. The precision points are:

$$\Omega = \{(3,3), (2.759, 3.363), (2.372, 3.663), (1.890, 3.862), (1.355, 3.943)\}. \tag{6}$$

The design vector has nine variables (Eq. 3). The suggested upper and lower values for each one of them are:

$$\begin{aligned} &r_1, r_2, r_3, r_4 \in [0, 50], \\ &r_{cx}, r_{cy} \in [-50, 50], \\ &x_0, y_0, \theta_0 = 0. \end{aligned} \tag{7}$$

The single-objective problem described in Eq. 1 is considered, subject to the constraints shown in Eq. 2.

3 Constrained Success History Based Adaptive DE with Linear Population Size Reduction

Proposed in [14], C-LSHADE is an algorithm focused on solving constrained optimization problems. Its components were borrowed from previous proposals: the mutation strategy was acquired from JADE [15], the historical memory 1.1 and the linear population size reduction function were inherited from L-SHADE [13]. In order to solve constrained problems, the Feasibility Rules constraint-handling technique was added [1]. Its components are briefly detailed, but a full explanation can be found in [14].

Parameter control based on historical memory. A historical memory is composed of M_{CR} and M_F structures of H dimensions for control parameters CR and F. Parameters CR_i and F_i of each individual are calculated by randomly selecting a memory space with index $r_i \in [1, H]$ as well as using Eqs. 8 and 9 corresponding to each one of them:

$$CR_i = \begin{cases} 0 & if\ M_{CR_{ri}} = \perp \\ randn_i(M_{CR_{ri}}, 0.1) & otherwise, \end{cases} \tag{8}$$

$$F_i = randc_i\left(M_{F_{ri}}, 0.1\right), \tag{9}$$

where $\perp = -1$ is a threshold, $randn_i$ a normal distribution and $randc_i$ a Cauchy distribution. If CR_i exceeds its limits, it is biased to the nearest. Similarly, when $F_i \geq 1$, is truncated to 1 and if $F_i \leq 0$ is regenerated. $CR_{i,g}$ and F_i that produced successful solutions are stored in S_{CR} and S_F structures. In the same way, the difference between objective functions values is stored in a similar structure. With the stored information, the memory content is updated as indicated in the Algorithm 1.

***Current-to-pbest* mutation strategy.** It includes information of the best individual with the aim to improve the convergence by varying the diversity of the population; and a p parameter that limits the selection space to control the convergence of the method during the search process. A representation of such operator is the expressed in Eq. 10:

$$v_{i,g} = x_{i,g} + F_i \cdot \left(x^p_{Best_g} - x_{i,g}\right) + F_i \cdot (x_{r1,g} - x_{r2,g}), \tag{10}$$

where $x^p_{Best_g}$ is randomly selected from the $100p\%$ of the population and $p \in [0, 1]$.

Survivor Selection. C-LSHADE uses a constraint-handling technique called Feasiblility Rules, which is composed of the following three conditions:

- Between two infeasible individuals, the one with the smallest sum of constraint violation (SVR or ϕ_x) is selected. SVR is expressed in Eq. 11.
- A feasible individual is preferable over an infeasible one.
- Between two feasible individuals, the one with the best objective function value is preferred.

$$\phi_x = \sum_{j=1}^{m} max(0, g_j(x)). \tag{11}$$

Update of historical memory spaces. The updating of the averages contained in the memory is performed by Algorithm 1. In this, the index $k \in [1, H]$ is associated with the memory space to be updated. At the beginning k = 1, this is increased when update memory is performed and restored if $k > H$. Moreover, $mean_{WL}$ is remitted to the Lehmer's weighted average (Eq. 12) where wk refers to the difference between fitness functions values in order to provide information on the adaptation of parameters.

$$mean_{WL}(S_F) = \frac{\sum_{k=1}^{|S_F|} wk \cdot S^2_{F,k}}{\sum_{k=1}^{|S_F|} wk \cdot S_{F,k}}, \quad wk = \frac{\Delta fx}{\sum_{k=1}^{|S_F|} \Delta fx}, \quad \Delta fk = |f(u_{k,g}) - f(x_{k,g})|. \tag{12}$$

Linear Population Size Reduction (LPSR). It linearly reduces the population size with respect to the number of evaluations of the objective function, where its initial size is N_{init} and at the end is N_{min}. The population size for each generation is calculated according to Eq. 13:

$$NP_{G+1} = round\left[\left(\frac{N_{min} - N_{init}}{MAX\_NFE}\right)\right] * NFE + N_{init}, \tag{13}$$

Algorithm 1 Memory Update 1.1

if $S_{CR} \neq \emptyset$ y $S_F \neq \emptyset$ **then**
  if $M_{CR,k,g} = \perp$ *or* $max\,(S_{CR}) = 0$ **then**
    $M_{CR,k,g+1} = \perp$;
  else
    $M_{CR,k,g+1} = mean_{WL}\,(S_{CR})$;
  end if
  $M_{F,k,g+1} = mean_{WL}\,(S_F)$;
  $k++$;
  if $k > H$ **then**
    $k = 1$;
  end if
else
  $M_{CR,k,g+1} = M_{CR,k,g}$;
  $M_{F,k,g+1} = M_{F,k,g}$;
end if

where MAX_NFE is the maximun number of evaluations, and NFE is the current number of evaluations of the objective function. This mechanism is activated when $NP_{G+1} < NP_G$, where NP_G corresponds to number of individuals in the current population.

Algorithm 2 is a general representation of C-LSHADE.

Algorithm 2 C-LSHADE

Require: H, p, N_{init}, N_{min}
Ensure: $P(x)$
Begin
$NP = D * N_{init}$
Create $P(x_{i,0})$ where i = 1,...,NP and evaluate$f(x_{i,0})$
Set the content of $M_{CR,i}, M_{F,i}(i = 1, ..., H) = 0.5$
$g = 0, k = 1$
while stop criteria not met **do**
  Create $S_F = \emptyset, S_{CR} = \emptyset, S_{DIF} = \emptyset$
  Sort population indexes ascendingly
  for i=1 to NP **do**
    $r_i = randi\,(1, H)$
    Compute $CR_{i,g}$ y $F_{i,g}$ based on the Eqs. 8 and 9
    Create $u_{i,g}$ based on *current-to-pbest/1/bin* (Eq. 10)
    if $f(u_{i,g}) < f(x_{i,g})$ based on Factible Rules **then**
      $x_{i,g+1} = u_{i,g}$
      $S_{CR} = S_{CR} \cup CR_i$
      $S_F = S_F \cup F_i$
      $S_{DIF} = S_{DIF} \cup |f\,(u_{i,g}) - f\,(x_{i,g})\,|$
    end if
  end for
  Update memory based on the Algorithm 1
  Compute NP_{g+1} according to Eq. 13
  if $NP_{g+1} < NP$ **then**
    Sort population indexes ascendingly giving priority to the SVR and then to the fitness.
    Delete worse $NP - NP_{g+1}$.
  end if
  $g++$
end while

4 Results and Analysis

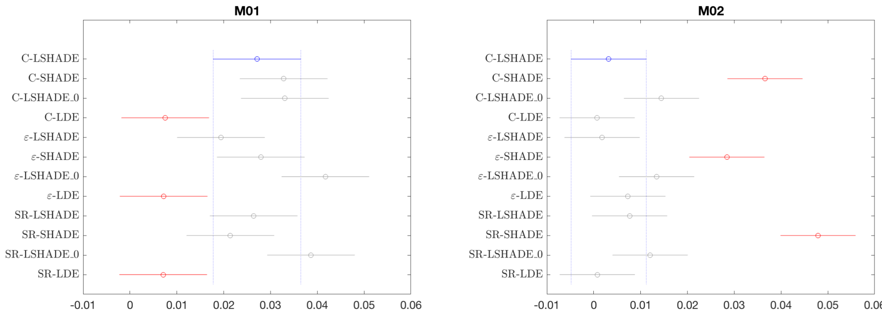
To study the C-LSHADE components, the following configurations were proposed:

- **Constraint handling technique:** Instead of the Feasibility Rules by ε-Constrained [11] and Stochastic Ranking [7] methods were adopted.
- **Linear population Size Reduction function:** Deactivate it.
- **Parameter adaptation scheme:** Replace historical memory update algorithm 1.1 by version 1.0 proposed in [12], and compute CR per individual and F at each generation.

Different variants of the algorithm were generated, grouped by the studied component and denoted as follows: (1) corresponds to versions that have different constraint-handling technique: LSHADE with ε-Constrained Method, ε-LSHADE, and LSHADE with Stochastic Ranking, SR-LSHADE; (2) variants without the population reduction mechanism: C-SHADE, ε-SHADE, and SR-SHADE; (3) variants with the historical memory version 1.0: C-LSHADE_0, ε-LSHADE_0, and SR-LSHADE_0; and (4) variants that compute CR and F dynamically: C-LDE, ε-LDE, and SR-LDE. A statistical comparison among the C-LSHADE variants was carried out to achieve the purpose of this study. The Kruskal-Wallis and the post-hoc Bonferroni tests were used. Each test was applied with 95% confidence. The experiments were performed on a computer with an Intel Core i7 - 2.5 GHz processor, 8 GB of RAM and 64-bit Windows 10 operating system. The algorithms and statistical analysis were developed in the M language using the MATLAB 2018a IDE. For all algorithms, 31 independent runs were performed to solve both optimization problems and the parameters recommended in [14] were used: $H = 6$, $p = 0.11$, $N_{init} = 18$ and $N_{min} = 4$. Likewise, the parameter values of the constraint handling techniques were taken from [7,**?**]: for Stochastic Ranking $Pf = 0.45$ and for ε-Constrained $cp = 0.5$, $\theta_0 = 0.2$, $Tc = 0.2MAX\_NFE$. The stop criterion was set at 400,000 maximun number of evaluations of the objective function (MAX_NFE) for M01 and 15,000 for M02. Case studies are treated as single-objective numerical optimization problems subject to constraints (Eq. 2) with the aim to minimize the trajectory tracking error. The complexity of the studies cases is high due to the amount of precision points that the coupler's point C must pass and the effort required to find a combination of design variables that allow a successful solution compared to the most known state-of-the-art algorithms. In general, all studied algorithms found feasible solutions in every independent run. Figure 2 shows the Bonferroni test results for the two test problems. There was significant differences in performance among the algorithm variants with different constraint-handling techniques (C-LSHADE, ε-LSHADE and SR-LSHADE) and those variants without the parameter adaptation scheme (C-LDE, ε-LDE and SR-LDE), of which the latter obtained better results for the M01 problem, see Figure 2a. Regarding this problem M01, all C-LSHADE mechanisms were removed and the good performance was still present by using any constraint-handling technique adopted in this paper. Concerning M02, those variants without a population reduction mechanism (C-SHADE, ε-SHADE and SR-SHADE) had a worse behavior, see Figure 2b. In contrast, the variants of group 4 (C-LDE, ε-LDE and SR-LDE) demonstrated better performance than the rest. Regarding problem MO2, the linear reduction is the only required mechanism by the algorithm to provide better results.

5 Conclusions and Future Work

This work proposed an empirical study of the C-LSHADE algorithm in order to determine the importance of its components in solving two mechatronic design optimization



(a) (b)

Fig. 2: Bonferroni post-hoc test based on final results. There are significant differences when the confidence intervals do not overlap. A variant is considered with a better performance when its confidence interval is closer to zero.

problems. The constraint-handling technique, the population size reduction and the historical memory version for parameter adaptation were the mechanisms under study. The overall results indicate that the only mechanism that must be present in the algorithm to provide competitive results, particularly for the second test problem, is the linear decreasing mechanism of the population size. In fact, a simplified version of the algorithm could successfully resolve the first test problem. As future work, the linear function for the population size reduction will be further analyzed and other case studies will be solved.

Acknowledgments. The first author acknowledges support from CONACyT through a scholarship to pursue graduate studies at LANIA.

References

1. Deb, K.: An efficient constraint handling method for genetic algorithms. Computer Methods in Applied Mechanics and Engineering 186(2-4), 311–338 (June 2000)
2. Hamidi, Y.E., Javash, M.S., Ettefagh, M.M., Doust, F.A.: Optimization of four-bar steering mechanism through Artificial Immune System (AIS) Algorithm. In: 12th International Conference on Control, Automation and Systems. pp. 913–917 (October 2012)
3. Hernández, B., del P. Pozos, M., Mezura, E., Portilla, E.A., Vega, E., Calva, M.B.: Two-Swim Operators in the Modified Bacterial Foraging Algorithm for the Optimal Synthesis of Four-Bar Mechanisms. Hindawi Publishing Corporation Computational Intelligence and Neuroscience p. 18 (January 2016)
4. Myszka, D.H.: Machines and mechanisms. Prentice Hall, 4th edn. (2004)
5. Prebil, I., Krašna, S., Ciglarič, I.: Synthesis of four-bar mechanism in a hydraulic support using a global optimization algorithm. Structural and Multidisciplinary Optimization 24(3), 246–251 (September 2002), https://doi.org/10.1007/s00158-002-0234-y
6. Rao, S.S.: Engineering Optimization: Theory and Practice. John Wiley and Sons, Inc., River Street,Hoboken, New Jersey, 4th edn. (June 2009)

7. Runarsson, T.P., Yao, X.: Stochastic ranking for constrained evolutionary optimization. IEEE Transactions on Evolutionary Computation 4(3), 284–294 (September 2000)
8. Sánchez-Márquez, Á., Vega-Alvarado, E., Portilla-Flores, E.A., Mezura-Montes, E.: Synthesis of a planar four-bar mechanism for position control using the harmony search algorithm. In: 11th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE). pp. 1–6 (September 2014)
9. Saravanan, R., Ramabalan, S., Godwin Raja Ebenezer, N., Dharmaraja, C.: Evolutionary multi criteria design optimization of robot grippers. Applied Soft Computing 9(1), 159 – 172 (2009)
10. Sleesongsom, S., Bureerat, S.: Four-bar linkage path generation through self-adaptive population size teaching-learning based optimization. Knowledge-Based Systems (August 2017)
11. Takahama, T., Sakai, S.: Constrained optimization by the ε constrained differential evolution with an archive and gradient-based mutation. In: IEEE Congress on Evolutionary Computation. pp. 1–9 (July 2010)
12. Tanabe, R., Fukunaga, A.: Success-history based parameter adaptation for Differential Evolution. In: 2013 IEEE Congress on Evolutionary Computation. pp. 71–78 (June 2013)
13. Tanabe, R., Fukunaga, A.: Improving the Search Performance of SHADE Using Linear Population Size Reduction. IEEE Congress on Evolutionary Computation pp. 1658–1665 (July 2014)
14. Zapata, M.F.: Control de Parámetros del Algoritmo Evolución Diferencial con Variantes Combinadas para la Solución de Problemas de Optimización en Mecatrónica. Master's thesis, Laboratorio Nacional de Informática Avanzada, Xalapa, Veracruz, México (Agosto 2017)
15. Zhang, J., Sanderson, A.C.: JADE: Adaptive Differential Evolution With Optional External Archive. IEEE Transactions on Evolutionary Computation 13(5), 945–958 (October 2009)