Applied Soft Computing 35 (2015) 333–348

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Hindrances for robust multi-objective test problems

Seyedali Mirjalili[a,b,*], Andrew Lewis[a]

[a] School of Information and Communication Technology, Griffith University, Nathan, Brisbane, QLD 4111, Australia
[b] Queensland Institute of Business and Technology, Mt Gravatt, Brisbane QLD 4122, Australia

ARTICLE INFO

Article history:
Received 29 August 2014
Received in revised form 9 April 2015
Accepted 29 May 2015
Available online 24 June 2015

Keywords:
Benchmark problem
Robust optimisation
Uncertainties
Multi-objective robust optimisation
Robust benchmark problem
Multi-objective optimisation

ABSTRACT

Despite the significant number of benchmark problems for evolutionary multi-objective optimisation algorithms, there are few in the field of robust multi-objective optimisation. This paper investigates the characteristics of the existing robust multi-objective test problems and identifies the current gaps in the literature. It is observed that the majority of the current test problems suffer from simplicity, so five hindrances are introduced to resolve this issue: bias towards non-robust regions, deceptive global non-robust fronts, multiple non-robust fronts (multi-modal search space), non-improving (flat) search spaces, and different shapes for both robust and non-robust Pareto optimal fronts. A set of 12 test functions are proposed by the combination of hindrances as challenging test beds for robust multi-objective algorithms. The paper also considers the comparison of five robust multi-objective algorithms on the proposed test problems. The results show that the proposed test functions are able to provide very challenging test beds for effectively comparing robust multi-objective optimisation algorithms. Note that the source codes of the proposed test functions are publicly available at www.alimirjalili.com/RO.html.

1. Introduction

Real world problems include and are surrounded by a variety of uncertainties. Such uncertainties are considered undesirable inputs for a system. Unfortunately, undesirable inputs often have significant negative impacts on the outputs of a system. Frequently, a system designed under laboratory conditions shows one set of outputs, while the same system provides very different outputs in a real environment. Failure in many projects originates from overlooking such uncertainties. Uncertainties can be classified into four types [1,2]: operating conditions, inputs, outputs, and constraints.

In the field of optimisation, the process of considering uncertainties when finding optimal solutions is called robust optimisation. In robust optimisation, the ultimate goal is to obtain an optimal design for a particular problem while achieving the least sensitivity to possible perturbations of any kind. In the field of evolutionary optimisation, there are two main methods of robust optimisation. We will discuss them in detail in Section 2. The general concept is to employ a robustness measure to evaluate the robustness of search agents over the course of iterations [3–5]. Then appropriate actions are taken to guide the search agents of evolutionary algorithm towards the robust optimum [6–8].

In addition to uncertainties, another important characteristic of real problems is that of multiple objectives. It is quite common in real problems that there is more than one objective to be optimised. Multi-objective optimisation is a popular and recent field of study [9]. It deals with finding a set of solutions for a particular problem which represent the best possible trade-offs between the problem's objectives [10]. Although there are studies in the literature for converting a multi-objective problem to a single-objective problem [11–13], it has been proven that maintaining multi-objective formulation of problems allows designers to optimise problems with different conflicting/non-conflicting objectives across a wide range of design parameters [14].

Robust optimisation in a multi-objective search space is more challenging than in a single-objective search space. In this case, an optimiser should search for the best trade-offs between the objectives and consider their sensitivity to possible perturbations. Despite the importance of robust optimisation, unfortunately there is little in the literature. Comparing the theoretical and practical studies in the literature of single and multiple objective optimisation, there is a negligible number of publications in robust optimisation. So, it seems robust optimisation (specifically robust multi-objective optimisation) is still in its early stages. This paper identifies a substantial gap in the literature of robust multi-objective optimisation, which is the extreme simplicity of the majority of test problems. Challenging test problems are essential tools for developing and benchmarking algorithms in the field of evolutionary optimisation. Lack of such difficult test functions may result in premature comparison between different algorithms. In

* Corresponding author. Tel.: +61 434555738.
E-mail addresses: seyedali.mirjalili@griffithuni.edu.au (S. Mirjalili), a.lewis@griffith.edu.au (A. Lewis).

http://dx.doi.org/10.1016/j.asoc.2015.05.037
1568-4946/

addition, challenging diverse test beds allow us to effectively test the performance of algorithms from different perspectives. This motivates our attempts to investigate the characteristics of the current robust multi-objective test problems, introduce five hindrances, and propose a set of 12 diverse robust multi-objective test problems. The contribution of the paper is the proposal of deceptive robust test functions, multi-modal robust test functions, and flat robust test functions. The use of biased search spaces in the field of robust optimisation is another contribution of the paper. Since the paper describes the process of designing robust multi-objective test functions in detail, it can be considered as a practical tutorial for constructing robust test problems as well. The rest of the paper is organised as follows.

Section 2 discusses the concepts of robust optimisation in a multi-objective search space and current methods of handling uncertainties using evolutionary multi-objective optimisation algorithms. Section 3 analyses the current test problems in the literature of robust multi-objective optimisation and identifies their weaknesses. The hindrances are introduced and employed to design new test problems in Section 4. In addition, Section 4 investigates the characteristics of the proposed test functions theoretically and by generating random solutions. To provide a comprehensive study, five robust multi-objective algorithms are employed in Section 5 to approximate the robust fronts of the proposed test functions. Finally, Section 6 concludes the work and suggests some directions for future studies.

2. Robust multi-objective optimisation

As mentioned in the preceding section, four elements of a system that may face uncertainty are: operating conditions, inputs, outputs, and constraints. In the first type of uncertainties, environmental and operating conditions vary when the system is operating in the real environment [15]. Operating conditions are considered as secondary inputs for a system (indirect), so these uncertainties can also be considered secondary. Examples of this type are temperature or pressure of water when a submarine is navigating underwater. Another example would be the angle of attack, temperature of air, or wind speed when an aircraft is flying. Such operating conditions sometimes become very important, especially for systems that are critical to the safety of human operators.

The second type of uncertainties occurs in the inputs of the systems [15,16]. Inputs refer to the primary inputs of the systems (design parameters). Examples of design parameters are the number of coils in a tension spring, number/length of blades in a propeller, or shape of airfoils along an aircraft's wing. The parameters may diverge from design values, which mostly happens during manufacturing processes. In this case, the designed parameters vary during production and directly affect the outputs of a system.

Outputs of a system also have the potential to face uncertainties [17,18]. Although fluctuation of other components of a system perturbs the outputs, the system itself may produce noisy outputs as well. This type of uncertainty is mostly due to the usage of meta-models, approximated models, or simulators that are used to calculate the outputs of a system. Obviously, real problems are modelled in computers, so the model itself has a slight discrepancy from reality. An example of this type is the simulation's tolerances in Computational Fluid Dynamics (CFD) problems. The difference between this type and others is its deterministic nature. Although using more accurate approximated models or meta-models can reduce the perturbations' intensity, the system always produces intrinsic noise. There is also another kind of system worth mentioning, called dynamic systems. Such systems' outputs dynamically change over time, which cause different outputs for similar inputs and operating conditions [19]. In other words, time is a key input that defines the outputs of dynamic systems.

The last type of uncertainty is applied to the constraints of a system [1]. Designing a system is usually undertaken while considering several constraints and limitations. The constraints are not distinct from other components of a system and may face perturbations. Uncertainty in constraints does not have direct impact on the system, yet it varies the boundaries of the system. As a result of perturbation here, the outputs of a system may become invalid due to the violation of constraints.

In this paper we concentrate on the uncertainty in parameter, which is, indeed, one of the most common concerns in engineering design problems. As discussed in the introduction, this type of uncertainty detrimentally varies the inputs of a system. In a single-objective search space, robust optimisation can be formulated as a minimisation problems as follows (without loss of generality):

$$\text{Minimise}: \quad f(\vec{x}+\vec{\delta}) \tag{2.1}$$

$$\text{Subject to}: \quad g_i(\vec{x}+\vec{\delta}) \geq 0, \quad i = 1, 2, \ldots, m \tag{2.2}$$

$$h_i(\vec{x}+\vec{\delta}) = 0, \quad i = 1, 2, \ldots, p \tag{2.3}$$

$$L_i \leq x_i \leq U_i, \quad i = 1, 2, \ldots, n \tag{2.4}$$

where $\vec{x}$ is the set of parameters, $\vec{\delta}$ indicates the uncertainty vector corresponding to each variable in $\vec{x}$, m is the number of inequality constraints, p is the number of equality constraints, and $[L_i, U_i]$ are the boundaries of ith variable.

Due to the nature of single-objective optimisation problems, there is one robust global optimum for a certain level of uncertainty. The robust optimum may change for different degrees of uncertainty. Therefore, the ultimate goal of a robust evolutionary algorithm is to find the optimum that has the least sensitivity to perturbation of parameters. The robust optimum obtained can be either the global optimum or one of the local optima in the search space.

Robust optimisation in a multi-objective search space is different and can be formulated as a minimisation problem as follows:

$$\text{Minimise}: \quad f(\vec{x}+\vec{\delta}) = \{f_1(\vec{x}+\vec{\delta}), f_2(\vec{x}+\vec{\delta}), \ldots, f_o(\vec{x}+\vec{\delta})\} \tag{2.5}$$

$$\text{Subject to}: \quad g_i(\vec{x}+\vec{\delta}) \geq 0, \quad i = 1, 2, \ldots, m \tag{2.6}$$

$$h_i(\vec{x}+\vec{\delta}) = 0, \quad i = 1, 2, \ldots, p \tag{2.7}$$

$$L_i \leq x_i \leq U_i, \quad i = 1, 2, \ldots, n \tag{2.8}$$

where $\vec{x}$ is the set of parameters, $\vec{\delta}$ indicates the uncertainty vector corresponding to each variable in $\vec{x}$, o is the number of objective functions, m is the number of inequality constraints, p is the number of equality constraints, and $[L_i, U_i]$ are the boundaries of ith variable.

It may be observed in these equations that the only difference is the number of objectives to be optimised, while all of them receive perturbation as undesirable inputs. Due to the multiple objectives, robust optimisation has a different meaning here. In this case, robust optimisation refers to the process of finding a set of solutions representing the best trade-offs between the objectives with the lowest susceptibility to perturbations of all of the parameters [20]. In addition, the comparison of results cannot be done based on inequality comparison operators anymore. A solution in a multi-objective search space is better than another, if it shows equal or better results in all of the objectives (and is at least better in one objective) while considering given perturbations. This is called robust Pareto dominance and defined as follows [20]: