Asset Analytics
Performance and Safety Management
Series Editors: Ajit Kumar Verma · P. K. Kapur · Uday Kumar

Kusum Deep
Madhu Jain
Said Salhi *Editors*

Decision Science in Action

Theory and Applications of Modern Decision Analytic Optimisation

Springer

Asset Analytics

Performance and Safety Management

Series editors

Ajit Kumar Verma, Western Norway University of Applied Sciences, Haugesund, Rogaland, Norway
P. K. Kapur, Centre for Interdisciplinary Research, Amity University, Noida, India
Uday Kumar, Division of Operation and Maintenance Engineering, Luleå University of Technology, Luleå, Sweden

The main aim of this book series is to provide a floor for researchers, industries, asset managers, government policy makers and infrastructure operators to cooperate and collaborate among themselves to improve the performance and safety of the assets with maximum return on assets and improved utilization for the benefit of society and the environment.

Assets can be defined as any resource that will create value to the business. Assets include physical (railway, road, buildings, industrial etc.), human, and intangible assets (software, data etc.). The scope of the book series will be but not limited to:

- Optimization, modelling and analysis of assets
- Application of RAMS to the system of systems
- Interdisciplinary and multidisciplinary research to deal with sustainability issues
- Application of advanced analytics for improvement of systems
- Application of computational intelligence, IT and software systems for decisions
- Interdisciplinary approach to performance management
- Integrated approach to system efficiency and effectiveness
- Life cycle management of the assets
- Integrated risk, hazard, vulnerability analysis and assurance management
- Adaptability of the systems to the usage and environment
- Integration of data-information-knowledge for decision support
- Production rate enhancement with best practices
- Optimization of renewable and non-renewable energy resources

More information about this series at http://www.springer.com/series/15776

Kusum Deep · Madhu Jain · Said Salhi
Editors

Decision Science in Action

Theory and Applications of Modern Decision Analytic Optimisation

Springer

Editors
Kusum Deep
Department of Mathematics
Indian Institute of Technology Roorkee
Roorkee, Uttarakhand, India

Madhu Jain
Department of Mathematics
Indian Institute of Technology Roorkee
Roorkee, Uttarakhand, India

Said Salhi
Kent Business School, Centre for Logistics and Heuristic Optimization (CLHO)
University of Kent
Canterbury, Kent, UK

ISSN 2522-5162 ISSN 2522-5170 (electronic)
Asset Analytics
ISBN 978-981-13-0859-8 ISBN 978-981-13-0860-4 (eBook)
https://doi.org/10.1007/978-981-13-0860-4

Library of Congress Control Number: 2018948716

This Springer imprint is published by the registered company Springer Nature Singapore Pte Ltd.
The registered company address is: 152 Beach Road, #21-01/04 Gateway East, Singapore 189721, Singapore

Contents

π **Fraction-Based Optimization of the PBM Antenna Benchmarks** 1
Richard A. Formato

Benchmark Function Generators for Single-Objective Robust Optimisation Algorithms 13
Seyedali Mirjalili and Andrew Lewis

Convergence of Gravitational Search Algorithm on Linear and Quadratic Functions 31
Anupam Yadav, Anita and Joong Hoon Kim

An Algorithm of Multivariant Evolutionary Synthesis of Nonlinear Models with Real-Valued Chromosomes 41
Oleg Monakhov and Emilia Monakhova

An Artificial Bee Colony Based Hyper-heuristic for the Single Machine Order Acceptance and Scheduling Problem 51
Sachchida Nand Chaurasia and Joong Hoon Kim

A New Evolutionary Optimization Method Based on Center of Mass 65
Jesús-Adolfo Mejía-de-Dios and Efrén Mezura-Montes

Adaptive Artificial Physics Optimization Using Proportional Derivative Controllers 75
Liping Xie, Jianchao Zeng, Qiongqiong Yang and Richard A. Formato

NSGA-II Based Decision-Making in Fuzzy Multi-objective Optimization of System Reliability 105
Hemant Kumar and Shiv Prasad Yadav

GA-Based Task Scheduling Algorithm for Efficient Utilization of Available Resources in Computational Grid 119
Shipra Singh, Anuradha Aggarwal, Harendera Kumar and Pradeep Kumar Yadav

Statistical Feature Analysis of Thermal Images from Electrical Equipment 127
Tamal Dutta, Deepjyoti Santra, Chee Peng-Lim, Jaya Sil and Paramita Chottopadhyay

Performance of Sine–Cosine Algorithm on Large-Scale Optimization Problems 139
Puneet Kumar Pal, Kusum Deep and Atulya K. Nagar

Necessary and Sufficient Optimality Conditions for Fractional Interval-Valued Optimization Problems 155
Indira P. Debnath and S. K. Gupta

Application of Constrained Spider Monkey Optimization to Solve Portfolio Optimization Problem 175
Kavita Gupta, Kusum Deep and Atulya K. Nagar

Optimal Configuration Selection in Reconfigurable Manufacturing System 193
Kamal Kumar Mittal, Pramod Kumar Jain and Dinesh Kumar

A Comparative Study of Regularized Long Wave Equations (RLW) Using Collocation Method with Cubic B-Spline 203
Nini Maharana, A. K. Nayak and Pravakar Jena

An Enhanced Fractal Dimension Based Feature Extraction for Thermal Face Recognition 217
Sandip Joardar, Arnab Sanyal, Dwaipayan Sen, Diparnab Sen and Amitava Chatterjee

Seismic Analysis of Multistoried Building with Optimized Damper Properties 227
Dipti Singh, Shilpa Pal and Abhishek Singh

Effect of Upper Body Motion on Biped Robot Stability 237
Ruchi Panwar and N. Sukavanam

Ant Colony Algorithm for Routing Alternate Fuel Vehicles in Multi-depot Vehicle Routing Problem 251
Shuai Zhang, Weiheng Zhang, Yuvraj Gajpal and S. S. Appadoo

Semidefinite Approximation of Closed Convex Set 261
Anusuya Ghosh and Vishnu Narayanan

About the Editors

Dr. Kusum Deep is a professor in the Department of Mathematics, Indian Institute of Technology Roorkee. Her research interests include numerical optimization, nature-inspired optimization, computational intelligence, genetic algorithms, parallel genetic algorithms, and parallel particle swarm optimization.

Dr. Madhu Jain is an associate professor in the Department of Mathematics, Indian Institute of Technology Roorkee. Her research interests include computer communications networks, performance prediction of wireless systems, mathematical modeling, and biomathematics.

Dr. Said Salhi is Director of the Centre for Logistics and Heuristic Optimization (CLHO) at Kent Business School, University of Kent, UK. Prior to his appointment at Kent in 2005, he served at the University of Birmingham's School of Mathematics for 15 years, where in the latter years he acted as Head of the Management Mathematics Group. He obtained his B.Sc. in Mathematics from the University of Algiers and his M.Sc. and Ph.D. in OR at Southampton (Institute of Mathematics) and Lancaster (School of Management), respectively. He has edited six special journal issues and chaired the European Working Group in Location Analysis in 1996 and recently the International Symposium on Combinatorial Optimization (CO2016) in Kent from September 1 to 3, 2016. He has published over 100 papers in academic journals.

π Fraction-Based Optimization of the PBM Antenna Benchmarks

Richard A. Formato

Abstract Real-world optimization problems often require an external "modeling engine" to compute fitnesses, and these programs often have much longer runtimes than evaluating fitnesses solely with built-in compiler routines. Using a stochastic optimizer on real-world problems can be quite challenging because every run returns a different "best" fitness. This issue is addressed by making many runs, often hundreds, possibly even thousands, in order to generate meaningful statistics, but doing so can be prohibitive with external modeling. And even then the statistical nature of the results may obscure true global extrema. Additionally, real-world problems do not come with well-defined, clearly appropriate objective functions (at least most of the time). The practitioner must define an appropriate function, which in itself can be a daunting task made more difficult using a stochastic optimizer. π fractions mitigate these issues by introducing pseudorandomness in an otherwise truly random metaheuristic, for example, genetic algorithm. This paper illustrates the utility of π fractions by using them in two different optimizers, one deterministic and the other probabilistic. These optimizers are applied with quite good results to the PBM antenna benchmarks, a set of difficult real-world engineering problems, thereby demonstrating the utility of π fractions in all types of optimizers.

Keywords Optimization · Global search and optimization · π fractions · CFO GASR · Genetic algorithm · PBM · PBM antenna benchmarks · Antenna Deterministic algorithm · Stochastic algorithm · Pseudorandomness

R. A. Formato (✉)
Consulting Engineer and Registered Patent Attorney of Counsel, Emeritus
Cataldo & Fisher, LLC, PO Box 1714, Harwich, MA 02645, USA
e-mail: rf2@ieee.org

K. Deep et al. (eds.), *Decision Science in Action*, Asset Analytics,
https://doi.org/10.1007/978-981-13-0860-4_1

1 Introduction

The utility of π fractions in global search and optimization is investigated by applying the π fraction-based algorithms Central Force Optimization (πCFO) and Genetic Algorithm with Sibling Rivalry (πGASR) to the Pantoja et al. [1] benchmarks (PBM). PBM is a group of typical real-world engineering problems that do not have known analytical solutions. They are designed to test the effectiveness of antenna optimization algorithms using the Numerical Electromagnetics Code [2] (NEC) as the modeling engine. A major concern when external modeling is required is having to make multiple runs if a stochastic optimizer is employed, for example, genetic algorithm. π fractions mitigate this issue by making stochastic algorithms pseudorandom, in effect deterministic. πCFO and πGASR data are compared to the published PBM data and to CFO implemented without π fractions. The results are quite good. They demonstrate the general utility of π fractions in global search and optimization, in particular in rendering deterministic an otherwise probabilistic algorithm.

2 The PBM Suite

PBM comprises five problems in which the antenna directivity is the fitness (objective function) to be maximized. Each problem has a unique landscape (fitness' topology over the decision space, DS). Four of the problems are two-dimensional (2D), while the fifth is $(N_{\mathrm{el}}-1)$D where N_{el} is the number of elements in a co-linear dipole array. Table 1 summarizes PBM's characteristics (λ is the free space wavelength), and the appendix contains geometries for the five antennas and perspective landscapes for the four 2D problems.

Table 1 Properties of the PBM benchmark problems

PBM benchmark	Problem characteristics (in each case objective is to maximize directivity)
1	Variable length center-fed dipole. 2D, unimodal, single global maximum, strong local maxima
2	Uniform 10-element array of center-fed $\frac{\lambda}{2}$-dipoles. 2D, added Gaussian noise, single global maximum, multiple strong local maxima
3	Eight-element circular array of center-fed $\frac{\lambda}{2}$-dipoles. 2D, highly multimodal, four global maxima
4	Vee Dipole. 2D, unimodal, single global maximum, "smooth" landscape.
5	Collinear N_{el}-element array of center-fed $\frac{\lambda}{2}$-dipoles. $(N_{el}-1)$ D, unimodal, single global maximum

3 π Fractions

The π fractions comprise a set of random numbers extracted from the constant π that are uniformly distributed on the interval [0, 1). They are generated by the Bailey, Borwein, and Plouffe hexadecimal digit extraction algorithm [3]. In an inherently deterministic algorithm like CFO, the π fractions can improve DS exploration by adding a measure of pseudorandomness [4, 5]. Used in an inherently stochastic algorithm like GASR [6], the π fractions render the algorithm effectively deterministic.

A major advantage of determinism is that every optimization run with the same setup yields precisely the same results. This characteristic allows the algorithm designer to quickly and with certainty evaluate the effects of changes such as different run parameters or different fitness functions. By contrast, such evaluations using a stochastic algorithm require many independent runs, often hundreds or thousands, to generate adequate statistics, and even then the results are imprecise because of their statistical nature. Having to make so many runs can be a very serious limitation in real-world optimization problems, especially ones that require long-running external modeling engines (NEC being an example with highly segmented antennas) [5].

Alternatives to the π fraction approach include using a compiler's built-in random number generator or a separately coded routine employing the same "seed" value on successive runs in order to generate a repeatable sequence of "random" numbers. These approaches, however, run the risk of creating undesirable bi-dimensional correlations in high dimensionality decision spaces, an effect seen, for example, in Halton and van der Corput low-discrepancy sequences [7]. In contrast, undesirable correlations in the π fractions are readily avoided by not using them in their order of occurrence [7].

In the present work, algorithms πCFO and πGASR both employ the following π fraction pseudocode instead of using a compiler's built-in random number generator. This procedure generates random real numbers $a \le r_i < b$ and mitigates the correlation problems discussed above (note that initialization values are completely arbitrary).

π_i : {π *fractions in order of occurrence; i=1,.., N_π* }
Initialize: $N_\pi \leftarrow 215\,830$: $init_1 \leftarrow 17$
$init_2 \leftarrow 22$: $inc \leftarrow 5$: $i \leftarrow init_1$
Procedure R_π: *generates a random number in [a,b) using π fractions*
$r_i = a + (b-a)\,\pi_i$
$i \leftarrow i + inc$
if $i > N_\pi$ then $i \leftarrow init_2$
End

Pseudocode π fraction Random Numbers

4 Algorithms Compared

Data from three algorithms, CFO, πCFO, and πGASR, are compared to the published PBM data. The PBM suite was initially studied using completely deterministic CFO without the pseudorandomness introduced by π fractions [8]. Initial probe accelerations were set to zero, the repositioning factor $\left(F_{\text{rep}}\right)$ was initialized and incremented deterministically, and a deterministic "Probe Line" Initial Probe Distribution (IPD) was used (for details, see [8], which includes the source code listing). By contrast, the present version, πCFO, is pseudorandomized using π fractions to compute the initial probe accelerations, F_{rep}, and the IPD. Run parameters were the same as in [8] with DS shrinking and early termination checking. Only a single πCFO run was made for each benchmark because the algorithm remains completely deterministic with the π fractions.

The second randomized algorithm applied to PBM is πGASR (Genetic algorithm with sibling rivalry, discussed in detail in [7]). Because GASR is inherently stochastic, calls to the compiler's built-in random number generator were replaced by the π fraction procedure above (run parameters otherwise the same as in [7]). The best πGASR fitness over multiple runs is reported here because the very purpose of using π fractions is to eliminate true randomness by replacing it with deterministic pseudorandomness.

A question raised in the initial PBM/CFO study was how well NEC recovered the published PBM data. Generally, NEC4 recovers those results well, but there are some noteworthy differences discussed in detail in [8] and briefly summarized here. On PBM #1 and #2, NEC4's computed directivities are slightly lower. But on problems #3 and #4, they are lower by a wider margin. The best agreement is on problem #5 where the NEC4 and PBM data show very good agreement. An explanation of these discrepancies is elusive (see [8] for further discussion). There are several possibilities, ranging from different versions of NEC to compiler differences to slight but important differences in the antenna models (for example, excitation source modeling). The validation data in [8] show that NEC4 can be used effectively to assess an algorithm's performance against PBM, but precise agreement is not to be expected.

5 Results

Table 2 compares the best fitness results for algorithms CFO, πCFO, and πGASR to the published PBM data. While there are slight differences, overall the maximum directivity data are quite similar for all five problems. Consistency is very high on PBM #5 and somewhat lower on the others. In all cases, the best fitness is close to the reported PBM data, but, as pointed out above, some (minor) questions arise concerning the accuracy of some of the published PBM results. A fair reading of these data is that all the tested algorithms provide maximum directivities close to

Table 2 Best fitness

PBM benchmark	Maximum directivity			
	PBM	CFO	πCFO	πGASR
1	3.32	3.20627	3.24340	3.25837
2*a* (no noise) 2*b* (noisy)	18.3[a] nr[b]	18.3654 18.6880	18.2810 19.7609	17.9473 18.8314
3	7.05[a]	6.48634	6.57766	6.57658
4	5.8[a]	5.71479	5.29663	5.29663
5 (6 el)	~11.25[c]	11.2202	11.2202	11.2202
5 (7 el)	nr	13.1826	13.0918	13.1826
5 (10 el)	~19[b]	19.0985	19.0985	19.0985
5 (13 el)	*nr*	25.0611	25.0035	25.0035
5 (16 el)	~31[b]	30.9742	30.9742	30.9742
5 (24 el)	~47[b]	46.8813	46.8813	46.8813

Notes [a]values estimated from the figures in [1]
[b]nr—not reported in [1]
[c]values marked with ~ are estimated from Fig. 13 in [1]

the actual maxima, and that the π fraction approach is effective in making stochastic πGASR deterministic so that only a single run actually is required instead of many.

Coordinates for the best fitnesses are shown in Table 3. These data also show a high degree of consistency between the tested algorithms except for πGASR on PBM #2a where the x_1 coordinate is different from the other two algorithms. This disagreement also shows up to a lesser degree in πGASR's maximum directivity in Table 2. On PBM #5, the dipole element separations are all close to 0.99λ with the greatest variability returned by algorithm πCFO. The differences in d_i , however, have no effect on the best fitness as seen in the remarkably consistent data in Table 2.

6 Conclusion

π fractions have been shown to be an effective approach to mitigating the inconsistency inherent in stochastic global search and optimization. The case for using π fractions is made by way of example using the PBM antenna benchmarks as representative real-world problems requiring an external modeling engine to calculate fitnesses. Stochastic algorithms require multiple runs to build meaningful statistics, often runs numbering in the hundreds. This requirement usually is not a limitation when optimizing analytical benchmarks because built-in routines are used, but it easily becomes a very significant impediment when a long-running external modeling engine is required as often is the case for real-world problems. π fractions address this concern by rendering deterministic (pseudorandom) an otherwise probabilistic

Table 3 Best fitness coordinates

PBM benchmark	PBM		CFO		πCFO		πGASR	
	x_1	x_2	x_1	x_2	x_1	x_2	x_1	x_2
1	2.58λ	0.63	2.5509λ	0.6181	2.5896	0.6195	2.5845	0.6198
2*a* (no noise)	~5.85λ	1.5730	5.9236λ	1.5569	5.9246	1.5554	6.9270	1.5467
2*b* (noisy)	nr[a]	nr	6.9360λ	1.5472	5.8877	1.5560	9.8907	1.5230
3	0.5	1.5730	0.4802	1.5733	2.4806	1.5611	1.5201	1.5704
4	1.5λ	0.834	1.4952λ	0.7110	1.4913	0.7176	1.4942	0.7317
Min/Max $d_i, i = 1, \ldots, N_{el} - 1$								
5	0.99λ		0.983/1λ		0.974/1.199λ		0.987/1λ	

Notes [a]nr—not reported in [1]

optimizer so that only a single run is required. π fractions also are useful in inherently deterministic algorithms by injecting a measure of pseudorandomness that may improve an algorithm's ability to explore the decision space.

Appendix: PBM Antenna Geometries and Objective Function Landscapes

Benchmark #1: Variable Length Center-Fed Dipole

The antenna geometry for Problem #1 is shown in Fig. 1. The objective is to maximize a center-fed dipole's directivity, D, as a function of its total length, L, and the polar angle, θ. A perspective view of the 2D landscape appears in Fig. 2. It is smoothly varying with a single global maximum and two local maxima of similar amplitude.

Benchmark #2: Uniform Dipole Array

The problem #2 antenna is the uniform array of half-wave dipoles shown in Fig. 3. All elements are center-fed with in-phase equal amplitude sources. The figure also shows the standard right-handed Cartesian coordinate system used by NEC, as well as the polar angle θ and azimuth angle ϕ. The objective is to maximize directivity $D(d, \theta)$ in the plane $\phi = 90°$ as a function of element separation d and polar angle θ with and without the presence of additive Gaussian noise. Figure 4a shows the landscape without noise and Fig. 4b with it. As in [1], noise is generated by

Fig. 1 Dipole

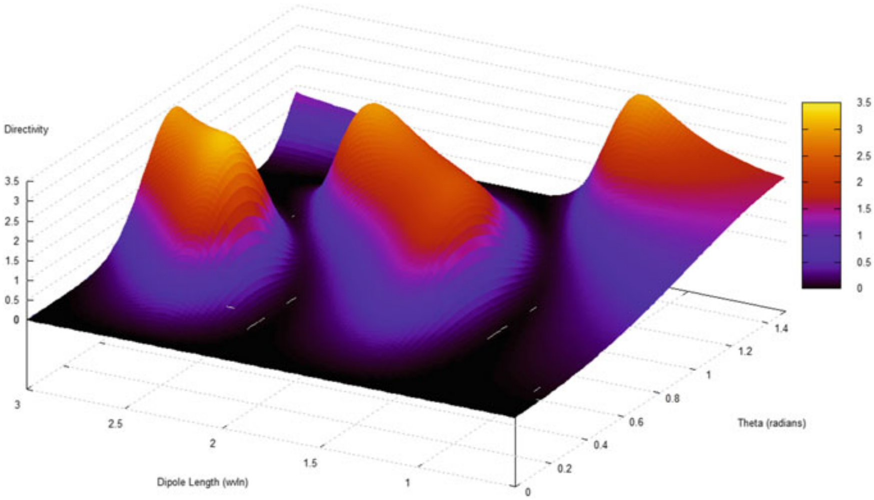

Fig. 2 Benchmark #1 topology, perspective view

adding to the NEC4-computed directivity a normally distributed zero-mean, 0.2-variance random variable z, here computed using the Box–Muller method [9, 10] as $z = \mu + \sigma\sqrt{-2\ln(s)}\,\cos(2\pi t)$, where μ and σ, respectively, are the mean (zero) and standard deviation (0.4472), and s and t are random variables uniformly distributed on [0, 1). s and t are generated using the π fraction random number pseudocode described above.

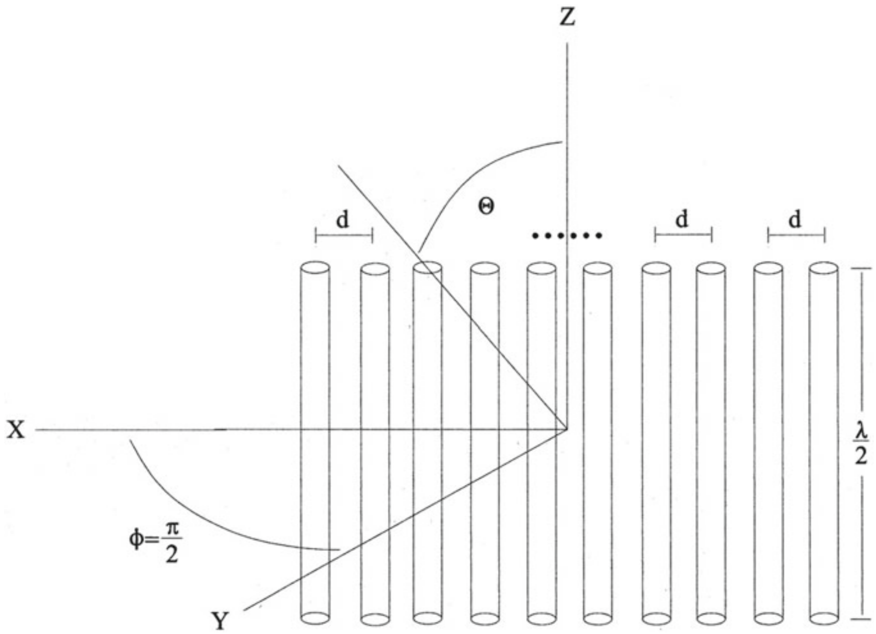

Fig. 3 Uniform array of half-wave center-fed in-phase dipoles

Benchmark #3: Circular Array of Half-Wave Dipoles

The antenna for problem #3 is the circular array of half-wave dipoles shown in Fig. 5. The array comprises eight dipoles parallel to the z-axis uniformly deployed on a one-wavelength radius circle. All elements are center fed by equal amplitude sources. But, following [1], the phase varies as $\alpha_n = -\cos\left[2\pi\,\beta\,(n-1)\right], n = 1, \ldots, 8$. The unit-amplitude excitation is therefore $V_n = \cos\alpha_n + j\,\sin\alpha_n$. The objective is to maximize the directivity $D(\beta, \theta)$ in the plane $\phi = 0°$ as a function of the dimensionless phase parameter $0 \le \beta \le 4$ and the polar angle θ. The range for β produces the four global maxima at $\left(\beta_i = i - 0.5, i = 1, \ldots, 4; \theta = \frac{\pi}{2}\right)$ as seen in the perspective topology plot in Fig. 6.

Benchmark #4: Vee Dipole

Benchmark #4 is the vee dipole antenna as shown in Fig. 7. It comprises two arms of equal length L_{arm} with inner angle 2α connected by a feed segment of length $2L_{\text{feed}}$ fed at its midpoint. The objective is to maximize the directivity $D(L_{\text{total}}, \alpha)$ along the $+X$-axis as a function of the total dipole length $0.5\lambda \le L_{\text{total}} = 2L_{\text{arm}}+2L_{\text{feed}} \le 1.5\lambda$ and the inner half-angle $\frac{\pi}{18} \le \alpha \le \frac{\pi}{2}$ with $L_{\text{feed}} = 0.01\lambda$.

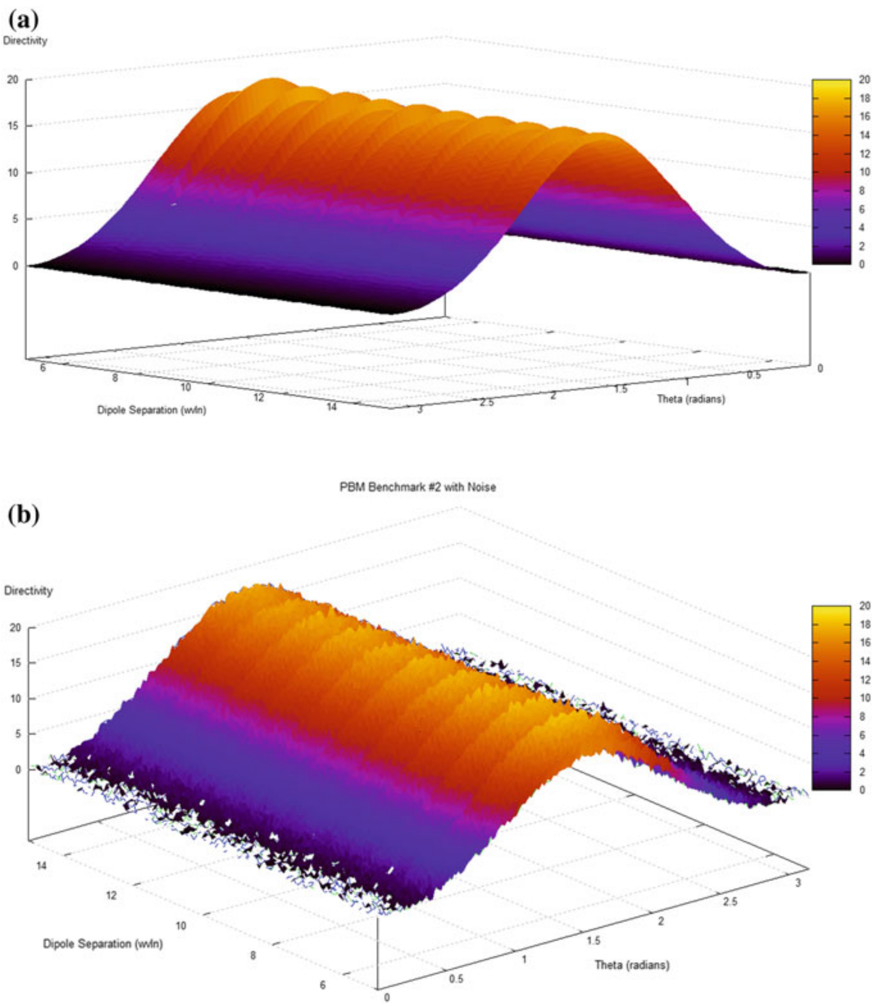

Fig. 4 **a** Uniform half-wave dipole array without noise, perspective view. **b** Uniform dipole array with additive Gaussian noise, perspective view

Topology of the Vee dipole's decision space appears in Fig. 8. This objective function is unimodal with a single global maximum at $D(L_{\text{total}}, \alpha) = (1.5\lambda, 0.834)$. The surface is smoothly varying without pronounced local maxima.

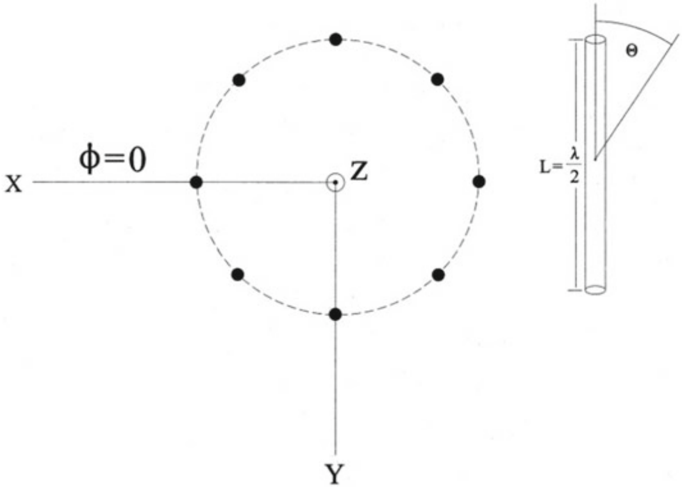

Fig. 5 Circular array of half-wave dipoles (1λ radius)

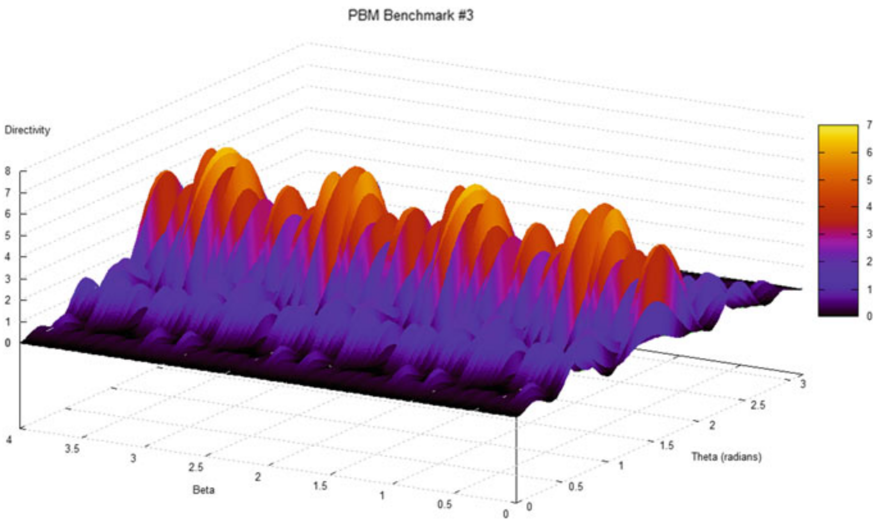

Fig. 6 Circular array landscape, perspective view

Benchmark #5:* N*-element Collinear Dipole Array

Benchmark #5 is a collinear array of N_{el} half-wave dipoles as shown in Fig. 9. All elements are center-fed in-phase with equal amplitudes sources. The objective is to maximize directivity $D(d_i, i = 1, \ldots, N_{\mathrm{el}} - 1)$ in the plane $\phi = 0°$ as a function of the element center-to-center spacings $0.5\lambda \leq d_i \leq 1.5\lambda$. Because there

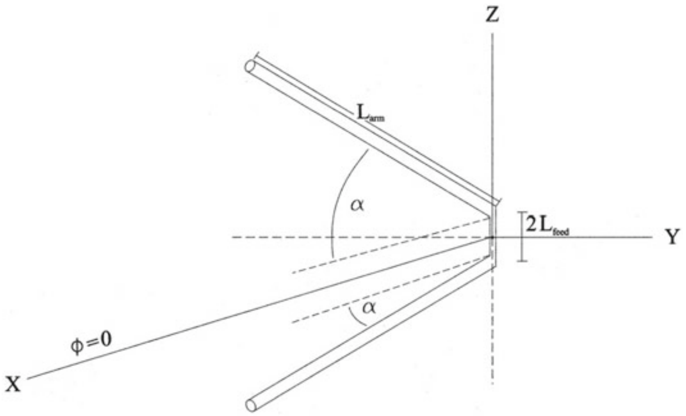

Fig. 7 Vee dipole

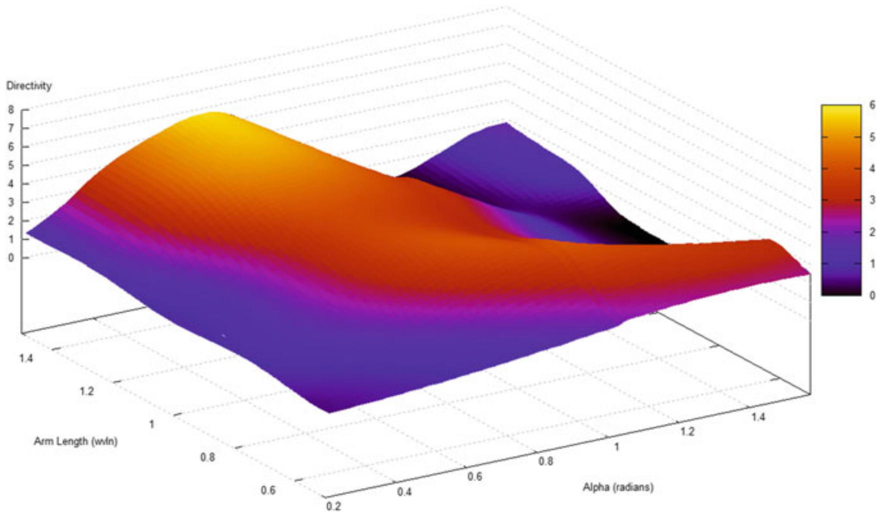

Fig. 8 Vee dipole decision space topology, perspective view

are $N_{el} - 1$ spacings in an N_{el} array, the dimensionality of this problem is $(N_{el} - 1)$D, unlike the previous four benchmarks each of which is 2D. As discussed at length in [1], maximum directivity occurs at $d_i = 0.99\lambda$, $\forall i$, independent of the number of elements, that is, with all dipoles spaced 0.99λ regardless of the array size. Of course, the value of the directivity does depend on the array size, increasing approximately in proportion to the length.

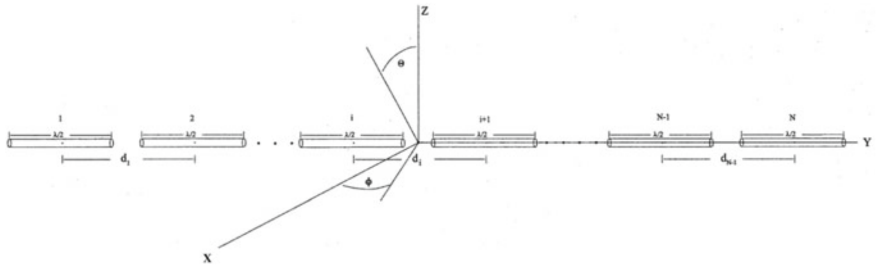

Fig. 9 N_{el}-element collinear dipole array

References

1. Pantoja, M.F., Bretones, A.R., Martin, R.G.: Benchmark antenna problems for evolutionary optimization algorithms. IEEE Trans. Antennas Propag. **55**(4), 1111–1121 (2007)
2. Burke, G.J.: Numerical Electromagnetics Code—NEC-4.2 Method of Moments, Part I: User's Manual. Lawrence Livermore Nat. Lab., Livermore, CA, Rep. LLNL SM-490875 (2011)
3. Bailey, D., Borwein, P., Plouffe, S.: On the rapid computation of various polylogarithmic constants. Math. Comp. **66**, 218, 903–913 (1997) [S 0025-5718(97)00856-9]
4. Formato, R.A.: Pseudorandomness in central force optimization. Br. J. Math. Comput. Sci. **3**(3), 241–264 (2013)
5. Formato, R.A.: Determinism in electromagnetic design & optimization—Part I: central force optimization. Forum for Electromagnetic Research Methods and Application Technologies (FERMAT) online at www.e-fermat.org (Formato-ART-2017_Vol19_Jan._Feb.-009)
6. Li, W.T., Shi, X.W., Hei, Y.Q., Liu, S.F., Zhu, J.: A hybrid optimization algorithm and its application for conformal array pattern synthesis. IEEE Trans. Ant. Prop. **58**(10), 3401–3406 (2010)
7. Formato, R.A.: Determinism in electromagnetic design & optimization—Part II: BBP-derived π fractions for generating uniformly distributed sampling points in global search and optimization algorithms. Forum for Electromagnetic Research Methods and Application Technologies (FERMAT) online at www.e-fermat.org (Formato-ART-2017_Vol19_Jan._Feb.-010)
8. Formato, R.A.: Central Force Optimization Applied to the PBM Suite of Antenna Benchmarks. *arXiv* online at https://arxiv.org/abs/1003.0221v1
9. Press, W.H., Flannery, B.P., Teukolsky, S.A., Vetterling, W.T.: Numerical Recipes: The Art of Scientific Computing, §7.2, Cambridge University Press, Cambridge, UK (1986)
10. Shinzato, T.: Box Muller Method (2007), online at: http://www.sp.dis.titech.ac.jp/~shinzato/boxmuller.pdf

Benchmark Function Generators for Single-Objective Robust Optimisation Algorithms

Seyedali Mirjalili and Andrew Lewis

Abstract Test problems are considered essential when designing optimisation algorithms. The two main conflicting characteristics of a proper test function are simplicity and complexity. The former feature is to allow analysing the behaviour of algorithms, whereas the latter is to mimic real-world problems. Despite the importance of the test functions, however, there are currently neither empirical studies on the suitability of the existing test functions nor benchmark generator to generate them in the field of robust optimisation. This motivates our attempts to analyse the current test functions and propose a new set of benchmark generators to generate test functions with different levels of difficulty. To examine the proposed test functions, robust particle swarm optimisation and robust genetic algorithms are used. The results and analysis first reveal the drawbacks of the current test functions as simplicity, low dimensionality, symmetric search space and lack of scalability. The results then demonstrate the merits of the proposed benchmark generators in alleviating these drawbacks and providing challenging test beds for robust optimisation algorithms.

Keywords Optimisation · Benchmark problems · Robust optimisation
Uncertainties · Test problems

1 Introduction

Meta-heuristics belong to the class of stochastic optimisation techniques, which have become very popular over the last decade. There are many studies in improving the current techniques or proposing new algorithms [1]. One of the common aspects of

S. Mirjalili (✉) · A. Lewis
Institute for Integrated and Intelligent Systems, Griffith University,
Nathan, Australia
e-mail: seyedali.mirjalili@griffithuni.edu.au

A. Lewis
e-mail: a.lewis@griffith.edu.au

K. Deep et al. (eds.), *Decision Science in Action*, Asset Analytics,
https://doi.org/10.1007/978-981-13-0860-4_2

these studies is the use of test problems for benchmarking purposes [2]. Test problems are mostly mathematical functions, which mimic the difficulties of real-world search spaces. The shape of test problem is highly associated with the assessment purposes of a designer. For example, a multi-modal mathematical function is useful for benchmarking the local optima avoidance mechanism of meta-heuristics [3].

In single-objective optimisation, the performance of an algorithm is measured mostly in terms of the accuracy of the obtained optimum and convergence rate. The terms exploration and exploitation are also used in the literature for these two issues [4]. There are several benchmark functions in the literature dividing into three main groups: unimodal [5], multi-modal [6], and composite [7, 8]. Unimodal problems have only one global optimum and there is no local optimum, so they have been designed to test the convergence speed and exploitation of an algorithm. The second group, however, has a huge number of local optima and is beneficial for examining the local optima avoidance ability and exploration. Finally, the composite test functions, which have a massive number local optima and composite shapes of unimodal and multi-modal, are helpful for benchmarking the balance of exploration and exploitation combined.

Despite the merits of the current test problems, there is a key factor in real problems called uncertainties that have attracted much less attentions in the literature. The uncertainties may occur in parameters, output, operating conditions and constraints [9, 10]. Considering such uncertainty during the optimisation process to minimise their negative impacts is called robust optimisation. To the best of our knowledge, there is a small number of test functions for benchmarking robust meta-heuristics. The authors have proposed several test functions for benchmarking robust algorithms in [1–4]. However, this work proposes three test function generators that allow designers to generate test functions with different difficulty levels. The remainder of the paper is organised as follows.

Section 2 discusses the preliminaries and essential definitions robust optimisation. A brief review of the current specific test problems for evaluating robust algorithms is also provided in Sect. 2. The proposed unconstrained benchmark problems are proposed in Sect. 3. Section 4 presents the experimental results of Robust Particle Swarm Optimisation (RPSO) and Robust Genetic Algorithm (RGA) on the current and proposed benchmark problems. Eventually, Sect. 5 concludes the work and recommends a number of future works.

2 Robust Optimisation

2.1 Methods of Handling Uncertainties in Parameters

As discussed above, an algorithm is called robust if it finds a solution that is error tolerant. To do so, uncertainties should be considered before, during, or after the optimisation process. Before the optimisation process, we have to identify the type

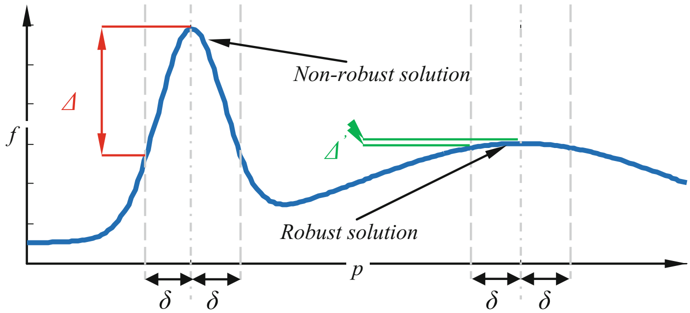

Fig. 1 A conceptual example of a robust optimum versus a non-robust optimum

and degree of uncertainties. For instance, if the resolution on a device is 1, 1 mm uncertainty might occur in the parameters. This means that if an optimisation algorithm might find a decision variable for a problem with the value of x, this value might fluctuate during manufacturing ($\pm$1mm). After identifying the type and level of uncertainty, we then consider this degree of error when evaluating the solutions during optimisation. The robustness of a solution can be tested after the optimisation for confirmation as well. In this work, we consider such undesirable perturbations since they are of the most common types of uncertainty.

Without the loss of generality, a robust optimisation problem when considering uncertainties in decision variables is formulated as a maximisation problem as follows:

$$\text{Maximise} : F\left(\vec{x} + \vec{\delta}\right) \tag{2.1}$$

$$\text{Subject to} : g_i\left(\vec{x} + \vec{\delta}\right) \geq 0, \quad i = 1, 2, 3, \ldots, m-1, m \tag{2.2}$$

$$h_j\left(\vec{x} + \vec{\delta}\right) = 0, \quad j = 1, 2, 3, \ldots, p-1, p \tag{2.3}$$

$$L_k \leq x_k \leq U_k, \quad k = 1, 2, 3, \ldots, n-1, n \tag{2.4}$$

where the vector $\vec{x}$ includes all parameters, $\vec{\delta}$ contains the maximum uncertainty for each parameter in $\vec{x}$, o is the number of objective functions, m is the number of inequality constraints, p is the number of quality constraints and $[L_i, U_i]$ is the boundary of *ith* variable.

A conceptual example of robust and non-robust optima is illustrated in Fig. 1. It can be seen that the left peak is not robust since perturbation (δ) of the solution at this point degrades the objective value significantly as opposed to the right peak. The right peak is not the global maximum, but it is robust and reliable in case of perturbations.

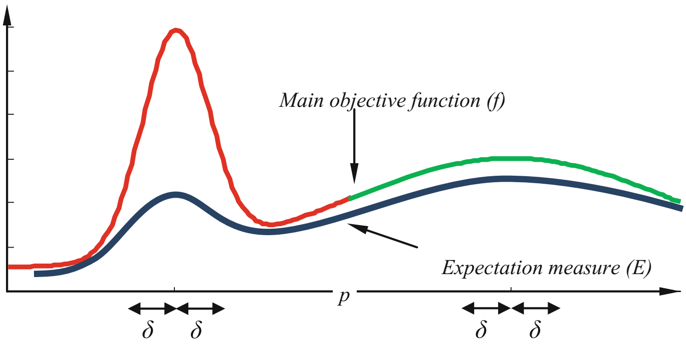

Fig. 2 A conceptual example of a landscape and its expected landscape using an expectation measure

To find such robust solutions, the current techniques can be divided into three classes [5]: expectation measures, variance measures and multi-objective [6]. Since the last method is out of the scope of this paper, the first two methods are discussed below.

In the first method, the average area in the vicinity of solution is calculated using the following integral if it is possible [5]:

$$\text{Maximise} : E(x) = \frac{1}{|B_\delta(x)|} \int\limits_{y \in B_\delta(x)} f(y)\mathrm{d}y \tag{2.5}$$

where $B_\delta(x)$ shows δ-radius neighbourhood of the solution x, and $|B_\delta(x)|$ indicates the hypervolume of the neighbourhood.

If the analytical integral is difficult to calculate, Monte Carlo estimation is a reliable alternative as follows:

$$E(x) = \frac{1}{H} \sum_{i=1}^{H} f(x + \delta_i) \tag{2.6}$$

where H is the number of samples.

In Eq. (2.6), it is evident that the average objective of H points is calculated as the average area of the neighbourhood around the solution x. No matter how the expectation measure is calculated, the key point is that the objective function is replaced. Figure 2 shows how this change of landscape causes a global optimum, which is not robust, to become a local optimum.

One of the first expectation measures in the field of stochastic optimisation was proposed by Deb and Gupta [5]. They named this method ‘type I’ robust optimisation. There are other expectation measures in the literature as well that follow the same concepts [7–12]. In most of them, different sampling techniques or equations are used to improve the accuracy of calculating the expected objective function.

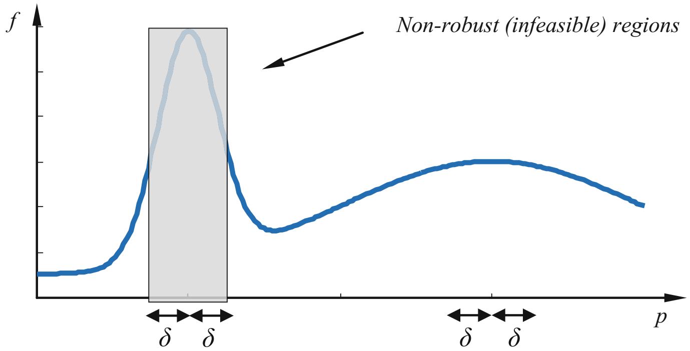

Fig. 3 A conceptual example of how a variance measure makes a non-robust region infeasible

In the second class of robust optimisation, there is no expectation measure to be replaced by the main objective function. Instead, there is a variance measure that is used as a constrained for an optimisation problem. This measure is defined as follows [5]:

$$\text{Maximise}: f(x) \tag{2.7}$$

$$\text{Subject to}: V(x) = \frac{||F(x) - f(x)||}{||f(x)||} \leq \eta \tag{2.8}$$

where $F(x)$ is as effective mean or the objective value of the worst solution between the H selected solutions, η is a vector of thresholds in [0, 1] and S indicates the feasible search space.

Equation (2.8) shows that the variance measure first calculates the difference between the average (or worst) of the objective values of points in the neighbourhood and the objective value of the current solution. It then divides it by the current objective value. This gives a number in the interval of [0, 1] and defines the robustness of a solution. With chanting η, a desired level of perturbation can be simulated for the solutions. Note that this type of robust optimisation is called 'type II'.

A conceptual example of the impacts of a variance measure is shown in Fig. 3. It can be seen that the non-robust global optimum is an infeasible solution when using a variance measure.

2.2 *Current Test Functions for Robust Optimisation*

In the field of robust optimisation, there is a large number of test functions. However, there is a small number of such test functions in the literature of robust optimisation.

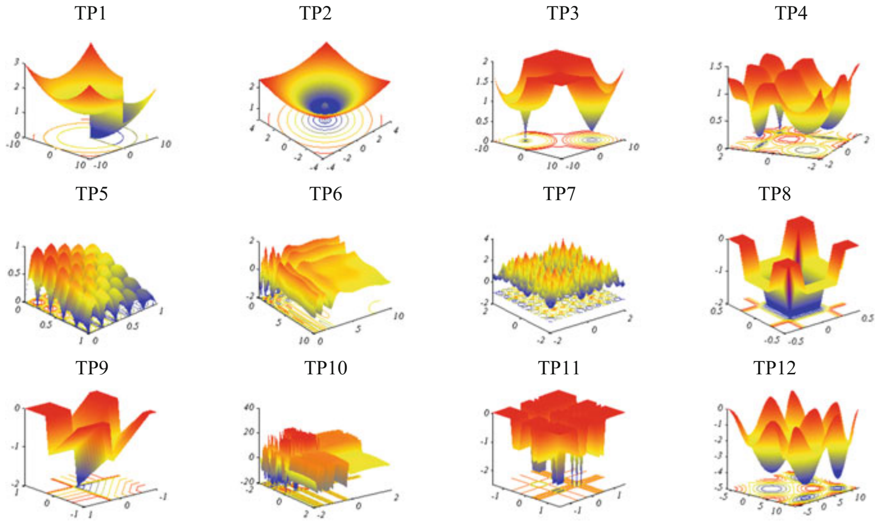

Fig. 4 A set of test functions obtained from the literature for benchmarking robust algorithms [12–15]

Thirteen test functions specifically designed to benchmark robust algorithms are presented in this subsection [12–15]. Figure 4 shows the shape and the details can be found in the original papers.

It is evident in Fig. 4 that most of these test functions are not highly multimodal. A large number of them have less than 10 dimensions and are not scalable as well. This prevents them from providing enough difficulties to benchmark a robust optimisation algorithm efficiently. The authors have proposed several test functions for benchmarking robust algorithms in [1–4]. However, this work proposes three test function generators that allow designers to generate test functions with diverse difficulty levels.

3 Proposed Benchmark Generators and Test Functions

This section proposes the benchmark generators.

3.1 Benchmark Function Generator I

This benchmark generator is designed to create a search landscape with two optima. The key point is that a designer is able to make the global optimum wide and narrow to change the robustness level. The mathematical equation is as follows:

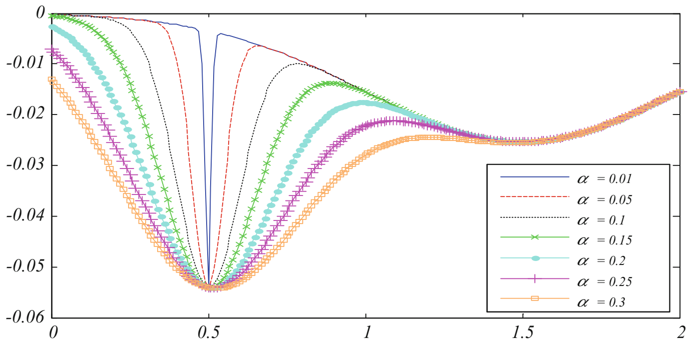

Fig. 5 Benchmark function generator I allows adjusting the robustness of the global optimum

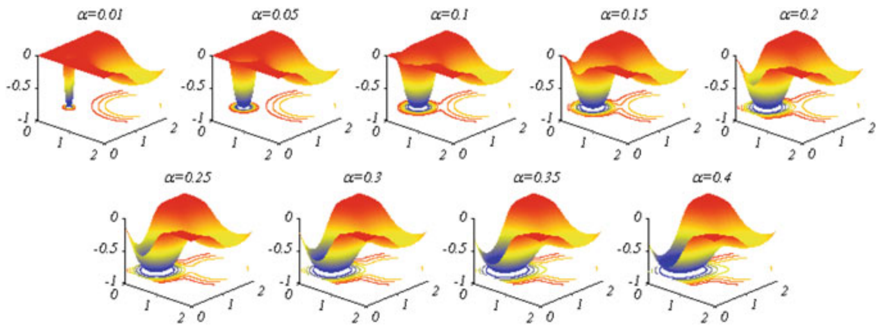

Fig. 6 Effect α of on the robustness of the global optimum

$$f(x) = \frac{1}{\sqrt{2\pi}} e^{-0.5\left(\frac{x-1.5}{0.5}\right)^2} + \frac{2}{\sqrt{2\pi}} e^{-0.5\left(\frac{x-0.5}{\alpha}\right)^2} \tag{3.1}$$

where α indicates the robustness of the global optimum by narrowing or widening it.

The shape of different test functions that be generated with this benchmark generator is shown in Fig. 5. It can be seen that the robustness of the global optimum is decreased proportional to the value of the parameter α.

The two-dimensional version of this function is defined as follows:

$$f(x, y) = \frac{1}{\sqrt{2\pi}} e^{-0.5\left(\frac{(x-1.5)^2+(y-1.5)^2}{0.5}\right)^2} + \frac{2}{\sqrt{2\pi}} e^{-0.5\left(\frac{(x-0.5)^2+(y-0.5)^2}{\alpha}\right)^2} \tag{3.2}$$

Figure 6 shows that the parameter α has the similar effect on the robustness of the global optimum compared to Eq. (3.1).

The benchmark generator I is defined as follows:

$$\text{Minimise}: f(x) = \left(\left(\frac{1}{\sqrt{2\pi}} e^{-0.5\left(\frac{\sum_{i=1}^{n}(x_i-1.5)^2}{0.5}\right)^2}\right) + \left(\frac{2}{\sqrt{2\pi}} e^{-0.5\left(\frac{\sum_{i=1}^{n}(x_i-0.5)^2}{\alpha}\right)^2}\right)\right) \tag{3.3}$$

$$\text{Where}: 0 \leq x_i \leq 2 \tag{3.4}$$

where n is the maximum number of variables.

The local optimum is always located on $(0.5, 0.5, \ldots, 0.5)$ and the global optimum is at $(1.5, 1.5, \ldots, 1.5)$. This benchmark generator offers a global optimum with alterable degree of robustness that would allow us to benchmark the performance of a robust algorithm in terms of favouring a robust solution. By changing the global optimum's robustness, algorithm designers would be able to observe the resistance of a robust meta-heuristic dealing with a non-robust global optimum. In addition, it may be seen in Eq. (3.3) that this benchmark generator is able to generate scalable test functions with desirable number of variables. The test functions generated by this benchmark generator have the following features:

- Test functions are not readily solvable by simple optimisation methods.
- The search space is non-linear, non-separable, and non-symmetric.
- The robustness of global optimum is alterable.
- The robustness of global optimum does not affect the optimal values of both local and global optima.
- Both local and global optima can play the role of the robust optimum based on the α parameter and considered perturbation in the parameters.
- Test functions are scalable.

3.2 Benchmark Generator II

The second benchmark generator generates a desirable number of local non-robust solutions. In other words, a multi-modal search space with one global optimum, one robust optimum and several local non-robust optima can be created by this benchmark generator. The mathematical formulation of this benchmark generator is as follows:

$$\text{Minimise}: f(x) = -G(x) \times H(x_1) \times H(x_2) + \omega \tag{3.5}$$

$$\text{Where}: H(x) = \frac{e^{-2x^2}\sin\left(\lambda \times 2\pi\left(x + \frac{\pi}{4\lambda}\right)\right) - x^{\beta}}{3} + 0.5 \tag{3.6}$$

$$G(x) = 1 + 10\frac{\sum_{i=3}^{N} x_i}{N} \tag{3.7}$$

$$0 \leq x_i \leq 1 \tag{3.8}$$

$$\lambda \geq 1 \tag{3.9}$$