A Reliable and Computationally Cheap Approach for Finding Robust Optimal Solutions

Seyedali Mirjalili*
School of Information and Communication Technology, Griffith University, Australia
ali.mirjalili@gmail.com

Andrew Lewis†
School of Information and Communication Technology, Griffith University, Australia
a.lewis@griffith.edu.au

ABSTRACT

Current robust optimisation techniques can be divided into two main groups: algorithms that rely on previously sampled points versus those that need additional function evaluations to confirm robustness of solutions during optimisation. This paper first identifies and investigates the drawbacks of these two methods: unreliability for the first and excessive computational cost for the second. A novel approach is then proposed to alleviate the drawbacks of both methods. The proposed method considers the number of suitable, previously sampled points in the parameter space as a key metric to decide whether a solution can be assumed to be a robust solution when relying on previously sampled points. This factor is treated as a constraint that prevents solutions with low numbers of suitable, previously sampled points from participating in the improvement of the next population. To prove the effectiveness of the proposed algorithm, the proposed method is implemented for Particle Swarm Optimisation (PSO) and applied to several test functions from the literature. The results show that the proposed approach is able to effectively improve the reliability of algorithms that rely of previously sampled points without the need for extra function evaluations.

Categories and Subject Descriptors

I.2.8 [**Computing Methodologies**]: ARTIFICIAL INTELLIGENCE—*Problem Solving, Control Methods, and Search*; G.1.6 [**Mathematics of Computing**]: NUMERICAL ANALYSIS—*Optimization*

Keywords

Robust optimisation; Uncertainty; Particle Swarm Optimization

*Seyedali Mirjalili is with the School of Information and Communication Technology, Griffith University, Australia.

†Andrew Lewis is with the School of Information and Communication Technology, Griffith University, Australia.

GECCO '15 July 11-15, 2015, Madrid, Spain

ACM ISBN 978-1-4503-3488-4/15/07.

DOI: http://dx.doi.org/10.1145/2739482.2764640

1. INTRODUCTION

The current robust optimization techniques [2, 6, 1, 3] can be divided into two main groups: algorithms that rely on previously sampled points versus those that need additional function evaluations to confirm robustness of solutions during optimisation. The literature lacks consideration of the status of sampled points in the vicinity of trial solutions during optimisation. This paper first identifies and investigates the drawbacks of these two methods: unreliability for the first and excessive computational cost for the second. A novel approach is then proposed to alleviate the drawbacks of both methods. It should be noted that this work only concentrates on handling uncertainty in parameters.

2. PROPOSED METHOD

Basically, the status of sampled points is defined based on the number, distribution, and distance to the trial solution. The best and most desirable case is to have many sampled points in the proximity of trial solutions with a uniform distribution. Among these three factors, the number of sampled points is more important. This is because the majority of meta-heuristics randomly explore and exploit the search space, so the distribution of the sampled points during optimisation can be considered as approximately uniform, given a sufficiently large number of samples. Also, the robustness of solutions is usually investigated based on the maximum possible perturbation in parameters, so the neighbourhood of a solution, within which robustness should be investigated, is fixed during optimisation. Therefore, the distance of sampled points in the neighbourhood to the trial solution is not as important as their existence. However, the number of sampled points in the neighbourhood of solutions is very important due to its substantial impact on the robustness of solutions. An algorithm that assumes there is always enough sampled points near all trial solutions will fail to decide whether a solution is non-robust when there is no desired sampled solutions.

The number of sampled points is considered and emphasised in this paper. As the number of sampled points increases, obviously the reliability of an implicit method rises. However, it is not obvious how many sampled points is sufficient to reliably calculate the robustness of solutions. To consider the number of suitable sampled points during optimisation, this paper considers it as a constraint as follows:

$$c(\vec{x}) = H \tag{1}$$

where H is the number of sampled points in the vicinity of x solution.

3. RESULTS

This section first integrates the proposed method in a PSO algorithm [4] and names it Constraint-Based Robust PSO (CBRPSO). This algorithm is then applied to several test functions to benchmark its performance. The mechanism of CBRPSO is identical to that of a normal PSO, but an expectation measure is employed that calculates the average of a solution and the sampled points in the vicinity. All the sampled points are stored in a repository and used to decide the robustness of solutions. The results are compared to an RPSO with the same robustness measure and repository for verification. As case studies, 8 functions are collected from the literature [5]. It should be noted that the 10-dimensional version of all test functions are employed and we utilise 20 search agents over 150 iterations to approximate their robust optima.

The average and standard deviation of the difference between the optimum obtained and the robust optimum ($|X^* - X|$) over 30 runs are presented in Table 1. This table shows that the CBRPSO exhibits much better performance on the majority of the test functions. The first pattern that can be seen in the results is the high discrepancy of the results on the unimodal robust test functions. The proposed constraint discards non-robust solutions, so the exploitation of non-robust optima is intrinsically lower than the normal RPSO. This fact causes failure of the CBRPSO in obtaining robust optima of one unimodal test function. Since the CBRPSO performs better than RPSO in the multi-modal test functions, however, it seems that the lower exploitation is not a substantial issue. In fact, a lower level of exploitation can prevent the CBRPSO from stagnation in non-robust optima during optimisation. Fig. 1 shows that the centralised and guided exploration and exploitation of the proposed method is good enough to cover the promising regions of the search space. Table 1 shows that the discrepancy of the results between CBRPSOP and RPSO on the multi-modal test functions is also noticeable. The CBRPSO tends to outperform the RPSO algorithm on the majority of the test functions. This is due to the fact that the proposed constraints make non-robust solutions infeasible and prevent them from deceptively leading other particles toward non-robust regions of the search space. To further investigate the effects of the proposed constraints on discarding non-robust solutions, we monitored the number of times that there was less than 1 or 4 sampled points in the neighbourhood of solutions. It was observed that the proposed constraint discards the non-robust solution in nearly 73 % and 67 % of the times on average when considering the minimum of 1 and 4 sampled points respectively. This shows that the robust algorithms that rely on previously sampled points are highly prone to wrongly assuming non-robust solutions as robust due to the lack of sampled points. The results show that the proposed constraint is able to prevent an algorithm from favouring non-robust solutions during optimisation.

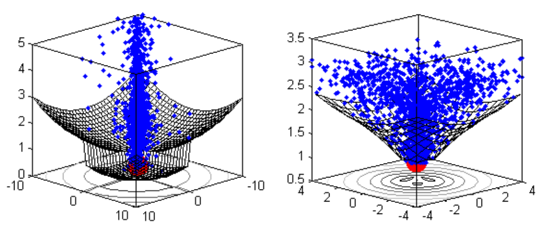

Figure 1: Centralised and guided exploration and exploitation of CBRPSO when solving TP1 and TP2

Table 1: Statistical results in the form of $ave \pm std$.

Test case	RPSO	CBRPSO
TP1	$0.0609 \pm 7.51E-05$	0.45 ± 0.199
TP2	$0.9600 \pm 4.80E-05$	0.0350 ± 0.0015
TP3	$0.3500 \pm 2.78E-05$	0.7950 ± 0.077
TP4	$0.37 \pm 2.306E-05$	0.299 ± 0.069
TP5	8.52 ± 1.3155	3.494 ± 0.84
TP6	7.278 ± 0.496	7.0550 ± 0.189
TP7	16.216 ± 0.344	15.636 ± 0.3118
TP8	128.989 ± 0.169	82.643 ± 17.360

4. CONCLUSIONS

The results showed that the current robustness measure and robust optimisation that relies on previously sampled points can be very vulnerable. From the results of the proposed CBRPSO algorithm, however, considering the number of sampled points as a constraint is able to improve the reliability of such methods significantly. The experimental results presented in this paper demonstrated that the proposed constraint prevents implicit sampling methods from making unreliable decisions during optimisation. Although making particles infeasible reduces exploration and exploitation, the results showed that this can be an advantage to prevent particles from being attracted toward non-robust solutions.

5. REFERENCES

[1] C. Barrico and C. Antunes. A new approach to robustness analysis in multi-objective optimization. In *7th International Conference on Multi-Objective Programming and Goal Programming (MOPGP 2006), Loire Valley, City of Tours, France, no. x*, pages 12–15, 2006.

[2] K. Deb and H. Gupta. Searching for robust pareto-optimal solutions in multi-objective optimization. *Lecture Notes in Computer Science*, 3410:150–164, 2005.

[3] A. Gaspar-Cunha, J. Ferreira, and G. Recio. Evolutionary robustness analysis for multi-objective optimization: benchmark problems. *Structural and Multidisciplinary Optimization*, 49(5):771–793, 2014.

[4] J. Kennedy and R. Eberhart. Particle swarm optimization. In *Neural Networks, 1995. Proceedings., IEEE International Conference on*, volume 4, pages 1942–1948 vol.4, Nov 1995.

[5] J. W. Kruisselbrink et al. *Evolution strategies for robust optimization.* Leiden Institute of Advanced Computer Science (LIACS), Faculty of Science, Leiden university, 2012.

[6] A. Saha, T. Ray, and W. Smith. Towards practical evolutionary robust multi-objective optimization. In *Evolutionary Computation (CEC), 2011 IEEE Congress on*, pages 2123–2130. IEEE, 2011.