Hybrid DE Algorithm With Adaptive Crossover Operator For Solving Real-World Numerical Optimization Problems

Gilberto Reynoso-Meza, Javier Sanchis, Xavier Blasco and Juan M. Herrero
Instituto Universitario de Automática e Informática Industrial
Universitat Politècnica de València
Camino de Vera s/n 46022, Valencia, Spain

***Abstract*—In this paper, the results for the CEC 2011 Competition on testing evolutionary algorithms on real world optimization problems using a hybrid differential evolution algorithm are presented. The proposal uses a local search routine to improve convergence and an adaptive crossover operator. According to the obtained results, this algorithm shows to be able to find competitive solutions with reported results.**

***Index Terms*—Differential Evolution algorithm, parameter selection, CEC competition.**

I. Introduction

In this paper, a hybrid differential evolution algorithm with a local search (LS) technique and an adaptive mechanism for crossover operator has been used to solve a set of single-objective optimization problems. Such problem set was defined for the CEC-2011 Competition and Special Session on testing evolutionary algorithms on real world optimization problems. The problem definitions and evaluation criteria are defined in [1]. This competition includes a set of 13 problems, with several diverse topics, which includes control systems, engineering design and aerospace applications (see table I).

The problem set consists of several single objective optimization statements, with different characteristics, number of decision variables and constraints. In general, a constrained single objective optimization is stated as follows:

$$\begin{aligned} \min_{\boldsymbol{x}\in\Re^n} J(\boldsymbol{x}) \in \Re^1 \\ s.t. \\ g_i(\boldsymbol{x}) \leq 0, i = 1, \ldots, p \\ h_j(\boldsymbol{x}) = 0, j = p+1, \ldots, r. \end{aligned} \quad (1)$$

where $\boldsymbol{x} \in \Re^n$ is known as the decision vector of n variables. A common approach to constraint handling in optimization problems is by means of penalty functions. An aggregate cost function is built using the original cost function and a penalty value according to the amount of the violation constraint. With this in mind, some of the problems have been re-formulated in the form:

$$\min_{\boldsymbol{x}\in\Re^n} \Im(\boldsymbol{x}) = J(\boldsymbol{x}) + f_{penalty}(\boldsymbol{x}, g_i(\boldsymbol{x}), h_j(\boldsymbol{x})) \in \Re^1 \quad (2)$$

Several problems in the set have been subject of research in order to generate new evolutionary algorithms strategies, due to their complexity [1]. Sometimes, such algorithms are highly specialized in a particular kind of problem, and the search for an algorithm of general purpose is worthwhile. Due to this fact, in this paper a proposal based on Differential Evolution (DE) algorithm is presented. The remainder of this paper is as follows: in section II the algorithm is presented. The experimental setup used to solve the proposed problems is shown in section III. In section IV the obtained results are commented and finally, some concluding remarks are given.

II. Evolutionary optimization algorithm

Nowadays, DE algorithm [2]–[4] is perhaps one of the most powerful evolutionary algorithms to solve real parameter optimization problems [4]. Several state-of-the-art evolutionary algorithms use DE as evolutionary mechanism [5]–[12]. For this reason, DE algorithm is used as proposal in this work.

A. *Differential Evolution Review*

Generally speaking, DE algorithm has three main steps:

1) Mutation: At generation k for each target (parent) vector $\boldsymbol{x}^i|_k$, a mutant vector $\boldsymbol{v}^i|_k$ is generated according to equation (3):

$$\boldsymbol{v}^i|_k = \boldsymbol{x}^{r_1}|_k + F(\boldsymbol{x}^{r_2}|_k - \boldsymbol{x}^{r_3}|_k) \quad (3)$$

Where $r_1 \neq r_2 \neq r_3 \neq i$ and F is known as the Scaling Factor.

2) Crossover: This operator is crucial due to its sensitivity to face separable and non-separable problems. One of the most popular mechanisms is the binary crossover. For each target vector $\boldsymbol{x}^i|_k$ and its mutant vector $\boldsymbol{v}^i|_k$, a trial (child) vector $\boldsymbol{u}^i|_k = [u_1^i|_k, u_2^i|_k, \ldots, u_n^i|_k]$ is calculated as follows:

$$u_j^i|_k = \begin{cases} v_j^i|_k & if \quad rand(0,1) \leq Cr \\ x_j^i|_k & \quad otherwise \end{cases} \quad (4)$$

where $j \in 1, 2, 3 \ldots n$ and Cr is the Crossover probability rate. In its standard version, at least one dimension must be exchanged. For separable problems, low values of Cr are recommended while for non-separable problems, high values are suggested [3].

978-1-4244-7835-4/11/$26.00 ©2011 IEEE

TABLE I: Problem set definitions.

Problem	Variables	Comments
P01	6	Parameter estimation for frequency-modulated (FM) sound waves.
P02	30	Lennard-Jones Potential Problem.
P03	1	Bifunctional catalyst blend optimal control problem.
P04	1	Optimal control of a non-linear stirred tank reactor.
P05	30	Tersoff potential function minimization problem (instance 1).
P06	30	Tersoff potential function minimization problem (instance 2).
P07	20	Spread spectrum radar polly phase code design.
P08	7	Transmission network expansion planning problem.
P09	122	Large scale transmission pricing problem.
P10	12	Circular antenna array design problem.
P11.1	120	Dynamic economic dispatch problem, instance 1
P11.2	216	Dynamic economic dispatch problem, instance 2
P11.3	6	Static economic load dispatch problem, instance 1
P11.4	13	Static economic load dispatch problem, instance 2
P11.5	15	Static economic load dispatch problem, instance 3
P11.6	40	Static economic load dispatch problem, instance 4
P11.7	140	Static economic load dispatch problem, instance 5
P11.8	96	Hydrothermal scheduling problem, instance 1
P11.9	96	Hydrothermal scheduling problem, instance 2
P11.10	96	Hydrothermal scheduling problem, instance 3
P12	26	Spacecraft trajectory optimization problem.
P13	22	Spacecraft trajectory optimization problem.

The binomial crossover (for $Cr < 1$) is not a rotationally invariant operator. For functions with a high dependency among search variables, DE could face some difficulties [4], and a rotationally invariant line recombination operator [13] was suggested as an alternative.

For each target vector $\boldsymbol{x}^i|_k$ and its mutant vector $\boldsymbol{v}^i|_k$, a trial (child) vector $\boldsymbol{u}^i|_k = [u_1^i|_k, u_2^i|_k, \ldots, u_n^i|_k]$ is calculated as:

$$\boldsymbol{u^i}|_{\boldsymbol{k}} = \boldsymbol{x^i}|_{\boldsymbol{k}} + F_i(\boldsymbol{v^i}|_{\boldsymbol{k}} - \boldsymbol{x^i}|_{\boldsymbol{k}}) \tag{5}$$

It is recommended $F_i = 0.5\cdot(F+1)$ as a good initial choice [13].

3) Selection: A greedy selection is used, where a child substitutes its father in the next generation if it has a better or equal fitness function value.

$$\boldsymbol{x}^i|_{k+1} = \begin{cases} \boldsymbol{u}^i|_k & if \quad f(\boldsymbol{u}^i|_k) \leq f(\boldsymbol{x}^i|_k) \\ \boldsymbol{x}^i|_k & \qquad otherwise \end{cases} \tag{6}$$

B. DE variant proposal

The proposal in this work is to use a hybrid DE algorithm with local search techniques and an adaptive parameter value for Cr. This proposal will hereafter be denoted as $DE-\Lambda_{Cr}$. This adaptive mechanism combines the binary crossover and the line recombination, to face the diversity in the problem set. Also, a population refreshment mechanism is used to avoid stagnation.

1) Local search strategy: A sequential quadratic programming (SQP) routine is incorporated to improve convergence as a local search mechanism. The SQP routine is executed in every child of the population with a probability α_{LS}. Local search strategies has shown to improve the performance of the classical DE algorithm [14]–[16].

2) Adaptive Parameter values: Triangular distributions Λ_F, Λ_{Cr} are used for both values F and Cr to calculate $f_F(\boldsymbol{x})|_k$ and $f_{Cr}(\boldsymbol{x})|_k$ for each child in the evolutionary process at generation k. In the case of F, it is used to incorporate some dither in the algorithm. Whilst the triangular distribution Λ_F is fixed, the triangular distribution Λ_{Cr} is adapted after a number $\gamma_{success}$ of recorded success. It will be said to have a success if a child substitutes its parent in the next generation. The minimum, maximum and medium value on such set of success is used for this purpose. The ideas behind this adaptive mechanism for the crossover are:

- Be able to detect a separable problem, choosing a binomial crossover operator with low values for Cr.
- Be able to detect non-separable problems, choosing a binomial crossover operator with high values for Cr.
- Be able to detect a strong-dependency on decision variables, and use a non-rotationally invariant line recombination.

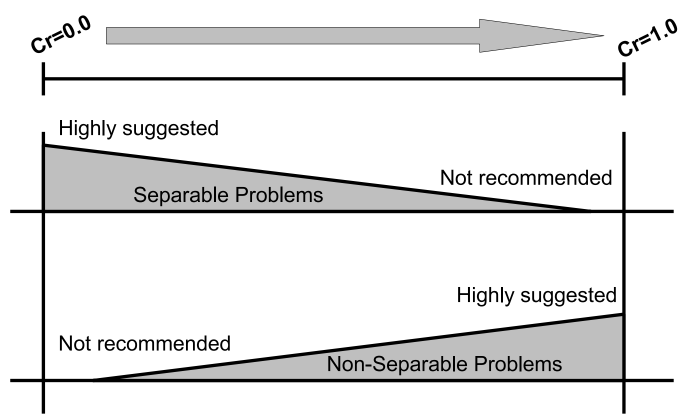

Fig. 1: Suggested use from literature for Cr values. With Cr=1.0, algorithm is fully rotationally invariant.

In this way, the algorithm will be able to detect if high values of Cr are useful, and furthermore, if a rotationally invariant crossover is required. A minimum base for Λ_{Cr} around its median value is incorporated to avoid stagnation around a single value. Also, a threshold α_{Cr} is defined to use directly the line recombination crossover if $f_{Cr}(\boldsymbol{x})|_k > \alpha_{Cr}$

3) Population management: An initial population $P(0)$ with N_p individuals is randomly selected from the decision space. A subpopulation of N_s individuals is used in each generation step. A population refreshment is incorporated when the difference of the interquartil range ($\boldsymbol{IQR}$) is below $\hat{\boldsymbol{V}} = \frac{Upper^* - Lower^*}{\gamma_{var}}$ where $Upper^*$ and $Lower^*$ are the upper and lower bounds since the last population refreshment. The refreshed population is initialized with N_R vector solutions in the range $\boldsymbol{x}^{median} \pm \gamma_{var} \cdot \hat{\boldsymbol{V}}$; the new initialization range for the next refreshment (if required) will be defined by $\boldsymbol{x}^{median} \pm \gamma_{var} \cdot \hat{\boldsymbol{V}}$ bounded to a minimum $\hat{\boldsymbol{V}}_{min}$. Interquartile difference is used instead variance, due to its robustness properties to exclude outliers.

The main steps of the hybrid DE are:

S1: Initialize $P(0)$ with N_p individuals randomly selected from the searching space.

S2: Evaluate initial population $P(0)$

S3: For $k = 1$: **MaxGen** or convergence criterion reached

- S3.1: Select randomly a subpopulation of $N_S(k)$ individuals with proposed solutions on $P(k)$.
- S3.2: Apply the fundamental DE operators on subpopulation $N_S(k)$ to get the offspring $O(k)$:
  1) Perform mutation operator (equation 3) with F selected from the triangular distribution.
  2) Perform crossover operator (equation 4) with Cr selected from triangular distribution. If $Cr > \alpha_{Cr}$, the arithmetic recombination is used.
- S3.4: For every child, perform a local search with a probability of α_{LS}
- S3.3: Evaluate offspring $O(k)$; if child $\leq$ parent, the parent is substituted by its child.
- S3.4: Record the number of success. If the cumulative number of success is bigger than $\gamma_{success}$, reset the cumulative number and recalculate the triangular distribution for Cr.
- S3.5: If $\boldsymbol{IQR} < \hat{\boldsymbol{V}}$, refresh the population with N_R individuals from $\boldsymbol{x}^{median} \pm \gamma_{var} \cdot \hat{V}$.

S4: Algorithm terminates.

III. Experimental Setup

The algorithm was implemented in Matlab© with an standard computer with 1.6GHz processor and 1.9 GB RAM. The following tuning parameters are selected for the $DE - \Lambda_{Cr}$ proposal:

Local Search parameter:

- $\alpha_{LS} = \frac{1}{100 \cdot n}$.

Adaptive mechanism:

- $\gamma_{success} = 15$;
- $\alpha_{Cr} = 0.95$;
- Minimum base for Λ_{Cr}: $median \pm 0.1$.
- $\Lambda_F = \{0.3, 0.4, 0.5\}$.
- $F_i = 0.5 \cdot (1 + 0.5) = 0.75$

Population Management:

- $N_R = 2 \cdot n$ (bounded to $[20, 120]$).
- $N_p = 5 \cdot n$ (bounded to $[50, 300]$).
- $N_s = 5 \cdot n$ (bounded to $[50, 300]$).
- $\gamma_{var} = 3$.
- $\hat{\boldsymbol{V}}_{min} = 0.05 \cdot \boldsymbol{x}^{upper} - \boldsymbol{x}^{lower}$.

IV. Discussion

In table II, the best result obtained by $DE - \Lambda_{Cr}$ proposal for each problem is shown, whilst in tables III, IV, V, VI and VII the obtained results for FES=5e4, FES=10e4, FES=15e4 are presented. The performance of some problems could be improved with different parameter tuning values. The competition discourages to use a different set of parameters for each problems and therefore, general values are used instead.

From reported results in the literature it is possible to notice some remarks on the algorithm performance:

- Regarding problem P01, the $DE - \Lambda_{Cr}$ proposal is capable to find low values for parameter estimation for FM sound waves, and it is competitive with other reported values [17].
- For problem subset P11, the $DE - \Lambda_{Cr}$ algorithm structure is able to detect the strong dependency on search variables. This strong dependency is due to the interdependency among variables, since the feasible space demands to have an specific load. It is not possible to find a better value than the reported in other works [18], but it is possible to have competitive results. As mentioned earlier, the constraint handling is incorporated in the problem definitions. A different constraint handling mechanism with the proposed algorithm could lead to better results.
- Regarding European Space Agency (ESA) problems (problems P12 and P13), notice that the algorithm has problems to find a reasonable solution for messenger problem (P12) with the allowed FEs. The better value reported from the ESA is 2.970 km/sec (recently found). For Cassini problem (P13), some executions are able to reach values below 9.0. The best reported results is 8.383 kg km/sec by MIDACO algorithm [19], which required 50 days to reach its solution.

V. Conclusion

The results of the proposed $DE - \Lambda_{Cr}$ algorithm for the CEC-2011 special session on real world optimization are presented. The problem set consist on 13 problems with diverse topics, including control systems, spacecraft trajectory optimization, economic load dispatch, engineering design, among others.

TABLE II: Best results achieved.

Problem	Best value (this paper)
P01	7.2093E-15
P02	-2.8423E+01
P03	1.1515E-05
P04	1.3772E+01
P05	-3.6845E+01
P06	-3.6845E+01
P07	6.6591E-01
P08	2.2000E+02
P09	3.2527E+05
P10	-2.1601E+01
P11.1	4.4847E+04
P11.2	1.0354E+06
P11.3	1.5445E+04
P11.4	1.8028E+04
P11.5	3.2730E+04
P11.6	1.2313E+05
P11.7	1.7124E+06
P11.8	9.2280E+05
P11.9	9.2717E+05
P11.10	9.2303E+05
P12	1.1814E+01
P13	8.9624E+00

The $DE - \Lambda_{Cr}$ proposal use a local search routine to improve converge and an adaptive parameter mechanism for the crossover operator. According with the reported results, the algorithm is capable to find competitive solutions compared with the reported results from the literature.

Acknowledgment

This work was partially supported by the FPI-2010/19 grant from the Universitat Politècnica de València and the project DPI2008-02133/DPI from the Spanish Ministry of Science and Innovation.

References

[1] S. Das and P. Suganthan, "Problem definitions and evaluation criteria for cec 2011 competition on testing evolutionary algorithms on real world optimization problems," Jadavpur university and Nanyang Technological University, Tech. Rep., 2011.

[2] R. Storn and K. Price, "Differential evolution: A simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, pp. 341 – 359, 1997.

[3] R. Storn, "Sci: Differential evolution research: Trends and open questions," U. K. C. (Ed.), Ed. Springer, Heidelberg, 2008, vol. LNCS 143, pp. 1 – 31.

[4] S. Das and P. N. Suganthan, "Differential evolution: A survey of the state-of-the-art," *Evolutionary Computation, IEEE Transactions on*, 2010.

[5] S. Kukkonen and J. Lampinen., "Gde3: The third evolution step on generalized differential evolution," vol. 1. IEEE Congress on Evolutionary Computation (CEC 2005), 2005, pp. 443 – 450.

[6] J. Brest, B. Boskovic, S. Greiner, V. Zumer, and M. S. Maucec, "Performance comparison of self-adaptive and adaptive differential evolution algorithms," vol. 11, no. 7, pp. 617 – 629, 2007.

[7] A. Qin and P. Suganthan, "Self-adaptive differential evolution algorithm for numerical optimization," vol. 2. Evolutionary Computation, 2005. The 2005 IEEE Congress on, 2005, pp. 1785 – 1791.

[8] V. Huang, A. Qin, P. Suganthan, and M. Tasgetiren, "Multi-objective optimization based on self-adaptive differential evolution algorithm," pp. 324 – 331, 2007.

[9] R. Mallipeddi, P. Suganthan, Q. Pan, and M. Tasgetiren, "Differential evolution algorithm with ensemble of parameters and mutation strategies," *Applied Soft Computing*, vol. 11, no. 2, pp. 1679 – 1696, 2011, the Impact of Soft Computing for the Progress of Artificial Intelligence.

[10] B. Qu and P. Suganthan, "Multi-objective evolutionary algorithms based on the summation of normalized objectives and diversified selection," *Information Sciences*, vol. 180, no. 17, pp. 3170 – 3181, 2010, including Special Section on Virtual Agent and Organization Modeling: Theory and Applications.

[11] Q. Zhang and H. Li, "Moea/d: A multiobjective evolutionary algorithm based on decomposition," *Evolutionary Computation, IEEE Transactions on*, vol. 11, no. 6, pp. 712 – 731, 2007.

[12] G. Reynoso-Meza, "Design, coding and implementation of a multiobjective optimization algorithm based on differential evolution with spherical pruning: applications for system identification and controller tuning." Master's thesis, Universidad Politécnica de Valencia., 2009. [Online]. Available: http://personales.alumno.upv.es/gilreyme

[13] K. V. Price, *An introduction to differential evolution*. Maidenhead, UK, England: McGraw-Hill Ltd., UK, 1999, pp. 79 – 108.

[14] N. Noman and H. Iba, "Accelerating differential evolution using an adaptive local search," *Evolutionary Computation, IEEE Transactions on*, vol. 12, no. 1, pp. 107 –125, feb. 2008.

[15] F. Neri and V. Tirronen, "Scale factor local search in differential evolution," *Memetic Computing*, vol. 1, pp. 153–171, 2009, 10.1007/s12293-009-0008-9.

[16] A. LaTorre, S. Muelas, and J.-M. Peña, "A mos-based dynamic memetic differential evolution algorithm for continuous optimization: a scalability test," *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, pp. 1–13, 2010, 10.1007/s00500-010-0646-3.

[17] F. Herrera and M. Lozano, "Gradual distributed real-coded genetic algorithms," *Evolutionary Computation, IEEE Transactions on*, vol. 4, no. 1, pp. 43 –63, Apr. 2000.

[18] L. Coelho and V. Mariani, "Combining of chaotic differential evolution and quadratic programming for economic dispatch optimization with valve-point effect," *Power Systems, IEEE Transactions on*, vol. 21, no. 2, pp. 989 – 996, May 2006.

[19] M. Schlueter, J.-J. Ruckmann, and M. Gerdts, "Non-linear mixed-integer-based optimisation technique for space applications," Poster ESA 1st Networking/Partnering Day 2010, January 2010.

TABLE III: Function values achieved when FES=5e4, FES=10e4, FES=15e4 for parameter estimation for FM sound waves (P01), Lennard-Jones Potential (P02), bifunctional catalyst blend optimal control (P03), optimal control of a non-linear stirred tank reactor (P04) and Tersoff potential minimization Si(B) problems.

		P01	P02	P03	P04	P05
5e4 FES	Worst	1.4813E+01	-2.2032E+01	1.1515E-05	2.1025E+01	-2.4250E+01
	Median	3.8779E-09	-2.7447E+01	1.1515E-05	2.0959E+01	-3.2957E+01
	Best	1.6147E-10	-2.8423E+01	1.1515E-05	1.3772E+01	-3.6845E+01
	Mean	2.1870E+00	-2.6525E+01	1.1515E-05	1.9524E+01	-3.2358E+01
	Std	4.6326E+00	1.7450E+00	2.2930E-17	2.4861E+00	3.2663E+00
10e4 FES	Worst	1.1757E+01	-2.6362E+01	1.1515E-05	2.1012E+01	-3.1484E+01
	Median	5.0886E-11	-2.7545E+01	1.1515E-05	2.0958E+01	-3.4252E+01
	Best	3.5443E-12	-2.8423E+01	1.1515E-05	1.3772E+01	-3.6845E+01
	Mean	1.0978E+00	-2.7527E+01	1.1515E-05	1.9343E+01	-3.4237E+01
	Std	3.1751E+00	6.5994E-01	1.1207E-17	2.6044E+00	1.5187E+00
15e4 FES	Worst	1.1757E+01	-2.6443E+01	1.1515E-05	2.1002E+01	-3.1484E+01
	Median	1.2362E-11	-2.7545E+01	1.1515E-05	1.7487E+01	-3.4870E+01
	Best	7.2093E-15	-2.8423E+01	1.1515E-05	1.3772E+01	-3.6845E+01
	Mean	8.7697E-01	-2.7731E+01	1.1515E-05	1.7339E+01	-3.4720E+01
	Std	3.0439E+00	4.9035E-01	6.1087E-18	2.9761E+00	1.4469E+00

TABLE IV: Function values achieved when FES=5e4, FES=10e4, FES=15e4 for Tersoff potential minimization Si(C) (P06) spread spectrum radar (C07), transmission network expansion planning (C08), large scale transmission pricing problem (C09) and circular antenna array design (C10) problems.

		P06	P07	P08	P09	P10
5e4 FES	Worst	-2.4495E+01	1.1839E+00	2.2000E+02	1.3610E+06	-1.0889E+01
	Median	-3.2897E+01	9.8081E-01	2.2000E+02	3.4789E+05	-1.6165E+01
	Best	-3.6845E+01	8.3380E-01	2.2000E+02	3.2528E+05	-2.1572E+01
	Mean	-3.2239E+01	9.8858E-01	2.2000E+02	4.5612E+05	-1.6736E+01
	Std	3.5197E+00	1.0315E-01	0.0000E+00	2.6085E+05	4.0332E+00
10e4 FES	Worst	-3.1693E+01	1.1263E+00	2.2000E+02	5.6363E+05	-1.0939E+01
	Median	-3.4165E+01	9.3720E-01	2.2000E+02	3.3088E+05	-1.6198E+01
	Best	-3.6845E+01	7.1698E-01	2.2000E+02	3.2527E+05	-2.1586E+01
	Mean	-3.4183E+01	9.2967E-01	2.2000E+02	3.4660E+05	-1.6751E+01
	Std	1.3086E+00	1.0051E-01	0.0000E+00	4.7757E+04	4.0388E+00
15e4 FES	Worst	-3.4165E+01	1.0361E+00	2.2000E+02	3.6130E+05	-1.0940E+01
	Median	-3.4316E+01	8.9739E-01	2.2000E+02	3.3088E+05	-1.6198E+01
	Best	-3.6845E+01	6.6591E-01	2.2000E+02	3.2527E+05	-2.1601E+01
	Mean	-3.5033E+01	8.8477E-01	2.2000E+02	3.3547E+05	-1.6756E+01
	Std	1.0287E+00	1.0571E-01	0.0000E+00	1.1809E+04	4.0437E+00

TABLE V: Function values achieved when FES=5e4, FES=10e4, FES=15e4 for Economic dispatch load problems (P11.1 ∽ P11.5).

		P11.1	P11.2	P11.3	P11.4	P11.5
5e4 FES	Worst	5.1872E+04	1.9581E+06	1.5448E+04	1.8196E+04	3.2953E+04
	Median	4.7680E+04	1.0407E+06	1.5446E+04	1.8088E+04	3.2842E+04
	Best	4.5458E+04	1.0376E+06	1.5445E+04	1.8028E+04	3.2773E+04
	Mean	4.7747E+04	1.1586E+06	1.5446E+04	1.8099E+04	3.2850E+04
	Std	1.3962E+03	2.9044E+05	7.9715E-01	4.0566E+01	5.0620E+01
10e4 FES	Worst	4.8019E+04	1.2968E+06	1.5448E+04	1.8196E+04	3.2939E+04
	Median	4.6316E+04	1.0376E+06	1.5446E+04	1.8088E+04	3.2810E+04
	Best	4.4847E+04	1.0354E+06	1.5445E+04	1.8028E+04	3.2762E+04
	Mean	4.6423E+04	1.0498E+06	1.5446E+04	1.8098E+04	3.2816E+04
	Std	7.9974E+02	5.1927E+04	8.5993E-01	4.0541E+01	3.9643E+01
15e4 FES	Worst	4.8019E+04	1.2810E+06	1.5448E+04	1.8196E+04	3.2854E+04
	Median	4.6232E+04	1.0375E+06	1.5446E+04	1.8087E+04	3.2789E+04
	Best	4.4847E+04	1.0354E+06	1.5445E+04	1.8028E+04	3.2730E+04
	Mean	4.6321E+04	1.0490E+06	1.5446E+04	1.8096E+04	3.2790E+04
	Std	8.5566E+02	4.8789E+04	8.4385E-01	3.9913E+01	3.0447E+01

TABLE VI: Function values achieved when FES=5e4, FES=10e4, FES=15e4 for Economic dispatch load problems (P11.6 ∽ P11.10).

		P11.6	P11.7	P11.8	P11.9	P11.10
5e4 FES	Worst	1.2553E+05	1.9712E+06	9.2842E+05	9.4618E+05	9.5266E+05
	Median	1.2445E+05	1.8502E+06	9.2519E+05	9.3229E+05	9.2550E+05
	Best	1.2359E+05	1.7124E+06	9.2390E+05	9.2717E+05	9.2312E+05
	Mean	1.2440E+05	1.8653E+06	9.2533E+05	9.3282E+05	9.2710E+05
	Std	5.0530E+02	5.7404E+04	1.1126E+03	3.9035E+03	6.7496E+03
10e4 FES	Worst	1.2520E+05	1.9250E+06	9.2680E+05	9.3507E+05	9.2597E+05
	Median	1.2404E+05	1.8483E+06	9.2413E+05	9.3079E+05	9.2427E+05
	Best	1.2313E+05	1.7124E+06	9.2307E+05	9.2717E+05	9.2312E+05
	Mean	1.2400E+05	1.8391E+06	9.2442E+05	9.3061E+05	9.2436E+05
	Std	4.4568E+02	4.5459E+04	8.2878E+02	1.7486E+03	8.0871E+02
15e4 FES	Worst	1.2520E+05	1.9250E+06	9.2542E+05	9.3348E+05	9.2481E+05
	Median	1.2390E+05	1.8483E+06	9.2389E+05	9.2999E+05	9.2376E+05
	Best	1.2313E+05	1.7124E+06	9.2280E+05	9.2717E+05	9.2303E+05
	Mean	1.2389E+05	1.8386E+06	9.2393E+05	9.3001E+05	9.2382E+05
	Std	4.5430E+02	4.5074E+04	7.4822E+02	1.6287E+03	5.1463E+02

TABLE VII: Function values achieved when FES=5e4, FES=10e4, FES=15e4 for Messenger (P12) and Cassini (P13) problems.

		P12	P13
5e4 FES	Worst	2.1172E+01	3.0450E+01
	Median	1.7047E+01	2.1860E+01
	Best	1.1814E+01	1.0544E+01
	Mean	1.6943E+01	2.0483E+01
	Std	1.9908E+00	4.4830E+00
10e4 FES	Worst	1.8486E+01	2.3129E+01
	Median	1.5733E+01	1.5778E+01
	Best	1.1814E+01	8.9636E+00
	Mean	1.5888E+01	1.6422E+01
	Std	1.4035E+00	3.8442E+00
15e4 FES	Worst	1.7891E+01	2.1052E+01
	Median	1.5219E+01	1.4412E+01
	Best	1.1814E+01	8.9624E+00
	Mean	1.5360E+01	1.4909E+01
	Std	1.2133E+00	2.7634E+00