Balancing Survival of Feasible and Infeasible Solutions in Evolutionary Optimization Algorithms

Zhichao Lu,∗, Kalyanmoy Deb∗, and Hemant Singh†
∗ Electrical and Computer Engineering
Michigan State University,
East Lansing, MI 48823, USA
{luzhicha,kdeb}@msu.edu
www.coin-laboratory.com
† Engineering and Information Technology,
The University of New South Wales,
Australian Defence Force Academy, Canberra ACT 2610, Australia
h.singh@adfa.edu.au
COIN Report Number 2018002

Abstract—Handling constraints in any optimization algorithm is as important as an algorithm’s progress towards the optimal solution. Population-based optimization algorithms allow a flexible way to handle constraints by making a careful comparison of feasible and infeasible solutions present in the population. A previous approach, which emphasized feasible solutions infinitely more than the infeasible solutions, has been popularly applied for more than a decade, mostly with real-parameter genetic algorithms (RGAs). In this paper, we extended the approach in a way so as to make a balance between survival of feasible and infeasible solutions in a population. And the balancing was controlled through an additional parameter ($\alpha$) that could be pre-specified or adaptively updated as the algorithm progresses. The approach was incorporated in the RGA, a significant improvement in performance was observed in the g-series test problem suite and a real-world application problem (welded-beam). A further analysis on the effect of $\alpha$ was performed and some standard values of $\alpha$ are suggested based on our empirical findings.

Index Terms—Differential evolution, real-parameter genetic algorithm, constraint handling.

I. INTRODUCTION

One of the main reasons for the popularity of evolutionary optimization (EO) algorithms is their population approach. Due to availability of multiple solutions, (i) EO constitutes an implicit parallel search which makes the convergence towards the optimal region in a fast manner [10], (ii) EO methods are capable and efficient in solving problems which possess multiple optimal solutions, such as in multimodal problems [9], [12], and multi-objective optimization problems [5], [2], [7], and (iii) EO methods allow a systematic comparison of good over bad solutions. It is the third aspect which we highlight in this paper and suggest an improved constraint handling procedure for population-based EO algorithms.

In 2000 [4], the second author proposed parameter-less constraint handling strategy by exploiting the population of solutions which has received a lot of attentions so far, despite existence of many other approaches [1], [11], [14], [15]. Besides its working on many problems, another attraction of the approach is that it is free of any additional parameter. In a tournament selection involving two population members, the approach selected any feasible solution over an infeasible solution, no matter how close the infeasible solution is to the constraint boundary. Although this worked in many problems, the approach has been criticized for its one-sided search effort in which search towards the optimum can only continue from the feasible region. After 17 years of its publication, we now propose a relaxation of the parameter-less approach in which certain infeasible solutions are provided better fitness over certain feasible solutions by introducing a parameter. However, over a number of standard constraint test problems, we have found a suitable value of the parameter which produced a better performance, not only in comparison with the original parameter-less approach, but also from most other parameter values. The balanced infeasibility-feasibility (BIF) approach is implemented in the real-parameter genetic algorithm (RGA) [6].

In the remainder of this paper, we present the main idea behind the proposed balanced infeasibility-feasibility based constraint handling EO (or BIF-EO) method in Section II. A description of the overall algorithm is also provided in the section. Then the experiment setup and test problem description are provided in subsection III-A. Thereafter, we present results of BIF-EO on RGA in subsection III-B and the parameter analysis in subsection III-C. Conclusions are drawn in Section IV.
II. Balanced Infeasibility-Feasibility Based EO (BIF-EO)

For a \( n \)-variable minimization problem with \( J \) inequality constraints of the following type (equality constraints are excluded here for brevity):

Minimize \( f(x) \),
subject to \( g_j(x) \leq 0, \ j = 1, \ldots, J \),
\[ x_j^{(L)} \leq x_i \leq x_j^{(U)}, \ i = 1, \ldots, n, \]
the overall constraint violation for a solution \( x \) is defined as follows:
\[
CV(x) = \sum_{j=1}^{m} \langle g_j(x) \rangle.
\]
where \( \langle \rangle \) returns the absolute value of the operand if the operand is positive, and returns a value zero otherwise. It is clear that \( CV(x) \) is a non-negative quantity and for feasible solutions \( CV(x) = 0 \). In most practical constrained optimization problems, one or more critical constraints become active (\( CV=0 \)) at the optimal solution, meaning that the optimum lies on the intersection of these critical constraints. Deb [4] defined a fitness function to handle constraints as follows:
\[
F(x) = \begin{cases} 
    f(x) & \text{if } CV(x) \leq 0, \\
    f_{\text{worst}} + f(x) & \text{otherwise}.
\end{cases}
\]
The parameter \( f_{\text{worst}} \) is not a user-defined parameter, but is the objective function value of the worst feasible solution in the current population. In the event of no feasible solution in the population, \( f_{\text{worst}} \) is set to zero. A careful look at the above equation suggests that the approach is parameter-less. Figure 1 illustrates the fitness assignment for five solutions (marked with circles on the \( x \)-axis, two feasible and three infeasible) in the presence of one constraint. The objective function \( (f(x)) \) is the dashed line, and the fitness function \( (F(x)) \) is the same dashed line for feasible solutions and the solid linear line for the infeasible solutions. It is clear that every feasible population member (objective value less than or equal to \( f_{\text{worst}} \)) has a better fitness value than any infeasible population member. This is due to the fact that every infeasible solution has a fitness of \( f_{\text{worst}} \) in addition to its constraint violation \( (CV(x)) \). This makes an infinite selection pressure for the feasible solutions in a population irrespective of the closeness of the infeasible solution to the constraint boundary.

The main idea of our proposed approach is to make a suitable balance between infeasible and feasible solutions by allowing some worse feasible solutions (having large \( f \)) to be worse than some infeasible solutions having low \( CV \) values. If this can be established, then certain near-constraint infeasible solutions will also be emphasized along with certain low-\( f \) feasible solutions, thereby instituting a more efficient search from both feasible and infeasible regions. Since population members can now be survived on both sides of the critical constraint boundaries, their recombination and mutation may result in a quicker discovery of near-optimal solutions.

Figure 2 illustrates our proposed approach.
\[
F(x) = \begin{cases} 
    f(x) & \text{if } CV(x) \leq 0, \\
    f_{\text{worst}} \cdot CV(x) + f_m & \text{otherwise,}
\end{cases}
\]
where
\[
f_m = f_{\text{worst}} - \alpha \cdot (f_{\text{worst}} - f_{\text{best}}),
\]
\[
CV(x) = \kappa \cdot (CV(x) - CV_{\text{best}}) + f_{\text{best}},
\]
\[
\kappa = \begin{cases} 
    1 & \text{if } (f_{\text{best}} = f_{\text{worst}}) \\
    f_{\text{worst}} - f_{\text{best}} & \text{if } (CV_{\text{best}} = CV_{\text{worst}}), \\
    CV_{\text{worst}} - CV_{\text{best}} & \text{otherwise}.
\end{cases}
\]
The parameters \( f_{\text{best}} \) and \( f_{\text{worst}} \) are the best and worst feasible objective value in the population and \( CV_{\text{best}} \) and \( CV_{\text{worst}} \) are the \( CV \)-values of the best and the worst infeasible solutions in the population. Note that these parameters are not user-supplied, but are computed from the current population statistics. However, the approach introduces a new parameter \( \alpha \), which must be set by the user. If \( \alpha = 0 \), \( f_m = f_{\text{worst}} \) and the approach is similar to the original parameter-less approach. However, if \( \alpha = 1 \), the infeasible solution with the best \( CV \)-value has equivalent fitness value as the feasible solution with the best function value, and all the infeasible solutions in the population becomes comparable with the feasible solutions. However, the above approach guarantees that even in the case of \( \alpha = 1 \), the fitness of the worst infeasible solution (having \( CV_{\text{worst}} \)) is worse than the fitness of the worst feasible solution (having \( CV_{\text{worst}} \) objective value).

The above mathematical description of the proposed method works as follows. The fitness function for the feasible region is identical to the objective function, as it was for the original strategy. However, for infeasible region the proposed method is different. First, the slope of a line joining the \((0, f_{\text{best}})\) and \((CV_{\text{best}}, f_{\text{worst}} + CV_{\text{best}})\) is computed. Then, a linear fitness function from \((CV_{\text{best}}, f_m)\) is extended for higher constraint-violated infeasible solutions. The solid red line in the figure shows the proposed fitness function for infeasible solution.

A description of the proposed algorithm is presented in Table I. It is clear that the procedure is systematic and easy to implement. It’s worth noting that both the original and our newly proposed constraint handling approaches are helpful when there are mix of feasible and infeasible members in the population. Otherwise, function values or constraint violations can be simply used as fitness values, in case of all population members are feasible or infeasible respectively.

III. RESULTS

In this section, we explain our experimental setup and test problems used in this study. Thereafter, we
Fig. 1. The parameter-less constraint handling scheme: five solid circles are solutions in a GA population.

Fig. 2. The new proposed constraint handling scheme: five solid circles are solutions in a GA population.

<table>
<thead>
<tr>
<th>Table I</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step-by-step procedure for our proposed constraint handling strategy.</strong></td>
</tr>
<tr>
<td><strong>Step</strong></td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
</tbody>
</table>

A. Experiment Setup

We evaluate the original and the new proposed constraint handling methods on ten benchmark functions that involve only inequality constraints, from the g-series constrained real-parameter test suite proposed in CEC 2006. A quick summary of the considered problems is provided in Table II, the detailed objective and constraint function description can be found in [13]. We also include a well-studied constrained real-world application problem in the literature, the Welded Beam problem[3]. The formulation of the Welded Beam problem consist of four variables $\vec{x} = (h,l,t,b)$ and five inequality constraints:

Minimize $f_w(\vec{x}) = 1.10471h^2l + 0.04811tb(14.0 + l)$,
subject to

$g_1(\vec{x}) = 13600 - \tau(\vec{x}) \geq 0,$
$g_2(\vec{x}) = 30000 - \sigma(\vec{x}) \geq 0,$
$g_3(\vec{x}) = b - h \geq 0,$
$g_4(\vec{x}) = P_c(\vec{x}) - 6000 \geq 0,$
$g_5(\vec{x}) = 0.25 - \delta(\vec{x}) \geq 0.$

(5)

The terms $\tau(\vec{x}), \sigma(\vec{x}), P_c(\vec{x})$, and $\delta(\vec{x})$ are given below:
The recombination mechanism in this RGA consists of other methods, and the sigma sharing functionality, discussed in the original parameter-less paper, is not used for this study. In this section, the results from our new proposed constraint handling method are shown with parameter \( a = 0.6 \) only, the effect of parameter \( a \) is discussed in the following section. The problem number used in the original parameter-less approach paper \([4]\) is provided in the parentheses.

According to the results summary in Table III, our new proposed method has a dominant advantage in the median function values achieved as compared to the parameter-less approach. Our new method achieves better median function values in seven, even in three and slightly worst in one, of the 11 considered benchmark functions. Furthermore, our new proposed method with RGA manages to converge to be very close to the true optimum in g02 in some of runs; and on the other hand, the parameter-less approach fails in all of the 31 trials.

Figure 4 provides a justification for the better performance of SBX operator in the presence of near-constraint infeasible solutions. When a feasible (point C) and an infeasible (point F) is recombined by SBX, there is 50% probability of creating both children inside the parents (totalling 100%), thereby making an excellent opportunity to create solutions between the feasible and infeasible solutions. If both parents are near the optimum and they are on the opposite side of the optimum, SBX under \( \alpha > 0 \) performs well. SBX-like operator does not work well when compared to vector-wise (DE-like) operators when the search proceeds from one side towards the optimum.

\[
\tau(\bar{x}) = \sqrt{\left(\tau'(\bar{x})\right)^2 + \left(\tau''(\bar{x})\right)^2 + \frac{l\tau'(\bar{x})\tau''(\bar{x})}{\sqrt{0.25l^2 + (h+l)^2}}},
\]

\[
\sigma(\bar{x}) = \frac{504000}{\sqrt{l}}.
\]

\[
P_c(\bar{x}) = 64746.022(1 - 0.0282346t)tb^3,
\]

\[
\delta(\bar{x}) = \frac{2.1952}{t^b},
\]

where

\[
\tau'(\bar{x}) = \frac{6000}{\sqrt{l}}
\]

\[
\tau''(\bar{x}) = \frac{6000(14+0.5l)\sqrt{0.25l^2 + (h+l)^2}}{2(0.707l\sqrt{l^2 + 0.25(h+l)^2})}
\]

Each experiment consists of 31 independent runs of simulations with pre-specified parameters. The crossover and mutation probabilities are fixed at 0.9 and 0.1 respectively for all the experiments. The population size is proportional to the number of variables; we use ten individuals per variable for each of the problems. And the maximum generation allowed is 5,000.

**B. RGA Procedure**

To better understand the behavior characteristics of our new proposed method, we return to the original real-coded genetic algorithm (RGA) code, in which the parameter-less method was initially implemented \([4]\).

The RGA code progresses as shown in Figure 3. The recombination mechanism in this RGA consists of the simulated binary crossover (SBX) \([6]\) paired with a polynomial mutation \([8]\) to create offsprings. The constraint handling schemes (both the original and the newly proposed) are implemented in the fitness assignment stage, then the fitness values are used for selecting the parents in the subsequent stage. It is worth noting that this particular version of the RGA code is a non-elitist-preserving implementation of the genetic algorithm, in which the offspring population created from genetic operations immediately becomes the parents population for the next generation. As a result, the elitist at any generation need not necessary to survive to mate with other parents to reproduce offspring in the next generation. However, this code stores the elitist found so far separately for solution reporting purpose.

We first present the function values achieved by each method with RGA on the 11 benchmark functions considered. The parameter settings are identical for both methods, and the sigma sharing functionality, discussed in the original parameter-less paper, is not used for this

We use the same approach to record the function evaluations required for each method to achieve a certain accuracy level. The accuracy level is calculated as the percentage difference between the obtained solution and the best known solution. The number in the \(|\) denotes the number of successful trials out of 31 and the number in the \(\) denotes the percentage savings in function evaluations (negative number means extra instead of saving). From the results in Table IV, we could observe

![Fig. 3. Main components of the rga process, constraint handling process is performed in the Fitness Assignment stage.](image)

![Fig. 4. A sketch explaining the performance of SBX under predominantly feasible and a mix of feasible-infeasible populations.](image)
that our new proposed constraint handling method also helps in expediting the convergence of the RGA in majority of the test problems considered. In addition, our BIF constraint handling method helps improving the success rate of the RGA in reaching certain accuracy level (for example, in g02, g06, g18). First of all, the new method reaches 0.1% accuracy within the true optimum in nine of the 11 considered benchmark functions and on the other hand, the parameter-less method only manages to converge in six of them. Secondly, the new method achieve 1% accuracy level with fewer function evaluations in majority of the considered problems.

A typical example (g04) of the generational convergence comparison between the two methods are provided in Figure 5, with the corresponding generational feasible fraction of the parent population provided in Figure 6. From this example, one could clearly observes that the new proposed constraint handling method not only expedites the convergence (in terms of median function value accuracy) of the algorithm, but also provides more robust performance (smaller span of the error-bar) along the generations. And one could also observes that the new method preserves the infeasible solution candidates more aggressively than the parameter-less method depending on the parameter $\alpha$ value, as shown in Figure 6. Please aware that because of the non-elitist-preserving nature of the RGA algorithm, even the parameter-less approach is not necessarily reaching 100% feasibility in the parents population in all the problems.

Another example (g18), in which the new method takes more function evaluations to reach 1% accuracy within the true optimum, of the generational convergence and feasibility fraction comparison is shown in Figure 7 and 8. This example shows an interesting observation that promoting more infeasible solutions (the new method) could slow down the convergence at the beginning, but lead to a better solution in the end.

We provide the last example on the real-world application, the welded beam problem. One could clearly observe that our new constraint handling approach (with $\alpha = 0.6$) converges not only to a better quality optimum in a faster pace, but also with less fluctuations in the performance across the runs, as shown in Figure 9. And the Figure 10 compares the feasible fraction of the child population at every generations from the two constraint handling approaches considered in this paper.
TABLE IV
AVERAGED FUNCTION EVALUATION OVER 31 RUNS COMPARISON TO THE PARAMETER-LESS APPROACH FROM [4], ACCURACY IS DEFINED AS $\frac{|f(x) - f(x^*)|}{|f(x^*)|}$. (‘-’ INDICATES THAT NO RUN IS CONVERGED TO THAT ACCURACY LEVEL).

<table>
<thead>
<tr>
<th>Prob. No.</th>
<th>Max. F.E.</th>
<th>Parameter-less</th>
<th>New Approach (a=0.6)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>&lt;0.01</td>
<td>&lt;0.001</td>
</tr>
<tr>
<td>g04 (6)</td>
<td>250000</td>
<td>5.00E+01[31]</td>
<td>3.91E+03[31]</td>
</tr>
<tr>
<td>g08</td>
<td>100000</td>
<td>2.82E+02[28]</td>
<td>4.16E+02[27]</td>
</tr>
<tr>
<td>g09 (5)</td>
<td>350000</td>
<td>2.65E+03[31]</td>
<td>5.25E+03[31]</td>
</tr>
<tr>
<td>g24</td>
<td>100000</td>
<td>2.42E+02[31]</td>
<td>8.29E+02[31]</td>
</tr>
<tr>
<td>Weld</td>
<td>400000</td>
<td>4.05E+03[31]</td>
<td>2.83E+04[31]</td>
</tr>
</tbody>
</table>

Fig. 7. Convergence plot of test problem g18 showing accuracy achieved $|f(x) - f(x^*)|$ versus generations, markers (‘o’ and ‘x’) indicate the mean, and the error bars indicate the min. and max. over 31 runs.

Fig. 8. Feasibility plot of test problem g18 showing the averaged feasible fraction over 31 runs of the child population versus generations.

Fig. 9. Convergence plot of welded beam problem showing accuracy achieved $|f(x) - f(x^*)|$ versus generations, markers (‘o’ and ‘x’) indicate the mean, and the error bars indicate the min. and max. over 31 runs.

Fig. 10. Feasibility plot of welded problem showing the averaged feasible fraction over 31 runs of the child population versus generations.

C. Parameter Study

As explained in previous sections, our new proposed constraint handling method requires an additional parameter $\alpha$ to the standard genetic algorithm parameters (population size, crossover probability, etc.). This parameter controls the aggressiveness of including infeasible solutions, whose value ranges in the interval [0,1]. Setting $\alpha$ to zero makes the infeasible solution with the best CV-value having the same fitness as the feasible solution with the worst function value, essentially, equivalent to the parameter-less method. On the other hand, setting $\alpha$ to one makes the infeasible solution with the best CV-value having the same fitness value as the feasible solution.
with the best function value, as a result, all infeasible solutions become comparable to the feasible solutions. However, our approach ensures the infeasible solution with the worst CV-values at any generation is worse than the feasible solution with the worst function value regardless of the value of $\alpha$.

To better understand the effect of $\alpha$ to our BIF-EO, we have conduct 31 independent runs using our approach with a value from 0 to 1 with 0.1 step size on each of the 11 considered test problems. For brevity, we only present the empirical results with $\alpha = [0.2, 0.4, 0.6, 0.8, 1.0]$ comparing to the parameter-less approach in six of the test problems in Table V - VI. And the convergence comparison from the median run is also provided in Figure 11 - 16.

Based on our experiments, a marginal improvement can be observed from using small values ($< 0.2$) of $\alpha$ and as the value of $\alpha$ increases, the improvement over the original parameter-less approaches also increases and peak around value 0.5 or 0.6. Hence, we suggest setting the $\alpha$ to its mid-range values [0.4, 0.7] as the initial trial for most of the test problems in general. Furthermore, the most aggressive settings of $\alpha (= 1)$ results in failure of solving many of the problems, hence, setting $\alpha$ to 1 is strongly not recommended as the initial trial for solving any arbitrary problems.

IV. CONCLUSIONS

In this paper, we have proposed an extension of a popular parameter-less constraint handling approach proposed more than a decade ago. The approach attempts to make a balance between survival of infeasible and feasible solutions in an evolving population. The idea is appealing, particularly for problems, in which an one-sided search from feasible region may turn out to be inefficient. When both approaches are implemented with a real-parameter genetic algorithm, the proposed balanced approach has produced much better results than the original parameter-less approach. An investigation of the working principles of RGA method reveals that whereas the presence of points around an optimum (whether feasible or infeasible) makes a significance difference in creating near-optimal solutions by the very nature of RGA’s recombination (SBX) operator. A parametric study has also shown that a specific range of values of the balancing parameter performed well on most test problems considered in this study. The performance enhancement with the proposed approach on the ten benchmark function of g-series test suite and the welded beam problem justified our initial insertion about the much needed balance between infeasible and feasible solutions in certain EO algorithms. For brevity, only problems with inequality constraints are studied in this paper, however, problems with equality constraints can be firstly transformed into inequality constraints, then applied our proposed method. Furthermore, our proposed BIF constraint handling approach can be easily extended to non-dominated-sorting based EAs like NSGA-II to solve constrained multi-objective problems.

REFERENCES

TABLE V
Parameter a study in test problem g01, g06 and g10, results compare the best, median and worst function value achieved with various a settings.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>g01</th>
<th>g06</th>
<th>g10</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Best</td>
<td>Median</td>
<td>Worst</td>
</tr>
<tr>
<td>Parameter-less</td>
<td>-15.0000</td>
<td>-15.0000</td>
<td>-12.4528</td>
</tr>
<tr>
<td>a = 0.2</td>
<td>-15.0000</td>
<td>-15.0000</td>
<td>-12.4528</td>
</tr>
<tr>
<td>a = 0.4</td>
<td>-15.0000</td>
<td>-15.0000</td>
<td>-12.4529</td>
</tr>
<tr>
<td>a = 0.6</td>
<td>-15.0000</td>
<td>-15.0000</td>
<td>-12.4529</td>
</tr>
<tr>
<td>a = 0.8</td>
<td>-15.0000</td>
<td>-15.0000</td>
<td>-12.4530</td>
</tr>
<tr>
<td>a = 1.0</td>
<td>-13.4735</td>
<td>-7.4998</td>
<td>-4.4864</td>
</tr>
</tbody>
</table>

TABLE VI
Parameter a study in test problem g04, g18 and g24, results compare the best, median and worst function value achieved with various a settings.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>g04</th>
<th>g18</th>
<th>g24</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Best</td>
<td>Median</td>
<td>Worst</td>
</tr>
<tr>
<td>Parameter-less</td>
<td>-30665.5371</td>
<td>-30665.5352</td>
<td>-30665.5312</td>
</tr>
<tr>
<td>a = 0.2</td>
<td>-30665.5391</td>
<td>-30665.5352</td>
<td>-30665.5312</td>
</tr>
<tr>
<td>a = 0.4</td>
<td>-30665.5371</td>
<td>-30665.5352</td>
<td>-30665.5312</td>
</tr>
<tr>
<td>a = 0.6</td>
<td>-30665.5391</td>
<td>-30665.5371</td>
<td>-30665.5352</td>
</tr>
<tr>
<td>a = 0.8</td>
<td>-30665.5371</td>
<td>-30665.5371</td>
<td>-30665.5352</td>
</tr>
<tr>
<td>a = 1.0</td>
<td>-30665.541</td>
<td>-30665.5352</td>
<td>-30665.4551</td>
</tr>
</tbody>
</table>

Fig. 11. Convergence comparison of test problem g01 showing accuracy achieved ($|f(x) - f(x^*)|$) versus generations.

Fig. 12. Convergence comparison of test problem g06 showing accuracy achieved ($|f(x) - f(x^*)|$) versus generations.

Fig. 13. Convergence comparison of test problem g10 showing accuracy achieved ($|f(x) - f(x^*)|$) versus generations.

Fig. 14. Convergence comparison of test problem g04 showing accuracy achieved ($|f(x) - f(x^*)|$) versus generations.

Fig. 15. Convergence comparison of test problem g18 showing accuracy achieved ($|f(x) - f(x^*)|$) versus generations.

Fig. 16. Convergence comparison of test problem g24 showing accuracy achieved ($|f(x) - f(x^*)|$) versus generations.