An Efficien Constraint Handling Approach for Optimization Problems with Limited Feasibility and Computationally Expensive Constraint Evaluations

Md Asafuddoula
University of New South Wales
Canberra, ACT 2610, Australia
Md.Asaf@.adfa.edu.au

Tapabrata Ray
University of New South Wales
Canberra, ACT 2610, Australia
t.ray@.adfa.edu.au

Ruhul Sarker
University of New South Wales
Canberra, ACT 2610, Australia
r.sarker@.adfa.edu.au

ABSTRACT

Existing optimization approaches adopt a *full evaluation policy*, i.e. all the constraints corresponding to a solution are evaluated throughout the course of search. Furthermore, a common sequence of constraint evaluation is used for all the solutions. In this paper, we introduce a scheme of constraint handling, wherein every solution is assigned a random sequence of constraints and the evaluation process is aborted whenever a constraint is violated. The solutions are sorted based on two measures i.e. the number of satisfie constraints and the violation measure. The number of satisfie constraints takes a precedence over the amount of violation. We illustrate the performance of the proposed scheme and compare it with other state-of-the-art constraint handling methods within a framework of differential evolution. The results are compared using *g-series* test functions for inequality constraints. The results clearly highlight the potential savings offered by the proposed method.

Categories and Subject Descriptors

I [**Computing Methodologies**]: MISCELLANEOUS

General Terms

Algorithm

Keywords

Constraint Handling, Constraint Sequencing, Fitness Evaluation

1. INTRODUCTION

Constraint handling is an important area of research and various forms of constraint handling schemes have been proposed in literature. The performance of all population-based stochastic optimization algorithms are known to be affected by the presence of constraints. The nonlinearity, multi-modality and the feasibility space associated with each constraint is likely to be different. Constraint handling methods can be broadly categorized in four different types i.e. use of penalty functions, repair schemes, use of decoders and the separation of objective function and constraints [1]. More recent methods maintain infeasible solutions such as through stochastic ranking, ϵ based comparisons or adaptive penalty function formulations [2]. However, in all such formulations and implementations, a *full evaluation policy* is adopted, i.e. for every solution, its constraint violation (CV) measure is computed which is the sum of all constraint violations. An important question is "why do we spend computational resources to evaluate constraints of a solution, when it has already violated a constraint ?". Assuming that one is only interested in a feasible solution (preferably optimum) at the end of the search process, it is important to investigate the worth of evaluating infeasible solutions i.e. the cost of evaluation versus the knowledge gained to steer the search. Other followup questions include "what is the best sequence to evaluate the constraints ?" and "is there a benefi in using different sequence of constraints ?". This paper attempts to understand the cost-benefit of *partial evaluation policy* i.e. aborting evaluation of constraints if the solution has already violated one constraint. Above discussion becomes more relevant in the context of optimization problems involving computationally expensive constraint evaluations. The study assumes that the constraints can be evaluated independently of one another.

An optimization algorithm has been introduced based on a partial evaluation policy. The solutions in the population are evaluated based on a random sequence of constraints. The search using multiple constraint sequences offer the potential to reach different regions of the search space. The proposed scheme has been implemented using a framework based on differential evolution [3].

GECCO'13 Companion, July 6–10, 2013, Amsterdam, The Netherlands.
ACM 978-1-4503-1964-5/13/07.

2. PROPOSED ALGORITHM

A population of N individuals is initialized. The variables of i^{th} individual are initialized as follows:

$$x_{j,i} = x_{j,min} + rand_{i,j}[0,1).(x_{j,max} - x_{j,min}) \quad (1)$$

where $j = 1, 2, ...D$ is the number of variables; $x_{j,max}$ and $x_{j,min}$ are the upper and the lower bounds of j^{th} variable. For a problem with m constraints, each individual is assigned a random sequence of constraints for evaluation. Every individual of the population is evaluated using its prescribed constraint sequence. Whenever a constraint is violated, the evaluation is aborted. The term *number of function evaluations* referred in the paper is the sum of the number of evaluated constraints and objective function evaluations [4].

In order to generate an offspring solution, the firs parent is selected sequentially, the second and third parents are selected randomly from the entire population. In the recombination process, a binomial crossover [2] has been used to generate the offspring solution. The fitnes of a solution is determined as follows:

$$fitness(\xi) = \begin{cases} f(\mathbf{x}), & \mathbf{x} \in \Re^n \\ c_i, & i = 1, 2, ...m \end{cases} \quad (2)$$

where c_i is the constraint violation measure of m number of constraints. The equality constraints are transformed into a set of inequalities as $|h_j(\mathbf{x}) - \delta| \leq 0$ (assuming δ is small positive quan-

Algorithm 1 DE-CS

SET: NT_{max}{Total number of function evaluation}, N{Size of population}, CR{Crossover rate}, F{A Mutation scale factor}, $Evalcount = 0$

Initialize the population of individuals and assign a random constraint sequence to each individual;
Evaluate the solutions following the above assigned sequence of constraints; $Update(Evalcount)$;
while $(Evalcount \leq NT_{max})$ **do**
  for *i=1:N* **do**
    Select $P_1 = i$ i.e. the i^{th} parent and two other parents P_2 and P_3 randomly s.t. $P_1 \neq P_2 \neq P_3$;
    Generate an offspring using recombination;
    Evaluate the offspring using the sequence of P_1; $Update(Evalcount)$;
    The offspring is compared with solutions in the population for replacement based on fitness
  end for
end while

*$Evalcount$ denotes the sum of objective and all individual constraint evaluations

tity). Here, c_i denotes the constraint satisfaction vector where

$$c_i = \begin{cases} 0, & \text{if } i^{\text{th}} \text{ constraint satisfied } i = 1, 2, ..., m \\ g_i(\mathbf{x}), & \text{if } i^{\text{th}} \text{ constraint violated, } i = 1, 2, ..., q \\ |h_j(\mathbf{x}) - \delta|, & \text{if } i^{\text{th}} \text{ constraint violated, } i = q+1, ..., m \end{cases} \tag{3}$$

For every solution in the population, one can compute the number of satisfie constraints (NS) and the amount of violation (V). With the number of constraints satisfie taking a precedence over the violation value, a sorting would yield the ranks of the individual solutions. For example assume a population, containing 4 solutions (S1, S2, S3, S4). The constraint violation matrix would assume a form illustrated in Table 1 with S3 identifie as the best and S1 the worst.

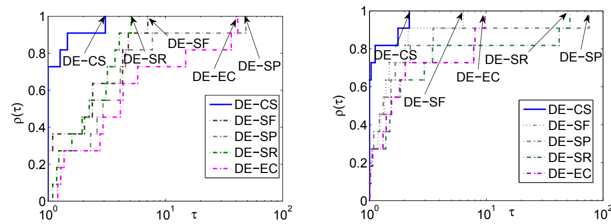
Table 1: Ranking of 4 individuals in the population in presence of 3 constraints

Initial order				NS	V	Final rank
$S1\ (g_1, g_2, g_3)$	5	–	–	0	5	4
$S2\ (g_2, g_3, g_1)$	0	3	–	1	3	2
$S3\ (g_1, g_3, g_2)$	0	0	1	2	1	1
$S4\ (g_2, g_1, g_3)$	2	–	–	0	2	3

3. EXPERIMENTAL RESULTS

The above section illustrated the principles of constraint sequencing and partial evaluation. In this section we objectively evaluate its performance on *CEC-2006* [5] benchmarks. We also include the results obtained by using stochastic ranking (SR) [6], self adaptive penalty (SP) [7], superiority of feasibility (SF) [8] and epsilon constraint (EC) [9] within the same framework of DE. Results based on performance profile are included for a more objective comparison. A population size of 50 is used for all the problems and the results are computed based on 30 independent runs. A f xed value of $CR = 0.9$ and $F = 0.5$ have been set for all the cases resulting the number of function evolutions (i.e. *NFEs*) of $4800 * (N * m)$, where N is the size of the population and m is the number of constraints.

In this experiment, we observe how quickly a feasible solution appears in the population–the function evaluation to reach a feasible solution and the computational time required to achieved the feasible solution. A performance profil [10] is computed for a more objective comparison between the strategies. The results clearly indicate the superiority of DE-CS over other strategies in terms of *NFEs* and computational time. Figure 1 shows the value of $\rho(\tau)$ for $r_{p,s} \leq \tau$ of the normalized performance ratio [10] i.e. (a) the number of function evaluation (b) computational time. One can observe from the figur that DE-CS outperforms with other strategies in terms of both.



(a) NFEs (b) computational time

Figure 1: Performance profile of DE-CS and others

4. CONCLUSION

In this paper, a scheme of constraint handling has been introduced within the framework of differential evolution utilizing the concepts of partial evaluation and constraint sequencing. The performance of the algorithm is subsequently assessed on 11 well known constrained single objective optimization benchmarks. The results on the test problems clearly indicate that the approach is computationally efficien and better than existing strategies for constraint handling.

5. ACKNOWLEDGEMENT

The second author would like to acknowledge the support of Future Fellowship offered by the Australian Research Council.

6. REFERENCES

[1] M. Schoenauer and S. Xanthakis, "Constrained GA optimization," in *Proceedings of the Fifth International Conference on Genetic Algorithms (ICGA-93)*, S. Forrest, Ed., University of Illinois at Urbana-Champaign. San Mateo, California: Morgan Kauffman Publishers, July 1993, pp. 573–580.

[2] C. A. C. Coello, "Theoretical and numerical constraint handling techniques used with evolutionary algorithms: A survey of the state of the art," *Computer Methods in Applied Mechanics and Engineering*, vol. 191, no. 11-12, pp. 1245–1287, January 2002.

[3] R. Storn and K. Price, "Differential evolution – a simple and efficien adaptive scheme for global optimization over continuous spaces," *Technical report TR-95-012, International Computer Science Institute, Berkeley, CA*, 1995.

[4] A. Asafuddoula, T. Ray, and R. Sarker, "A self-adaptive differential evolution algorithm with constraint sequencing," *in Proceedings AI 2012: Advances in Artificia Intelligence*, vol. 7691 of Lecture Notes in Artificia Intelligence, pp. 182–193, 2012.

[5] J. J. Liang, T. P. Runarsson, E. Mezura-Montes, M. Clerc, P. Suganthan, C. A. Coello, and K. Deb, "Problem definition and evaluation criteria for the cec 2006 special session on constrained real-parameter optimization," *Technical Report, Nanyang Technological University*, Dec 2005.

[6] T. P. Runarsson and X. Yao, "Stochastic ranking for constrained evolutionary optimization," *IEEE Transactions on Evolutionary Computation*, vol. 4, no. 3, pp. 284–294, September 2000.

[7] B. Tessema and G. G. Yen, "A self adaptative penalty function based algorithm for constrained optimization," in *2006 IEEE Congress on Evolutionary Computation (CEC'2006)*. Vancouver, BC, Canada: IEEE Press, July 2006, pp. 950–957.

[8] K. Deb, "An efficien constraint handling method for genetic algorithms," *Computer Methods in Applied Mechanics and Engineering*, vol. 186, no. 2/4, pp. 311–338, 2000.

[9] T. Takahama and S. Sakai, "Constrained optimization by the ϵ constrained differential evolution with an archive and gradient-based mutation," in *2010 IEEE Congress on Evolutionary Computation (CEC'2010)*. Barcelona, Spain: IEEE Press, July 18–23 2010, pp. 1680–1688.

[10] E. D. Dolan and J. J. More, "Benchmarking optimization software with performance profiles" *Mathematical Programming*, vol. 91, pp. 201–213, 2002.