Improving Clonal Colony Optimization to evolve Robust Solutions

Jorge Maturana*
Universidad Austral de Chile

Fernando Vergara
Universidad Austral de Chile

Cristian Rojas
Universidad Austral de Chile

Frédéric Saubion
Université d'Angers, France

Felipe Vargas
Universidad Austral de Chile

Daniel Vidal
Universidad Austral de Chile

ABSTRACT

In this article we work with a recently introduced metaheuristic for robust optimization, inspired by the structure and behavior of biologic clonal colonies. We propose some improvements to increase their exploration and the exploitation capabilities. Our approach is compared to other robust optimization techniques, focusing in how the population is managed during the search.

Categories and Subject Descriptors

I.2 [**Artificial Intelligence**]: Miscellaneous—*Algorithms, Evolutionary computing*

Keywords

Robust optimization, Evolutionary computing, clonal colonies

1. INTRODUCTION

A central issue when solving real world problems is to deal with changing conditions. *Robust optimization* not only searches for a good quality solution, but also aims to maintain a high quality level in presence of unexpected changes in control variables, noisy data or approximations that replace expensive objective functions [3].

In real problems, it is desirable to find solutions with different trade-off between quality and robustness in order to provide decision makers with a wide range of alternatives to choose from. Different methods can be used in robust optimization, including explicit or implicit averaging or considering quality and robustness in a mutiobjective approach [1]. In this article we use *Clonal Colony Optimization* (CCO) [5], a recently proposed metaheuristic to perform robust optimization.

2. CLONAL COLONY OPTIMIZATION

Biologic clonal colonies are composed by a set of linked plants which collaborate by sharing resources. Plants specialize, ranging from exploiting resources on rich zones to explore the environment with shots. The whole organism is thus able to both efficiently explore wide areas and benefit from rich zones of the soil.

CCO maintains several solutions (plants), grouped in linked sets (colonies), which cover different zones of the search space. These colonies collaborate by sharing their fitness and evolve through iterative cycles of destruction and extension, as outlined in the following pseudo-code.

```
Program CCO()
  Seed()
  while(stop condition is not satisfied)
    ForEach(Colony c in the yard)
      Purge(c)
      Extend(c)
      Split(c)
    Reallocate()
  Distill()
```

The `Seed` method initially spreads a number of plants in the search space, that grow up to form entire colonies later. `Purgue` erases plants, with a bias to select both old and low-quality ones. `Extend` restores erased plants, by allowing the better ones (w.r.t. fitness) to give birth to new plants nearby. `Split` finds out whether a colony extends towards opposite directions in order to cut it and let both parts evolve separately. `Reallocate` ensures that the total number of plants simultaneously alive stay limited. Finally, `Distill` identifies the best plants in terms of both quality and robustness at the end of the search (see [5] for further details of CCO). Robustness of solutions in CCO can be assessed by measuring fitness homogeneity among plants in the same colony.

3. IMPROVEMENTS TO CCO

A key global parameter in CCO is δ, which is positively related to the size that the colonies are allowed to reach. Two innovations were included in CCO: (1) we have set δ dynamic and colony-level, with lower values in good zones and higher in bad ones, to respectively encourage a finer exploitation or a quick exploration; and (2) prevent colony overlaps to avoid over-exploiting the same area. If more than 50% of the area covered by the colony A is also covered by the colony B and the mean fitness of A is lower than these of B, A is erased and replaced by a seed randomly placed in the search space.

Comparisons with previous version of CCO have shown that these modifications increase exploration levels without

*Corresponding author, *jorge.maturana@inf.uach.cl*. This research was supported by Project DID UACh S-2010-50 and Project ECOS-CONICYT C10E07

GECCO'12 Companion, July 7–11, 2012, Philadelphia, PA, USA.
ACM 978-1-4503-1178-6/12/07.

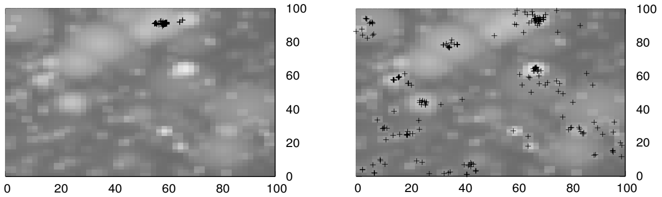

Figure 1: Typical placement of individuals once GA (GA2 shown at left) and CCO (right) are finished. Brigther zones stands for higher fitness values.

compromising the quality. This is only observed when both methods are combined, since no good results were obtained by applying them separately. We used a random-generated search space in $\Re^2$ containing local optima with different levels of robustness, inspired from NK fitness landscape [4].

4. BENCHMARK COMPARISON

We have compared our approach with three robust optimization approaches based on Genetic Algorithms, among those presented in [2]. *GA1* (single disturbed evaluation) evaluates disturbed individuals instead of actual ones, to penalize individuals in rugged zones. *GA2* (reevaluate only the best) evaluates the whole population once, then reevaluates 20% of the best individuals another 4 times to calculate a mean value, next 20% 3 times and so on. *GA3* (using neighbors) calculates a weighted mean of fitness individuals in the population, decreasing the weight as individuals fall apart. Algorithms stop when 10 000 calls to fitness function were done. Both GAs and CCO have 200 individuals (or plants) over a search space with 800 local optima.

Figure 1 shows the typical distribution of individuals at the end of GAs and CCO. Note that CCO plants are more diversified and placed in optima with different trade-off of quality and robustness, as expected.

Figure 2 summarizes 30 executions of each algorithm. The x axis represents the number of invocations made to the objective function, divided into eight intervals to show the evolution of the population in terms of quality. Each column is divided into quartiles (boxes) according to their fitness (y axis). The width of boxes is proportional to phenotipical diversity, i.e., thinner boxes indicate higher robustness.

GA1 obtains solutions with a fixed trade-off of quality and robustness. Despite the robustness of the solution found, GA1 notoriously dismiss high-quality individuals, due to its evaluation schema. *GA2* obtains the better results among GAs, somewhat favoring quality over robustness, and showing that making several calls to the evaluation function in one generation may be a good investment. *GA3* is similar to *GA1*, showing a notorious concentration in central quartiles, but without neglecting (nor improving) the best individuals. *CCO* obtains better values than GAs, without losing genotypic diversity. Solutions in the range $[0.2, 0.4]$ are as robust as those found by GA1, but the quality improvement is substantially better that those of GAs while phenotypic diversity remain high, as the height of boxes suggest.

5. CONCLUSIONS

A new configuration of CCO has been proposed, combining an adaptive δ parameter and a criterion to avoid overlapping colonies, increasing its capacity to explore without compromising the quality of obtained solutions.

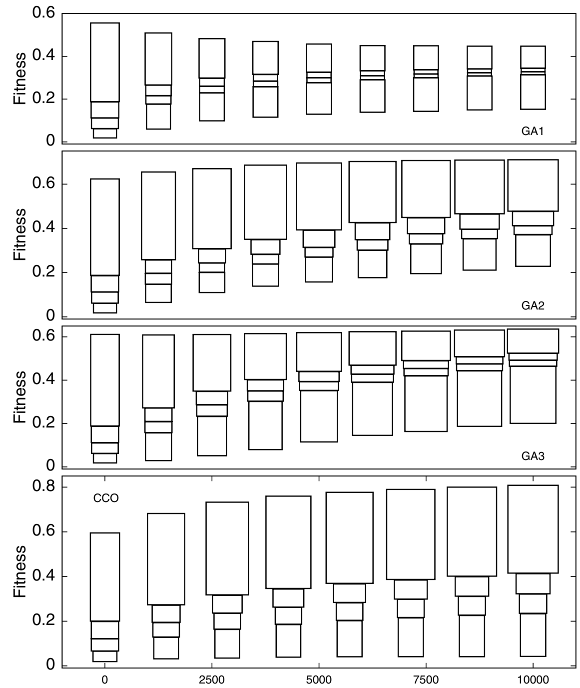

Figure 2: Average fitness distribution and diversity throughout the search for tested algorithms.

We verified the benefits of our approach w.r.t specialized GAs, namely the maintenance of diversity, in order to obtain solutions with different trade-off of quality and robustness, to provide the user with more alternatives of solution from a single run. Dynamic δ allows to specialize the search, exploiting in good zones and exploring in less interesting ones. CCO is able to perform intensification and diversification at the same time. In the future we plan to scale CCO to deal with multi-dimensional problems and different encodings, such as permutations.

6. REFERENCES

[1] H.-G. Beyer and B. Sendhoff. Robust optimization – a comprehensive survey. *Computer Methods in Appl.Mech. and Eng.*, 196(33–34):3190–3218, 2007.

[2] J. Branke. Creating robust solutions by means of evolutionary algorithms. In *Proc. of the 5th Intl.Conf. on Parallel Problem Solving from Nature*, 1998.

[3] K. Deb and H. Gupta. Searching for robust pareto-optimal solutions in multi-objective optimization. *Evol. Multicrit.Opt.*, 3410:150–164, 2005.

[4] S. Kauffman and S. Levin. Towards a general theory of adaptive walks on rugged landscapes. *J. Theor. Biol*, 128 (1):11–45, 1987.

[5] J. Maturana and F. Vergara. Robust optimization by means of vegetative reproduction. In *Intl.Conf. on Adapt. and Intell.Sys., LNAI 6943*, pages 404–415. Springer, 2011.