

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

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

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

ISSN 2522-5162

Asset Analytics

ISBN 978-981-13-0859-8

<https://doi.org/10.1007/978-981-13-0860-4>

ISSN 2522-5170 (electronic)

ISBN 978-981-13-0860-4 (eBook)

Library of Congress Control Number: 2018948716

© Springer Nature Singapore Pte Ltd. 2019

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

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, Qionggiong 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, Dwaipayana 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

© Springer Nature Singapore Pte Ltd. 2019
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 (π CFO) and Genetic Algorithm with Sibling Rivalry (π GASR) 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. π CFO and π GASR 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 $(N_{el}-1)D$ 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 π CFO and π GASR both employ the following π fraction pseudocode instead of using a compiler's built-in random number generator. This procedure generates random real numbers $a \leq r_i < b$ and mitigates the correlation problems discussed above (note that initialization values are completely arbitrary).

$\pi_i : \{ \pi \text{ fractions in order of occurrence; } i=1, \dots, N_\pi \}$

Initialize: $N_\pi \leftarrow 215\,830 : \text{init}_1 \leftarrow 17$

$\text{init}_2 \leftarrow 22 : \text{inc} \leftarrow 5 : i \leftarrow \text{init}_1$

Procedure R_π : *generates a random number
in $[a,b)$ using π fractions*

$r_i = a + (b-a) \pi_i$

$i \leftarrow i + \text{inc}$

if $i > N_\pi$ *then* $i \leftarrow \text{init}_2$

End

Pseudocode π fraction Random Numbers

4 Algorithms Compared

Data from three algorithms, CFO, π CFO, and π GASR, 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 (F_{rep}) 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, π CFO, 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 π CFO run was made for each benchmark because the algorithm remains completely deterministic with the π fractions.

The second randomized algorithm applied to PBM is π GASR (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 π GASR 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, π CFO, and π GASR 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	π CFO	π GASR
1	3.32	3.20627	3.24340	3.25837
2a (no noise)	18.3 ^a	18.3654	18.2810	17.9473
2b (noisy)	nr ^b	18.6880	19.7609	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 ^avalues estimated from the figures in [1]

^bnr—not reported in [1]

^cvalues marked with ~ are estimated from Fig. 13 in [1]

the actual maxima, and that the π fraction approach is effective in making stochastic π GASR 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 π GASR 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 π GASR'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 π CFO. 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 bench- mark	PBM		CFO		π CFO		π GASR	
	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
2a (no noise)	$\sim 5.85\lambda$	1.5730	5.9236λ	1.5569	5.9246	1.5554	6.9270	1.5467
2b (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 $d_i, i = 1, \dots, N_{el} - 1$								
5	0.99 λ		0.983/1 λ		0.974/1.199 λ		0.987/1 λ	

Notes ^anr—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 $\phi = 90^\circ$ 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

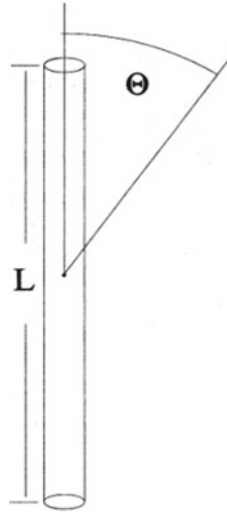


Fig. 1 Dipole

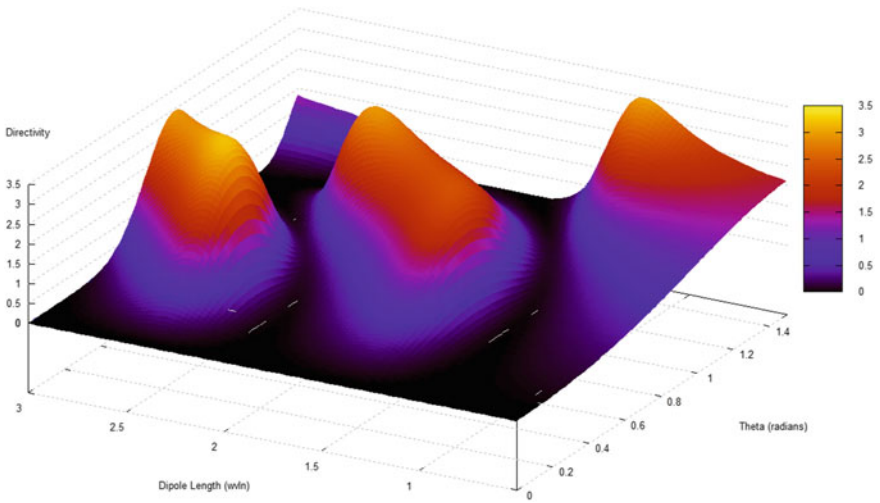


Fig. 2 Benchmark #1 topology, perspective view

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.

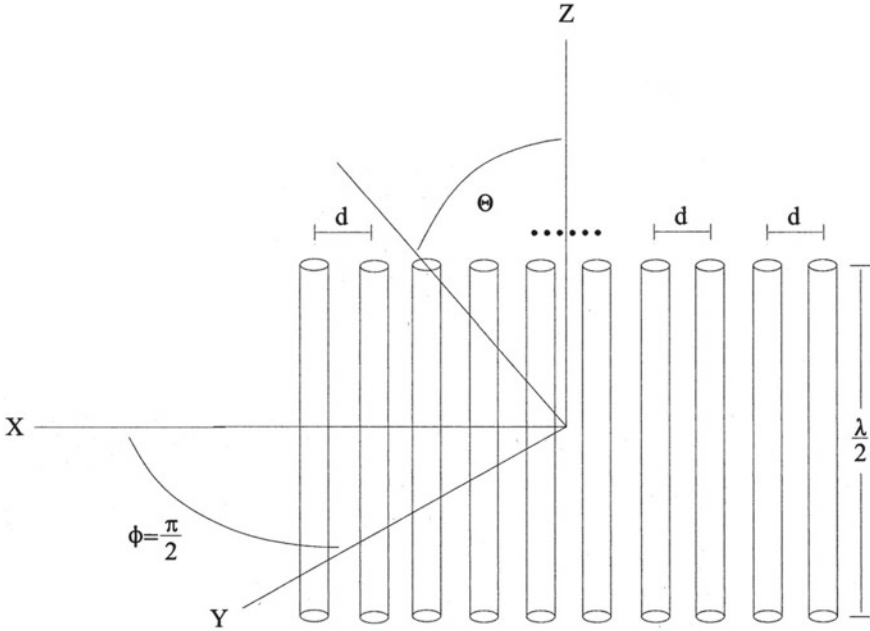


Fig. 3 Uniform array of half-wave center-fed in-phase dipoles

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 $\alpha_n = -\cos[2\pi\beta(n-1)]$, $n = 1, \dots, 8$. 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 $\phi = 0^\circ$ as a function of the dimensionless phase parameter $0 \leq \beta \leq 4$ and the polar angle θ . The range for β produces the four global maxima at $(\beta_i = i - 0.5, i = 1, \dots, 4; \theta = \frac{\pi}{2})$ 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 $0.5\lambda \leq L_{\text{total}} = 2L_{\text{arm}} + 2L_{\text{feed}} \leq 1.5\lambda$ and the inner half-angle $\frac{\pi}{18} \leq \alpha \leq \frac{\pi}{2}$ with $L_{\text{feed}} = 0.01\lambda$.

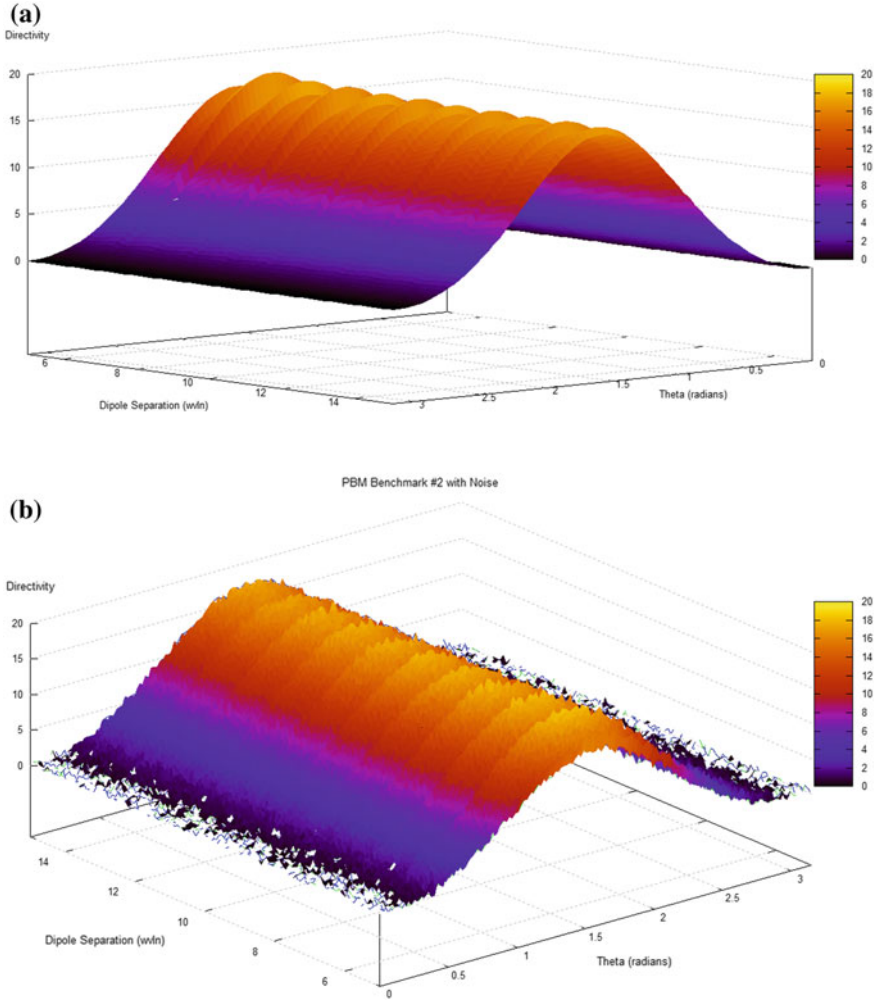


Fig. 4 **a** Uniform half-wave dipole array without noise, perspective view. **b** Uniform dipole array with additive Gaussian noise, perspective view

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

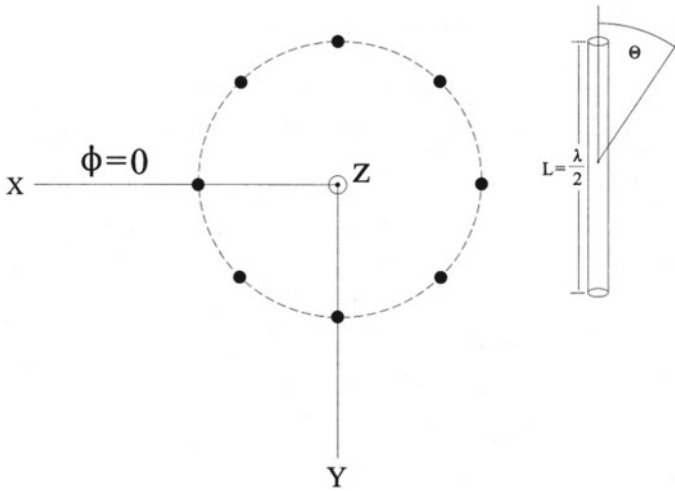


Fig. 5 Circular array of half-wave dipoles (1λ radius)

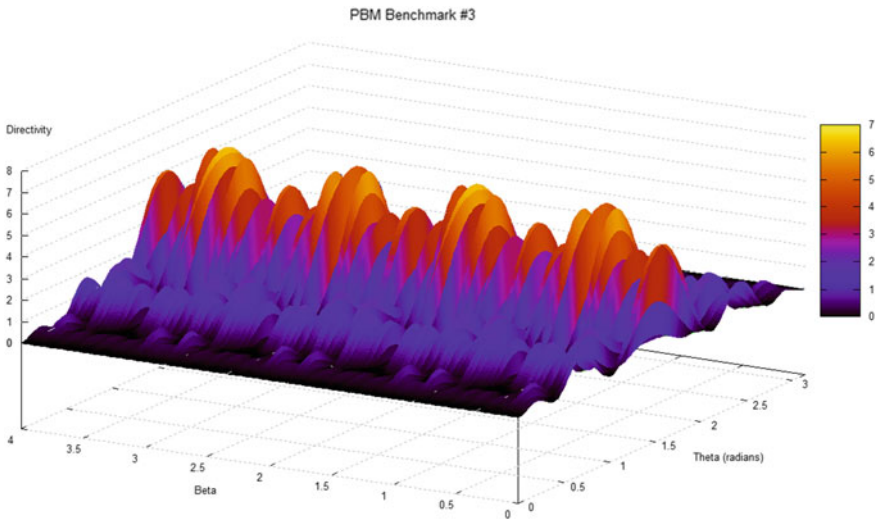


Fig. 6 Circular array landscape, perspective view

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 $D(d_i, i = 1, \dots, N_{el} - 1)$ in the plane $\phi = 0^\circ$ as a function of the element center-to-center spacings $0.5\lambda \leq d_i \leq 1.5\lambda$. Because there

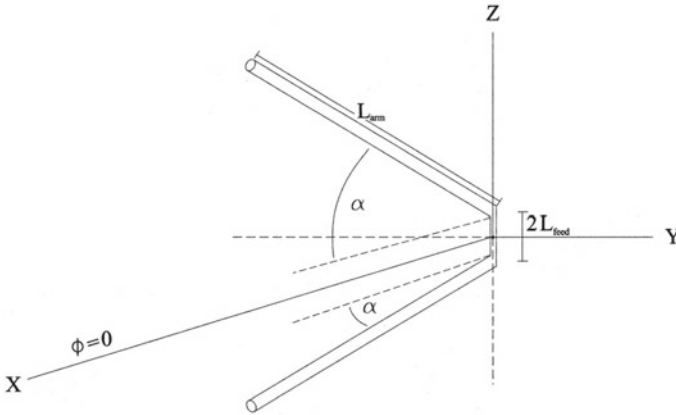


Fig. 7 Vee dipole

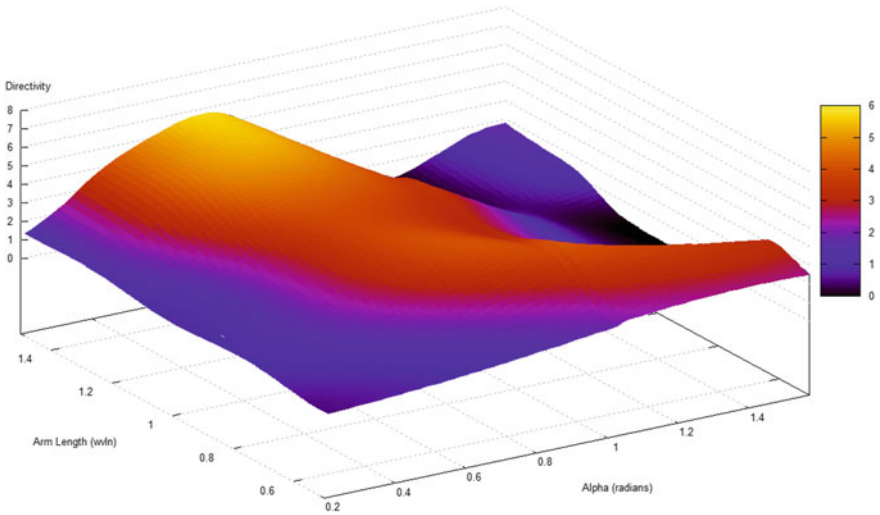


Fig. 8 Vee dipole decision space topology, perspective view

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.

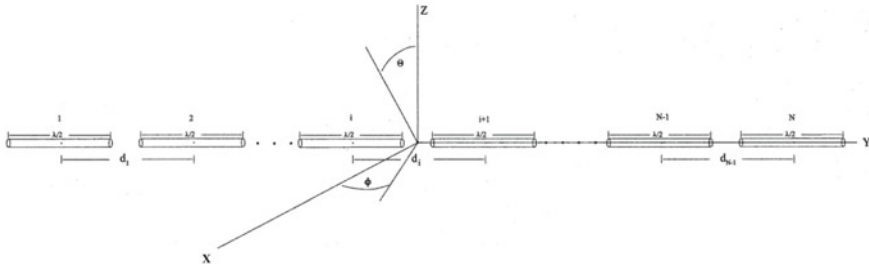


Fig. 9 N_{el} -element collinear dipole array

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

© Springer Nature Singapore Pte Ltd. 2019
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

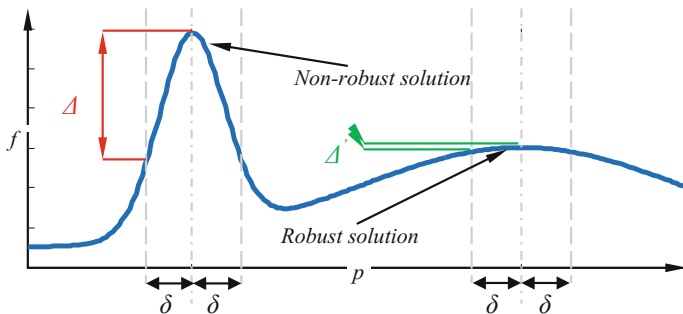


Fig. 1 A conceptual example of a robust optimum versus a non-robust optimum

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 ($\pm 1\text{mm}$). 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:

$$\text{Maximise : } F(\vec{x} + \vec{\delta}) \quad (2.1)$$

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

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

$$L_k \leq x_k \leq U_k, \quad k = 1, 2, 3, \dots, n - 1, n \quad (2.4)$$

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 i th 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.

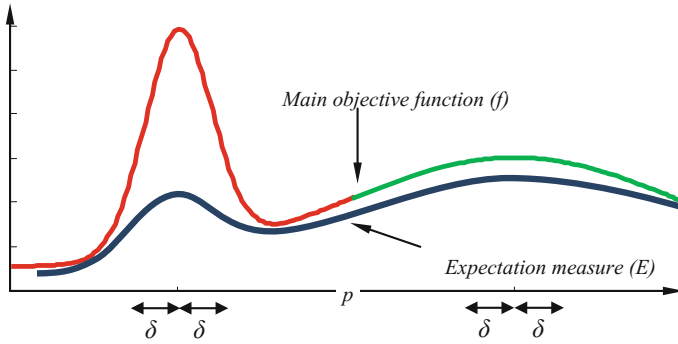


Fig. 2 A conceptual example of a landscape and its expected landscape using an expectation measure

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]:

$$\text{Maximise : } E(x) = \frac{1}{|B_\delta(x)|} \int_{y \in B_\delta(x)} f(y) dy \quad (2.5)$$

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:

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

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.

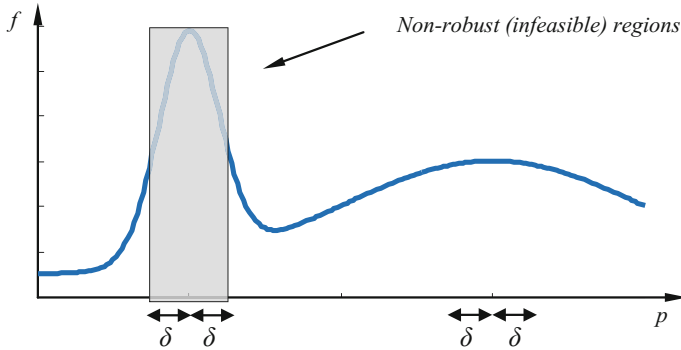


Fig. 3 A conceptual example of how a variance measure makes a non-robust region infeasible

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.

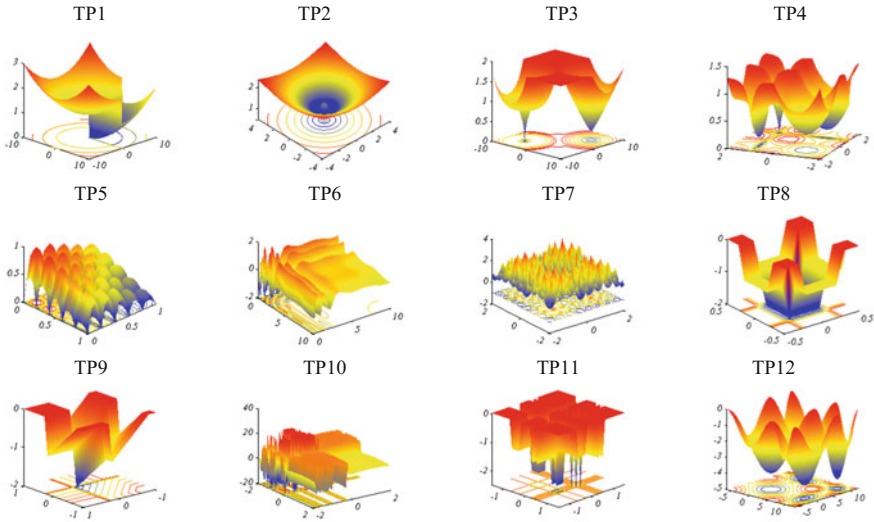


Fig. 4 A set of test functions obtained from the literature for benchmarking robust algorithms [12–15]

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 multi-modal. 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:

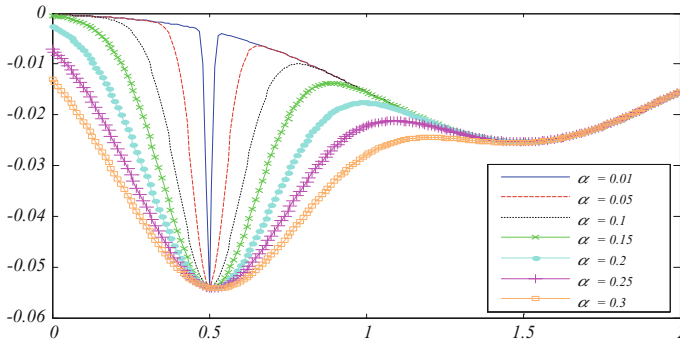


Fig. 5 Benchmark function generator I allows adjusting the robustness of the global optimum

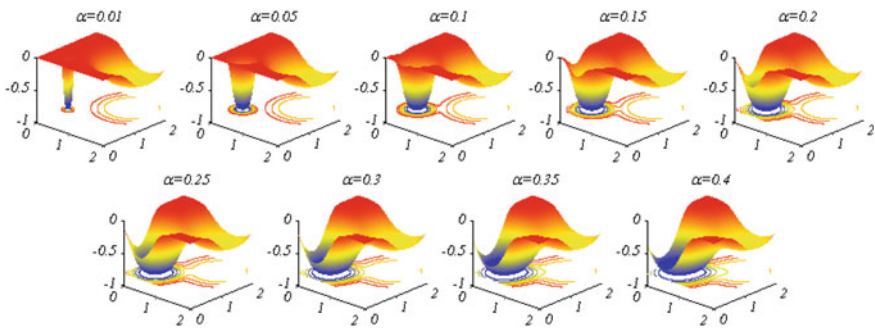


Fig. 6 Effect α of on the robustness of the global optimum

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

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:

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

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:

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

$$\text{Where : } 0 \leq x_i \leq 2 \quad (3.4)$$

where n is the maximum number of variables.

The local optimum is always located on $(0.5, 0.5, \dots, 0.5)$ and the global optimum is at $(1.5, 1.5, \dots, 1.5)$. 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:

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

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

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

$$0 \leq x_i \leq 1 \quad (3.8)$$

$$\lambda \geq 1 \quad (3.9)$$

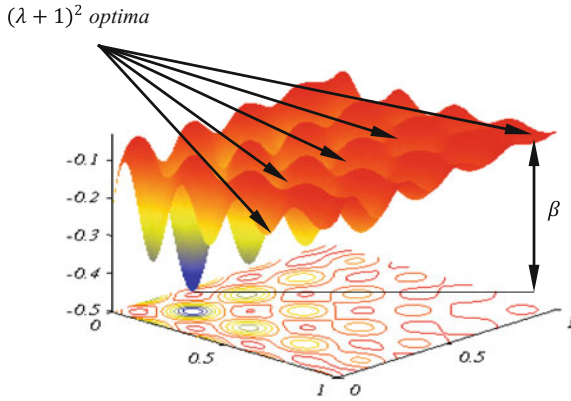


Fig. 7 Shape of the search space with controlling parameters constructed by benchmark generator II

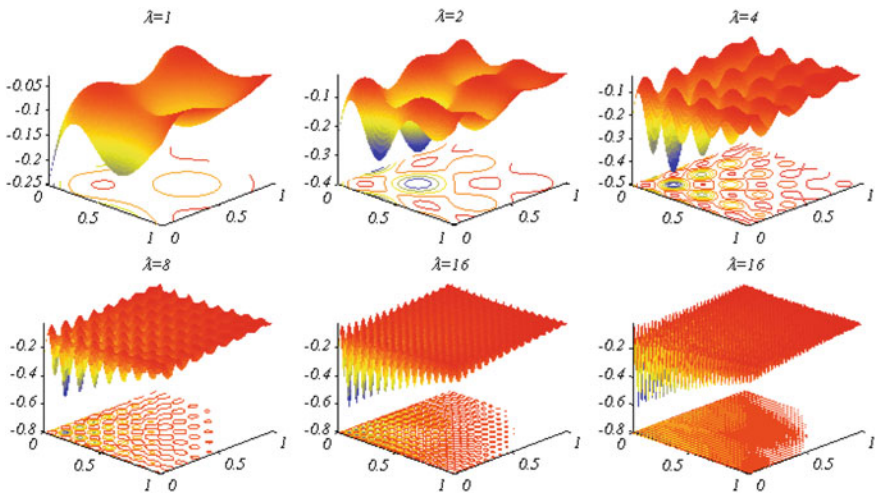


Fig. 8 Effect of parameter λ on the shape of search space

$$\beta \geq 1 \tag{3.10}$$

As may be seen in Fig. 7, this benchmark generator allows generating $(\lambda + 1)^2$ number of local optima through the search space. The effect of this parameter on the landscape can be observed in Fig. 8. This figure shows that the search space becomes more challenging as λ increases.

Another characteristic of this benchmark generator is scalability. The function $G(x)$ is responsible for supporting three or more variables. Since $G(x)$ is a kind of penalty function, an algorithm should find zero values for $x_3 - x_n$. The characteris-

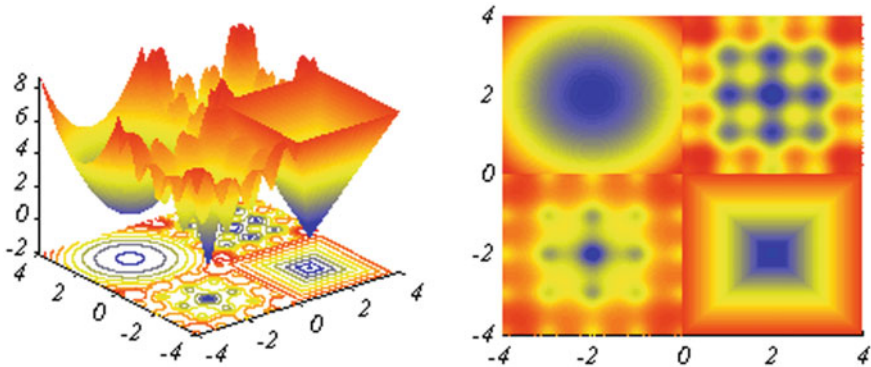


Fig. 9 An example of the search space that can be constructed by the benchmark generator III

tics of the test functions generated by this benchmark generator are summarised as follows:

- Test functions are not readily solvable by simple optimisation methods.
- The search space is non-linear, non-separable and non-symmetric.
- The number of local optima can be adjusted.
- The robustness of optima is increased inversely proportional to the objective value. The last worst local optimum can be the most robust optimum when considering a certain level of uncertainty.
- Test functions are scalable.

3.3 Benchmark Generator III

This benchmark generator aggregates four current test functions in the literature of global optimisation. It divides the search space into four sections and allows user to define different functions in each section. The mathematical formulation is as follows:

$$\text{Minimize : } f(x) = \begin{cases} f_1(x) & (x_1 \leq 0) \wedge (x_2 \geq 0) \\ f_2(x) & (x_1 \geq 0) \wedge (x_2 \leq 0) \\ f_3(x) & (x_1 > 0) \wedge (x_2 > 0) \\ f_4(x) & (x_1 < 0) \wedge (x_2 < 0) \end{cases} \quad (3.11)$$

Any type of function with robust and non-robust optima can be integrated instead of f_1 to f_4 . For instance, Fig. 9 shows a search space constructed using sphere, Ackley, Rastrigin and pyramid-shaped functions. It is evident from the figure that the sphere function has the most robust optimum.

In order to provide scalability for this benchmark function generator, there can be two possibilities. Each of the subfunctions can be chosen with different numbers of variable or the function $G(x)$ from the second proposed benchmark function generator can be multiplied by the results of each function as follows:

$$f(x) = \begin{cases} f_1(x) \times G(x) & (x_1 \leq 0) \wedge (x_2 \geq 0) \\ f_2(x) \times G(x) & (x_1 \geq 0) \wedge (x_2 \leq 0) \\ f_3(x) \times G(x) & (x_1 > 0) \wedge (x_2 > 0) \\ f_4(x) \times G(x) & (x_1 < 0) \wedge (x_2 < 0) \end{cases} \quad (3.12)$$

$$G(x) = 1 + 10 \frac{\sum_{i=3}^N x_i}{N} \quad (3.13)$$

The characteristics of the test functions generated by this benchmark generator are summarised as follows:

- Test functions are not readily solvable by simple optimisation methods.
- The search space is non-linear, non-separable and non-symmetric.
- There can be desirable number of local, global and robust optima.
- Test functions are scalable.

3.4 Test Functions Generated by the Benchmark Generator

With the proposed benchmark generators, three test functions are generated in this subsection. Table 1 presents name, specifications and search landscape of the three proposed test functions. Note that we name them TP14, TP15 and TP16 since we will be comparing them with TP1 to TP12.

The next section investigates the effectiveness of these test functions in practice.

4 Results and Discussion

In order to test the difficulties of test functions, PSO and GA are employed in this section. We use 20 search agents for each algorithm and allow them to find the robust optima of the current/proposed test functions over 500 iterations. In addition, we require PSO and GA to consider 10% fluctuation in the parameters of search agents to simulate uncertainties. Handling uncertainties are done by an expected measure that is calculated by re-sampling and averaging 50 random solutions in the neighbourhood of search agents at each iteration. The statistical results are provided in Table 1 in the form of average \pm standard deviation.

Table 1 Generated test functions

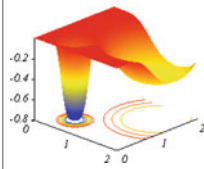
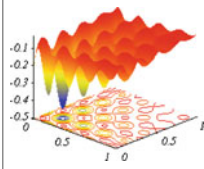
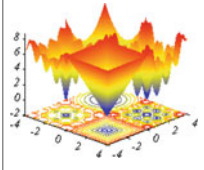
Name	Specifications	Search space
TP14	<p>Benchmark generator I: $\alpha = 0.05$ Dimension = 2 Search space : $\vec{x} \in [-4, 4]^N$ Input noise : $\vec{\delta} \sim \vec{U}(-0.5, 0.5)$ Robust optimum fitness ≈ -0.4 Robust optimum location : (1.5, 1.5)</p>	
TP15	<p>Benchmark generator II: $\lambda = 4$ $\beta = 1$ Dimension = 2 Search space : $\vec{x} \in [0, 1]^N$ Input noise : $\vec{\delta} \sim \vec{U}(-0.02, 0.02)$ Robust optimum fitness ≈ -0.05 Robust optimum location $\approx (0.98, 0.98)$</p>	
TP16	<p>Benchmark generator III: $f_1(\vec{x}) = \sum_{i=1}^N x_i^2$ $f_2(\vec{x}) = \max_i \{ x_i , 1 \leq i \leq N\}$ $f_3(\vec{x}) = \sum_{i=1}^N x_i^2 [x_i^2 - 10 \cos(2\pi x_i) + 10]$ $f_4(\vec{x}) = -20e^{-0.2\sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}} - e^{\left(\frac{1}{N} \sum_{i=1}^N \cos(2\pi x_i)\right)} + 20 + e$ $N=2$ Dimension = 2 Search space: $\vec{x} \in [-4, 4]^N$ Input noise: $\vec{\delta} \sim \vec{U}(-0.2, 0.2)$ Robust optimum fitness ≈ -0.8 Robust optimum location = (-2, 2)</p>	

Table 2 Statistical results of algorithms

Algorithm	TP1	TP2	TP3
PSO	$0.0186 \pm 5.20E-05$	0.9274 ± 0.0044	0.2107 ± 0.0166
GSA	0.0195 ± 0.0001722	0.9272 ± 0.0065	0.2155 ± 0.0021
	TP4	TP5	TP6
PSO	$0.294960 \pm 2.8E-05$	$4.2E-09 \pm 8.72E-25$	-1.3366 ± 0.48765
GSA	0.295021 ± 0.000119	0.0201 ± 0.006302	-1.8867 ± 0.030969
	TP7	TP8	TP9
PSO	0.3843 ± 0.1770	-2.0000 ± 0.0000	-1.61374 ± 0.000716
GSA	0.2708 ± 0.0186	-2.0000 ± 0.0000	-1.61023 ± 0.001679
	TP10	TP11	TP12
PSO	-19.1423 ± 3.1883	-1.500 ± 0.7071	-3.9994 ± 0.00055
GSA	-20.0798 ± 0.2038	-1.000 ± 0.0000	-3.9995 ± 0.000545
	TP13	TP14	TP15
PSO	-1.2306 ± 0.078981	0.12319 ± 0.46783	-0.39279 ± 0.002322
GSA	-1.2689 ± 0.051873	-0.1480 ± 0.029463	-0.39135 ± 0.002133
	TP16		
PSO	$-0.44723 \pm 4.42E-06$		
GSA	$-0.44721 \pm 2.05E-05$		

The box plots of the statistical results of Table 2 are illustrated in Fig. 10. The first thing that may be observed in the results is the similar performance of PSO and GA on a number of test functions such as TP4, TP8 and TP11. This shows that most of the current test functions can be readily solved, so the performance of robust algorithms cannot be benchmarked thoroughly.

The results of both algorithms on TP1 and TP9 follow similar behaviours. These two functions are unimodal and the robust optima are located near to the global optima. Since we consider $\delta = 1$ for TP1, the robust optimum is located in $[0.2, 0.2]$ with the value of 0.02. The global optimum is at $[0, 0]$ with the value of 1. The results show that the GA algorithm provides better results as Table 2 shows. The shape of the TP1 is very similar to that of TP9, in which there is one global optimum (located at $[0.2, 0.2]$ with the value of 0) and the robust optimum is located based in the same valley based on the degree of perturbation. In our case study, we considered $\delta = 0.2$, so the robust optimum is at $[0, 0]$ with the value of -1.6 . The results of Table 2 again suggest that the GA algorithm performs better than PSO in finding the robust optimum.

In contrast to the results of the TP1 to TP13, Table 2 shows that the results of RPSO and RGA on TP14 to TP16 are very different compared to other results. None of the algorithms found the robust optimum for TP14 and TP16. These results show that the test functions generated by the benchmark generators are very challenging test beds. The reason for poor performance of both algorithms is due to employed expectation measure. The expectation measure calculates the average fitness of 50

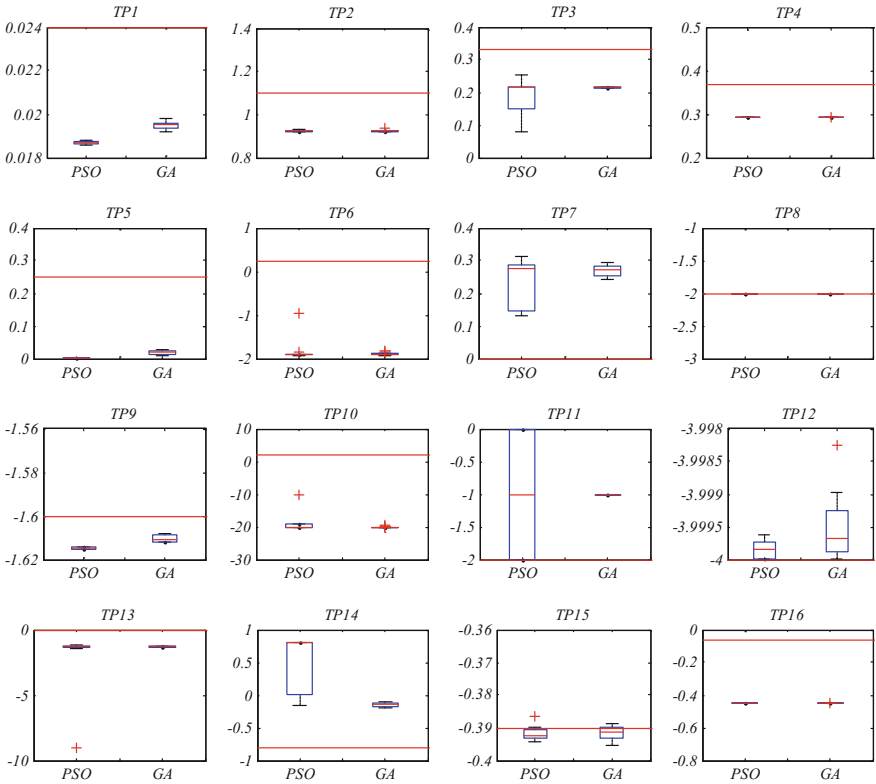


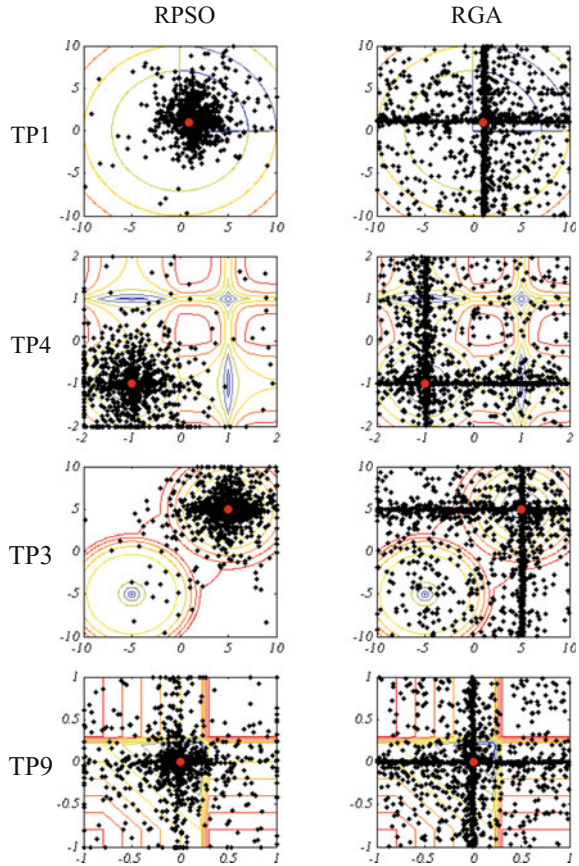
Fig. 10 Box plots of PSO and GA

neighbouring solutions for each candidate solution. The results can be improved with increasing the number of sampled point. However, this is out of the scope of this work since the main objective was the comparison of algorithms on the test functions.

To further observe the behaviour of both algorithms, we ran each algorithm over 100 iterations and illustrate some of the search history in Figs. 11 and 12. The first distinct behaviour of algorithms is that both algorithms show reasonable exploration of the search space and exploitation near the robust optima. According to the results of Table 1, it seems that GA balances exploration and exploitation more appropriately for finding the robust optimum. However, it is clear from these results that these two test functions are simple for both algorithms and there is no significant superiority for the GA algorithm.

Figure 12 shows the search history of both algorithms when solving the three proposed test functions. Search history of RPSO and RGA on TP14 shows that both algorithms are able to find the robust optimum with the considered perturbation. However, they failed to find the optimum in TP15. In addition, RPSO outperforms RGA in TP16. These results evidence the difficulties of the proposed test functions

Fig. 11 Search histories on TP1, TP3, TP4 and TP9



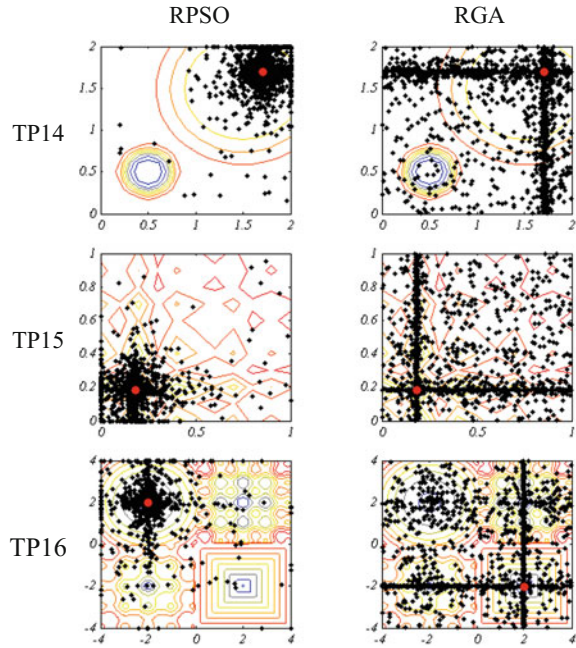
compared to other test functions. With increasing the number of variables, obviously, the difficulties of the test functions would also be increased.

To sum up, the results first demonstrates that the current text functions are readily solvable by robust meta-heuristics. The simplicity is the main reason for this. However, the results proved that the proposed test functions can provide much more challenging test beds for effectively benchmarking the performance of robust meta-heuristics.

5 Conclusion

This paper contributed to the literature of robust optimisation with proposing three benchmark generators. The gaps targeted were simplicity, low dimensionality, high symmetrically, and lack of scalability of the current test function for evaluating robust

Fig. 12 Search histories on TP14, TP15 and TP16



single-objective optimisation algorithms. In order to alleviate these shortcomings, the paper proposed three novel benchmark function generators for generating test functions with different characteristics. The paper employed the RPSO and RGA to prove the disadvantages of the current test functions and demonstrate the merits of the proposed benchmark function generator and benchmark functions. The results showed that the proposed test functions are able to benchmark the performance of robust algorithms effectively from different perspectives: resistance of an algorithm in converging towards non-robust but global solutions, and ability to avoid non-robust local optima. The paper also considered the comparison of RPSO and RGA on the rest functions. It was observed that the RGA shows high exploration, whereas the RPSO provides good exploitation around robust optima.

For future work, it is recommended to integrate some constraints to the benchmark function generator in order to generate constrained robust test functions.

References

1. Mirjalili, S., Lewis, A.: Hindrances for robust multi-objective test problems. *Appl. Soft Comput.* **35**, 333–348 (2015)
2. Mirjalili, S., Lewis, A.: Novel frameworks for creating robust multi-objective benchmark problems. *Inf. Sci.* **300**, 158–192 (2015)

3. Mirjalili, S., Lewis, A.: Obstacles and difficulties for robust benchmark problems: a novel penalty-based robust optimisation method. *Inf. Sci.* **328**, 485–509 (2016)
4. Mirjalili, S.: Shifted robust multi-objective test problems. *Struct. Multidisciplinary Optim.* **52**, 217–226 (2015)
5. Deb, K., Gupta, H.: Introducing robustness in multi-objective optimization. *Evol. Comput.* **14**, 463–494 (2006)
6. Ray, T.: Constrained robust optimal design using a multiobjective evolutionary algorithm. In: *Proceedings of the 2002 congress on evolutionary computation, 2002. CEC'02*, pp. 419–424
7. Tsutsui, S., Ghosh, A.: Genetic algorithms with a robust solution searching scheme. *Evol. Comput. IEEE Trans.* **1**, 201–208 (1997)
8. Wiesmann, D., Hammel, U., Bäck, T.: Robust design of multilayer optical coatings by means of evolution strategies. In: *IEEE Transactions on Evolutionary Computation* (1998)
9. Jin, Y., Sendhoff, B.: Trade-off between performance and robustness: an evolutionary multi-objective approach. In: *Evolutionary Multi-Criterion Optimization*, pp. 237–251 (2003)
10. Deb, K., Gupta, H.: Searching for robust Pareto-optimal solutions in multi-objective optimization. *Lect. Notes Comput. Sci.* **3410**, 150–164 (2005)
11. Gaspar-Cunha, A., Covas, J.A.: Robustness in multi-objective optimization using evolutionary algorithms. *Comput. Optim. Appl.* **39**, 75–96 (2008)
12. Kruisselbrink, J.W.: *Evolution strategies for robust optimization*. Leiden Institute of Advanced Computer Science (LIACS), Faculty of Science, Leiden university (2012)
13. Branke, J.: *Creating robust solutions by means of evolutionary algorithms*, pp. 119–128 (1998)
14. Dippel, C.E.J.: *Using particle swarm optimization for finding robust optima* (2010)
15. Kruisselbrink, J., Emmerich, M., Bäck, T.: An archive maintenance scheme for finding robust solutions. *Parallel Problem Solving from Nature–PPSN XI*, pp. 214–223 (2011)

Convergence of Gravitational Search Algorithm on Linear and Quadratic Functions



Anupam Yadav, Anita and Joong Hoon Kim

Abstract Convergence characteristic of any optimization algorithm is a very important aspect. Several studies have been performed to discuss the convergence of non-deterministic optimization algorithms. In this article, the convergence of gravitational search algorithm (GSA) is discussed over linear and quadratic functions. A theoretical proof of convergence for GSA is provided for linear and quadratic functions. The article ensures the convergence of GSA over linear and quadratic functions.

Keywords Convergence · Gravitational Search · Optimization

1 Introduction

The modeling of optimization problems played a major role in engineering over the years, as many real-life engineering problems can be modeled as an optimization problem. To achieve the optimal solution of these problems always remains an important point. In order to provide the better results of these problems, many optimization algorithms are proposed in the literature. Since the traditional deterministic techniques had their own limitations, therefore, non-deterministic optimization algorithms came into the existence. Current status of research in the area of non-deterministic optimization algorithms is moving on the wheels of the swarm intelligence, evolutionary computation, and other hybrid methodologies. Based on these ideas, a significant number of optimization algorithms are proposed such as genetic algorithm [1], particle swarm optimization [2, 3], differential evolution [4], artificial

A. Yadav · Anita · J. H. Kim (✉)

Department of Sciences and Humanities, National Institute
of Technology Uttarakhand, Srinagar (Garhwal), Uttarakhand 246174, India
e-mail: jaykim@korea.ac.kr

Anita

e-mail: annusajwan93@gmail.com

A. Yadav · Anita · J. H. Kim

School of Civil, Environmental and Architectural Engineering,
Korea University, Seoul 136-713, South Korea

© Springer Nature Singapore Pte Ltd. 2019

K. Deep et al. (eds.), *Decision Science in Action*, Asset Analytics,
https://doi.org/10.1007/978-981-13-0860-4_3

bee colony [5], gravitational search algorithm [6–10], and many more. As the no free lunch theorem concludes that no algorithm can solve all kind of optimization problems, the choice of a particular optimization algorithm always remains a big question. Many methodologies are proposed in the literature to judge the optimization ability of any optimization algorithm such as its testing of benchmark problems, computational time, number of functions evaluation, success performance, success rate, and convergence. Each mentioned point is very important to understand the quality of any optimization algorithm. In this article, it is focused to study the convergence behavior of algorithms. Some articles are reported in the literature to discuss the convergence of non-deterministic techniques. One of them is the study performed in the case of PSO [11]. Based on the inspiration from the above article, the convergence behavior of gravitational search algorithm (GSA) is discussed over linear and quadratic functions. The next section provides a brief detail of the gravitational search algorithm, and afterward some theoretical establishments are made to study the convergence behavior of GSA.

2 Gravitational Search Algorithm

Gravitational search algorithm is a recent optimization algorithm inspired by the natural laws of motion. It is designed from the idea of Newton's basic laws of motion and active gravitational force between two masses. As GSA is also a population-based optimization algorithms, the population of the GSA is termed as agents and these agents mimic the celestial bodies in the universe. The mass of each agent is defined in a very specific manner as a function of fitness values, and these masses are designed to work as moving bodies with some acceleration which follows the following two natural laws of motion:

1. *Law of gravity*: The working force of attraction between two bodies is directly proportional to the product of their masses and inversely proportional to the square of their distances [12].
2. *Law of motion*: The force exerted upon anybody is directly proportional to the acceleration of the body.

These two laws are the basis of the GSA and they are inspired from the Newton's law of gravity and law of motion, respectively. Let the position of the i th agent at any instant t in a D -dimensional search space be $X_i^t(x_{i1}^t, x_{i2}^t, \dots, x_{iD}^t)$ for $i = 1 \dots n$. The force of attraction on the i th agent to j th agent is defined as follows:

$$F_{ijD}^t = G^t \times \frac{M_{pi}^t \times M_{aj}^t}{R_{ij}^t} \times (x_{jd}^t - x_{id}^t) \quad (1)$$

where M_{pi} is the passive gravitational mass related to agent i , M_{aj} is the active gravitational mass related to agent j , G^t is gravitational constant, and R_{ij} is the Euclidian distance between two agents i and j at any time t . The Euclidian distance between the two agents i and j is given by the following equation:

$$R_{ij}^t = \|X_i^t, X_j^t\|_2 \quad (2)$$

The gravitational constant G^t is defined as in Eq. 3

$$G^t = G^{t_0} \times \exp\left(\left(-\alpha \frac{iter}{itermax}\right)\right) \quad (3)$$

where G^{t_0} is the initial value of the gravitational constant, $iter$ is the current iteration, $itermax$ is the total number of iterations, and α is a constant.

2.1 Formulation of Gravitational Search Algorithm

The total working force of attraction by the i th agent at time t in a D -dimensional space is given by Eq. 4

$$F_{id}^t = \sum_{j=1, i \neq j}^{ps} rand() F_{ijd}^t \quad (4)$$

where $d = 1, 2, \dots, D$ and $rand()$ is a random number in the interval $[0,1]$. The law of motion says the acceleration of i th agent is given by the following equation:

$$ac_{id}^t = \frac{F_{id}^t}{M_{ii}^t} \quad (5)$$

where M_{ii}^t is the inertial mass of the i th agent. The velocity and position of agents are calculated as follows (Table 1):

$$V_{id}^{t+1} = rand() \times V_{id}^t + ac_{id}^t \quad (6)$$

$$x_{id}^{t+1} = x_{id}^t + V_{id}^{t+1} \quad (7)$$

The gravitational and inertial mass will be updated with the help of following equations:

$$M_{ai} = M_{pi} = M_{ii} \text{ for } i = 1, 2, \dots, ps \quad (8)$$

$$m_i^t = \frac{fit_i^t - worst^t}{best^t - worst^t} \quad (9)$$

$$M_i^t = \frac{m_i^t}{\sum_{i=1}^{ps} m_i^t} \quad (10)$$

where fit_i^t represents the fitness value of the i th agent at time t , and $best^t$ & $worst^t$ may be defined in the following equations:

Table 1 Working procedure of GSA

Gravitational Search Algorithm
<p>Step (1) Initialization Randomly initialize all the particles ($X_1^t, X_2^t, \dots, X_{ps}^t$) of agent size ps in search range $[X_{min}, X_{max}]$ Set iteration $t=0$ Evaluate the fitness values ($fit_1^t, fit_2^t, \dots, fit_{ps}^t$) of $X(\text{Agent})$ Calculate G^t, $best^t$, $worst^t$ and M_i^t for $i = 1, 2, \dots, ps$. Calculate the total force in each direction F_i^t Calculate the ac_i^t and velocity.</p> <p>Step(2) Reproduction and Updating While (Stopping Criterion is not satisfied) do for $i=1: ps$ do $V_i^{t+1} = rand() \times V_i^t + ac_i^t$ $X_i^{t+1} = X_i^t + V_i^{t+1}$ Evaluate the fitness values (fit_i^t) of $X(\text{Agent})$ Update G_i^t, $best_i^t$, $worst_i^t$ and M_i^t. Using feasibility based rule. Calculate the total force in each direction F_i^t Calculate the ac_i^t and velocity. end of for end while</p>

$$best^t = \min(fit_j^t), j \in \{1, \dots, ps\} \quad (11)$$

$$worst^t = \max(fit_j^t), j \in \{1, \dots, ps\} \quad (12)$$

The flowchart of GSA is depicted in Fig. 1.

The next section proposes the idea of convergence of GSA over linear and quadratic optimization problems.

3 Convergence of Gravitational Search Algorithm

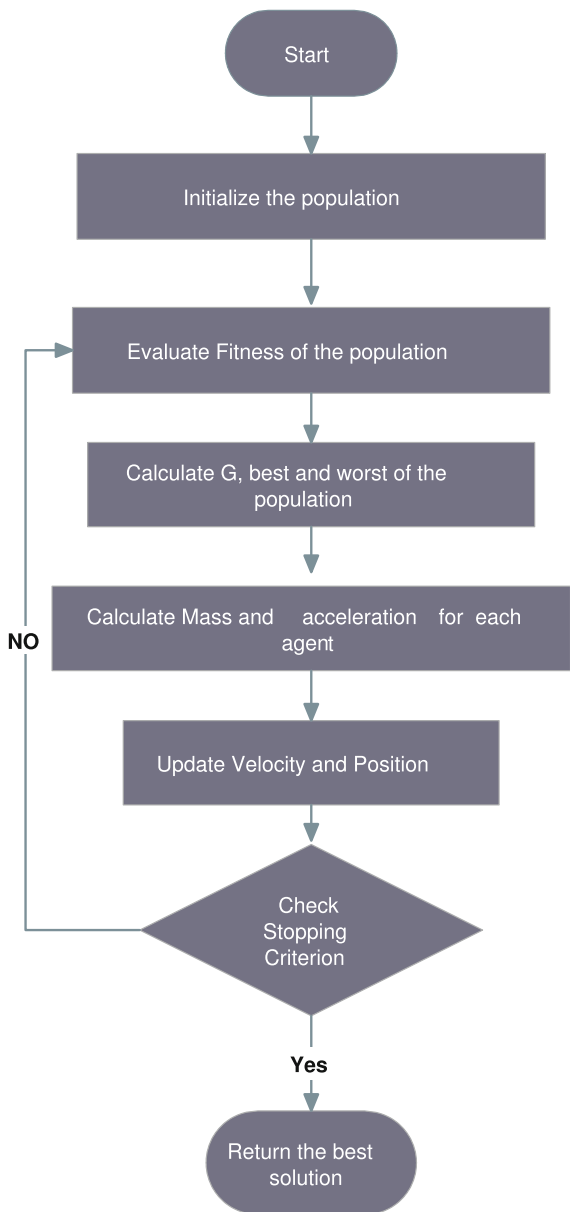
In order to establish theoretical discussion on the convergence of GSA, the following mathematical computations are performed by creating the difference equation of position update procedure. As we know, the velocity update equation of GSA is as follows:

$$V^{t+1} = rand() \times V^t + ac^t \quad (13)$$

Position update equation

$$X^{t+1} = X^t + V^{t+1} \quad (14)$$

Fig. 1 Flowchart of GSA



using Eqs. 13 and 14

$$X^{t+1} - X^t = c_1(X^t - X^{t-1}) + \frac{F^t}{M^t} \quad (15)$$

where $c_1 = rand()$. Now using Eq. 4 and in Eq. 15, we get

$$X^{t+1} - X^t = c_1(X^t - X^{t-1}) + \frac{f(X^t G)}{cX^t} \quad (16)$$

where f is the fitness function and c is the constant. Equation 16 may be written as

$$X^{t+1} - (1 + c_1)X^t + c_1X^{t-1} + \frac{c_2f(X^t)}{X^t} = 0 \quad (17)$$

where $c_2 = \frac{G}{c}$. Let at $t = 0$ and $t = 1$ the position of the particle is $X^{t=0} = X^0$ and $X^{t=1} = X^1$. Characteristic equation of Eq. 17 is given by

Case 1: When f is linear

$$\lambda^2 - (1 + c_1)\lambda + c_1 = 0, \quad (18)$$

$$\implies \lambda_1 = 1, \lambda_2 = c_1,$$

The explicit solution of the recurrence relation Eq. 17 is given by

$$X^t = a_1 \times c + a_2 \times \lambda_1^t + a_3 \times \lambda_2^t \quad (19)$$

where $a_1 = -c$, $a_2 = X^0 - c - \frac{X^0 - X^1}{1 - c_1}$ and $a_3 = \frac{X^0 - X^1}{1 - c_1}$. The necessary and sufficient condition for the convergence of Eq. 17 is

$$\lim_{t \rightarrow +\infty} (a_1 \times c + a_2 \lambda_1^t + a_3 \lambda_2^t) = 0 \quad (20)$$

if $||\lambda_2|| < 1$ then the condition of the convergence will be satisfied and Eq. 17 will converge to $a_1 c$.

Case 2: When f is quadratic

Then, the form of Eq. 17 will be

$$X^{t+1} - (1 + c_1 + c_2)X^t + c_1X^{t-1} = 0 \quad (21)$$

The characteristic equation of Eq. 21 will be

$$\lambda^2 - (1 + c_1 + c_2)\lambda + c_1 = 0, \quad (22)$$

$$\implies \lambda_1 = \frac{1+c_1+c_2+\nu}{2}, \lambda_2 = \frac{1+c_1+c_2-\nu}{2}$$

where $\nu = \sqrt{(1 + c_1 + c_2)^2 - 4c_1}$. Hence, the explicit solution of Eq. 21 is

$$X^t = a_2 \times \lambda_1^t + a_3 \times \lambda_2^t \tag{23}$$

where $a_1 = \frac{(\nu - \lambda_1)X^0 - X^1}{\nu}$, $a_2 = \frac{\lambda_1 X^0 - X^1}{\nu}$.

Again necessary and sufficient condition for the convergence of Eq. 21 will be

$$\lim_{t \rightarrow +\infty} (a_2 \lambda_1^t + a_3 \lambda_2^t) = 0 \tag{24}$$

This will hold when $\max(|\lambda_1|, |\lambda_2|) < 1$ and in this case it will converge to zero.

In order to check the convergence of GSA numerically. The convergence plot of GSA is plotted for the following quadratic and linear order functions:

$$F_1(x) = \sum_{i=1}^n x_i^2$$

$$F_2(x) = \sum_{i=1}^n |x_i|$$

F_1 is a quadratic function and F_2 is a linear order function. As a case study, it has been tried to justify the theoretically established result. Multiple runs of GSA are plotted against the iteration for F_1 and F_2 , and they show a very clear convergence toward the optimal solution in Figs. 2 and 3.

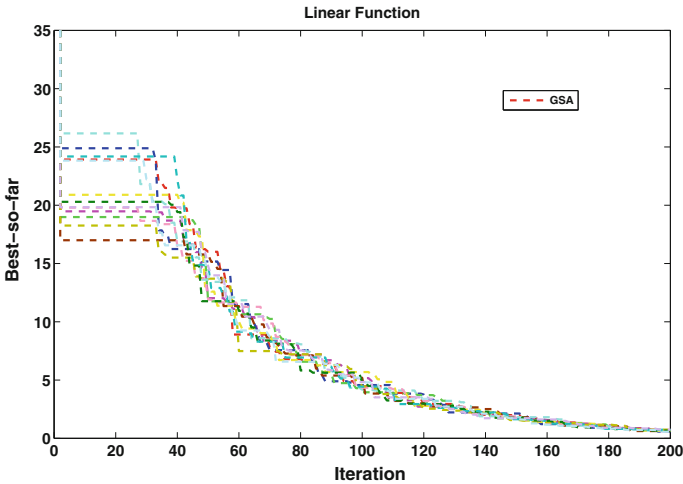


Fig. 2 Convergence of GSA over linear function

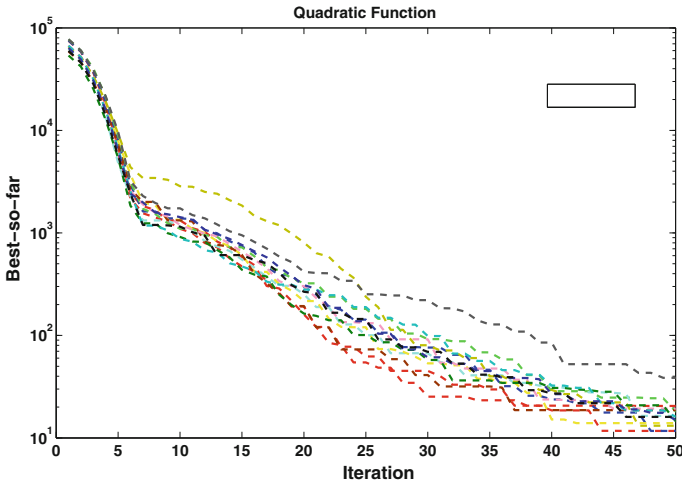


Fig. 3 Convergence of GSA over Quadratic function

4 Conclusion

In this article, the convergence of GSA is studied. Apart from numerical convergence, a theoretical proof of the convergence of GSA is established by using the concept of difference equations. It has been proved that the GSA has the guaranteed convergence; in this case, the problem has linear and quadratic functions. As a case study, one linear and quadratic function is tested using GSA algorithms and their fitness convergence is plotted against the iterations which justifies the theoretical proof of the convergence.

References

1. Goldberg, D.E., Samtani, M.P.: Engineering optimization via genetic algorithm. In: Electronic Computation: ASCE, pp. 471–482 (1986)
2. Eberhart, R., Kennedy, J.: A new optimizer using particle swarm theory. In: Micro Machine and Human Science, 1995. MHS'95, Proceedings of the Sixth International Symposium on, IEEE, pp. 39–43 (1995)
3. Yadav, A., Deep, K.: Shrinking hypersphere based trajectory of particles in pso. Appl. Mathematics and Comput. **220**, 246–267 (2013)
4. Storn, R., Price, K.: Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. J. Global Optim. **11**(4), 341–359 (1997)
5. Karaboga, D., Basturk, B.: A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm. J. Global Optim. **39**(3), 459–471 (2007)
6. Rashedi, E., Nezamabadi-Pour, H., Saryazdi, S.: Gsa: a gravitational search algorithm. Inf. Sci. **179**(13), 2232–2248 (2009)

7. Yadav, A., Deep, K., Kim, J.H., Nagar, A.K.: Gravitational swarm optimizer for global optimization. *Swarm Evolutionary Comput.* **31**, 64–89 (2016)
8. Yadav, A., Deep, K.: Constrained optimization using gravitational search algorithm. *Natl. Acad. Sci. Lett.* **36**(5), 527–534 (2013)
9. Yadav, A., Deep, K.: An efficient co-swarm particle swarm optimization for non-linear constrained optimization. *J. Comput. Sci.* **5**(2), 258–268 (2014)
10. Yadav, A., Yadav, N., Kim, J.H.: A study of harmony search algorithms: Exploration and convergence ability. *Harmony Search Algorithm* 53–62 (2016)
11. Van Den Bergh, F., Engelbrecht, A.P.: A study of particle swarm optimization particle trajectories. *Inf. Sci.* **176**(8), 937–971 (2006)
12. Newton, I.: *The Principia: Mathematical Principles of Natural Philosophy*. Univ of California Press (1999)

An Algorithm of Multivariant Evolutionary Synthesis of Nonlinear Models with Real-Valued Chromosomes



Oleg Monakhov and Emilia Monakhova

Abstract We propose a new multivariant evolutionary algorithm for solving the problem of construction of nonlinear models (mathematical expressions, functions, algorithms, and programs) based on the given experimental data, sets of variables, basic functions, and operations. The proposed algorithm of multivariant evolutionary synthesis of nonlinear models includes a linear representation of a chromosome by real variables, simple operations in decoding of a genotype into a phenotype for interpreting a chromosome as a sequence of instructions, and also a multivariant method for presenting a set of models (expressions) using a single chromosome. We compare the proposed algorithm with the standard genetic programming algorithm (GP) and the Cartesian genetic programming (CGP) one. We show that the proposed algorithm exceeds the GP and CGP algorithms both in the time required for search for a solution (more than by an order of magnitude in the most cases) and in the probability of finding a given model.

Keywords Multivariant evolutionary synthesis · Genetic algorithm · Genetic programming · Cartesian genetic programming

1 Introduction and Basic Definitions

We consider a solution to the problem of building nonlinear models in the form of mathematical expressions, functions, formulas, algorithms, and programs based on the given experimental data, sets of variables, basic functions, and operations. The problem consists in finding a mathematical expression f^* which best describes a nonlinear computation model defined by a set of input (X) and output (Y) experimental data, i.e., it is necessary to select a function $Y = f^*(X)$ that represents the dependence of Y on X with a minimum error. Sometimes, this problem is called symbolic regression or system identification. The search of the expression f is performed based on

O. Monakhov (✉) · E. Monakhova
Institute of Computational Mathematics and Mathematical Geophysics SB RAS,
Pr. Lavrentieva, 6, Novosibirsk 630090, Russia
e-mail: monakhov@rav.sccc.ru

a given set of basic functions and operations $F_1 = \{f_i \mid f_i : R \times \dots \times R \rightarrow R\}$ and a given set of variables and constants $T_1 = \{x_i, c_i\}$, which are used for automatic creation of analytical expressions (formulas) representing the model, and computer programs for their computation. We assume that the objective function (fitness function) FF calculates the sum of the squared deviations of the output data of the expression $Y'_i = f(X_i)$ from the given reference values Y_i for certain subsets of the input data of the expression X_i , $1 \leq i \leq N$:

$$FF = \sum_{i=1}^N (f(X_i) - Y_i)^2,$$

where N is the amount of the experimental data. The goal of the algorithm is to determine $\min_{f \in D(F_1, T_1)} FF(f)$, where $D(F_1, T_1)$ is the set of models defined by a set of basic functions and operations and a set of variables and constants.

The known approaches to this problem include the genetic programming (GP) [1, 3, 7], which is focused primarily on solving the problems of automatic synthesis of programs on the basis of learning data through evolutionary search for models that minimize the representation error. Chromosomes have tree structures that are automatically generated using genetic operators in the GP and represent after an interpretation some expressions and computer programs of various sizes and complexities that implement the expressions.

This paper presents a new algorithm for multivariant evolutionary synthesis (MVES) of nonlinear models. In Sect. 2, we describe a new algorithm for multivariant evolutionary synthesis of nonlinear models. In Sect. 3, we compare our algorithm with two known systems of genetic programming and show that the MVES algorithm has a higher efficiency of evolutionary search. We close with some concluding remarks and suggestions for future work in Sect. 4.

2 Algorithm of Multivariant Evolutionary Synthesis

The multivariant evolutionary synthesis (MVES) algorithm is based on evolutionary computations and simulation of natural selection in a population of individuals, each being points in the space of solutions of the optimization problem, but not a only solution, as in the standard genetic programming algorithm (GP) [1, 2] and in Cartesian genetic programming (CGP) [4, 5], which has a representation of a program in the form of a finite graph. Individuals are data structures (chromosomes), namely, sequences of real-valued numbers, which encode mathematical expressions (formulas and programs). Each population is a set of chromosomes, and each chromosome in this algorithm defines a set of expressions (formulas) arising from it after decoding. The basic idea of the synthesis algorithm is evolutionary transformation of a set of chromosomes (formulas) in the process of natural selection for “the strongest” to survive. In our case, these individuals are expressions that have the smallest value

of the objective function. The algorithm begins with the generation of an initial population. All the individuals in this population are created randomly, and then the best individuals are selected through decoding of the genotype (chromosome) into a phenotype (expression) and calculation of the fitness function. To create the next-generation population (subsequent iteration), new individuals are produced via genetic operations of selection, mutation, and crossover. We assume that the objective function FF calculates the sum of the squared deviations of the output data of the expression $Y'_i = f(X_i)$ from the given reference values Y_i for certain subsets of the input data. In practice, if a few solutions with the same value of the objective function are obtained, then a solution with the minimum estimate of structural complexity is chosen, i.e., with the less total length (the sum of the number of elements) of the solution formulas.

The stages of simulation of the evolution process in the MVES algorithm are as follows.

1. Creating an initial population from randomly generated solutions (chromosomes) as sequences of real-valued numbers. Note that a solution is presented in the chromosome in an encrypted form, i.e., as a genotype.
2. Evaluating the population by the fitness function, which shows how well each individual solves the given problem. In so doing, a genotype is decoded to a phenotype to interpret the chromosome as a program for calculation of the fitness function.
3. Creating a next-generation population using the following evolutionary operators as in the standard genetic algorithm (GA) :
 - 3.1. Selection of the best solution in the population and copying the chromosome into the next generation.
 - 3.2. Creation of new chromosomes by the crossover method.
 - 3.3. Creation of new chromosomes by the mutation method.
4. Repeating points 2 and 3 until a decision meeting a specified criterion is found or the maximum number of generations is reached.

In the MVES algorithm, a new approach is suggested to decode the main data structures, chromosomes. This approach relies on representing the chromosome linear structure as a sequence of three-address instructions and producing a linear operator structure for calculation of expressions (formulas) encoded in the chromosome. To this end, a sequence of real-valued numbers (a chromosome) is divided into groups of three elements (triplets) (h_1, h_2, h_3) with $0 < h_i < 1$. Each such group is interpreted as a three-address instruction as follows: $\langle oper \rangle \langle adr1 \rangle \langle adr2 \rangle$, where the operation $oper$ is applied to operands in the instructions with the numbers $adr1$ and $adr2$, which are calculated by the following formulas:

$$\begin{aligned}
 oper &= \lfloor h_1 * |F| \rfloor, \\
 adr1 &= \lfloor (h_2 * (I - 1)) \rfloor + 1, \\
 adr2 &= \lfloor (h_3 * (I - 1)) \rfloor + 1, \\
 \text{if } oper = 0 \text{ then } adr1 &= \lfloor (h_2 * |T|) \rfloor + 1,
 \end{aligned}
 \tag{1}$$

where $|F|$ is the cardinal number of the set of basic functions, $oper$ is the element number in this set, i.e., the number of the function being executed in the current instruction, I is the number of the current instruction, $adr1$, $adr2$ are the numbers of the preceding instructions, the execution results of which are used as operands in the current instruction, and $|T|$ is the cardinal number of the terminal set. If $oper=0$, then the instruction is interpreted as a loading operator, and the terminal symbol with the number $adr1$ is loaded.

Thus, decoding of a chromosome into an expression (function) results in representation of the function in the form of interpretable code. Each instruction of the code will be considered as a separate function, which includes all the preceding instructions. The first operation will always call in the variable of this function. The execution runs top-down; only precedent records instructions with lower numbers being possible arguments of the instructions. Hence, the execution turns out to be linear. The genetic solution in this case is a set of expression functions, i.e., sequences from the first operator to each current one. This enables, in contrast to the standard GP, simultaneous evaluation of a set of expressions in the form of sequence of operators and reduction in the time of search for the optimal solution. Here, the estimate of a certain chromosome from the set of variants obtained is selected as the estimate of expression that has a minimum value of the objective function. The variables and constants in the formula f make a set of terminal symbols T , and operations used in the formula f make a set of nonterminal ones F .

Let $f = (x + 2)/e^{ax-5}$. For $F = \{L, +, -, /, *, exp, sin\}$ and $T = \{x, a, 2, 5\}$, we have $|F|=7$ and $|T|=4$ (L is the operation of loading (call) of a variable or a constant from the set T the number of which was calculated from the first address). An example of representation of this expression of f in the form of a sequential operator structure with an interpretable code and in a symbolic form is shown in Table 1, where I is the instruction number, K is the chromosome (0.11, 0.14, ..., 0.91) divided into triplets, M is the result of decoding of the triplet via simple operations (1), CM is the instruction itself in a mnemonic record, E is the resulting expression (formula) for this instruction in a symbolic form, and FC is the expression value for each instruction (for $x = 1$, $a = 2$).

Below, there is an example of decoding the instruction $I = 5$ by formulas (1): $K = 0.66:0.91:0.08$; the operation number $oper = [0.66 * 7] = 4$ means the multiplication operation $*$ in F , wherein the first element L has the number 0 here and the first operand $adr1 = [0.91 * (5 - 1)] + 1 = 4$, for the instruction with the number $I_1=4$ we have $E_4 = a$ and $F_4 = 2$, and the second operand $adr2 = [0.08 * (5 - 1)] + 1 = 1$, for the instruction with the number $I_2=1$ we have $E_1 = x$ and $F_1 = 1$, which implies $M_5=4:4:1$, $CM_5 = * 4, 1$, $E_5 = E_4 * E_1 = ax$, and $F_5 = F_4 * F_1 = 2$.

Note that for the given chromosome K , the set of expressions E is found; the value of the objective function is calculated for each expression, and chromosome fitness is defined as the fitness of the best expression encoded by this chromosome. Thus, unlike the GP and CGP algorithms, which encode one solution in a chromosome, this algorithm encodes several solutions in a chromosome, which is also done in the multi-expression programming (MEP) [6], but not using the chromosome as a simple

Table 1 Example of representation of expression in the form of chromosome (K), sequence of operators (CM), and symbols (E)

<i>I</i>	<i>K</i>	<i>M</i>	<i>CM</i>	<i>E</i>	<i>FC</i>
1	0.11:0.14:0.97	0:1:-	$L x$	$E_1 = x$	$F_1 = 1$
2	0.13:0.55:0.15	0:3:-	$L 2$	$E_2 = 2$	$F_2 = 2$
3	0.19:0.16:0.79	1:1:2	+ 1, 2	$E_3 = x + 2$	$F_3 = 3$
4	0.08:0.31:0.47	0:2:-	$L a$	$E_4 = a$	$F_4 = 2$
5	0.66:0.91:0.08	4:4:1	* 4, 1	$E_5 = ax$	$F_5 = 2$
6	0.04:0.79:0.81	0:4:-	$L 5$	$E_6 = 5$	$F_6 = 5$
7	0.35:0.80:0.87	2:5:6	-5, 6	$E_7 = ax - 5$	$F_7 = -3$
8	0.77:0.96:0.19	5:7:-	exp 7	$E_8 = e^{ax-5}$	$F_8 = 0, 0498$
9	0.52:0.26:0.91	3:3:8	/ 3, 8	$E_9 = (x + 2)/e^{ax-5}$	$F_9 = 60, 2566$

vector of real-valued numbers and a standard GA for their evolution, and instead of this organizing evolution on a set of programs.

The process of evolution of population of chromosomes (the vectors of reals) in the MVES involves standard genetic algorithm operators. Let us define an initial population consisting of M arbitrary random vectors of real-valued numbers of a given length l , each number in the interval $(0, 1)$ and l is a multiple of 3. Then, we apply the genetic operators of mutation, crossover, and selection to this population.

Mutation is a replacement of each element of the vector, independent of the others, with a random number in the interval $(0, 1)$ with a probability $p_m \in [0, 1]$.

Crossover. From a population consisting of M vectors, arbitrary pairs are selected M times, and with the probability $p_c \in [0, 1]$ a pair is subjected to the crossover operation as follows: this pair of vectors is divided in an arbitrary position into two parts, which are swapped.

Selection. The objective functions of the new vectors obtained through mutation and crossover are calculated, and the vectors (chromosomes) are decoded into expressions and programs as described above. If their objective functions are less than those of some vectors of the population, then “the worst” vectors in the population (with large values of the objective function) are replaced with the “best” ones, i.e., those having the smallest value of the objective function.

For search for the optimum of a given objective function FF , the iterative process of calculations in the genetic algorithm is organized as follows.

The first iteration: generation of an initial population. A random operator creates all the individuals of the population, which produce the values for each element of the vector in the initial population (these values are uniformly distributed in the interval $(0, 1)$), with subsequent calculation of the objective function.

Intermediate iteration: the step from the current population to the next one. The essence of the algorithm is the creation of a new generation of individuals on the basis of the current population using operations of mutation, crossover, and selection. At

each iteration, there are made M (the size of the population) attempts to select pairs of individuals, which are subjected to operations of crossover (with a probability p_c), mutation (with a probability p_m), and selection.

Last iteration: stopping criterion. The algorithm is completed when a vector with $FF = 0$ is found or after a given number of iterations (generations) t .

Note that the algorithm developed for the multivariant evolutionary synthesis of nonlinear models combines the advantages of the genetic algorithm (simple genetic operations on real-valued vectors, the constant size of which prevents the effect of unreasonable growth of expressions) and genetic programming (automatic synthesis of mathematical expressions and computer programs of various sizes and complexities that implement these expressions). The MVES algorithm combines a linear representation of a chromosome, simple operations (1) in decoding of a genotype into a phenotype for interpretation of a chromosome as a sequence of commands, and also the multivariant method for representing a set of models (expressions) through a single chromosome.

3 Experimental Results

To compare the efficiency of the multivariant evolutionary synthesis algorithm and other genetic programming algorithms for the problem of search for an analytical description of a model on the basis of the given experimental data, the sets of variables, basic functions and operations, we select ten test functions:

Test1: $\sin(\exp(\sin(\exp(\sin(x))))))$, Test2: $\sin(x^2 + x^4)$, Test3: $\sin(x^3) + e^x$, Test4: $x^4 + x^3 + x^2 + x$, Test5: $(x + a)/\sin(2x - 4)$, Test6: $2\sin(x)\exp(a)$, Test7: $\sin(x) + \sin(a^2)$, Test8: $x^5 - 2x^3 + x$, Test9: $\sin(x) + \sin(x^2 + x)$, and Test10: $\sin(x^2)\exp(x) - 1$.

The values of each function at 20 random points in the range $(0, 2)$ were used in the experiments. The set of basic functions for Test1–Test4 is $F_1 = \{+, *, \sin, \exp\}$, for Test5–Test10— $F_2 = \{+, -, *, /, \sin, \exp\}$, the terminal symbols are for Test1–Test4 and Test8–Test10— $T_1 = \{x\}$, and for Test5–Test7— $T_2 = \{x, a, 2, 5\}$.

The following parameters of the compared algorithms are applied: a crossover probability of 0.80, a mutation probability of 0.15, a population size of 100, and a maximum number of generations of 250. The length of the chromosomes is 30; in the GP, the initial depth of the expression tree is set equal to 6; in CGP, the dimensions of the functional network are set as $32 = 2*16$; the maximum number of evolution strategy generations (1+4)-ES for the functions Test1–Test4 is equal to 10,000, for Test5–Test10—75,000. The program was executed 100 times for each algorithm and each test function, and the results were averaged.

The algorithm of the multivariant evolutionary synthesis has been implemented using the resources of the Information and Computing Center (ICC) of the Novosibirsk National Research State University (NSU) on 12-core processor Intel Xeon X5670, 2.93 GHz (Westmere). We compare the MVES algorithm implemented in MATLAB with the following open implementations of genetic programming algorithm (GP)

Table 2 Frequency of successful search for test functions

	GP	CGP	MVES
Test1: $\sin(\exp(\sin(\exp(\sin(x))))))$	0.9	0.25	1
Test2: $\sin(x^2 + x^4)$	0.05	0.35	1
Test3: $\sin(x^3) + e^x$	0.67	0.12	1
Test4: $x^4 + x^3 + x^2 + x$	0.317	0.27	1
Test5: $(x + a)/\sin(2x - 4)$	0	0	0.035
Test6: $2\sin(x)\exp(a)$	0.03	0.71	1
Test7: $\sin(x) + \sin(a^2)$	0.05	0.875	1
Test8: $x^5 - 2x^3 + x$	0.083	0.17	1
Test9: $\sin(x) + \sin(x^2 + x)$	0	0.5	1
Test10: $\sin(x^2)\exp(x) - 1$	0	0.25	0.32

[1, 2] and the Cartesian genetic programming (CGP) [4, 5] in MATLAB: for GP - GPLAB v.4.02 (<http://gplab.sourceforge.net>), for CGP - cgpmatlab (<http://www.cartesiangp.co.uk/resources>).

One of the main indicators used to measure the efficiency of evolutionary algorithms is the probability (frequency) of success, i.e., the probability that the algorithm has detected (synthesized) an expression coinciding exactly with the reference function. This is the ratio of the number of successful experiments, when the algorithm found a correct expression, to the total number of experiments with the given parameters. Table 2 presents the probabilities (frequencies) of success for different algorithms. It can be seen from the table that the probability of success in the MVES is higher than with other algorithms in all cases. This result is explained by the ability of the MVES to represent several expressions in one chromosome, which leads to a higher chance of finding a solution.

The second indicator used in this paper to measure the effectiveness of evolutionary algorithms is the average time of search for test functions, i.e., the average algorithm execution time before the algorithm has detected (synthesized) a specified expression or has fulfilled a required number of generations. Table 3 presents the average time (in sec.) of search for test functions for the compared algorithms. The algorithms can be arranged in order of descending solution time as follows: $GP > CGP > MVES$. This table shows that the MVES has the smallest time for finding a solution (more than by an order of magnitude in most cases). This result can be explained by (1) the simplicity of genetic operations and chromosome structures in the MVES and (2) the direct execution of instructions during decoding of chromosomes “on the fly”, without obtaining the whole expression separately in the form of a string and interpreting the latter, as in the GP and the CGP algorithms.

Table 3 Average time of search for test functions

	GP	CGP	MVES
Test1: $\sin(\exp(\sin(\exp(\sin(x))))))$	23.6	4.4	0.26
Test2: $\sin(x^2 + x^4)$	309.7	3.8	0.26
Test3: $\sin(x^3) + e^x$	118.6	4.1	0.3
Test4: $x^4 + x^3 + x^2 + x$	783.2	3.9	0.27
Test5: $(x + a)/\sin(2x - 4)$	11265	277	3.6
Test6: $2\sin(x)\exp(a)$	4425	135.7	0.32
Test7: $\sin(x) + \sin(a^2)$	4746	106.7	0.33
Test8: $x^5 - 2x^3 + x$	5540	217.9	0.88
Test9: $\sin(x) + \sin(x^2 + x)$	10468	203.3	0.49
Test10: $\sin(x^2)\exp(x) - 1$	9377	214.7	2.5

4 Conclusion and Future Work

We have considered a new approach to addressing the problem of construction of nonlinear models (mathematical expressions, functions, algorithms, and programs) on the basis of the given experimental data, sets of variables, basic functions, and operations. A multivariant evolutionary synthesis algorithm has been developed for such models. The algorithm combines the advantages of genetic algorithm and genetic programming. It uses a linear representation of chromosome, simple operations in decoding of a genotype into a phenotype for interpretation of chromosome as a command sequence, and also the multivariant method for representing a set of models (expressions) using a single chromosome. The proposed multivariant evolutionary synthesis algorithm has been realized and compared with the standard genetic programming algorithm using a tree representation of chromosome and the Cartesian genetic programming algorithm using a representation of a program in the form of a finite graph. The experiments show that the proposed approach exceeds the GP and CGP algorithms both in the time of search for a solution (more than an order of magnitude in the most cases) and in the probability of finding a preset function (model). The main limitation of the research is using it for the synthesis of simple test functions as models. In the future, we will apply our approach to solve some of the real problems for more complex nonlinear models from the fields of biology, financial mathematics, physics, and chemical processes.

References

1. Koza, J.: Genetic Programming II: Automatic Discovery of Reuseable Programs. MIT Press, Cambridge, Massachusetts (1996)
2. Koza, J.R.: Genetic programming as a means for programming computers by natural selection. *Statist. Comput.* **4**, 87–112 (1994)

3. Langdon, W.B., Poli, R.: Foundations of Genetic Programming. Springer, Heidelberg (2002)
4. Miller, J.F.: Cartesian Genetic Programming. Springer, Heidelberg (2011)
5. Miller, J.F., Thomson, P.: Cartesian Genetic Programming. In: Proceedings of the 3rd European Conference on Genetic Programming. LNCS, vol. 1802, pp. 121–132, Springer, Heidelberg (2000)
6. Oltean M.: Multi Expression Programming, Technical Report, Babes-Bolyai Univ, Romania (2006)
7. Poli, R., Langdon, W.B., McPhee, N.F.: A Field Guide to Genetic Programming. Lulu.com, San Francisco (2008)

An Artificial Bee Colony Based Hyper-heuristic for the Single Machine Order Acceptance and Scheduling Problem



Sachchida Nand Chaurasia and Joong Hoon Kim

Abstract This paper presents an artificial bee colony based hyper-heuristic for solving the order acceptance and scheduling (OAS) problem in a single machine environment. The OAS problem gives the flexibility to accept or reject an order where the systems have limited production capacity and on-time delivery constraints. The OAS problem, which is a typical \mathcal{NP} -hard problem, becomes more complex when a sequence-dependent setup time is incurred between two consecutive orders. Solving an \mathcal{NP} -hard problem through exact approaches is computationally expensive and they fail to solve large-size instances. Therefore, we proposed hyper-heuristic in which artificial bee colony (ABC) algorithm is employed as a search methodology for the OAS problem. Hyper-heuristic works on the search space of heuristics, whereas ABC algorithm works on the solution space of the problem. A guided heuristic, which works on search space of heuristics, is developed to search the best heuristic from a set of heuristics residing at the lower level of hyper-heuristic. The proposed approach is compared with the state-of-the-art approaches. The computational results show that the integration of ABC algorithm into hyper-heuristic outperformed the other approaches in terms of average and minimum deviation from the upper bound.

Keywords Artificial bee colony · Order acceptance and scheduling Optimization · Evolutionary algorithm · Guided mutation · Single machine Sequence-dependent setup time · Hyper-heuristic

S. N. Chaurasia
Research Center for Disaster Prevention Science and Technology,
Korea University, Seoul 02841, South Korea
e-mail: snchaurasiacs@gmail.com

J. H. Kim (✉)
School of Civil, Environmental and Architectural Engineering,
Korea University, Seoul 02841, South Korea
e-mail: jaykim@korea.ac.kr

1 Introduction

In a competitive production system, limited production capacity and tight delivery constraints force the system to accept a limited number of orders so that all the accepted orders can be delivered on time and also can generate maximum revenue from the processed orders. A build-to-order (BTO), also referred to as make-to-order, is a manufacturing system which gives flexibility to the system to customize the processing of the orders. The OAS problem [1] addresses both acceptance of orders and scheduling of the accepted orders. A balanced trade-off between acceptance and rejection of orders increases the chance of maximum revenue gain from the production system. The acceptance of an order and on-time delivery maximize earn revenue and also goodwill of customers. On the other hand, rejection of an order avoids the production overload and boosts the production.

In the last two decades, different characteristics and objectives based on OAS problems have been studied. For the detailed study of OAS problems and algorithms to solve them, the literature survey on OAS problem by Keskinocak and Tayur [2] and Slotnick [3] can be referred. The OAS problem can be considered as a mix of subset selection and permutation problem. The joint decision, acceptance of orders and scheduling of the accepted orders, of the OAS problem makes it a complex problem. The OAS problem is proven to \mathcal{NP} -hard [4–6] due to its complex decision procedure.

The OAS problem has application in many real-world complex optimization problems such as printing [7], lamination [4], laundry services [8], and steel production [9]. The OAS problem has two folds—first is to determine which orders should be accepted and the second is to determine the sequence of processing of the accepted orders. The number of incoming orders is fixed, and all the orders are put at a time in a processing sequence.

The OAS problem is solved by the *CPLEX* solver with 3600 s of limited time using mixed-integer linear programming formulation [4]. Cesaret et al. [10] developed three algorithms for the OAS problem, viz., a modified apparent tardiness cost rule-based approach (called *m-ATCS*), a simulated annealing based algorithm (called *ISFAN*), and a tabu search-based algorithm (called *TS*). The tabu search (TS) uses the problem-specific information to make a compound move such as iterative drop, add, and insertion operations. A database keeps the record of two factors. The first factor keeps the status of accepted as well as rejected orders, whereas the second factor keeps the record of a processing sequence of the orders. Using the information from the database, the acceptance and scheduling of orders are done simultaneously. Further, a local search is applied to each solution generated by the TS algorithm to improve the solution quality.

In [11], Lin and Ying developed a swarm-based artificial bee colony algorithm to solve the OAS problem. The neighborhood solutions are produced by either crossover [12] or IG algorithm [13]. The IG algorithm has two phases—first is destruction phase and the second is construction phase.

In [14], Chaurasia and Singh proposed two metaheuristics, viz., steady-state genetic algorithm (SSGA) and evolutionary algorithm with guided mutation (EA/G). Both SSGA and EA/G were coupled with a problem-specific local search to improve the solution. The local search was applied to a solution where the fitness difference between the global best and the current fitness is more than some fixed percentage of the global best.

In this paper, we present an artificial bee colony based hyper-heuristic for the OAS problem in a single machine environment. In the last two decades, hyper-heuristics have received huge attention from the research community and practitioner due to their flexibility toward the selection of heuristic. Hyper-heuristic is considered as an automated heuristic which works on the search space of the heuristic rather than the solution space of the problem.

The major contribution of this paper is as follows: (i) an ABC-based hyper-heuristic is presented for order acceptance scheduling problem; (ii) different from the ABC of [11], the search procedure of scout bee is improved to avoid worse solution. Hereafter, the proposed approach will be referred to as ABC-HH.

We have compared our ABC-HH with the TS [10], ABC [11], hybrid SSGA, and hybrid EA/G [14] approaches on the same test instances.

The remainder of this paper is organized as follows: Sect. 2 describes the problem formulation of the OAS problem. Section 3 is focused on the overview of ABC algorithm and hyper-heuristic. Section 4 describes our ABC algorithm-based hyper-heuristic for the OAS problem. Section 5 is dedicated to the computational results and their analysis. Finally, conclusion and some recommendation for future studies are provided in Sect. 6.

2 Problem Description

We followed the same formulation notation, as given in [11, 14], to describe the order acceptance and scheduling (OAS) problem in a single machine environment. The OAS problem is formally defined as follows:

N	number of incoming orders.
R_i	released date of order i . Processing of order i cannot start before its release date.
P_i	processing time of order i .
C_i	completion time of order i .
D_i	due date of order i .
\overline{D}_i	dead line of order i .
S_{ij} ($i \neq j$)	a sequence-dependent setup time between orders i and j is incurred when order i is processed before order j in the sequence.
E_i	the revenue gain for order i .
$W_i = \frac{E_i}{(D_i - \overline{D}_i)}$, $D_i < \overline{D}_i$	the per unit time tardiness penalty of an order i .

Based on the above information about the orders, the objective of the OAS problem is finding a sequence of all the accepted orders on the single machine that maximizes the total net revenue (TNR). It is assumed that all the orders have equal precedence and will be processed without any interruption. In other words, processing of any order can start any time (after the release date of that order) and once the processing starts, it cannot be stopped before its completion. The mathematical formulation of the objective function of the OAS problem is given in Eq. (1).

$$TNR = \max \sum_{i=1}^N x_i * (E_i - W_i * T_i) \quad (1)$$

where N is the number of orders, $T_i = \max(0, C_i - D_i)$ is the tardiness of an order i . $x_i \forall i \in \{1, 2, \dots, N\}$ are boolean variables. $x_i = 1$ means order i is accepted and $x_i = 0$ means order is rejected. Here, it is to be noted that any order i , where $C_i \geq \bar{D}_i$, is rejected by setting its associated boolean variable x_i to zero.

3 Overview of Hyper-heuristic and ABC

3.1 Hyper-heuristic

Many problem-independent heuristics and metaheuristics have been developed and successfully applied to combinatorial optimization problems. However, difficulties come when one needs to reconstruct an algorithm or have to do time-consuming parameter tuning for the already existing algorithm for a new problem or even for a new instance for the same problem. To overcome the shortcoming of heuristics or metaheuristics, an automated heuristic, called hyper-heuristic, is used as an alternative to solve the problem with least changes in the, already, existing algorithm. Hyper-heuristic uses a heuristic, called search methodology, to search a heuristic or construct a heuristic [15–17]. Generally, hyper-heuristics perform their tasks at two levels—higher level and lower level. The higher level works on search space and it is independent of the problem domain knowledge and mainly engaged in constructing or generating a best possible heuristic from the set of heuristics which reside at the lower level of hyper-heuristic. The lower level directly works on the solution space of the problem and each low-level heuristic can search the solution space, modify the solution, and construct a new solution using the problem domain knowledge [18].

3.2 Artificial Bee Colony (ABC) Algorithm

The artificial bee colony (ABC) algorithm is a recent addition to the class of swarm intelligence algorithms which is developed by Karaboga [19] in 2005. The ABC

algorithm is inspired by the intelligence behavior of natural honey bees in the search of nectar sources. The collaborative and co-operative work of natural honey bees swarm is mapped to the artificial bee colony algorithm. The real bees are distributed into three groups—employed bees, onlooker bees, and scout bees. Employed bees do the job of exploitation of the food source. Employed bees bring the nectar of food sources to the hive and share the information with onlooker bees in the form of dance in the dance area of the hive. The nature and duration of the dance depend on the nectar content of its food source as well as the location of food source with respect to the hive. Onlooker bees watch several dances and choose the best food source. Hence, good food sources, always, have more chance to get selected by onlooker bees. Scout bees do the job of exploration and randomly explore the surroundings of the hive for a new food source. Whenever onlooker and scout bees search a food source, it becomes employed bee.

The ABC algorithm begins with a certain number of randomly generated food sources, and each of these food sources is associated with an employed bee. Each generation of ABC consists of two phases, viz., the employed bee phase and the onlooker bee phase. In employed phase, each employed bee search a new food source in the neighborhood of its associated food source. The employed moves to the new food source if the nectar content of the new food source is higher than its associated food source.

After the completion of employed bee phase, the onlooker bee phase begins and each onlooker bee selects a good food source by applying a probability-based selection method such as roulette wheel selection method and binary tournament selection method and then searches a new food source in the neighborhood of its associated food source. After determining all the food sources, each onlooker bee takes a move to the best food source.

If the food source has not improved for the predetermined number of generations *limit*, then that food source is abandoned and it becomes a scout bee. This scout bee becomes employed bee by associating with a new food source. For a detailed study on the ABC algorithm and its applications, interested readers may refer to [20, 21].

4 ABC-Based Hyper-heuristic (ABC-HH) for the OAS Problem

The proposed ABC-HH approach is designed in such a way that it maintains trade-off between exploration and exploitation in the search process. To the best of authors' knowledge, this is the first ABC-based hyper-heuristic to solve the OAS problem. Hyper-heuristics are automated search method which explore the search space of the solution by. At the higher level of the hyper-heuristic, an ABC algorithm is employed as a search methodology, whereas at the lower level, a set of heuristics which directly work on the solution space of the problem is employed. The subsequent subsections describe the components of the proposed ABC-HH approach.

Algorithm 1 Pseudo-code of ABC-HH

```

1: Randomly generate initial population of  $N_{eb}$  solutions;
2:  $b_{sol} \leftarrow$  Select the best solution from initial solution population;  $\triangleright b_{sol}$  is the global best solution
3: while (termination condition not satisfied) do
4:   for ( $i \leftarrow 1$  to  $N_{eb}$ ) do
5:      $\overline{eb} \leftarrow$  Determine_Neighboring_Solution( $eb_i$ );  $\triangleright$  Use Algorithm 2 to select a heuristic
       and then apply that heuristic on  $eb_i$  to generate a new solution  $\overline{eb}$ 
6:     if ( $fitness(\overline{eb}) > fitness(eb_i)$ ) then
7:        $eb_i \leftarrow \overline{eb}$ ;
8:     else
9:       if ( $eb_i$  has not changed over last limit generations) then
10:         $eb_i \leftarrow$  Randomly_Generate_Solution();  $\triangleright$  Use initial population generation
            method to generate a new solution
11:      end if
12:    end if
13:    if ( $fitness(eb_i) > fitness(b_{sol})$ ) then
14:       $b_{sol} \leftarrow eb_i$ ;
15:    end if
16:  end for
17:  for ( $i \leftarrow 1$  to  $N_{ob}$ ) do
18:     $s_i \leftarrow$  bst( $eb_1, eb_2, \dots, eb_{N_{eb}}$ );  $\triangleright$  bst is a binary tournament selection function that
        returns an index of an employed bee from the set of employed bees  $N_{eb}$ 
19:     $ob_i \leftarrow$  Determine_Neighboring_Solution( $eb_{s_i}$ );  $\triangleright$  Algorithm 2 is used to
        select a heuristic and then the selected heuristic is applied to the solution  $eb_{s_i}$  to generate a new
        solution  $ob_i$ 
20:    if ( $fitness(ob_i) > fitness(b_{sol})$ ) then
21:       $eb_{s_i} \leftarrow ob_i$ ;
22:    end if
23:  end for
24:  if ( $fitness(ob_i) > fitness(b_{sol})$ ) then
25:     $b_{sol} \leftarrow ob_i$ ;
26:  end if
27: end while
28: return  $b_{sol}$ ;

```

4.1 Solution Representation

We followed the same solution representation method as used in [11, 14]. Each solution $\eta = (\eta_1, \eta_2, \dots, \eta_N)$ is a linear permutation of all the N orders. An order η_i in the permutation sequence is accepted if the completion time, C_{η_i} , is less than the dead line \overline{D}_{η_i} .

4.2 Fitness Evaluation

The fitness function is same as the objective function (Eq. 1) and evaluates the fitness of only accepted orders in the permutation η . In Algorithm 1, *fitness* is a function that calculates the fitness of the accepted orders using Eq. 1.

4.3 Initial Population

The initial population is generated randomly with the motive to maintain diversity in the population. A random sequence or permutation η of N orders is generated.

4.4 Determination of Neighborhood Solution

Neighborhood solution is determined by using *higher level search methodology*. The *higher level search methodology* returns a heuristic H_i , and then the heuristic H_i is applied on the solution which is passed in the *Determine Neighboring Solution()* function in Algorithm 1.

I. **Higher level search methodology:** Higher level search methodology is the higher level of hyper-heuristic, and it is used to select a heuristic from a set of heuristics which reside at the lower level of hyper-heuristic. The proposed higher level search methodology uses two steps to select a heuristic. The first step is credit assignment rule and the second step is heuristic selection rule. *credit assignment rule* is used to assign probability to each heuristic h_i . In *heuristic selection rule*, a heuristic is selected using the probability vector δ .

A: **Credit assignment rule:** In a two-dimensional matrix of W rows and H columns, where W and H represent the size of window and number of heuristics, respectively, f_{Hij} represents the fitness returned by the heuristic H_i at stage j in the current generation. After each generation, the fitness matrix f_{matrix} is updated using first-in-first-out (FIFO). FIFO means, when the $W + 1$ fitness is appended into the window, then the first will be removed.

$$f_{matrix} = \begin{matrix} & & \text{\textit{H heuristics}} \\ & \text{\textit{(W) Window size}} & \left[\begin{array}{cccc} f_{1H_1} & f_{1H_2} & \cdots & f_{1H_h} \\ f_{2H_1} & f_{2H_2} & \cdots & f_{2H_h} \\ \vdots & \vdots & \ddots & \vdots \\ f_{WH_1} & f_{WH_2} & \cdots & f_{WH_h} \end{array} \right] \end{matrix}$$

The f_{matrix} is initialized using $W \times H$ number of solutions. The probability δ_{H_i} is initialized using Eq. (2). f_{jH_i} is the fitness of H_i at stage j and $\max(f_{jH_i})$ is the best fitness returned by heuristic H_i for $j = 1, 2, \dots, W$.

$$\delta_{H_i} = \frac{\max(f_{jH_i})}{\sum f_{jH_i}}, j = 1, 2, \dots, W, \quad i = 1, 2, \dots, H \quad (2)$$

The probability vector δ_H is updated after each generation using Eq. (3). In Eq. (3), Z ($1 \leq Z \leq W$) is size of partial window, i.e., last Z rows in f_{matrix} . For example, after each generation, the probability vector is updated using Eq. (3).

Variable Z is equal to the size of *parent*.

$$\delta_{H_i} = (1 - \lambda) \times \delta_{H_i} + \lambda \times \frac{\max(f_{jH_i})}{\sum_{k=1}^Z f_{kH_i}}, i = 1, 2, \dots, H, j = 1, 2, \dots, W \quad (3)$$

where λ is a learning rate. The larger the value of λ more contribution from the current window, the smaller value of λ means more contribution from the parent window.

- B: Heuristic selection rule:** It is inspired by the *guided mutation* of [22]. The guided heuristic (GH) (Algorithm 2) is developed to choose a heuristic from a set of heuristics. Similar to *guided mutation*, the guided heuristic (GH) uses both the global information which is stored in the form of probability δ_H and the location information about the fitness of the parent heuristics to generate a neighborhood solution. In Algorithm 2, a random value *rand* is generated uniformly in $[0, 1]$ and if the value *rand* is less than the probability, δ_{H_i} , of heuristic H_i then the heuristic H_i is applied to generate a new solution. Otherwise, the heuristic is selected which has highest probability among all the heuristics.

Algorithm 2 Guided heuristic

```

1: for each heuristic  $H_i \in H$  do
2:   if ( $rand < \delta_{H_i}$ ) then
3:     return heuristic  $H_i$ 
4:   else
5:     return the heuristic,  $H_i$ , whose probability,  $\delta_{H_i}$ , is highest among all the heuristics;
6:   end if
7: end for

```

- II. Lower level heuristics:** The lower level of hyper-heuristic consists of a set of heuristics. Each heuristic H_i has its own advantage. Some of the heuristics are developed with the purpose to explore the solution space, and some are developed with the purpose to exploit the solution space. For example, heuristic H_1 is used to explore the solution space. Heuristic H_3 is used to exploit the solution space. The detailed description of all the heuristics is given in Table 1.

5 Computational Results

The proposed ABC-HH has been implemented in C language and executed on a Linux-based system having 3.30 GHz Intel Core i5-4590 processor and 4GB RAM. gcc 5.4.0 compiler with O3 flag has been used to compile the C code. For the developed ABC-HH approach, we set N_{eb} (number of employed bees) = 50,

Table 1 Constructive low-level heuristics for the OAS problem

H1	Mutation operator [14]: The mutation operator is used to maintain diversity in the solution population. The proposed mutation operator begins with randomly deleting a fixed number, M_d , of orders and then the deleted orders are assigned with the help of <i>Assignment operator</i> [14]
H2	Multi-swap [23]: Two different positions of orders are selected uniformly at random and orders at the selected positions are swapped. This process is repeated mw_p of times
H3	Crossover operator [14]: As the OAS problem has characteristics of subset selection and permutation problem, we developed a problem-specific crossover to generate a new solution. After crossover operation, some orders remain unsigned. These unsigned orders are assigned with the help of <i>Assignment operator</i> [14]
H4	Local Search [14]: Improve the solution iteratively by interchanging the adjacent orders

N_{ob} (number of onlooker bees) = 50, $limit = 10$, $\lambda = 0.70$, $mw_p = 4$, and $M_d = 4$. The number of generation is set to 250.

The proposed ABC-HH approach has been tested on the same instances which have been used for the state-of-the-art approaches [10, 11, 14]. The details about the test instances can be found in [10]. The evaluation criterion of the performance of the approaches is the percentage deviation of total net revenue (TNR) returned by each approach from the upper bound (UB) on each of the 250 instances of with 15 orders. The percentage deviation is calculated using Eq. (4).

$$\%Deviation \text{ from } UB = \frac{(UB - TNR)}{UB} \times 100\% \quad (4)$$

where TNR indicates the total net revenue obtained by the approach under consideration and UB is the upper bound obtained by Cesaret et al. [10]. We have reported the maximum, average, and minimum percentage (%) deviations from the UB of various approaches on each group of 10 instances with same N (number of orders), τ (tardiness factor), and R (due date range). Table 2 reports the results obtained by TS [10], ABC [11], HSSGA [14], EA/G-LS [14], and ABC-HH approaches for the instances with 15 orders. In Table 2, Max., Min., and Avg. are maximum, minimum, and the average deviation from UB , respectively. Columns of Table 2 also report the number of optimal solutions found in each group of 10 instances by each of the 5 approaches and their average execution times. As TS [10] and ABC [11], HSSGA [14], and EA/G-LS [14] approaches were executed on the systems whose configuration is different from the system used to execute ABC-HH approach, and therefore, execution times of different approaches in these tables cannot be compared directly.

Table 2 Performance of various approaches on instances with 15 orders

N = 15		% Deviations from UB												Number of optimal solutions							
τ	R	TS			ABC			HSSGA			EA/G-LS			ABC-HH			TS	ABC	HSSGA	EA/G-LS	ABC-HH
		Max.	Avg.	Min.	Max.	Avg.	Min.	Max.	Avg.	Min.	Max.	Avg.	Min.	Max.	Avg.	Min.					
0.1	0.1	3.57	2.70	1.19	3.55	2.34	0.00	3.55	1.96	0.00	3.55	2.07	0.00	2.96	1.33	0.00	0	1	1	1	1
	0.3	13.73	3.19	0.00	7.19	3.01	1.20	6.21	2.76	0.00	6.54	2.85	0.00	12.56	2.51	0.00	1	0	1	1	1
	0.5	3.65	1.71	0.00	3.65	1.59	0.00	3.65	1.46	0.00	3.65	1.49	0.00	3.65	0.92	0.00	1	1	3	3	3
	0.7	4.45	1.77	0.00	3.86	0.99	0.00	3.87	1.01	0.00	4.45	1.33	0.00	4.84	0.84	0.00	1	5	6	3	3
	0.9	6.08	1.32	0.00	6.08	1.16	0.00	6.08	1.16	0.00	6.08	1.16	0.00	4.31	0.55	0.00	4	5	5	5	5
0.3	0.1	8.05	4.42	3.36	5.75	3.21	0.00	5.75	2.77	0.00	6.90	3.06	0.00	5.75	2.59	0.00	0	1	1	1	1
	0.3	11.33	4.92	1.57	11.33	3.94	1.08	11.33	3.73	0.00	11.33	3.73	0.00	11.33	3.64	0.00	0	0	1	1	1
	0.5	8.04	4.61	1.57	7.76	4.05	1.08	6.88	3.88	0.00	7.76	4.05	0.00	7.76	3.72	0.00	0	0	0	0	0
	0.7	7.57	3.97	1.55	7.57	2.98	1.17	7.57	2.91	1.17	7.57	3.03	1.17	7.57	2.44	0.00	0	0	0	0	0
	0.9	7.10	3.79	0.00	7.10	2.94	0.00	7.10	2.94	0.00	7.10	2.94	0.00	4.22	1.22	0.00	2	2	3	3	3
0.5	0.1	12.56	7.48	2.47	12.56	7.00	2.47	12.56	7.00	2.47	12.56	7.00	2.47	12.56	6.54	2.47	0	0	0	0	0
	0.3	13.73	7.68	4.26	13.73	7.51	4.27	13.73	6.89	3.05	13.73	6.89	3.05	8.62	6.41	3.85	0	0	0	0	0
	0.5	15.29	9.35	5.81	15.29	9.17	4.07	15.29	9.10	3.88	15.29	9.11	4.07	15.29	8.80	3.88	0	0	0	0	0
	0.7	11.00	6.47	1.16	11.00	6.09	0.09	11.00	6.04	0.09	11.00	6.04	0.09	11.00	6.16	0.00	0	0	1	1	1
	0.9	18.49	6.36	0.10	18.39	6.09	0.10	18.39	6.09	0.10	18.39	6.09	0.10	18.39	5.41	0.00	1	1	1	1	1
0.7	0.1	4.06	0.70	0.00	2.21	0.30	0.05	0.10	0.08	0.00	0.10	0.08	0.00	0.10	0.08	0.00	7	9	10	10	10
	0.3	2.78	0.57	0.07	0.10	0.09	0.06	1.25	0.21	0.07	0.10	0.09	0.06	0.10	0.07	0.00	7	10	9	10	10
	0.5	2.40	0.43	0.07	0.10	0.08	0.00	0.13	0.09	0.07	0.10	0.08	0.00	0.77	0.13	0.00	7	10	10	10	10
	0.7	8.28	2.05	0.08	8.28	1.03	0.03	8.28	0.90	0.03	8.28	1.03	0.03	8.28	1.24	0.00	3	6	9	8	8
	0.9	0.27	0.11	0.08	0.10	0.08	0.00	0.10	0.08	0.00	0.10	0.08	0.00	0.27	0.08	0.00	9	8	10	10	10

(continued)

6 Conclusions

In this work, we have presented an artificial bee colony based hyper-heuristic for the order acceptance and scheduling (OAS) problem in a single machine environment with release date and sequence-dependent setup time. The developed approach is compared with the state-of-the-art approaches, viz., tabu search, artificial bee colony algorithm, steady-state genetic algorithm, and evolutionary algorithm with a guided mutation. The computational results show that the hyper-heuristic can be used as an alternative to heuristics and metaheuristics. As a future work, we would like to develop other swarm- and evolutionary-based hyper-heuristics for the OAS problem in the multi-machine environment. Similar to the proposed ABC algorithm-based hyper-heuristic, other metaheuristics such as genetic algorithm, particle swarm optimization, etc. based on hyper-heuristic can be proposed to solve the OAS problem.

Acknowledgements This work was supported by the grant from The National Research Foundation (NRF) of Korea, funded by the Korea government (MSIP) (No. 2016R1A2A1A05005306).

References

1. Guerrero, H., Kern, G.: How to more effectively accept and refuse orders. *Production and Inventory Management* **29**, 59–62 (1988)
2. Keskinocak, P., Tayur, S.: Due date management policies. *Handbook of Quantitative Supply Chain Analysis, International Series in Operations Research & Management Science* **74**, 485–554 (2004)
3. Slotnick, S.: Order acceptance and scheduling: A taxonomy and review. *European Journal of Operational Research* **212**, 1–11 (2011)
4. Oğuz, C., Salmana, F., Yalçın, Z.: Order acceptance and scheduling decisions in make-to-order systems. *International Journal of Production Economics* **125**, 200–211 (2010)
5. Ghosh, J.: Job selection in a heavily loaded shop. *Computers & Operations Research* **24**, 141–145 (1997)
6. Slotnick, S., Morton, T.: Order acceptance with weighted tardiness. *Computers & Operations Research* **34**, 3029–3042 (2007)
7. Herbots, J., Herroelen, W., Leus, R.: Dynamic order acceptance and capacity planning on a single bottleneck resource. *Naval Research Logistics* **54**, 874–889 (2007)
8. Xiao, Y.Y., Zhang, R.Q., Zhao, Q.H., Kaku, I.: Permutation flow shop scheduling with order acceptance and weighted tardiness. *Applied Mathematics and Computation* **218**, 7911–7926 (2012)
9. Rom, W., Slotnick, S.: Order acceptance using genetic algorithms. *Computers & Operations Research* **36**, 1758–1767 (2009)
10. Cesaret, B., Oğuz, C., Salman, F.: A tabu search algorithm for order acceptance and scheduling. *Computers & Operations Research* **39**, 1197–1205 (2012)
11. Lin, W., Ying, K.C.: Increasing the total net revenue for single machine order acceptance and scheduling problems using an artificial bee colony algorithm. *Journal of the Operational Research Society* **64**, 293–311 (2013)
12. Davis, L.: *Handbook of Genetic Algorithms*. Van Nostrand Reinhold, New York (1991)
13. Ruiz, R., Stützle, T.: A simple and effective iterated greedy algorithm for the permutation flowshop scheduling problem. *European Journal Operational Research* **177**, 2033–2049 (2007)

14. Chaurasia, S.N., Singh, A.: Hybrid evolutionary approaches for the single machine order acceptance and scheduling problem. *Applied Soft Computing Journal* **52**, 725–747 (2017)
15. Burke, E.K., Hyde, M., Kendall, G., Ochoa, G., Özcan, E., Woodward, J.R.: *A Classification of Hyper-heuristic Approaches*, pp. 449–468. Springer US, Boston, MA (2010)
16. Burke, E.K., Gendreau, M., Hyde, M., Kendall, G., Ochoa, G., Özcan, E., Qu, R.: Hyper-heuristics: a survey of the state of the art. *Journal of the Operational Research Society* **64**(12), 1695–1724 (2013)
17. Sabar, N.R., Ayob, M., Kendall, G., Qu, R.: Automatic design of a hyper-heuristic framework with gene expression programming for combinatorial optimization problems. *IEEE Transactions on Evolutionary Computation* **19**(3), 309–325 (2015)
18. Burke, E., Kendall, G., Newall, J., Hart, E., Ross, P., Schulenburg, S.: *Hyper-Heuristics: An Emerging Direction in Modern Search Technology*, pp. 457–474. Springer US, Boston, MA (2003)
19. Karaboga, D.: An idea based on honey bee swarm for numerical optimization. Tech. rep. (2005)
20. Karaboga, D., Akay, B.: A survey: algorithms simulating bee swarm intelligence. *Artificial Intelligence Review* **31**(1), 61–85 (2009)
21. Karaboga, D., Gorkemli, B., Ozturk, C., Karaboga, N.: A comprehensive survey: artificial bee colony (abc) algorithm and applications. *Artificial Intelligence Review* **42**(1), 21–57 (2014)
22. Zhang, Q., Sun, J., Tsang, E.: An evolutionary algorithm with guided mutation for the maximum clique problem. *IEEE Transactions on Evolutionary Computation* **9**, 192–200 (2005)
23. Sundar, S., Singh, A.: A swarm intelligence approach to the early/tardy scheduling problem. *Swarm and Evolutionary Computation* **4**, 25–32 (2012)

A New Evolutionary Optimization Method Based on Center of Mass



Jesús-Adolfo Mejía-de-Dios and Efrén Mezura-Montes

Abstract Physical phenomena have been the inspiration for proposing different optimization methods such as electro-search algorithm, central force optimization, and charged system search among others. This work presents a new optimization algorithm based on some principles from physics and mechanics, which is called Evolutionary Centers Algorithm (ECA). We utilize the center of mass definition for creating new directions for moving the worst elements in the population, based on their objective function values, to better regions of the search space. The efficiency of the new approach is showed by using the CEC 2017 competition benchmark functions. We present a comparison against the best algorithm (jSO) in such competition and against a classical method (SQP) for nonlinear optimization. The results obtained are promising.

Keywords Optimization · Center of mass · Evolutionary algorithm
Physics-inspired

1 Introduction

Nowadays, real-world optimization problems are complex to solve due to different sources of difficulty, e.g., highly nonlinear objective function and constraints and large number of variables. There are several population-based algorithms, which are competitive to solve optimization problems [4]. Two main types can be distinguished: evolutionary algorithms (EAs), e.g., genetic algorithms, differential evolution, etc., [3, 10, 13], and swarm intelligence, e.g., artificial bee colony, ant system, particle swarm optimization, etc. [7, 8]. In this work, we are focused on EAs.

J.-A. Mejía-de-Dios · E. Mezura-Montes (✉)
Artificial Intelligence Research Center, University of Veracruz,
Sebastián Camacho 5, Centro, Xalapa 91000 Veracruz, Mexico
e-mail: emezura@uv.mx
URL: <https://www.uv.mx/ciia/>

J.-A. Mejía-de-Dios
e-mail: jesusmejded@gmail.com

EAs have provided successful results when solving complex bound-constrained optimization problems [13]. However, most popular EAs usually are those which design keeps simple and their number of parameters is low so as to facilitate the fine-tuning process when a particular problem is solved.

Motivated by the above mentioned, we propose a physics-inspired algorithm based on the center of mass concept on a D -dimensional space for real-parameter single-objective optimization. The general idea is to promote the creation of an irregular body using K mass points in the current population, then the center of mass is calculated to get a new direction for the next population.

Single-objective optimization problems are defined as follows: for an objective function $f(\mathbf{x})$, an algorithm needs to find the variables of a vector \mathbf{x} that minimizes or maximizes the function f . It is assumed that the number of variables in \mathbf{x} is D , i.e., $\mathbf{x} = (x_1, x_2, \dots, x_D)$. The search space is assumed to be convex, where each variable has its boundaries $x_{j,\min}, x_{j,\max}$ for $j = 1, 2, \dots, D$. Problems are often found where the objective function is not explicitly known, then classical optimization methods in this type of problem are hardly applicable [6].

There are different algorithms based on biological or physical metaphors with different characteristics. Some of them use the current population distribution to generate new solutions, i.e., swarm intelligence algorithms such as particle swarm optimization (PSO) [8] and the artificial bee colony (ABC) [7]. There are also algorithms inspired by physical phenomena such as Newton's Law of Universal Gravitation (CFO) [2, 5]. The relationship among those algorithms is their mathematical formulation for generating solutions through an iterative process:

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \mathbf{v}_{i+1}, \quad (1)$$

where each algorithm updates \mathbf{v}_{i+1} as follows:

– PSO:

$$\mathbf{v}_{i+1} = \omega \mathbf{v}_i + c_1 r_{1,i} (\mathbf{x}_{pbest,i} - \mathbf{x}_i) + c_2 r_{2,i} (\mathbf{x}_{gbest,i} - \mathbf{x}_i),$$

where ω is a inertia weight used for balancing the global search and local search; $\mathbf{x}_{pbest,i}$ and $\mathbf{x}_{gbest,i}$ are the best position reached by solution i so far and the best solution in the population, respectively; c_1 and c_2 are two positive constants; and $r_{1,i}, r_{2,i}$ are random numbers with uniform distribution in the range $[0, 1]$.

– ABC:

$$\mathbf{v}_{i+1} = \phi_i (\mathbf{x}_i - \mathbf{x}_r),$$

where \mathbf{x}_i is the current solution, \mathbf{x}_r is a randomly chosen solution, and ϕ_i is a randomly produced number with uniform distribution in the interval $[-1, 1]$.

– CFO:

$$\mathbf{v}_{i+1} = \omega \mathbf{v}_i + \lambda \mathbf{F}_i / m_i,$$

where \mathbf{v}_i is the current solution, λ is a uniformly distributed random variable in $[0, 1]$, ω is the user-defined weight $0 < \omega < 1$, m_i , F_i are the mass and force functions, respectively, both defined by the authors.

In the three previous cases, the \mathbf{v} value depends on the population distribution at current generation i .

Section 2 describes our algorithm and how it relates to what has been described above. Section 3 presents the results obtained. Section 4 summarizes our conclusions and Sect. 5 indicates the future work.

2 Evolutionary Centers Algorithm

In this section, evolutionary centers algorithm (ECA) is detailed. Also, experiments are presented.

2.1 Motivation

The center of mass is a geometric property of any object. Intuitively, it is the average location of the weight of an object. We can completely describe the motion of any object through space in terms of the translation of the center of mass of the object from one place to another, and the rotation of the object about its center of mass if it is free to rotate. This is the motivation for using the center of mass concept, and we translate the population to places where the mass of the entire population is maximum.

We present ECA details. First, we introduce the center of mass in physics terms [9, 12].

Definition 1 The center of mass is the unique point \mathbf{c} at the center of a distribution of mass $U = \{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_K\}$ in a space that has the property that the weighted sum of position vectors relative to this point is zero. That is,

$$\sum_{i=1}^K m(\mathbf{u}_i)(\mathbf{u}_i - \mathbf{c}) = 0, \quad \text{implies} \quad \mathbf{c} = \frac{1}{M} \sum_{i=1}^K m(\mathbf{u}_i)\mathbf{u}_i, \quad (2)$$

where $m(\mathbf{u}_i)$ is the mass of \mathbf{u}_i and M is the sum of the masses of vectors in U . Here, m is a nonnegative function.

Note 1 Similar as in statistics, the center of mass is the mean location of a distribution of mass in space.

The concept of *center of mass* is, by far, not new. It was introduced by the ancient Greek physicist, mathematician, and engineer Archimedes of Syracuse. Archimedes

worked with some assumptions about gravity in a uniform field, so as to get the mathematical properties of what we now call the center of mass [9].

For this work, the following proposition is required for ensuring stability and keep ECA solutions into the convex space.

Proposition 1 *If \mathbf{c} is the center of mass of a system of particles U , then for all $\mathbf{u} \in U$:*

$$d(\mathbf{c}, \mathbf{u}) \leq \text{diam}(U).$$

Here, $\text{diam}(U) := \sup\{d(\mathbf{u}, \mathbf{v}) \mid \mathbf{u}, \mathbf{v} \in U\}$.

In other words, the center of mass of U is never out of the minimum convex set that contains U . We are assuming Euclidean distance and $U \subset \mathbb{R}^D$ [14].

In this work, the objective function of the optimization problem represents the mass of each solution in the population, i.e., we set $f = m$. Without loss of generality, we assume that we want to maximize the nonnegative function f .

2.2 Algorithm Description

For each solution \mathbf{x}_i in the population $P = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$ of N solutions, we select a subset $U \subset P$ with K solutions; then, from U we obtain the center of mass \mathbf{c} . After that, based on a randomly chosen solution $\mathbf{u}_r \in U$, and the already generated center of mass \mathbf{c} , we generate a direction to locate a new solution \mathbf{h}_i . We suggest using the following strategy:

$$\mathbf{h}_i = \mathbf{x}_i + \eta_i(\mathbf{c} - \mathbf{u}_r), \quad (3)$$

where

$$\mathbf{c}_i = \frac{1}{W} \sum_{\mathbf{u} \in U} f(\mathbf{u}) \cdot \mathbf{u}, \quad W = \sum_{\mathbf{u} \in U} f(\mathbf{u}). \quad (4)$$

Note 2 If f is constant, then the center of mass of U is the geometric center of U . That is, assume that $f(\mathbf{x}) = \alpha$ for every $\mathbf{x} \in \mathbb{R}^D$, with α a positive constant. The center of mass is

$$\mathbf{c}_i = \frac{1}{K\alpha} \sum_{\mathbf{u} \in U} \alpha \cdot \mathbf{u} = \frac{1}{K} \sum_{\mathbf{u} \in U} \mathbf{u}. \quad (5)$$

Thus, for a constant mass function, we have the center of mass converging to the geometric center. In real-world problems, functions can be flat in some regions, and then this algorithm may find some difficulties when dealing with such issue.

Note 3 The bias is given by Eq. (4) because for a solution with the highest mass, the position of the center of mass is nearest to its position, see Fig. 1.

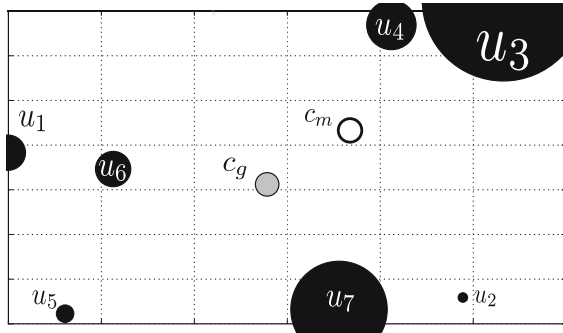


Fig. 1 c_m is center of mass, c_g is geometric center of black points. Black point radius is its mass. Note the bias given by the weighted sum

Algorithm 1 ECA pseudocode

```

1: procedure ECA( $K = 7$ ,  $\eta_{\max} = 2$ )
2:    $N \leftarrow 2K * D$ 
3:   Generate and evaluate start population  $P$  with  $N$  elements
4:   while the end criterion is not achieved do
5:      $A = \emptyset$ 
6:     for each  $\mathbf{x}$  in  $P$  do
7:       Generate  $U \subset P$  such that  $\text{card}(U) = K$ 
8:       Calculate  $\mathbf{c}$  using  $U$  with (4)
9:        $\eta \leftarrow \text{rand}(0, \eta_{\max})$ 
10:       $\mathbf{h} \leftarrow \mathbf{x} + \eta * (\mathbf{c} - \mathbf{u})$  where  $\mathbf{u} \in U$  random
11:      if  $f(\mathbf{x}) < f(\mathbf{h})$  then
12:        Append  $\mathbf{h}$  in  $A$ 
13:      end if
14:    end for
15:     $P \leftarrow$  best elements in  $P \cup A$ 
16:  end while
17:  Report best solution in  $P$ 
18: end procedure

```

Note that ECA has only two parameters: the number of neighbors K and the step size η_{\max} . For large K values, ECA could converge faster, and we suggest $K = 7$, a value obtained experimentally. Figure 2 shows a representation of ECA solution update.

2.3 Experiments

Algorithm 1 details the procedure for the implementation of ECA. Such algorithm was coded in C language using a PC with quad-core 2.4 GHz CPU and 8 GB of RAM, and it was tested in 30 functions of CEC 2017 competition on real-parameter single-objective optimization [1].

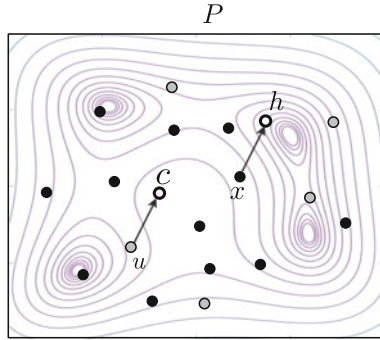


Fig. 2 Schematic diagram representing a generation of ECA. Gray points represent elements in U

Table 1 Representative functions from CEC 2017 benchmark. This set of functions is shifted and rotated. The search range is $[-100, 100]^D$

Function	Formula
Bent cigar function	$f_1(\mathbf{x}) = x_1 + 10^6 \sum_{i=2}^D x_i^2$
Sum of different power functions	$f_2(\mathbf{x}) = \sum_{i=1}^D x_i ^{i+1}$
Zakharov function	$f_3(\mathbf{x}) = \sum_{i=1}^D x_i^2 + \left(\sum_{i=1}^D 0.5x_i\right)^2 + \left(\sum_{i=1}^D 0.5x_i\right)^4$
Rastrigin function	$f_5(\mathbf{x}) = 10D + \sum_{i=1}^D (x_i^2 - 10 \cos(2\pi x_i))$
High conditioned elliptic function	$f_{11}(\mathbf{x}) = \sum_{i=1}^D (10^6)^{\frac{i-1}{D-1}} x_i^2$
Discus function	$f_{12}(\mathbf{x}) = 10^6 x_1^2 + \sum_{i=2}^D x_i^2$
Griewanks function	$f_{15}(\mathbf{x}) = \sum_{i=1}^D \frac{x_i^2}{4000} - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$

For this experimentation, $D = 10$ was considered. Here, the optimal values for the test functions are known (see Table 1 where representative functions are shown). There is also a maximum number of evaluations equal to $10,000D$. The parameters in all experiments were $K = 7$, η_i is a uniform random number between in $(0, 2]$. The size of the population was $N = 2K * D$.

3 Results

The statistical results obtained by ECA are reported in Table 2. It is worth noting that ECA obtained results close to the optimum while reporting low standard deviation values. Therefore, ECA behavior can be considered as robust and suitable to deal with different types of search spaces. Furthermore, we compared ECA against a nonlinear optimization algorithm (SQP) [11], and the most competitive algorithm in the CEC 2017 competition on real-parameter single-objective optimization (jSO), which is

Table 2 Results of 51 independent runs of ECA on CEC17 problems for $D = 10$

f	Best	Worst	Median	Mean	Std.
f_1	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
f_2	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
f_3	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
f_4	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
f_5	9.94967E-01	1.54772E+01	7.95967E+00	7.93134E+00	3.77745E+00
f_6	4.74662E-07	1.87951E-03	1.87537E-05	7.68970E-05	2.64095E-04
f_7	1.11988E+01	2.87323E+01	1.82470E+01	1.79819E+01	4.13487E+00
f_8	0.00000E+00	1.39919E+01	3.97988E+00	5.20411E+00	3.44675E+00
f_9	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
f_{10}	2.48759E+01	1.08470E+03	7.38475E+02	7.15419E+02	1.72816E+02
f_{11}	0.00000E+00	6.57982E+00	9.94986E-01	1.40699E+00	1.58245E+00
f_{12}	0.00000E+00	2.55756E+02	1.14822E+02	7.23057E+01	6.74642E+01
f_{13}	0.00000E+00	1.36689E+01	2.44396E+00	3.57464E+00	3.36532E+00
f_{14}	0.00000E+00	9.86504E+00	9.94959E-01	1.62895E+00	2.14527E+00
f_{15}	4.70850E-03	3.29460E+00	1.13397E+00	1.03424E+00	7.22193E-01
f_{16}	3.90059E-01	2.42227E+01	2.14109E+00	3.42342E+00	3.81635E+00
f_{17}	6.04248E+00	4.77547E+01	3.68447E+01	3.65033E+01	6.29621E+00
f_{18}	1.91438E-02	2.54001E+00	4.26493E-01	5.91709E-01	5.31956E-01
f_{19}	3.15884E-02	1.56140E+00	2.90525E-01	5.22549E-01	4.57284E-01
f_{20}	1.30976E+00	4.53821E+01	2.69242E+01	2.45399E+01	9.82172E+00
f_{21}	1.00000E+02	2.04138E+02	1.00000E+02	1.02042E+02	1.44386E+01
f_{22}	0.00000E+00	1.01678E+02	1.15631E+01	4.87450E+01	4.90438E+01
f_{23}	3.43302E-08	3.20754E+02	3.09754E+02	3.03630E+02	4.32143E+01
f_{24}	1.98982E-07	3.31138E+02	1.00000E+02	1.11616E+02	5.65212E+01
f_{25}	3.97743E+02	4.43546E+02	3.98009E+02	3.99730E+02	8.83947E+00
f_{26}	3.00000E+02	3.00000E+02	3.00000E+02	3.00000E+02	0.00000E+00
f_{27}	3.88861E+02	3.97791E+02	3.93436E+02	3.92839E+02	1.83721E+00
f_{28}	3.00000E+02	3.00000E+02	3.00000E+02	3.00000E+02	0.00000E+00
f_{29}	2.31919E+02	2.87909E+02	2.57749E+02	2.57882E+02	9.95923E+00
f_{30}	3.94649E+02	4.08051E+02	3.95237E+02	3.98513E+02	5.55498E+00

an adaptive algorithm based on differential evolution. The comparison based on 51 independent runs by each algorithm is presented in Table 3. As we can see, ECA, based on the 95%-confidence Wilcoxon rank-sum test, was able to outperform jSO in five test functions; it reached similar results in seven test problems, and finally

Table 3 Comparison of results between ECA, jSO, and SQP in $D = 10$ CEC 2017 test problems. Wilcoxon rank-sum test ($\alpha = 0.05$) was computed. “+” means that ECA outperformed jSO/SQP in the function in the corresponding row, “-” means that jSO/SQP outperformed ECA, and “≈” means that no significant difference was observed between algorithms. Note that SQP algorithm was outperformed by ECA in 29 of 30 functions

f	ECA		jSO	SQP	
f_1	0.00000E+00	≈	0.00000E+00	3.220300E-04	+
f_2	0.00000E+00	≈	0.00000E+00	6.065700E+20	+
f_3	0.00000E+00	≈	0.00000E+00	0.000000E+00	≈
f_4	0.00000E+00	≈	0.00000E+00	6.253000E-01	+
f_5	7.93134E+00	-	1.67777E+00	2.685479E+02	+
f_6	7.68970E-05	-	0.00000E+00	9.300540E+01	+
f_7	1.79819E+01	-	1.20817E+01	5.279684E+02	+
f_8	5.20411E+00	-	1.91188E+00	1.641353E+02	+
f_9	0.00000E+00	≈	0.00000E+00	4.765000E+03	+
f_{10}	7.15419E+02	-	3.83851E+01	1.582000E+03	+
f_{11}	1.40699E+00	-	0.00000E+00	8.694410E+01	+
f_{12}	7.23057E+01	-	3.55067E-01	5.889026E+02	+
f_{13}	3.57464E+00	≈	2.68638E+00	2.617198E+02	+
f_{14}	1.62895E+00	-	1.36563E-01	1.054084E+02	+
f_{15}	1.03424E+00	-	3.00324E-01	9.494020E+01	+
f_{16}	3.42342E+00	-	5.49544E-01	6.348346E+02	+
f_{17}	3.65033E+01	-	5.25569E-01	8.089248E+02	+
f_{18}	5.91709E-01	-	2.17729E-01	1.050214E+02	+
f_{19}	5.22549E-01	-	7.72037E-03	8.497336E+02	+
f_{20}	2.45399E+01	-	3.36657E-01	5.496386E+02	+
f_{21}	1.02042E+02	+	1.42465E+02	3.480549E+02	+
f_{22}	4.87450E+01	+	1.00000E+02	1.470100E+03	+
f_{23}	3.03630E+02	-	3.01261E+02	6.704390E+02	+
f_{24}	1.11616E+02	+	2.96919E+02	4.196612E+02	+
f_{25}	3.99730E+02	+	4.12195E+02	4.360669E+02	+
f_{26}	3.00000E+02	≈	3.00000E+02	1.591800E+03	+
f_{27}	3.92839E+02	-	3.89468E+02	4.350638E+02	+
f_{28}	3.00000E+02	+	3.40596E+02	4.037759E+02	+
f_{29}	2.57882E+02	-	2.34365E+02	2.005400E+03	+
f_{30}	3.98513E+02	-	3.94521E+02	3.957100E+08	+
Mean	116.9022		99.03207	2.021900E+19	

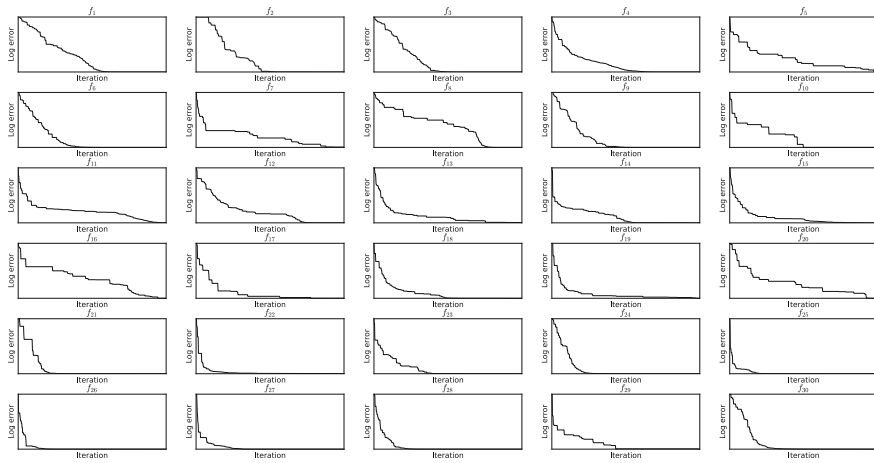


Fig. 3 Convergence graphs at the run located in the median of 51 independent runs. Log scale is used for visualization purposes

ECA was outperformed by jSO in 18 functions. ECA was then competitive in 12 test problems. Moreover, ECA is more simple to implement than jSO and requires less mechanisms to operate. Note that ECA outperformed SQP most of the time.

Figure 3 shows ECA convergence graphs. Those plots show that ECA is able to converge fast in most cases, which can be suitable for computationally expensive real-world optimization problems.

4 Conclusions

A new metaheuristic optimization algorithm, denoted as evolutionary centers algorithm, inspired by the center of mass of a system of particles was proposed. The results showed the capability of ECA to consistently reach the vicinity of the global optima in different types of search spaces. ECA also provided a competitive, but still not better, performance against the winner of the CEC 2017 competition on real-parameter single-objective optimization. ECA is a simple algorithm which requires the fine-tuning of just two parameters, besides the population size.

5 Further Work

Implementing a self-adaptive technique for the ECA parameters and solving constrained optimization problems are part of the future work derived from this current research.

References

1. Awad, N., Ali, M., Q., B., Liang, J., Suganthan, P.: Problem definitions and evaluation criteria for the cec 2017 special session and competition on single objective bound constrained real-parameter numerical optimization (2016)
2. Biswas, A., Mishra, K., Tiwari, S., Misra, A.: Physics-inspired optimization algorithms: a survey. Hindawi Publishing Corporation (2013)
3. Brest, J., Sepesy-Mauec, M., Bokovi, B.: Single objective real-parameter optimization: algorithm jso. In: IEEE Congress on Evolutionary Computation (CEC) pp. 1311–1317 (2017)
4. Fleming, P., Purshouse, R.: Evolutionary algorithms in control systems engineering: a survey. In: Elsevier Science Ltd. (2002)
5. Formato, R.A.: Central force optimization: a new metaheuristic with applications in applied electromagnetics. *Progr. Electromagnetics Res.* pp. 425–491 (2007)
6. Jamil, M., Yang, X.: A literature survey of benchmark functions for global optimization problems. *Int. J. Math. Modell. Numer. Optimisation* **4**, 150–194 (2013)
7. Karaboga, D.: An idea based on honey bee swarm for numerical optimization. Erciyes University. Technical report, Computer Engineering Department, Engineering Faculty (2005)
8. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proc of IEEE International Conference on Neural Network pp. 1942–1948 (1995)
9. Kleppner, D., Kolenkow, R.: *An Introduction to Mechanics*, 2nd edn. McGraw-Hill (1973)
10. Mitchell, M.: *An introduction to genetic algorithms*. MIT Press, Cambridge, MA (1996)
11. Nocedal, J., Wright, S.: *Numerical Optimization*. Springer (2006)
12. Serway, R., Jewett, J.: *Principles of Physics: a Calculus-Based Text*. 4th edn. Thomson Learning (2016)
13. Storn, R., Price, K.: Differential evolution – a simple and efficient adaptive scheme for global optimization over continuous spaces. *J. Global Optim.* (1995)
14. Walter, R.: *Principles of Mathematical Analysis*, 3rd edn. International Series in Pure and Applied Mathematics. McGraw-Hill, New York (1976)

Adaptive Artificial Physics Optimization Using Proportional Derivative Controllers



Liping Xie, Jianchao Zeng, Qiongqiong Yang and Richard A. Formato

Abstract APO (Artificial Physics Optimization) is a physicomimetics-inspired population-based global search and optimization heuristic that can be modeled as a second-order dynamical system. A central concept of physicomimetics is that the tools and techniques of modern physics and engineering may be applied directly to optimization algorithms such as APO. The extended algorithms described in this paper are a realization of this concept. Using the state-space Z-transform, APO's performance is improved by introducing backward and forward PDCs (Proportional Derivative Controllers). Algorithm APO-PD1 employs a backward PDC architecture that allows each particle to predict its location in the optimization landscape based on its then current state of motion. An error signal computed from the distance between the particle's predicted position and the swarm-weighted position is used to adjust the particle's velocity through the decision space (DS) with the result that APO-PD1 is measurably better than APO. APO-PD2 further improves APO by utilizing the same error signal in a forward PDC architecture in which both the particle's current state of motion and its trajectory history are used to predict its future location. This modification improves performance even more by allowing the swarm's particles to change trajectories more quickly. Numerical experiments on a suite of widely employed high-dimensionality benchmarks show that APO-PD2 outperforms both APO-PD1 and APO.

L. Xie · Q. Yang

Complex System and Computational Intelligence Laboratory,
Taiyuan University of Science and Technology, Taiyuan 030024, Shanxi, P.R. China
e-mail: lipingxie1978@163.com

Q. Yang

e-mail: 47793010@qq.com

J. Zeng

School of Computer Science and Control Engineering,
North University of China, Taiyuan, China
e-mail: zengjianchao2015@163.com

R. A. Formato (✉)

Cataldo & Fisher, LLC, P.O. Box 1714, Harwich, MA 02645, USA
e-mail: rf2@ieee.org

© Springer Nature Singapore Pte Ltd. 2019

K. Deep et al. (eds.), *Decision Science in Action*, Asset Analytics,
https://doi.org/10.1007/978-981-13-0860-4_7

Keywords Artificial physics optimization · APO · Global search and optimization
Physicomimetics · PD controller · PDC · Gravitational force · Virtual force
Newton's law

1 Introduction

Because analytical solutions either are not possible or require excessive computer run time, many difficult global search and optimization problems (GSO) are addressed using heuristic algorithms. These algorithms usually are proffered without a firm mathematical basis, at least initially, and fall into two general categories, evolutionary algorithms (EA) and swarm intelligence (SI) algorithms. They are widely used for difficult “real world” problems in a wide range of disciplines, typically engineering, science, medicine, and economics, and they provide excellent results. Most heuristics are based on natural phenomena that often but not always are drawn from biology. SI and EA algorithms mimic some natural process to create an intelligent and efficient search strategy without necessarily knowing anything about the topology of a fitness (objective) function defined on the Decision Space (DS). The topology is referred to as the problem's “landscape”.

An example of a well-established and widely used biology-based EA is the Genetic Algorithm (GA) [1]. It comprises several “operators” that simulate the Darwinian strategy of “survival of the fittest” in the evolution of living organisms. Operations of inheritance, selection, crossover, and mutation, all of which occur in Nature, are applied in a heuristic algorithm that “evolves” a solution to the GSO problem. By contrast, SI algorithms analogize the collective behavior of a group of animals or insects, typical examples being schooling fish, flocking birds, or swarming bees [2, 3]. A well-established and widely used heuristic is Particle Swarm Optimization (PSO) [4, 5], which was the first SI algorithm used in GSO. It mimics the swarming behavior of birds searching for food. Many other heuristics are biology-based and extend even to mimicking the behavior of individual cells, as, for example, do Artificial Immune System (AIS) and Clonal Selection Algorithm (CSA) [6, 7]. AIS simulates the processes involved in mammalian immunology, while CSA mimics clonal selection in the body's immune response to “non-self” cells invading a living organism.

While biology-based algorithms are quite popular and effective, recent years have seen the development of a new class of heuristics that analogize physical laws. These algorithms are inspired by the laws of physics drawn typically from the fields of classical and quantum mechanics, electricity and magnetism, and thermodynamics. Examples include Simulated Annealing (SA) [8], Central Force Optimization (CFO) [9], Artificial Physics Optimization (APO) [10], Gravitational Search Algorithm (GSA) [11], Electromagnetism-like algorithm (EM) [12, 13], Quantum-Inspired Genetic Algorithm (QGA) [14], Quantum-Inspired Particle Swarm Optimization (QPSO) [15], and Big Bang-Big Crunch (BB-BC) [16]. SA, one of the earliest physics-based algorithms, simulates the statistical mechanics of thermal equilibrium. The “Q” algo-

rithms QGA and QPSO invoke quantum physics in their analogies to Nature, while BB-BC simulates the grand-scale creation and destruction of the physical Universe. On a more macro level, EM uses Coulomb's Law of electric force between static charges to find GSO solutions.

Another field of physics that has become popular in developing new heuristics is gravitational kinematics, the study of the motion of masses in a gravitational field. Examples include APO, CFO, and GSA which invoke modified forms of Newton's Universal Law of Gravitation. These algorithms create a group or swarm of "probes," "particles," or "individuals", each of which has an associated "mass," that move through the GSO problem's landscape on paths governed by the force of gravity as defined on the problem's metaphorical DS. For example, CFO's force law resembles Newton's. Its gravitational force is proportional to the product of two particles' masses divided by the separation distance raised to the second power (but the algorithm designer is free to change that exponent). An inverse proportionality is used in GSA, while direct proportionality is invoked in APO. Of course, the algorithm designer is free to modify these force laws in any desired manner because the DS is metaphorical in nature. For example, while the CFO and GSA force laws create only an attractive gravitational force, the APO force can be either attractive or repulsive. In the real Universe, of course, gravity is attractive and mass always positive definite. These algorithms also differ in another important respect. CFO is inherently deterministic, but is easily randomized, whereas GSA and APO are inherently stochastic and cannot easily be made deterministic.

APO was motivated by two considerations: (i) how well Physicomimetics or "Artificial Physics" (AP) [17] performed in controlling the behavior of multi-robot systems and (ii) the growing popularity and effectiveness of other physics-based algorithms. In the AP environment ("AP space"), each robot senses various environmental parameters, for example, other individuals' velocity and mass, and responds by moving through space along trajectories governed by algorithmically created "virtual" forces. The philosophy underlying AP is that any metaphorical system formulated using the laws of physics can be analyzed and controlled using all the tools and approaches common in state-of-the-art engineering and physics, whatever their nature, empirical or theoretical. The work described in this paper is an example; it uses a linear system theory to analyze APO.

AP's robots are considered to be physical "particles" or "individuals" that are characterized by their mass, vector velocity (speed and direction), and momentum, their motion being determined by a gravitational force law that analogizes the real force of gravity. Its physicomimetic framework creates the virtual forces that move individuals through DS searching for the objective function's extrema, a process that is analogous to real masses moving through our physical Universe. The general AP force law has the same functional form as Newton's and is defined as

$$F = G \frac{m_i m_j}{r^p}. \quad (1)$$

The force between particles i and j with masses m_i and m_j is F (magnitude), r is their separation distance, and the force is bounded from above as $F < F_{\max}$. Exponent p is a user-defined parameter, usually in the range $[-5, 5]$. While its value is 2 in our Universe, in AP space the user is free to assign an entirely different force variation (in APO, for example, $p = -1$ so that the force actually is proportional to r instead of varying inversely to some power). In most cases the “gravitational constant” G is set at initialization, but, of course, its value is determined by the algorithm designer and in some implementations it is variable. The physicomimetics AP framework has been used successfully to control distributed robot swarms and to perform a variety of tasks such as robot formation [18], obstacle avoidance [19], and coverage [20, 21]. AP also is effective in GSO as demonstrated by APO [22].

APO solves GSO problems, and its structure resembles that of many other GSO algorithms. Like many physics-based heuristics, each feasible GSO solution in APO is a metaphorical “particle” or “individual” characterized by the physical attributes of mass and motion (vector velocity and position in DS). A random sample of feasible solutions comprises APO’s initial population or “swarm.” The “mass” of a particle in APO-space is a *user-defined* function of the value of the objective function (“fitness”) being optimized, not necessarily the fitness itself. APO’s virtual force law preferentially drives individuals toward other particles with larger masses with the result that the swarm generally migrates toward regions in DS, where the fitnesses are better. This process constitutes APO’s search mechanism through the GSO problem’s landscape.

There are two important user-defined properties in a metaphorical AP space, namely (i) the gravitational force law, and (ii) the definition of “mass,” both of which are key elements in APO’s performance. The balance between “exploitation” and “exploration,” that is, local versus global search, is determined by how well the user-defined force law drives particles to efficiently search DS. Specifying different values in APO for exponent p in Eq. (1) results in quite different search mechanisms.

The definition of “mass” in APO-space is another critical factor in the algorithm’s performance. The virtual forces created in APO tend to move individuals toward others with larger mass (better fitness) and away from those with less mass (worse fitness). If, for example, APO is used to solve a GSO maximization problem its user-defined particle “mass” is some mathematical function whose value increases with *increasing* objective function value (fitness), whereas just the opposite occurs in a minimization problem, that is, APO’s particle mass then increases with *decreasing* fitness. In both cases, the use of a linear force law increases the gravitational force of attraction proportionately with the product of the masses as shown in Eq. (1).

Of course, the algorithm designer can choose any desired function to define “mass” in APO-space, and some will be better than others for certain GSO problems or perhaps even for classes of problems, for example, unimodal versus multimodal, low versus high dimensionality, and so on. Published work provides some guidance by suggesting basic requirements and by proposing specific algorithm design methodologies [23, 24]. Often, the curvilinear nature of the proposed mass functions is used to group candidate functions into various categories, representative examples being

linear, convex, or concave. Numerical experiments with APO show that concave functions generally outperform the other types.

APO's gravitational force may be *attractive* or *repulsive* which uniquely, and significantly, distinguishes it from other gravitational algorithms. Each particle in the swarm exerts an attractive force on all other particles with less mass (lower fitness) while repelling any particle whose mass is greater. The “best” individual (particle with the greatest mass in the swarm) attracts all others, but itself is neither attracted nor repelled by the others. Compared to other force laws, APO's linear attractive–repulsive law results in more efficient searches in the regions of DS with better fitnesses.

The unique attractive–repulsive force law is illustrated in Fig. 1, where it is applied to three particles in a two-dimensional (2-D) DS. The individuals labelled i , j , and l , respectively, have objective function values $f(X_i)$, $f(X_j)$, $f(X_l)$ with corresponding masses m_i , m_j , m_l . When APO performs minimization, fitnesses that are related as $f(X_l) < f(X_i) < f(X_j)$ creates masses whose relationship is as $m_l > m_i > m_j$. The larger circles in Fig. 1 correspond to greater mass, that is, “bigger” individuals that produce greater attraction. APO's attractive–repulsive force law results in individual i being *attracted* by individual l but *repelled* by individual j , thus exerting both an attractive force F_{il} and a repulsive force F_{ij} on individual i . In metaphorical APO-space, as in the real Universe, the motion of particle i is determined by the *total* applied force, F_i , which is the *vector sum* of forces F_{il} and F_{ij} . F_i is updated step by step throughout the APO run, and it determines particle i 's velocity at each step, hence its trajectory through DS.

APO has been effectively applied to many GSO problems, among them multi-objective optimization [25], constrained optimization [26, 27], and swarm robot search [28, 29]. But, like most if not all other EAs, APO can exhibit premature convergence under certain circumstances, in particular on complex high-dimensionality problems. However, techniques are available to address this behavior. For example, the earliest versions of APO retained only the best particle's then current position in DS, thereby ignoring its search history even though that information likely would be useful in improving the algorithm's exploration. The algorithm consequently was modified to include fitness history in an extended version, EAPO [30], which has

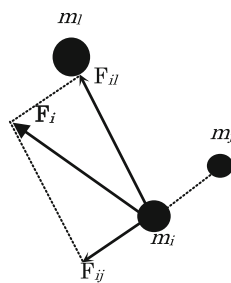


Fig. 1 Typical 2D APO decision space

been proven to converge using linear system theory [31]. An improved vector model of APO is described in [32], and [33] extends it to include a multidimensional search strategy that improves local exploitation. Yet another extension combines dissipative structure theory and population diversity to mitigate stagnation in APO's swarm evolution [34]. Further enhancements include introducing two selection schemes for v_{\max} (constant and adaptive) that were investigated using a group of high-dimensionality problems [35]. Simulation results show that adaptive v_{\max} generally yields better results.

Invoking the philosophy of physicomimetics, APO has been proven to converge using the theory of linear systems that are widely employed in various engineering disciplines. APO was modeled as a discrete-time linear system in which each particle's position is treated as a stochastic vector. Each possible APO algorithm was uniquely characterized by a nonnegative 3-tuple of run parameters $\{m_i, w, G\}$, and its convergence guaranteed when these parameters satisfy explicitly developed conditions. Parameter selection guidelines also were developed [36]. Of particular importance is the gravitational constant G because it has a significant impact on APO's convergence. Two strategies for specifying G were studied and reported, a constant value and an adaptive one [37]. A series of numerical experiments with recognized benchmarks showed that the algorithm with adaptive G outperforms a constant G implementation. Perhaps most importantly, the work demonstrating convergence shows that APO is a controllable and observable second-order dynamic system just as one would expect in the physicomimetics framework.

This paper recasts the APO architecture using another technique in the linear system toolbox, the Z -transform. Backward and forward proportional derivative (PD) controllers are introduced that improve an APO particle's ability to utilize historical data to anticipate and respond quickly to status changes. These characteristics result in improving APO's exploration, that is, its global search capability. Section 2 describes the second-order dynamical system framework for APO. Section 3 describes a version of APO that includes a *backward* PD controller (algorithm APO-PD1) and analyzes its convergence properties. Section 4 describes a parallel development of APO with a *forward* PD controller (algorithm APO-PD2). Section 5 compares the performance of these three algorithms using several recognized benchmark functions, and Sect. 6 presents conclusions and suggestions for future work.

2 The APO Algorithm

The APO algorithm addresses the GSO problem of locating the global minima of an objective function $f(x)$ defined on a bounded hyperspace; that is, determine

$$\min\{f(X) : X \in \Omega \subset R^n\}, f : \Omega \subset R^n \rightarrow R \quad (2)$$

wherein $\Omega := \{X | x_k^{\min} \leq x_k \leq x_k^{\max}, k = 1, \dots, n\}$ is the (bounded) region of feasible (allowable) solutions (DS), in which

- n problem dimensionality.
- x_k^{\max} upper bound for each DS dimension.
- x_k^{\min} lower bound for each dimension.
- $f(X)$ pointer to the function being minimized

The problem's "landscape" (topology over DS) is defined as $L = \Omega \cup f(X), X \in \Omega$.

2.1 The APO Framework

APO comprises three main procedures: (a) *Initialization*, (b) *Force Calculation*, and (c) *Motion* whose algorithmic framework (pseudocode) appears in Fig. 2. Throughout the remainder of this paper, the following terms are used interchangeably: "step", "time step", "iteration", and "generation", "individual", and "particle", and "swarm" and "population."

Initialization creates a randomly selected swarm of particles in the n -dimensional DS (upper case N_{pop} is the number of individuals). The individuals' velocities usually are initialized to zero (of course, nonzero values also can be used at the algorithm designer's discretion). The function pointer $f(X)$ is used to calculate each individual's fitness (objective function value). The location in DS (position vector) of the particle with the greatest fitness at step t is denoted X_{best} (global best position at that iteration).

Fig. 2 APO algorithmic framework

```

Begin
Initialize population: both position  $x$  and velocity  $v$ ;
Set parameters,  $N_{pop}$ ,  $n$ ,  $w$  and  $G$ , etc.;
Iteration=1;
While (termination criterion is not met)
Do
Evaluate all individuals using corresponding fitness
function;
Update the global best position  $X_{best}$ ;
Calculate the mass using equation(3);
Calculate the component force using equation(4);
Calculate the total force using equation(5);
Update the velocity  $v$  using equation(6);
Update the position  $x$  using equation(7);
Iteration= Iteration+1;
End Do (until termination criterion is met)
End

```

In *Force Calculation*, the APO force law is used to calculate the *total* gravitational force exerted on each particle based on the “masses” of all particles and the distances between them. The APO force law extends the physicomimetics force law of Eq. (1). Of course, before applying the force law “mass” in APO space must be defined. Mass is a *user-defined* function of the value of the objective function to be optimized. In minimization $m_i = g(f(X_i))$, where $m_i \in (0, 1]$ and the function g is greater than or equal to zero, bounded, and monotonically decreasing. The mass is normalized to the interval $(0, 1]$ as a matter of convenience. There exists a plethora of functions meeting these requirements, some undoubtedly better than others for specific GSO problems or perhaps classes of problems. The essential requirement is that the best individual’s mass has the *largest* value, that is, $m_{\text{best}} = 1$, and all other particles with worse (lower) fitnesses have *smaller* mass values. The following mass functions provide typical examples: $g_1(x) = e^{-x}$, $g_2(x) = \arctan(x)$, $g_3(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$, $x \in [-\infty, +\infty]$ in which $g_k(x) \in [a, b]$ is mapped onto the interval $(0, 1]$. Of course, which specific mass function is chosen does depend on whether maximization or minimization is the goal. It must be monotonically increasing or decreasing, respectively, for maximization or minimization thereby guaranteeing greater mass with better fitness and consequently a greater attractive force.

Equation (3) provides an example of a suitable minimization mass function that was successfully used in previous APO implementations and which is adopted in this formulation:

$$m_i = e^{\frac{f(X_{\text{best}}) - f(X_i)}{f(X_{\text{worst}}) - f(X_{\text{best}})}}, \quad \forall i \quad (3)$$

in which $f(X_{\text{best}})$ is the value of the objective function at the position of individual “best”, where $\text{best} = \arg\{\min f(X_i), i \in S\}$, and $f(X_{\text{worst}})$ is the objective function’s value at the position of individual “worst”, where $\text{worst} = \arg\{\max f(X_i), i \in S\}$, and S is the set of particle indices $\{1, \dots, N_{\text{pop}}\}$. Each individual’s mass varies from step by step as APO evolves, and the order of the differences in Eq. (3) insures that the exponent is in the interval $[-1, 0]$ as desired.

APO’s next step is computing *component by component* the vector forces exerted on each individual by all other individuals using APO’s unique proportional force law. Hence,

$$F_{ij,k} = \begin{cases} Gm_i m_j (x_{j,k} - x_{i,k}) & \text{if } f(X_j) < f(X_i) \\ -Gm_i m_j (x_{j,k} - x_{i,k}) & \text{if } f(X_j) \geq f(X_i) \end{cases}, \quad \forall i \neq j \text{ and } i \neq \text{best} \quad (4)$$

where $F_{ij,k}$ is the k th component of force exerted on an individual i by individual j , and $x_{i,k}$ and $x_{j,k}$ are the k th-dimension coordinates of particles i and j , respectively. If $f(X_j) < f(X_i)$ Eq. (4) shows that X_j attracts X_i because force $F_{ij,k}$ is attractive. However, if $f(X_j) \geq f(X_i)$, then X_j repels X_i because force $F_{ij,k}$ is repulsive. Since the forces are vector quantities, the k th *component* of the total force $F_{i,k}$ exerted on an individual i by all other particles is computed by summing over all other particles, that is,

$$F_{i,k} = \sum_{j=1}^{N_{\text{pop}}} F_{ij,k} \quad \forall i \neq \text{best} \quad (5)$$

The *total* force on each particle is the vector sum of all the forces created by every other particle. Note that under Eq. (4)'s force law each particle neither attracts nor repels itself (the total *self-exerted* force is zero). Equation (5), therefore, can *include* particle *i*'s self-exerted force because the addend is zero. Note, too, that particle *best* is *excluded* because it is neither attracted nor repelled by other individuals (its position is fixed). Excluding *best* is equivalent to setting the total force exerted on it to zero.

APO's final procedure is *Motion*, which computes the movement of the swarm's particles through DS. The total force is used to calculate each individual's "velocity", which is used to update the particle's position. This calculation is made for each particle in the swarm. Equations (6) and (7), respectively, are used to update the velocity and coordinates of the individual *i* at the next iteration at a time *t* + 1.

$$v_{i,k}(t+1) = wv_{i,k}(t) + \alpha \frac{F_{i,k}}{m_i}, \quad \forall i \neq \text{best} \quad (6)$$

$$x_{i,k}(t+1) = x_{i,k}(t) + v_{i,k}(t+1), \quad \forall i \neq \text{best} \quad (7)$$

where $v_{i,k}(t)$ and $x_{i,k}(t)$, respectively, are the *k*th components of particle *i*'s velocity and coordinates at the previous iteration (generation) *t*. The quantity α is a random variable (RV) uniformly distributed on [0,1]. The *inertia weight* $0 \leq w < 1$ is a user-specified parameter that determines how easily the previous velocity can be changed. Larger values result in greater velocity changes. Each particle's motion through DS is restricted so that it remains in the domain of feasible solutions $x_{i,k} \in [x_k^{\min}, x_k^{\max}]$. Its velocity is similarly constrained $v_{i,k} \in [v_k^{\min}, v_k^{\max}]$. Importantly, under the scheme in Eqs. (6) and (7), the best individual's position is fixed. It does not move away from its current position nor does its velocity change.

Once the positions of all particles have been updated as described, the corresponding objective function fitnesses are updated at each individual's new location. A new best individual is determined by the new fitness values with its position vector replacing X_{best} from the previous generation APO's processes of *Force Calculation* and *Motion* are repeated until some termination criterion is met, a variety of which are typically used. Commonly employed criteria are a specified maximum number of iterations or some number of successive iterations with no substantial change in particle *best*'s position or its corresponding mass.

2.2 Modeling APO as a Second-Order Dynamical System

APO can be analyzed by considering it to be a second-order dynamical system. In linear system theory, a second-order system is described by the following differential

equation: $\frac{d^2y}{dt^2} + 2\zeta\omega_n \frac{dy}{dt} + \omega_n^2 y = k_{dc}\omega_n^2 u(t)$ in which $k_{dc}\omega_n^2 u(t)$ is the *forcing function*, k_{dc} the *DC gain*, ω_n the *natural frequency*, ζ the *damping ratio*, and t the continuous *time variable*. The time domain system may be Laplace transformed into the frequency domain as follows: $Y(s) = G(s)X(s)$, where $Y(s)$ and $X(s)$, respectively, are the system output and input, and $L\{\cdot\}$ is the Laplace transform with (complex) frequency variable s . The corresponding transfer function is $G(s) = \frac{k_{dc}\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$. For discrete-time systems, the Laplace transform is replaced by the Z -transform. The system's behavior as $t \rightarrow \infty$ may be obtained by applying the Final Value Theorem (FVT). If $\lim_{t \rightarrow \infty} f(t)$ exists then the system's time domain behavior as $t \rightarrow \infty$ may be computed in the frequency domain as $\lim_{t \rightarrow \infty} f(t) = \lim_{s \rightarrow 0} (s \cdot L\{f(t)\})$ with an analogous result for the discrete-time Z -transform case.

The first step in applying linear system theory to APO is rewriting the velocity update Eq. (6) by substituting Eq. (5) to obtain

$$v_{i,k}(t+1) = wv_{i,k}(t) + \alpha \sum_{j=1}^{N_{\text{pop}}} F_{ij,k}/m_i, \quad \forall i \quad (8)$$

Define $N_i = \{j | f(X_j) \leq f(X_i), \forall j \in S\}$, in which, as before, S is the set of all individuals, N_i is the subset of all particles with fitnesses better than individual i 's fitness, and $M_i = \{j | f(X_j) > f(X_i), \forall j \in S\}$, where M_i is the subset of all particles whose fitnesses are worse than individual i 's. It is evident that $S = N_i \cup M_i$.

With these definitions, APO's velocity update Eq. (8) becomes

$$\begin{aligned} V_{i,k}(t+1) &= wV_{i,k}(t) + \sum_{\substack{j=1 \\ j \neq i}}^{N_{\text{pop}}} \alpha_j F_{ij,k}/m_i \\ &= wV_{i,k}(t) - \sum_{j \in N_i} \alpha_j Gm_j (X_{i,k}(t) - X_{j,k}(t)) \\ &\quad + \sum_{j \in M_i} \alpha_j Gm_j (X_{i,k}(t) - X_{j,k}(t)) \\ &= wV_{i,k}(t) + \left(\sum_{j \in M_i} \alpha_j Gm_j - \sum_{j \in N_i} \alpha_j Gm_j \right) X_{i,k}(t) \\ &\quad + \sum_{j \in N_i} \alpha_j Gm_j X_{j,k}(t) - \sum_{j \in M_i} \alpha_j Gm_j X_{j,k}(t) \end{aligned} \quad (9)$$

And, with the following definitions

$$G_{N_i} = \sum_{j \in N_i} \alpha_j Gm_j \quad (10)$$

$$G_{M_i} = \sum_{j \in M_i} \alpha_j Gm_j \quad (11)$$

$$G_{NM_i} = G_{N_i} - G_{M_i} \quad (12)$$

$$Q_{i,k} = \frac{1}{G_{NM_i}} \left(\sum_{j \in N_i} \alpha_j G m_j X_{j,k}(t) - \sum_{j \in M_i} \alpha_j G m_j X_{j,k}(t) \right) \quad (13)$$

Eq. (9) simplifies to

$$V_{i,k}(t+1) = w V_{i,k}(t) + G_{NM_i} (Q_{i,k} - X_{i,k}) \quad (14)$$

It is apparent from Eq. (14) that $Q_{i,k}$ represents the *swarm-weighted* position relative to particle i which can be used to compute the *total* force exerted on particle i by all other individuals in the swarm.

Substituting Eq. (14) into (7) yields the following relation for the position vector

$$X_{i,k}(t+1) = (w+1)X_{i,k}(t) - wX_{i,k}(t-1) + G_{NM_i}(Q_{i,k}(t) - X_{i,k}(t)) \quad (15)$$

Z-transforming Eq. (15) yields

$$X_{i,k}(z) = \frac{G_{NM_i} z}{z^2 - (w+1)z + w} (Q_{i,k}(z) - X_{i,k}(z)) \quad (16)$$

Defining

$$H(z) = \frac{G_{NM_i} z}{z^2 - (w+1)z + w} \quad (17)$$

Eq. (16) simplifies to

$$X_{i,k}(z) = H(z)(Q_{i,k}(z) - X_{i,k}(z)) \quad (18)$$

This last step completes APO's characterization as a linear system, because Eq. (18) defines the second-order linear dynamical system shown diagrammatically in Fig. 3.

APO can be thought of as a second-order dynamical system with time-varying input $Q_{i,k}$ and output $X_{i,k}(z)$. This system has a unique equilibrium point that exists only for particle *best*. It occurs when $X_{j,k}^{j \in M_i} = X_{j,k}^{j \in N_i} = Q_{i,k}$. As the swarm converges all particles except the *best* gradually approach $Q_{i,k}$, but equilibrium cannot be reached until an individual actually becomes *best*. Convergence alone, however, does not assure discovering a global optimum because it is possible for the swarm

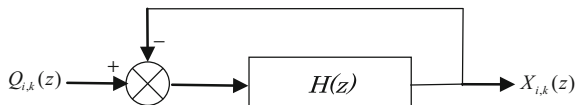


Fig. 3 APO system architecture

to converge prematurely on a suboptimal solution. Such local trapping is common across all optimization algorithms, and APO is no exception. It, therefore, is advantageous if particle *best* can influence other individuals to explore different solutions by having them move toward *best* while exploring for still better solutions along the way. If the probability of locating the global best position as its equilibrium point is high, then APO should exhibit good global convergence while avoiding premature convergence. This objective is achieved by introducing into APO a proportional derivative controller (PDC), because a PDC is capable of tracking and responding quickly to changes in the system's inputs.

3 APO with Backward PD Controller

In this section, a backward PDC is introduced into APO, and the resulting algorithm is named APO-PD1.

3.1 APO-PD1 Model

APO-PD1's linear system architecture appears in Fig. 4. The controller $C(z)$ creates an error signal that is used to alter the system's behavior so as to minimize the error. In this implementation, the error is the difference between a particle's predicted position and the swarm-weighted position which is used to modify the particle's velocity through the optimization problem's landscape.

The PDC is $C(z)$ defined by the following Z-transform whose "controller coefficient" or "control gain" is K_p and whose "derivative gain" is T_D :

$$C(z) = K_p \left(1 + T_D \frac{z-1}{z} \right) \quad (19)$$

This system's output, $X_{i,k}(z)$, is given by

$$X_{i,k}(z) = [Q_{i,k}(z) - X_{i,k}(z) \cdot C(z)] \cdot H(z) \quad (20)$$

which can be written as

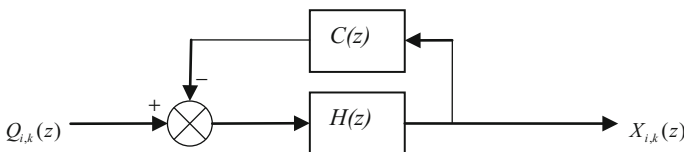


Fig. 4 APO-PD1 system architecture

$$X_{i,k}(z) = \frac{H(z)}{1 + C(z) \cdot H(z)} Q_{i,k}(z) \quad (21)$$

Substituting Eq. (19) into (21) yields

$$\begin{aligned} X_{i,k}(z) &= \frac{G_{NM_i} z}{z^2 - (w+1)z + w + G_{NM_i} K_p (z + T_D z - T_D)} Q_{i,k}(z) \\ &= \frac{G_{NM_i} z}{z^2 + [G_{NM_i} K_p (1 + T_D) - (w+1)]z + w - G_{NM_i} K_p T_D} Q_{i,k}(z) \end{aligned} \quad (22)$$

APO's time-domain behavior is obtained by inverting Z -transformed Eq. (22), which results in the following expression for the swarm particle's positions:

$$\begin{aligned} X_{i,k}(t+1) &= (w+1)X_{i,k}(t) - wX_{i,k}(t-1) + G_{NM_i} [Q_{i,k}(t) \\ &\quad - K_p(1 + T_D)X_{i,k}(t) + K_p T_D X_{i,k}(t-1)] \end{aligned} \quad (23)$$

The associated velocity update equation may be written as follows by noting that $V_{i,k}(t+1) = X_{i,k}(t+1) - X_{i,k}(t)$ and $V_{i,k}(t) = X_{i,k}(t) - X_{i,k}(t-1)$:

$$\begin{aligned} V_{i,k}(t+1) &= X_{i,k}(t+1) - X_{i,k}(t) \\ &= (w+1)X_{i,k}(t) - X_{i,k}(t) - wX_{i,k}(t-1) + G_{NM_i} [Q_{i,k}(t) \\ &\quad - K_p X_{i,k}(t) - K_p T_D (X_{i,k}(t) - X_{i,k}(t-1))] \\ &= w(X_{i,k}(t) - X_{i,k}(t-1)) + G_{NM_i} [Q_{i,k}(t) - K_p X_{i,k}(t) - K_p T_D \Delta X_{i,k}(t)] \\ &= wV_{i,k}(t) + G_{NM_i} [Q_{i,k}(t) - K_p (X_{i,k}(t) + T_D \Delta X_{i,k}(t))] \end{aligned} \quad (24)$$

or, equivalently,

$$V_{i,k}(t+1) = wV_{i,k}(t) + G_{NM_i} [Q_{i,k}(t) - K_p (X_{i,k}(t) + T_D V_{i,k}(t))] \quad (25)$$

Equation (25) is the final form of the velocity update equation for algorithm APO-PD1.

The term $X_{i,k}(t) + T_D V_{i,k}(t)$ represents each particle's "predicted" position based on its trajectory history, and is defined as the new quantity $X'_{i,k}(t) = X_{i,k}(t) + T_D V_{i,k}(t)$. The derivative gain $T_D \in [0, 1]$ can be thought of as a parameter that controls this prediction step. The backward PDC allows each individual in the swarm to "predict" its future position based on its previous history. The particle then adjusts its velocity through DS based on an "error signal" that is computed as the distance between the predicted position and the swarm-weighted position. Numerical testing shows that this approach improves APO's performance.

3.2 Convergence Analysis of APO-PD1

Convergence of the APO-PD1 algorithm is analyzed in this section. The equation describing the system architecture of Fig. 4 may be written as

$$\begin{aligned} \frac{X_{i,k}(z)}{Q_{i,k}(z)} &= \frac{H(z)}{1 + C(z) \cdot H(z)} \\ &= \frac{G_{NM_i} z}{z^2 - (w + 1 - K_P G_{NM_i} - K_P G_{NM_i} T_D)z + w - K_P G_{NM_i} T_D} \end{aligned} \quad (26)$$

where

$$\begin{aligned} X_{i,k}(z) &= \frac{G_{NM_i} z}{z^2 - (w + 1 - K_P G_{NM_i} - K_P G_{NM_i} T_D)z + w - K_P G_{NM_i} T_D} Q_{i,k}(z) \end{aligned} \quad (27)$$

Without loss of generality, and as a matter of convenience, the controller coefficient may be set equal to unity ($K_P = 1$ hereafter).

Applying the FVT discussed in Sect. 2.2,

$$\begin{aligned} \lim_{t \rightarrow \infty} x_{i,k}(t) &= \lim_{z \rightarrow 1} (z - 1)x_{i,k}(z) \\ &= \lim_{z \rightarrow 1} (z - 1) \left[\frac{G_{NM_i} z}{z^2 - (w + 1 - K_P G_{NM_i} - K_P G_{NM_i} T_D)z + w - K_P G_{NM_i} T_D} \cdot \frac{z}{z - 1} \cdot Q_{i,k} \right] \\ &= Q_{i,k} \end{aligned} \quad (28)$$

which is equivalent to

$$\lim_{t \rightarrow \infty} x_{i,k}(t) = Q_{i,k} = \frac{1}{G_{NM_i}} \left(\sum_{j \in N_i} \alpha_j G m_j X_{j,k}(t) - \sum_{j \in M_i} \alpha_j G m_j X_{j,k}(t) \right) \quad (29)$$

Equation (29) may be written as

$$\begin{aligned} G_{NM_i} \lim_{t \rightarrow \infty} x_{i,k}(t) - \left(\sum_{j \in N_i} \alpha_j G m_j X_{j,k}(t) - \sum_{j \in M_i} \alpha_j G m_j X_{j,k}(t) \right) &= 0 \\ \Rightarrow \left(\sum_{j \in N_i} \alpha_j G m_j - \sum_{j \in M_i} \alpha_j G m_j \right) \lim_{t \rightarrow \infty} x_{i,k}(t) - \left(\sum_{j \in N_i} \alpha_j G m_j X_{j,k}(t) - \sum_{j \in M_i} \alpha_j G m_j X_{j,k}(t) \right) &= 0 \\ \Rightarrow \sum_{j \in N_i} \alpha_j G m_j \left(\lim_{t \rightarrow \infty} x_{i,k}(t) - X_{j,k}(t) \right) - \sum_{j \in M_i} \alpha_j G m_j \left(\lim_{t \rightarrow \infty} x_{i,k}(t) - X_{j,k}(t) \right) &= 0 \end{aligned} \quad (30)$$

and because $\sum_{j \in N_i} \alpha_j G m_j$ and $\sum_{j \in M_i} \alpha_j G m_j$ in Eq. (30) are nonzero RV's, algorithm APO-PD1 converges to X_{best} ($X_{\text{best}} \in N_i$) if and only if $\lim_{t \rightarrow \infty} x_{i,k}(t) = X_{j,k}^{j \in N_i} = X_{j,k}^{j \in M_i}$. This analysis provides the precise conditions that guaranty algorithm APO-PD1's convergence. It realizes the physicomimetics concept that the tools and tech-

niques of modern physics and engineering, in this case, linear system theory, can be effectively applied to GSO problems.

3.3 APO-PD1 Procedure

Pseudocode for both algorithms APO-PD1 and APO-PD2 appears in Fig. 5, because they differ only in the position/velocity update in Step (3) as discussed below.

4 APO with Forward PD Controller

APO is extended in this section by introducing a forward PD controller. The new implementation is algorithm APO-PD2.

4.1 APO-PD2 Model

The APO/PDC architecture is shown diagrammatically in Fig. 6. In this case, the controller $C(z)$ creates a different error signal that includes the particle's and the swarm's trajectory history which, as before, is used to modify the particle's velocity through the optimization problem's landscape.

This system's output $X_{i,k}(z)$ is

$$X_{i,k}(z) = C(z)H(z)[Q_{i,k}(z) - X_{i,k}(z)] \tag{31}$$

- (1): Initialize each coordinate $x_{i,k} \in [x_{\min}, x_{\max}]$ and velocities $v_{i,k} \in [v_k^{\min}, v_k^{\max}]$ with random numbers.
- (1.1): calculate the fitness for each individual and select the best and worst, X_{best} and X_{worst} .
- (2): Compute the total force exerted on each individual.
 - (2.1): calculate the mass of each individual at time t , Eq. (3).
 - (2.2): calculate G_{swi} , Eq. (12).
 - (2.3): calculate $Q_{i,k}$, Eq. (13).
- (3): Update each particle's velocity and position vector using Eqs. (7) and either Eq. (25) [APO-PD1] or Eq. (35) [APO-PD2].
- (4): Compute each individual's fitness and update the global best and worst positions, X_{best} and X_{worst} .
- (5): If termination criteria are met, output the best solution; otherwise, go to (2).

Fig. 5 APO-PD1/APO-PD2 pseudocode

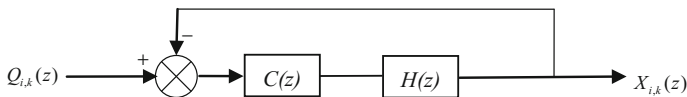


Fig. 6 APO-PD2 system architecture

Substituting Eqs. (17) and (19) into (31) yields

$$X_{i,k}(z) = K_P(1 + T_D \frac{z-1}{z}) \frac{G_{NM_i} z}{z^2 - (w+1)z + w} [Q_{i,k}(z) - X_{i,k}(z)] \quad (32)$$

Following the same procedure used for APO-PD1, Z-transforming Eq. (32) and inverting the result yields the following time domain expression for the swarm's particle positions

$$\begin{aligned} X_{i,k}(t+1) &= (w+1)X_{i,k}(t) - wX_{i,k}(t-1) \\ &\quad + G_{NM_i} K_P [(1 + T_D)(Q_{i,k}(t) - X_{i,k}(t)) - T_D(Q_{i,k}(t-1) - X_{i,k}(t-1))] \end{aligned} \quad (33)$$

Noting that $V_{i,k}(t+1) = X_{i,k}(t+1) - X_{i,k}(t)$, $V_{i,k}(t) = X_{i,k}(t) - X_{i,k}(t-1)$, $\Delta Q_{i,k}(t) = Q_{i,k}(t) - Q_{i,k}(t-1)$, and $\Delta X_{i,k}(t) = X_{i,k}(t) - X_{i,k}(t-1)$, APO-PD2's velocity update equation becomes

$$\begin{aligned} V_{i,k}(t+1) &= X_{i,k}(t+1) - X_{i,k}(t) \\ &= (w+1)X_{i,k}(t) - X_{i,k}(t) - wX_{i,k}(t-1) \\ &\quad + G_{NM_i} K_P [(1 + T_D)(Q_{i,k}(t) - X_{i,k}(t)) - T_D(Q_{i,k}(t-1) - X_{i,k}(t-1))] \\ &= w(X_{i,k}(t) - X_{i,k}(t-1)) + G_{NM_i} K_P [(Q_{i,k}(t) - X_{i,k}(t)) \\ &\quad + T_D(Q_{i,k}(t) - X_{i,k}(t)) - T_D(Q_{i,k}(t-1) - X_{i,k}(t-1))] \\ &= wV_{i,k}(t) + G_{NM_i} K_P [(Q_{i,k}(t) - X_{i,k}(t)) + T_D(Q_{i,k}(t) \\ &\quad - Q_{i,k}(t-1)) - T_D(X_{i,k}(t) - X_{i,k}(t-1))] \\ &= wV_{i,k}(t) + G_{NM_i} K_P [(Q_{i,k}(t) - X_{i,k}(t)) + T_D \Delta Q_{i,k}(t) - T_D \Delta X_{i,k}(t)] \\ &= wV_{i,k}(t) + G_{NM_i} K_P [(Q_{i,k}(t) + T_D \Delta Q_{i,k}(t)) - (X_{i,k}(t) + T_D \Delta X_{i,k}(t))] \end{aligned} \quad (34)$$

which may be simplified to

$$V_{i,k}(t+1) = wV_{i,k}(t) + G_{NM_i} K_P [(Q_{i,k}(t) + T_D \Delta Q_{i,k}(t)) - (X_{i,k}(t) + T_D \Delta X_{i,k}(t))] \quad (35)$$

Without loss of generality, as a matter of convenience, the time step can be set to unity ($\Delta t = 1$) with the results $\Delta Q_{i,k}(t) = V_{Q_{i,k}}(t) \cdot \Delta t = V_{Q_{i,k}}(t)$ and $\Delta X_{i,k}(t) = V_{i,k}(t) \cdot \Delta t = V_{i,k}(t)$. Equation (35) then becomes

$$V_{i,k}(t+1) = wV_{i,k}(t) + G_{NM_i} K_P [(Q_{i,k}(t) + T_D V_{Q_{i,k}}(t)) - (X_{i,k}(t) + T_D V_{i,k}(t))] \quad (36)$$

In (36) $V_{Q_{i,k}}(t) = \Delta Q_{i,k}(t)$ is swarm $Q_{i,k}$'s velocity at time t . The quantity $Q_{i,k}(t) + T_D V_{Q_{i,k}}(t)$ is its predicted future position. Following the analysis of Sect. 2.2, the swarm-weighted predicted position taking into account the swarm's directional

history is given by the term $Q'_{i,k}(t) = Q_{i,k}(t) + T_D V_{Q_{i,k}}(t)$. As before, the derivative gain $T_D \in [0, 1]$ can be thought of as a user-specified *prediction factor* that controls the prediction step.

Taking into account its directional history and total force it experiences, particle i 's predicted position is given by $X'_{i,k}(t) = X_{i,k}(t) + T_D V_{i,k}(t)$, which simplifies Eq. (36) to

$$V_{i,k}(t+1) = w V_{i,k}(t) + G_{NM_i} K_P [Q'_{i,k}(t) - X'_{i,k}(t)] \quad (37)$$

Because $Q'_{i,k}(t)$ includes a history increment in $Q_{i,k}$, Eq. (37) implies that swarm $Q'_{i,k}(t)$'s velocity will be *greater* than $Q_{i,k}$'s in converging on individual *best*. Stated differently, the APO-PD2 architecture *increases* the probability of converging on the global optimum because the problem's landscape is explored more efficiently.

The forward PDC provides APO's swarm with more information that can be used to improve exploration. Each particle predicts its future position based on its own trajectory history as well as on the swarm-weighted position history. These data then are used in updating the particle's velocity. The effect of this procedure is to discourage quick convergence on the current swarm-weighted position, thus allowing each individual to adjust its velocity while taking into account its distance from the swarm-weighted's predicted position (the *error signal* introduced above). This process is rapid and generally improves APO-PD2's exploration.

4.2 Convergence Analysis of APO-PD2

As was done with APO-PD1, this section develops a proof of convergence for algorithm APO-PD2. It parallels the previous development. APO-PD2's architecture in Fig. 3 is described by the following equation:

$$\begin{aligned} \frac{X_{i,k}(z)}{Q_{i,k}(z)} &= \frac{C(z) \cdot H(z)}{1 + C(z) \cdot H(z)} \\ &= \frac{K_P G_{NM_i} (z + T_D z - T_D)}{z^2 - (w + 1 - K_P G_{NM_i} - K_P G_{NM_i} T_D)z + w - K_P G_{NM_i} T_D} \end{aligned} \quad (38)$$

which may be written as

$$X_{i,k}(z) = \frac{K_P G_{NM_i} (z + T_D z - T_D)}{z^2 - (w + 1 - K_P G_{NM_i} - K_P G_{NM_i} T_D)z + w - K_P G_{NM_i} T_D} Q_{i,k}(z) \quad (39)$$

Again applying FVT (see Sect. 2.2),

$$\begin{aligned}
\lim_{t \rightarrow \infty} x_{i,k}(t) &= \lim_{z \rightarrow 1} (z-1)x_{i,k}(z) \\
&= \lim_{z \rightarrow 1} (z-1) \left[\frac{K_P G_{NM_i} (z + T_D z - T_D)}{z^2 - (w+1 - K_P G_{NM_i} - K_P G_{NM_i} T_D)z + w - K_P G_{NM_i} T_D} \cdot \frac{z}{z-1} \cdot Q_{i,k} \right] \\
&= Q_{i,k}
\end{aligned} \tag{40}$$

Thus,

$$\lim_{t \rightarrow \infty} x_{i,k}(t) = Q_{i,k} = \frac{1}{G_{NM_i}} \left(\sum_{j \in N_i} \alpha_j G_{M_j} X_{j,k}(t) - \sum_{j \in M_i} \alpha_j G_{M_j} X_{j,k}(t) \right) \tag{41}$$

Rewriting

$$\begin{aligned}
G_{NM_i} \lim_{t \rightarrow \infty} x_{i,k}(t) - \left(\sum_{j \in N_i} \alpha_j G_{M_j} X_{j,k}(t) - \sum_{j \in M_i} \alpha_j G_{M_j} X_{j,k}(t) \right) &= 0 \\
\Rightarrow \left(\sum_{j \in N_i} \alpha_j G_{M_j} - \sum_{j \in M_i} \alpha_j G_{M_j} \right) \lim_{t \rightarrow \infty} x_{i,k}(t) - \left(\sum_{j \in N_i} \alpha_j G_{M_j} X_{j,k}(t) - \sum_{j \in M_i} \alpha_j G_{M_j} X_{j,k}(t) \right) &= 0 \\
\Rightarrow \sum_{j \in N_i} \alpha_j G_{M_j} (\lim_{t \rightarrow \infty} x_{i,k}(t) - X_{j,k}(t)) - \sum_{j \in M_i} \alpha_j G_{M_j} (\lim_{t \rightarrow \infty} x_{i,k}(t) - X_{j,k}(t)) &= 0
\end{aligned} \tag{42}$$

As in the previous development, because $\sum_{j \in N_i} \alpha_j G_{M_j}$ and $\sum_{j \in M_i} \alpha_j G_{M_j}$ in Eq. (42) are nonzero RV's, *if and only if* $\lim_{t \rightarrow \infty} x_{i,k}(t) = X_{j,k}^{j \in M_i} = X_{j,k}^{j \in N_i}$, then algorithm APO-PD2 converges to X_{best} ($X_{best} \in N_i$). *QED.*

4.3 The Procedure of APO-PD2

Pseudocode for algorithm APO-PD2 is shown in Fig. 5.

5 Numerical Experiments

Five recognized GSO benchmarks were used to compare the performance of algorithms APO-PD1, APO-PD2, and APO (collectively APO*). Three of the test functions are multimodal with many local optima, viz., Ackley and the two Penalized Functions. Rosenbrock is multimodal but with a few local optima, and likewise Schwefel Problem 2.26. The Rosenbrock is characterized by a banana-shaped landscape surrounding its global optimum and sometimes is referred to as the ‘‘Rosenbrock Banana Function.’’ Ackley, on the other hand, has a narrow basin containing its global optimum. Schwefel’s landscape is ‘‘multi-funnel,’’ one whose complexity is characterized by deep local optima that are distant from the global optimum. The Schwefel Problem 2.26 is an especially robust benchmark because it is difficult to locate the global extremum if an algorithm converges prematurely, which may be the

case if a substantial portion of APO*'s swarm falls into one of the Schwefel's deep local optima.

The test functions are defined as follows:

- Ackley

$$f_2(x) = -20 \exp(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}) - \exp(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)) + 20 + e, \text{ where } |x_i| \leq 32.0 \text{ and } f_2(x^*) = f_2(0, 0, \dots, 0) = 0$$

- Rosenbrock

$$f_1(x) = \sum_{i=1}^{n-1} (100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2), \text{ where } |x_i| \leq 30.0 \text{ and } f_1(x^*) = f_1(1, 1, \dots, 1) = 0$$

- Penalized #1

$$f_3(x) = \frac{\pi}{n} \{10 \sin^2(\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1})] + (y_n - 1)^2\} + \sum_{i=1}^{n-1} u(x_i, 10, 100, 4),$$

$$\text{where } |x_i| \leq 50.0, u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, & \text{if } x_i > a \\ 0, & \text{if } -a \leq x_i \leq a \\ k(-x_i - a)^m, & \text{if } x_i < -a \end{cases}, y_i = 1 + \frac{1}{4}(x_i +$$

$$1), \text{ and } f_3(x^*) = f_3(1, 1, \dots, 1) = 0$$

- Penalized #2

$$f_4(x) = 0.1 \{ \sin^2(3\pi x_1) + \sum_{i=1}^{n-1} (x_i - 1)^2 [1 + \sin^2(3\pi x_{i+1})] + (x_n - 1)^2 [1 + \sin^2(2\pi x_n)] \} + \sum_{i=1}^{n-1} u(x_i, 5, 100, 4)$$

$$\text{where } |x_i| \leq 50.0 \text{ and } f_4(x^*) = f_4(1, 1, \dots, 1) = 0$$

- Schwefel Problem 2.26

$$f_5(x) = - \sum_{i=1}^n (x_i \sin(\sqrt{|x_i|})), \text{ where } |x_i| \leq 500.0 \text{ and } f_5(x^*) = f_5(420.9687, 420.9687, \dots, 420.9687) \approx -418.9829n$$

All APO* runs were made with the same empirically determined parameters as follows:

- (1) $w = 0.9 - \frac{t}{\text{MAXITER}} \times 0.5$ (inertia weight), wherein t is the step (iteration) number and MAXITER the maximum number of steps that automatically terminates a run.
- (2) in APO the gravitational constant is $G = 10$ for all test functions.
- (3) in APO-PD1 and APO-PD2, the gravitational constant is $G = 10$ for Rosenbrock, Penalized #1 and Penalized #2; $G = 0.008$ for Schwefel Problem 2.26; and $G = 5$ for Ackley.
- (4) in APO-PD1, $K_p = 0.7$, $T_D = 0.1$; and in APO-PD2, $K_p = 0.1$, $T_D = 0.9$.
- (5) for 30-dimensional problems, $n = 30$, the swarm population is set to $N_{\text{pop}} = 30$; for higher dimensionality problems, $n = 50, 100, 200, 300$, the population was set to $N_{\text{pop}} = 100$.
- (6) velocity threshold v_k^{\min} set to DS's lower bound.
- (7) v_k^{\max} set to DS's upper bound.
- (8) the maximum number of steps for run termination was related to the problem's dimensionality as $\text{MAXITER} = 50n$.

Table 1 Ackley function

Dimension	Algorithm	Mean	STD
30	APO	3.507931e+000	1.116182e+000
	APO-PD1	5.887218e-016	0.000000e+000
	APO-PD2	5.887218e-016	0.000000e+000
50	APO	1.086667e+000	3.981407e-001
	APO-PD1	5.887218e-016	0.000000e+000
	APO-PD2	5.887218e-016	0.000000e+000
100	APO	6.143348e-002	2.980632e-002
	APO-PD1	5.887218e-016	0.000000e+000
	APO-PD2	5.887218e-016	0.000000e+000
200	APO	1.884757e+000	7.572034e-003
	APO-PD1	5.887218e-016	0.000000e+000
	APO-PD2	5.887218e-016	0.000000e+000
300	APO	2.823220e+000	1.348221e-001
	APO-PD1	5.887218e-016	0.000000e+000
	APO-PD2	5.887218e-016	0.000000e+000

Thirty separate runs were made for each numerical experiment, and the following performance data recorded: (i) average best function value (*Mean*) and (ii) its standard deviation (*STD*) over the 30 runs. The average best fitness as a function of iteration is plotted in Figs. 7, 8, 9, 10, and 11 using the same 20 sample points for each plot. Performance data for each algorithm are summarized in Tables 1, 2, 3, 4, and 5 in which the best returned values appear in **bold** type..

It is apparent from Table 1 that for the Ackley APO’s solutions lie only in the global optimum’s vicinity, while in contrast both APO-PD1 and APO-PD2 in fact locate the known global optimum. Their performance cannot be distinguished based on accuracy or efficiency (measured by the number of function evaluations, FEs). From the fitness plots in Figs. 7, 8, 9, 10, and 11 it is apparent that APO-PD1 and APO-PD2 converge at a similar rate.

All APO* variants had trouble with Rosenbrock, each returning a local minimum at nearly the same position $(0, 0, 0, \dots, 0)^n$ instead of the actual minimum’s location [fitness of zero at $(1, 1, 1, \dots, 1)^n$]. APO-PD2 did perform somewhat better than APO and APO-PD1 for $n \leq 50$, while for $n \geq 100$ APO-PD1/2’s performance was similar and better than APO’s.

For the other test functions, all APO* implementations converge on local extrema in the vicinity of the known global minimum. For $n \leq 50$, APO returned a slightly better solution than APO-PD1/2 on Rosenbrock, Schwefel, and both Penalized Functions. However, APO-PD1’s *STD* was slightly better than APO’s for Penalized #2, which indicates that APO-PD1 exhibits better stability than APO. By contrast, for $n \geq 100$ APO-PD1’s solutions were of better quality than APO’s. The performance of APO and APO-PD1, therefore, is mixed. By contrast, the test data show that

Table 2 Rosenbrock function

Dimension	Algorithm	Mean	STD
30	APO	2.899009e+001	7.045900e−003
	APO-PD1	2.899993e+001	7.178198e−005
	APO-PD2	2.894036e+001	1.461165e−002
50	APO	4.899675e+001	1.252448e−003
	APO-PD1	4.899980e+001	1.955173e−004
	APO-PD2	4.893962e+001	1.517006e−002
100	APO	1.002528e+002	8.417579e−001
	APO-PD1	9.900000e+001	0.000000e+000
	APO-PD2	9.893687e+001	1.478572e−002
200	APO	3.585296e+003	8.376577e+002
	APO-PD1	1.989984e+002	5.889333e−004
	APO-PD2	1.989351e+002	1.359424e−002
300	APO	6.645564e+004	1.261125e+004
	APO-PD1	2.989997e+002	2.978742e−004
	APO-PD2	2.989396e+002	1.388029e−002

Table 3 Schwefel problem 2.26

Dimension	Algorithm	Mean	STD
30	APO	−6.180069e+003	2.041997e+002
	APO-PD1	−6.120724e+003	2.000111e+002
	APO-PD2	−6.274354e+003	2.030197e+002
50	APO	−5.647666e+003	1.197372e+002
	APO-PD1	−8.161097e+003	2.253472e+002
	APO-PD2	−8.850338e+003	2.676209e+002
100	APO	−8.722049e+003	1.299973e+002
	APO-PD1	−1.290723e+004	5.839660e+002
	APO-PD2	−1.772176e+004	5.715952e+002
200	APO	−1.162446e+004	2.080767e+002
	APO-PD1	−2.210786e+004	9.548621e+002
	APO-PD2	−3.678811e+004	1.820679e+001
300	APO	−1.342185e+004	1.987825e+002
	APO-PD1	−3.186174e+004	1.288484e+003
	APO-PD2	−5.484869e+004	1.787830e+001

Table 4 Penalized 1 function

Dimension	Algorithm	Mean	STD
30	APO	1.467692e+000	2.335548e-002
	APO-PD1	1.579828e+000	2.873102e-002
	APO-PD2	1.430841e+000	3.742797e-002
50	APO	1.353991e+000	1.409452e-002
	APO-PD1	1.134589e+007	1.115519e+007
	APO-PD2	1.257534e+000	1.841566e-002
100	APO	1.296102e+000	4.272975e-003
	APO-PD1	1.283348e+000	1.016297e-002
	APO-PD2	1.210165e+000	1.079314e-002
200	APO	1.356963e+000	2.030752e-002
	APO-PD1	1.232687e+000	4.236398e-003
	APO-PD2	1.194904e+000	4.887244e-003
300	APO	1.600399e+000	6.083825e-002
	APO-PD1	1.218252e+000	2.278210e-003
	APO-PD2	1.186042e+000	2.894729e-003

Table 5 Penalized 2 function

Dimension	Algorithm	Mean	STD
30	APO	2.900961e+000	5.744124e-003
	APO-PD1	2.964067e+000	6.876373e-003
	APO-PD2	2.908425e+000	1.121417e-003
50	APO	4.897998e+000	6.254901e-003
	APO-PD1	4.964973e+000	7.184687e-003
	APO-PD2	4.902154e+000	7.414638e-004
100	APO	9.963455e+000	8.905394e-003
	APO-PD1	9.945704e+000	7.829821e-003
	APO-PD2	9.897933e+000	2.890774e-003
200	APO	2.252701e+001	9.613815e-001
	APO-PD1	1.992891e+001	7.335306e-003
	APO-PD2	1.989362e+001	2.383790e-003
300	APO	1.439755e+004	3.916375e+003
	APO-PD1	2.992930e+001	7.248343e-003
	APO-PD2	2.988707e+001	4.707260e-003

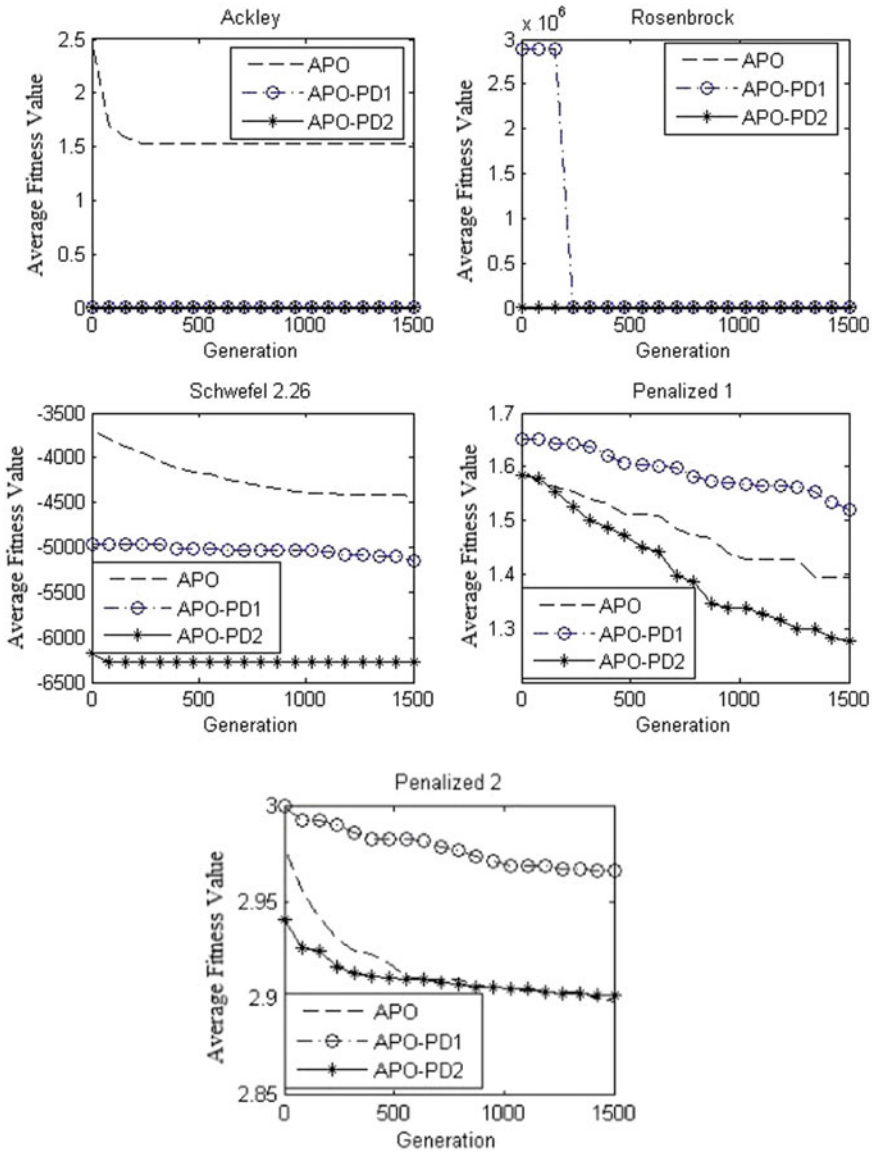


Fig. 7 Performance comparison among APO variants for $n = 30$ test functions

APO-PD2 outperforms both APO and APO-PD1 on all benchmarks over all dimensions, which likely reflects APO-PD2’s greater diversity that results in improved exploration.

On rate of convergence, the data show that APO converged much more slowly than the other variants on both $n = 30$ Penalized functions, on the $n = 50$ Rosenbrock, and

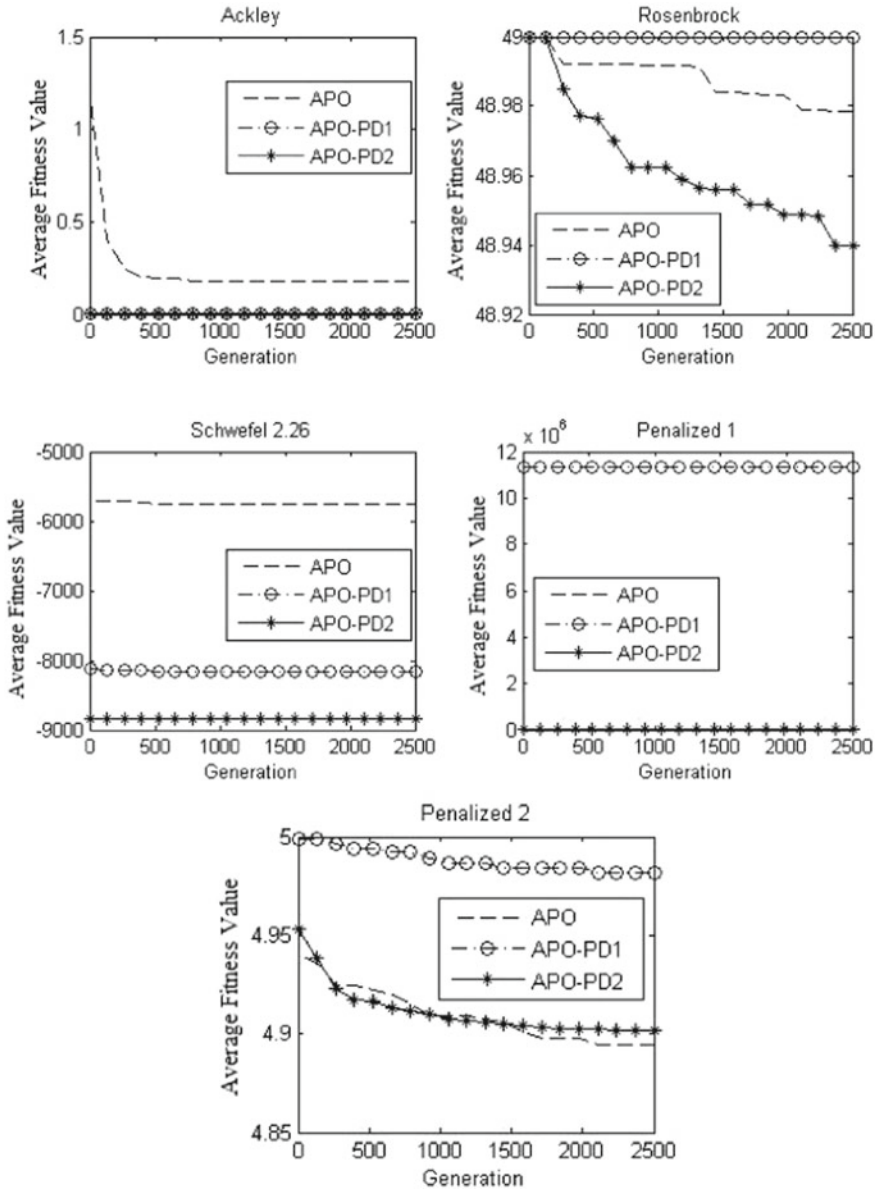


Fig. 8 Performance comparison among APO variants for $n = 50$ test functions

on the $n = 200$ Penalized #1. But its performance was better on the other benchmarks. APO-PD1 and APO-PD2 exhibit similarly rapid convergence.

The reasonable interpretation of these data is that generally APO-PD2 exhibits the best overall performance, while APO-PD1 performs better than APO, especially in

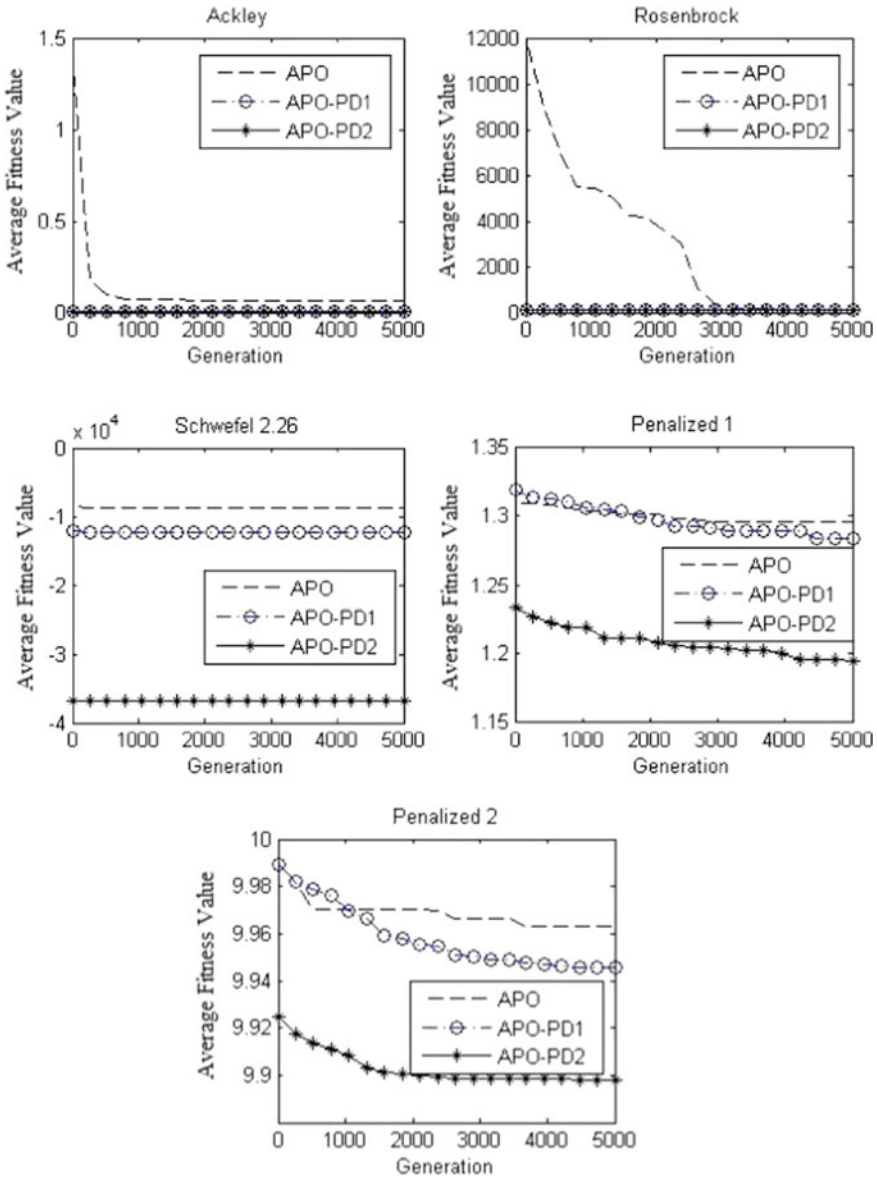


Fig. 9 Performance comparison among APO variants for $n = 100$ test functions

high dimensions. The physicomimetics approach of modeling APO as a second-order linear system has led to improved GSO algorithms and to proofs of the conditions under which these algorithms are guaranteed to converge.

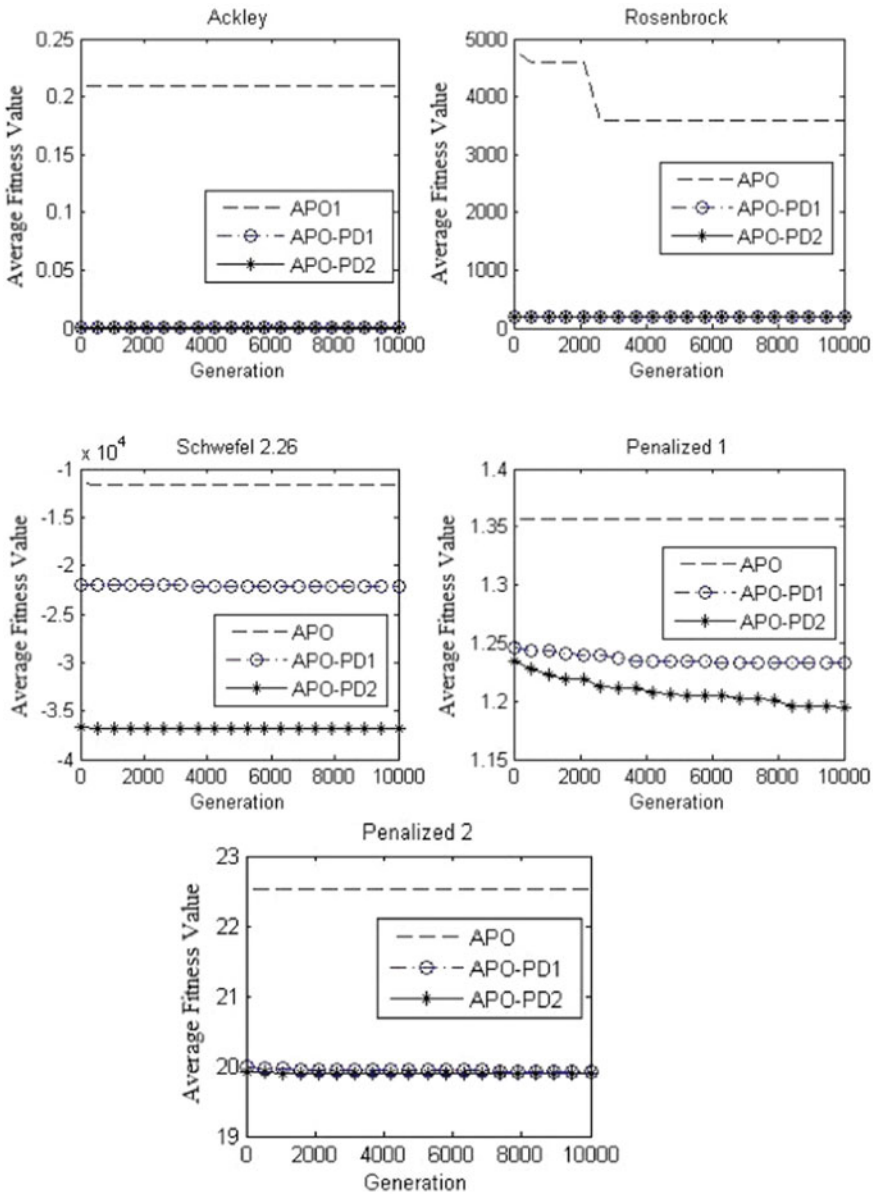


Fig. 10 Performance comparison among APO variants for $n = 200$ test functions

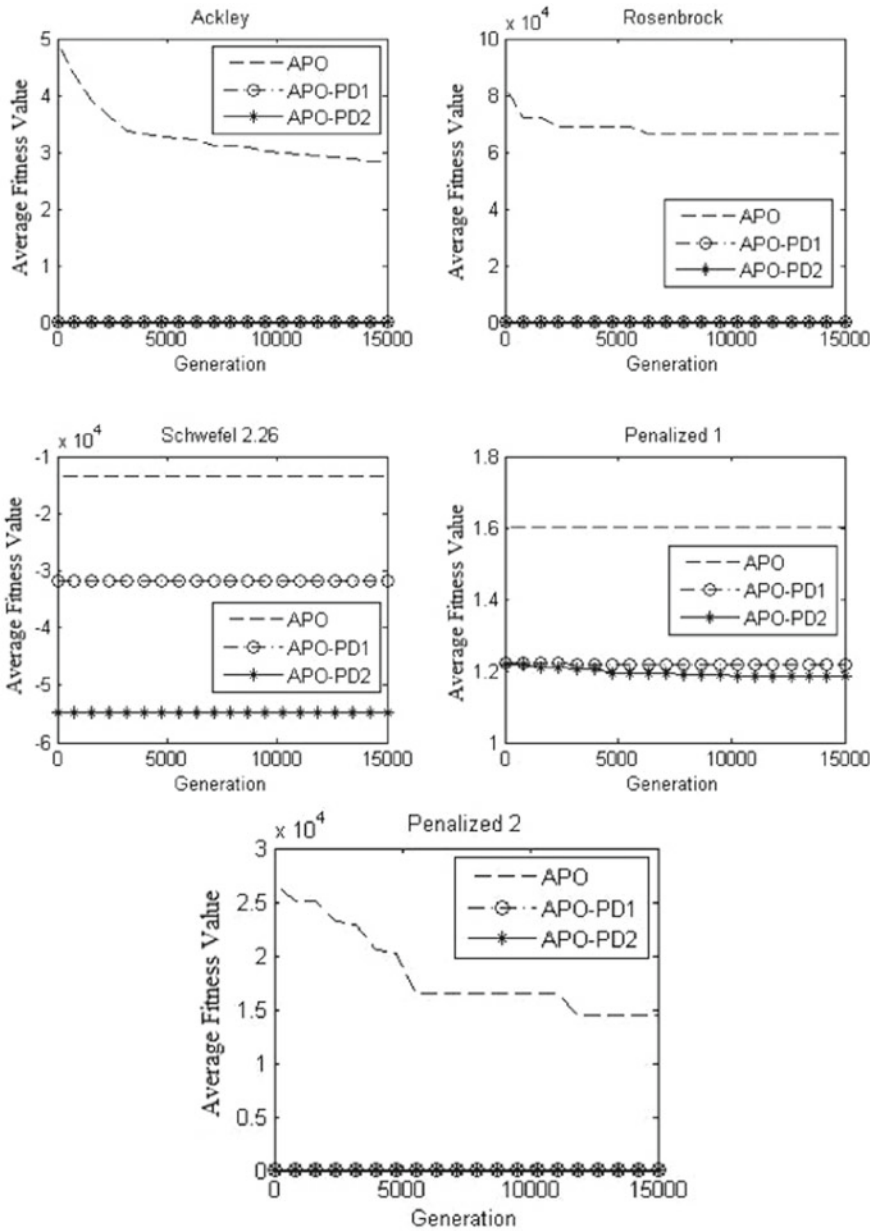


Fig. 11 Performance comparison among APO variants for $n = 300$ test functions

6 Conclusion and Future Work

This paper describes two new population-based, stochastic, swarm intelligent algorithms for multidimensional GSO: APO-PD1 and APO-PD2. Both algorithms are variants of the physicomimetics APO system created by introducing backward and forward PD controllers into the original APO architecture. APO's global performance is improved as a result. APO-PD1 utilizes a backward PDC to forecast each particle's future position in DS. APO-PD2 utilizes a forward PDC that predicts both the particle's future position as well as the future swarm-weighted position. This architecture results in a higher probability of converging on the global maximum, because it responds quickly to changes in an individual's fitness and location. Numerical experiments on recognized benchmark functions have shown that APO-PD2 exhibits faster convergence and better performance than both APO-PD1 and APO. These experiments also show that the prediction factor, T_D , is a particularly important parameter in addition to the gravitational constant G and the PDC coefficient K_p . It remains an open question as to how to best assign these parameter values. Future work will address this issue with a view toward further improving algorithm APO-PD2.

Acknowledgements This work was supported by the National Natural Science Foundation of China for Young Scientists under Grant Number 61403271 and by the Postdoctoral Scientific Research Starting Foundation of Taiyuan University of Science and Technology under Grant Number 20142022.

References

1. Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolution Programs. Springer Press, Berlin (1994)
2. Holland, J.H.: Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor (1975)
3. Bonabeau, E., Dorigo, M., Theraulaz, G.: Intelligence: From Natural to Artificial Intelligence. Oxford University Press, New York (1999)
4. Shah-Hosseini, H.: The intelligent water drops algorithm: a nature-inspired swarm-based optimization algorithm. *Int. J. Bio-Inspired Comput.* **1**(1/2), 71–79 (2009)
5. Eberhart, R., Kennedy, J.: New optimizer using particle swarm theory. In: Proceedings of the Sixth International Symposium on Micro Machine and Human Science, IEEE CS Press, Nagoya, Japan, pp. 39–43 (1995)
6. Xue, S.D., Zhang, J.H., Zeng, J.C.: Parallel asynchronous control strategy for target search with swarm robots. *Int. J. Bio-Inspired Comput.* **1**(3), 151–163 (2009)
7. Dasgupta, D.: Advances in artificial immune systems. *IEEE Comput. Intell. Mag.* **1**(4), 40–49 (2006)
8. Kirkpatrick, S., Gelatt, C., Vecchi, M.: Optimization by simulated annealing. *Science* **220**(4598), 671–680 (1983)
9. Formato, R.A.: Central force optimization: a new nature inspired computational framework for multidimensional search and optimization. *Nat. Inspired Coop. Strat. Optim. (NICSO)* **129**, 221–238 (2008)

10. Xie, L.P., Tan, Y., Zeng, J.C., Cui, Z.H.: Artificial physics optimization: a brief survey. *Int. J. Bio-Inspired Comput.* **2**(5), 291–302 (2010)
11. Rashedi, E., Nezamabadi-pour, H., Saryazdi, S.: GSA: a gravitational search algorithm. *Inf. Sci.* **179**, 2232–2248 (2009)
12. Birbil, S.I., Fang, S.-C.: An electromagnetism-like mechanism for global optimization. *J. Global Optim.* **25**(3), 263–282 (2003)
13. Rocha, A.M.A.C., Fernandes, E.M.G.P.: On charge effects to the electromagnetism-like algorithm. In: The 20th International Conference, EURO Mini Conference “Continuous Optimization and Knowledge-Based Technologies” (EurOPT-2008), Vilnius Gediminas Technical University Publishing House “Technika”, pp. 198–203 (2008)
14. Narayanan, A., Moore, M.: Quantum-inspired genetic algorithms. In: Proceedings of the IEEE International Conference on Evolutionary Computation (ICEC '96), pp. 61–66 (1996)
15. Sun, J., Xu, W., Feng, B.: A global search strategy of quantum behaved particle swarm optimization. In: Proceedings of the 2004 IEEE Conference on Cybernetics and Intelligent Systems, vol. 1, pp. 111–116 (2004)
16. Erol, O.K., Eksin I.: A new optimization method: Big Bang-Big Crunch. *Adv. Eng. Softw.* **37**(2), 106–111 (2006)
17. Spears, D.F., Kerr, W., et al.: An overview of physicomimetics. *Lect. Notes Comput. Sci.-State Art Ser.*, **3324**, 84–97 (2005)
18. Spears, W.M., Heil, R., Zarzhitsky, D.: Artificial physics for mobile robot formations. *Proc. IEEE Int. Conf. Syst. Man Cybern.* **3**, 2287–2292 (2005)
19. Kerr, W., Spears, D.F., Spears, W.M., et al.: Two formal gas models for multi-agent sweeping and obstacle avoidance. *Lect. Notes Artif. Intell.* **3228**, 111–130 (2005)
20. Spears, D.F., Kerr, W., Spears, W.F.: Physics-based robots swarms for coverage problems. *Int. J. Intell. Control Syst.* **11**(3), 11–23 (2006)
21. Xie, L.P., Zeng, J.C.: The performance analysis of artificial physics optimization algorithm driven by different virtual forces. *ICIC Express Lett. (ICIC-EL)*, **4**(1), 239–244 (2009)
22. Spears, W.M., et al.: *Physicomimetics: Physics-Based Swarm Intelligence*, pp. 549–573. Springer, Verlag Berlin Heidelberg Press, Berlin (2011)
23. Xie, L.P., Zeng, J.C., Cui, Z.H.: On mass effects to artificial physics optimization algorithm for global optimization problems. *Int. J. Innov. Comput. Appl.* **2**(2), 69–76 (2009)
24. Xie, L., Tan, Y., Zeng, J., Cui, Z.: The selection strategy of mass functions in artificial physics optimization algorithm. *Int. J. Model. Ident. Control* **18**, 226–233 (2013)
25. Wang, Y., Zeng, J.C., Cui, Z.H., He, X.J.: A novel constraint multi-objective artificial physics optimization algorithm and its convergence. *Int. J. Innov. Comput. Appl.* **3**(2), 61–70 (2010)
26. Xie, L., Yin, J., Zhang, H., Tan, Y.: Mass functions design of artificial physics optimization algorithm for constrained optimization problem. *Int. J. Comput. Appl. Technol.* **46**, 220–227 (2013)
27. Xie, L.P., Zeng, J.C.: A hybrid vector artificial physics optimization for constrained optimization problems. In: Proceedings-1st International Conference on Robot, Vision and Signal Processing, pp. 145–148 (2011)
28. Xie, L., Yang, G., Zeng, J., Cui, Z.: Swarm robots search based on artificial physics optimization algorithm. *Int. J. Comput. Sci. Math.* **4**, 62–71 (2013)
29. Xie, L., Yang, G., Zeng, J.: The model of swarm robots search with local sense based on artificial physics optimization. *Int. J. Comput. Sci. Math.* **4**(3), 222–230 (2013)
30. Xie, L.P., Zeng, J.C.: An extended artificial physics optimization algorithm for global optimization problem. In: Fourth International Conference on Innovative Computing, Information and Control (ICICIC 2009), 7–9 Dec 2009, Kaohsiung, Taiwan
31. Xie, L.P., Zeng, J.C., Formato, R.: Convergence analysis and performance of the extended artificial physics optimization algorithm. *Appl. Math. Comput.* **218**(8), 4000–4011 (2011)
32. Xie, L., Zeng, J., Cui, Z.: The vector model of artificial physics optimization algorithm for global optimization problems. In: Proceedings of the 10th International Conference on Intelligent Data Engineering and Automated Learning (IDEAL 2009), Spain, pp. 610–617 (2009)

33. Xie, L.P., Zeng, J.C., Cai, X.J.: A hybrid vector artificial physics optimization with multi-dimensional search method. In: Proceedings 2011 2nd International Conference on Innovations in Bio-Inspired Computing and Applications, pp. 116–119 (2011)
34. Yang, G., Xie, L., Tan, Y., Cui, Z.: Artificial physics optimization algorithm guided by diversity. *Int. J. Comput. Appl. Technol.* **46**, 369–375 (2013)
35. Xie, L., Tan, Y., Zeng, J.: A study on the effect of Vmax in artificial physics optimization algorithm with high dimension. In: The Second International Conference of Soft Computing and Pattern Recognition (SoCPaR 2011), Dalian, 14–16 Oct 2011
36. Xie, L., Tan, Y., Zeng, J., Cui, Z.: The convergence analysis of artificial physics optimization algorithm. *Int. J. Intell. Inf. Database Syst.* **5**(6), 536–554 (2011)
37. Xie, L.P., Zeng, J.C., Formato, R.: Selection strategies for gravitational constant G in artificial physics optimization based on analysis of convergence properties'. *Int. J. Bio-Inspired Comput.* **4**(6), 380–391 (2012)

NSGA-II Based Decision-Making in Fuzzy Multi-objective Optimization of System Reliability



Hemant Kumar and Shiv Prasad Yadav

Abstract This paper presents an approach to determine the optimal value of multi-objective optimization of a reliability-based system design problem. For this purpose, an over-speed protection system for a gas turbine is designed with mutually conflicting objectives such as the system reliability and system cost. This is a multi-objective nonlinear mixed integer programming problem subject to the upper limits on design constraints such as weight and volume. To solve the problem, a fuzzy approach is adopted to specify the goals in terms of the membership functions. This approach is effective in modeling the vague and imprecise information involved in the system. NSGA-II is employed to obtain the Pareto solutions efficiently. Finally, one out of these solutions is obtained by the decision-making methods such as TOPSIS and Shannon's entropy approach. The efficiency of the proposed approach is compared with the existing approach.

Keywords System reliability · Multi-objective optimization · Fuzzy optimization Membership function · NSGA-II · Crowding distance · Rank · Decision-making

1 Introduction

Reliability is characterized by the performance of a system under some specified conditions. It is a necessary aspect of an engineering system design. In many practical situations, a design engineer needs to improve the reliability with reduction of other resource consumptions such as cost, weight, and volume. Formulation of system design in multi-objective programming problem is a better adaptation in such situations. Many multi-objective approaches in reliability-based system design can be seen in [1–4]. Ideally, a multi-objective optimization presents a group of non-

H. Kumar (✉) · S. P. Yadav
Department of Mathematics, I.I.T. Roorkee, Roorkee 247667, Uttarakhand, India
e-mail: hemantkumar2654@gmail.com

S. P. Yadav
e-mail: spyorfma@gmail.com

© Springer Nature Singapore Pte Ltd. 2019
K. Deep et al. (eds.), *Decision Science in Action*, Asset Analytics,
https://doi.org/10.1007/978-981-13-0860-4_8

dominated solutions in the form of trade-offs, where the desired solution is then selected by some high-level information involved in the problem [5]. Classical optimization methods [6] are not able to fulfill such demands. Evolutionary algorithms [7] are useful alternatives in a multi-objective optimization problem, where a collective Pareto solutions is obtained simultaneously. The basic concepts and approaches of multi-objective evolutionary algorithms (MOEAs) can be viewed in Coello et al. [7]. Elitist non-dominated sorting genetic algorithm (NSGA-II) [6] is one of the second-generation MOEAs. It finds a much better convergence and spread of solutions near the true Pareto front [6] compared to two other elitist MOEAs such as PAES [8] and SPEA [9]. The applications of NSGA-II have now increased due to its elitism, parameter-less sharing approach, and low computational requirements [6]. Salazar et al. [10] showed the competency of NSGA-II to classify a set of optimal solutions (Pareto front) in solving constrained reliability problems. Wang et al. [11] used NSGA-II to solve multi-objective redundancy allocation problem (RAP) and compared their results with single-objective approaches. Kishore et al. [12] proposed an interactive approach to fuzzy multi-objective reliability optimization problem using NSGA-II. Safari [13] proposed a variant of NSGA-II in solving a multi-objective RAP. Khalili-Damghani et al. [14] proposed a decision-support system for multi-objective RAPs. Fuzzy-based multi-objective reliability problems are solved by Garg and Sharma [15] and Garg et al. [16] using PSO and GA. Recently, Sharifi et al. [17] present NSGA-II algorithm for solving multi-objective RAP for series-parallel and k-out-of-n subsystems with three objectives.

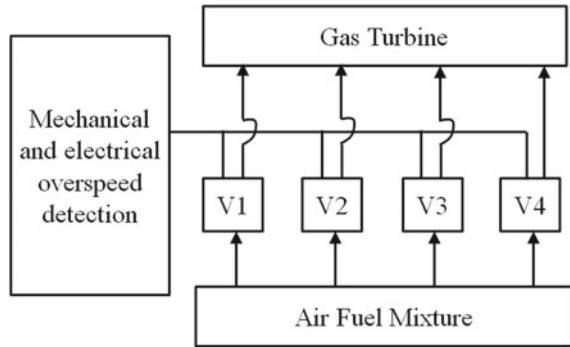
In this paper, a methodology is developed to achieve the optimal value of multi-objective reliability-based system design problem. First, the multi-objective problem of system design is formulated in the fuzzy environment and then solved by using NSGA-II. In order to find a concrete solution, decision-making methods such as TOPSIS [20] and Shannon's entropy [21] are implemented on the basis of the ideal and anti-ideal points (solutions) specified by the decision-maker. The optimal values are shown graphically in the objective space. The proposed method is compared with one of the existing approaches [15]. The rest of the paper is organized as follows. In Sect. 2, a mathematical model of the problem is constructed. Section 3 presents a concise depiction of the NSGA-II algorithm. In Sect. 4, the proposed methodology is described. Section 5 gives the results and with its discussion and Sect. 6 gives the conclusion.

2 Mathematical Model of the Problem

In this work, a four-stage over-speed protection system model [1] for a gas turbine is considered. The system diagram is shown in Fig. 1.

Over-speed detection is constantly arranged by the electrical and mechanical systems. When an over-speed occurs, the fuel supply goes cut off. In this way, four control valves ($V1-V4$) get locked. The control system is formed as a 4-stage *series* system. A constant failure rate occurs for all components in the system. The goal is

Fig. 1 A symbolic diagram of the over-speed protection system



to determine the optimal design variables R_j and $|X_j|$ at each stage j such that the minimization of the system cost and the maximization of the system reliability can be achieved simultaneously.

Notation:

- R_S System reliability;
- C_S cost of the total system;
- R_j reliability of a component at stage j ;
- $|X_j|$ number of the redundant component at stage j ;
- W_S total system weight;
- V_S total system volume;
- W_{lim} upper limit on the system weight;
- V_{lim} upper limit on the system volume;
- W_j weight of each component at stage j ;
- V_j volume of each component at stage j ;
- γ_j, δ_j physical quantities representing characteristics of each component at stage j ;
- M number of stages;
- τ operating time

The mathematical model of the problem is given as follows:

$$\text{Max } R_S = \prod_{j=1}^M [1 - (1 - R_j)^{|X_j|}], \tag{1}$$

$$\text{Min } C_S = \sum_{j=1}^M \gamma_j \left(\frac{-\tau}{\ln(R_j)} \right)^{\delta_j} [|X_j| + \exp(|X_j|/4)], \tag{2}$$

subject to

$$W_S = \sum_{j=1}^M W_j |X_j| \exp(|X_j|/4) \leq W_{lim}, \tag{3}$$

$$V_S = \sum_{j=1}^M V_j (|X_j|)^2 \leq V_{\text{lim}}, \quad (4)$$

$$1 \leq |X_j| \leq |X_{\text{max}}|, R_{\text{min}} \leq R_j \leq R_{\text{max}}, j = 1, 2, \dots, M; |X_j| \in \mathbb{Z}^+, R_j \in \mathbb{R}^+, \quad (5)$$

where $\exp(|X_j|/4)$ represents the interconnecting hardware, $|X_{\text{max}}|$ denotes the maximum number of components given at each stage, R_{min} and R_{max} denote the minimum and maximum values on the reliability of each component.

Assumptions:

- (i) The cost–reliability relation is

$$C(R_j) = \gamma_j \lambda_j^{-\delta_j} \quad (6)$$

- (ii) Each component of the system has a constant failure rate λ_j that follows an exponential distribution. The reliability of each component is obtained by

$$R_j(\tau) = \int_{\tau}^{\infty} \lambda_j e^{-\lambda_j \tau} d\tau = e^{-\lambda_j \tau} \quad (7)$$

From (6) and (7), the cost of each component is

$$C(R_j) = \gamma_j [-\tau / \ln(R_j)]^{\delta_j} \quad (8)$$

3 NSGA-II

Non-dominated sorting genetic algorithm (NSGA) was initially suggested by Srinivas and Deb [18]. It uses Goldberg's domination criterion [19] to assign ranks for the solutions and utilization of fitness sharing for maintaining the diversity in the solution set. It has some difficulty in regarding computational complexity, non-elitist approach, and highly dependent on the parameters of fitness sharing. Deb et al. [6] extended this algorithm in the form of NSGA-II by giving some new features like fast non-dominated sorting, crowding distance, and comparison operator.

NSGA-II assigns a rank for solutions employing non-dominated sorting procedure and emphasizes good solutions throughout this algorithm. The overall complexity governed by this process is $O(kN^2)$, where k and N denote the number of objectives and population size, respectively [6]. See Fig. 2a.

For maintaining the diversity in the solution set, NSGA-II calculates the crowding distance of each solution. It is basically defined as those solutions that contain the same rank. A partial order comparison operator is applied to determine a better

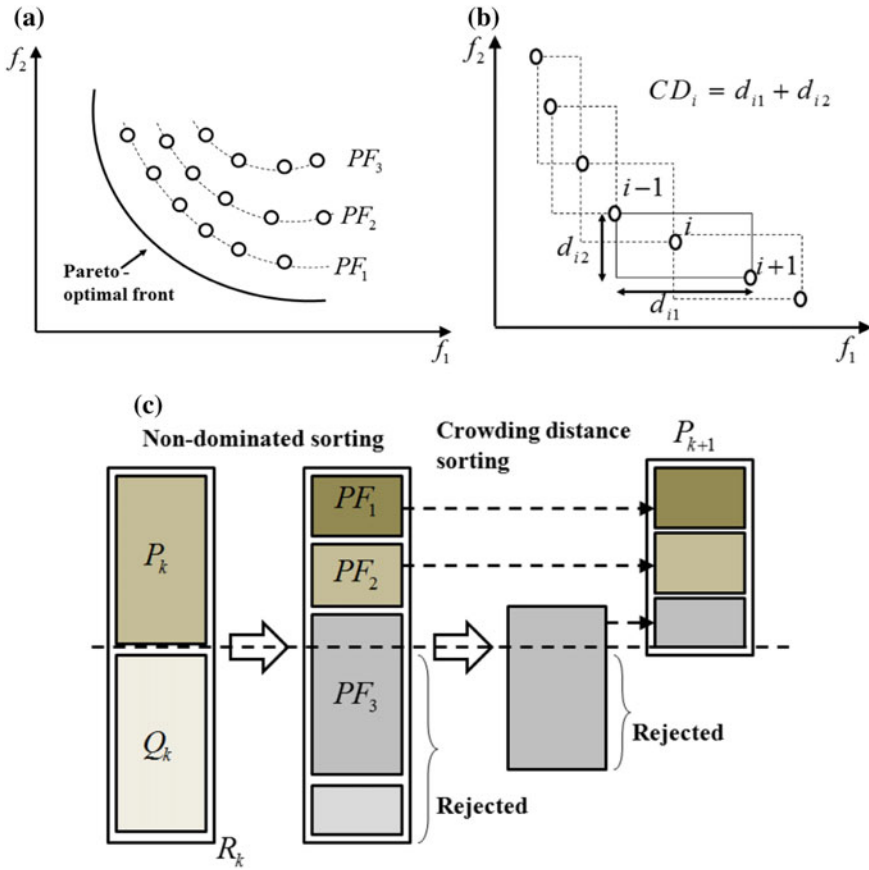


Fig. 2 a Sorting procedure of a population. b Crowding distance estimation of a solution. c Evaluation cycle of the NSGA-II algorithm

solution between two solutions. According to this operator, if both the solutions belong to the same rank, then preference is given to the solution that contains a higher crowding distance value. A higher crowding distance value gives the lesser crowded region and vice versa [6]. See Fig. 2b.

Deb et al. [6] proposed constraint-dominance based binary tournament selection method in constraint handling procedure. A search space is divided by the constraints into two regions—feasible and infeasible. Accordingly, a solution α is defined as a *constrained-dominate* to a solution β if

- (i) α is feasible and β is infeasible.
- (ii) α and β are infeasible, but α contains a lower overall constraint violation.
- (iii) α and β are feasible, but α dominates β .

The pseudocode of NSGA-II algorithm (See Fig. 2c) is given as follows:

- Step 1. Initializing randomly a parent population P_0 of size N . Setting $k = 0$.
- Step 2. Assigning fitness (rank) according to non-domination level and crowded-comparison operator.
- Step 3. **while** $k < \text{number of maximum generation}$ **do**
- (i) Creating an offspring population Q_k of size N applying reproduction, crossover, and mutation.
 - (ii) Combining via $R_k = P_k \cup Q_k$.
 - (iii) Sorting on R_k and classifying them into non-dominated fronts (Pareto fronts) $PF_i, i = 1, 2, \dots$, etc.
 - (iv) Setting a new population $P_{k+1} = \emptyset$ and $i = 1$.
while the parent population size $|P_{k+1}| + |PF_i| < N$ **do**
 - (i) Calculating the crowding distance of PF_i .
 - (ii) Adding the i th non-dominated front PF_i to the parent population P_{k+1} .
 - (iii) $i = i + 1$.**end while**
 - (v) Sorting the PF_i using the crowding distance-based comparison operator.
 - (vi) Filling the parent population P_{k+1} with the first $N - |P_{k+1}|$ solutions of PF_i .
 - (vii) Generating the offspring population Q_{k+1} .
 - (viii) Setting $k = k + 1$.
- end while**

Step 4. Collecting the non-dominated solutions in the vector P .

4 Proposed Methodology

The problem given in Sect. 2 is solved by the following steps:

Step 1. Constructing the membership functions of fuzzy objectives (Fig. 3).

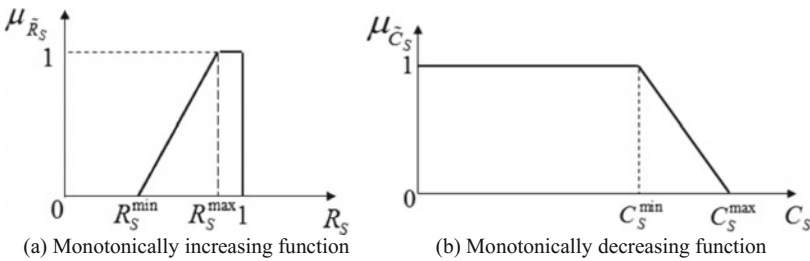


Fig. 3 Linear membership function for **a** system reliability **b** system cost

$$\mu_{\tilde{R}_S} = \begin{cases} 0, & R_S \leq R_S^{\min}, \\ \frac{R_S - R_S^{\min}}{R_S^{\max} - R_S^{\min}}, & R_S^{\min} < R_S < R_S^{\max}, \\ 1, & R_S \geq R_S^{\max}, \end{cases} \quad (9)$$

where R_S^{\min} and R_S^{\max} are the minimum and maximum values on the system reliability, respectively. This range is fixed by the decision-maker according to his/her requirements.

$$\mu_{\tilde{C}_S} = \begin{cases} 1, & C_S \leq C_S^{\min}, \\ \frac{C_S^{\max} - C_S}{C_S^{\max} - C_S^{\min}}, & C_S^{\min} < C_S < C_S^{\max}, \\ 0, & C_S \geq C_S^{\max}, \end{cases} \quad (10)$$

similarly, C_S^{\min} and C_S^{\max} are the minimum and maximum values on the system cost, respectively. This range is decided by the decision-maker according to his/her investment capacity.

Step 2. Formulating the problem in the form of fuzzy objectives.

$$\text{Maximize } (\mu_{\tilde{R}_S}, \mu_{\tilde{C}_S}) \quad (11)$$

subject to the constraints given in (3)–(5).

Step 3. Setting the parameters as given in Tables 1 and 2, and then applying the NSGA-II algorithm to get the Pareto front in (11).

Step 4. Constructing the decision matrix of objectives (criteria) as follows:

$$D = \begin{bmatrix} \mu_{\tilde{R}_S}^{11} & \mu_{\tilde{R}_S}^{21} & \dots & \mu_{\tilde{R}_S}^{m1} \\ \mu_{\tilde{C}_S}^{12} & \mu_{\tilde{C}_S}^{22} & \dots & \mu_{\tilde{C}_S}^{m2} \end{bmatrix}^T = [\mu_{\tilde{R}_S, \tilde{C}_S}^{ij}]; \quad i = 1, 2, \dots, m; \quad j = 1, 2. \quad (12)$$

Step 5. Finding the best alternative in the decision matrix given by (12).

To determine the best alternative, we apply the decision-making method as follows:

4.1 TOPSIS Approach

In the present work, we applied the TOPSIS method [20] on membership values of the objective functions. The ideal membership value is taken as 1 for the upper limit of each objective and the anti-ideal membership value is taken as 0 for the lower limit of each objective. The Euclidean distances of each membership value of the objective function from the anti-ideal and ideal points are calculated, respectively, as follows:

$$D_i^- = \sqrt{\sum_{j=1}^2 (\mu_{R_s, C_s}^{ij} - 0)^2}, \quad i = 1, 2, \dots, m \tag{13}$$

$$D_i^+ = \sqrt{\sum_{j=1}^2 (\mu_{R_s, C_s}^{ij} - 1)^2}, \quad i = 1, 2, \dots, m \tag{14}$$

In this method, D_i (relative closeness of i th alternative) is calculated as

$$D_i = \frac{D_i^-}{D_i^- + D_i^+} \tag{15}$$

Table 1 Designing data for the problem

Number of stages (M)		4				
$1 \leq X_j \leq 10, 0.5 \leq R_j \leq 1 - 10^{-6}, j = 1, 2, 3, 4; X_j \in \mathbb{Z}^+, R_j \in \mathbb{R}^+$						
		Stage	$10^5 \gamma_j$	δ_j	v_j	w_j
Upper limit on W_s	500.0	1	1.0	1.5	1	6
Upper limit on V_s	250.0	2	2.3	1.5	2	6
Operating time (τ)	1000 h	3	0.3	1.5	3	8
		4	2.3	1.5	2	7

Table 2 The parameter settings for the given problem

The parameters are set to NSGA-II and GA					
Population size	80	R_s^{\min}	R_s^{\max}	C_s^{\min}	max
Maximum generation	100	0.75	0.99	25	100
Crossover rate	0.9	Ideal point = (25, 0.99)			
Mutation rate	0.1	Anti-Ideal point = (100, 0.75)			
Random seed	0.1234				

Therefore,

$$A_i^{\text{best}} = \max(D_i). \quad (16)$$

4.2 Shannon's Entropy Approach

Entropy [21] is calculated to measure the disorder in the given discrete probability distribution of the system. It is observed that a broad distribution gives a more uncertainty than a sharply packed distribution. Consider H_{ij} in the decision matrix D as follows:

$$H_{ij} = \frac{\mu_{R_s, C_s}^{ij}}{\sum_{i=1}^m \mu_{R_s, C_s}^{ij}}, \quad i = 1, 2, \dots, m; \quad j = 1, 2. \quad (17)$$

Shannon's entropy is calculated by

$$E_j = -M \sum_{i=1}^m H_{ij} \ln H_{ij}, \quad M = 1/\ln(m) \quad (18)$$

The degree of deviation is obtained by

$$DV_j = 1 - E_j \quad (19)$$

The weight of j th fuzzy objective is calculated by

$$W_j = \frac{DV_j}{\sum_{j=1}^2 DV_j} \quad (20)$$

Finally,

$$Y_i = \sum_{j=1}^2 H_{ij} W_j; \quad i = 1, 2, \dots, m \quad (21)$$

Therefore,

$$A_i^{\text{best}} = \max(Y_i). \quad (22)$$

Formulation of the problem for the genetic algorithm (GA) [19]-based decision-making

$$\text{Maximize} \left(1 \wedge \frac{\alpha_1}{W_1} \right) \wedge \left(1 \wedge \frac{\alpha_2}{W_2} \right) \quad (23)$$

subject to

$$\alpha_1 = \mu_{\tilde{R}_S}, \alpha_2 = \mu_{\tilde{C}_S}, W_1, W_2 \in (0, 1] \quad (24)$$

$$W_S = \sum_{j=1}^M W_j |X_j| \exp(|X_j|/4) \leq W_{\text{lim}}, \quad (25)$$

$$V_S = \sum_{j=1}^M V_j (|X_j|)^2 \leq V_{\text{lim}}, \quad (26)$$

$$1 \leq |X_j| \leq 10, 0.5 \leq R_j \leq 1 - 10^{-6}, j = 1, 2, 3, 4; |X_j| \in \mathbb{Z}^+, R_j \in \mathbb{R}^+ \quad (27)$$

where \wedge represents *min* operator as the aggregate operator, W_1 and W_2 are the weights of the objectives suggested by the decision-maker, α_1 and α_2 are the degree of satisfaction of the objectives.

5 Results and Discussion

The problem presented in Sect. 2 is a RAP problem. A real number of encoding is used in a vector of design variables $[(R_1, |X_1|), (R_2, |X_2|), (R_3, |X_3|), (R_4, |X_4|)]$. The SBX and polynomial operators [5] are used for crossover and mutation, respectively. Based on rigorous experimentation, results are obtained in Table 3. In Table 3, the proposed approach is compared with heuristic method GA where the problem is converted to single objective using aggregation operator. To make a fair comparison, same parameters are used and equal weight given to each objective. One of the best solutions is chosen from 10 independent runs in GA. In Fig. 4, the results are displayed on the basis of membership functions. There are 29 solutions found by NSGA-II in the first front. The decision-making methods are applied on the basis of the Euclidean distances from the ideal and anti-ideal points. Figure 5 shows the Pareto front and the best results obtained by the decision-making methods such as TOPSIS and Shannon's entropy.

6 Conclusion

In this piece of work, an approach is developed to determine the optimal value of fuzzy multi-objective reliability-based system design. A mathematical model of real-world problem of the over-speed protection system is presented. To avoid any kind of aggregator operators, NSGA-II is employed to solve the problem. At the decision-

Table 3 Comparison of optimal solutions with the existing method

	NSGA-II based decision-making		GA-based decision-making
	TOPSIS	Shannon's entropy method	$W = [0.5, 0.5]$
(R_1, X_1)	(0.73947, 3)	(0.73349, 3)	(0.72315, 3)
(R_2, X_2)	(0.65557, 3)	(0.67206, 3)	(0.66142, 3)
(R_3, X_3)	(0.83512, 3)	(0.84847, 3)	(0.80802, 3)
(R_4, X_4)	(0.65311, 3)	(0.69319, 3)	(0.66382, 3)
$\mu_{\tilde{R}_s}$	0.669	0.728	0.739
$\mu_{\tilde{C}_s}$	0.731	0.659	0.709
R_s	0.91726	0.93192	0.92729
C_s	45.16	50.57	46.81
W_s	477.58	469.55	498.78
V_s	76.72	72.21	77.54

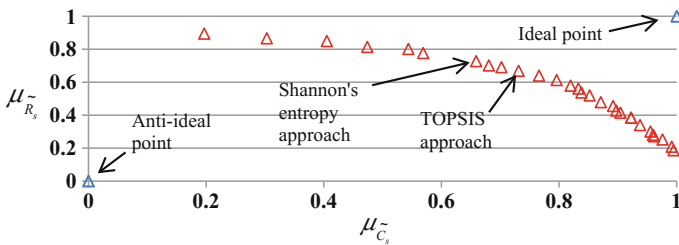


Fig. 4 The optimal values based on membership functions

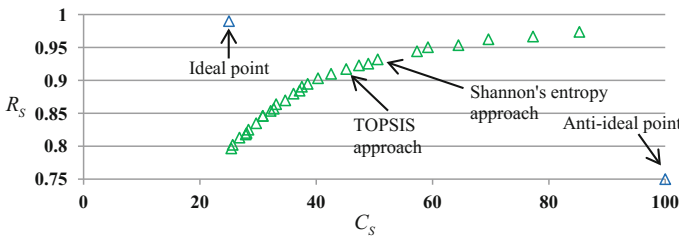


Fig. 5 The optimal values based on objective functions

making stage, we modify the decision-making methods in terms of membership function and find the best optimal value according to the Euclidean distances from the ideal and anti-ideal points (solutions) in the objective space. In order to show the efficiency of the proposed approach, it is compared with the existing approach. The obtained results are found encouraging. Thus, the proposed methodology can be a

better adaptation in finding the concrete solution in multi-objective reliability-based system design problem.

Acknowledgement The first author acknowledges the MHRD, Govt. of India, for providing the financial grant.

References

1. Dhingra, A.K.: Optimal apportionment of reliability & redundancy in series and system under multiple objectives. *IEEE Trans. Reliab.* **41**(4), 576–582 (1992)
2. Huang, H.Z.: Fuzzy multiobjective optimization decision-making of reliability of series system. *Micro. Reliab.* **37**(3), 447–449 (1997)
3. Rao, S.S., Dhingra, A.K.: Reliability and redundancy apportionment using crisp and fuzzy multi-objective approaches. *Reliab. Eng. Syst. Saf.* **37**(3), 253–261 (1992)
4. Ravi, V., Reddy, P.J., Zimmermann, H.J.: Fuzzy global optimization of complex system reliability. *IEEE Trans. Fuzzy Syst.* **8**(3), 241–248 (2000)
5. Deb, K.: *Multi-objective Optimization Using Evolutionary Algorithms*. Wiley, New York (2001)
6. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **6**(2), 182–197 (2002)
7. Coello, C.A., Lamount, G.B., Veldhuizen, D.A.V.: *Evolutionary Algorithms for Solving Multi-objective Problems*, 2nd edn. Springer Science+Business Media, LLC (2007)
8. Knowles, J., Corne, D.: The Pareto archived evolution strategy: a new baseline algorithm for multiobjective optimization. *Pro Cong Evol Comp Piscataway NJ IEEE Press* (1999). <https://doi.org/10.1109/CEC1999.781913,98-105>
9. Zitzler, E., Thiele, L.: An evolutionary algorithm for multi-objective optimization: the strength Pareto approach. Technical report 43, Zurich, Switzerland: Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH) (1998)
10. Salazar, D., Rocco, C.M., Galvan, B.J.: Optimization of constrained multiple objective reliability problems using evolutionary algorithms. *Reliab. Eng. Syst. Saf.* **91**, 1057–1070 (2006)
11. Wang, Z., Chen, T., Tang, K., Yao, X.: A multi-objective approach to redundancy allocation problem in parallel-series systems. *IEEE*, 582–589 (2009). doi: 978-1-4244-2959-2/09
12. Kishore, A., Yadav, S.P., Kumar, S.: Interactive fuzzy multiobjective optimization using NSGA-II. *OPSEARCH* **46**(2), 214–224 (2009)
13. Safari, J.: Multi-objective reliability optimization of series-parallel systems with a choice of redundancy strategies. *Reliab. Eng. Syst. Saf.* **108**, 10–20 (2012)
14. Khalili-Damghani, K., Abtahi, A.R., Tavana, M.: A decision support system for solving multi-objective redundancy allocation problems. *Qual. Reliab. Eng. Int.* **30**(8), 1249–1262 (2014)
15. Garg, H., Sharma, S.P.: Multi-objective reliability-redundancy allocation problem using particle swarm optimization. *Comput. Ind. Eng.* **64**(1), 247–255 (2013)
16. Garg, H., Rani, M., Sharma, S.P., Vishwakarma, Y.: Intuitionistic fuzzy optimization technique for solving multi-objective reliability optimization problems in interval environment. *Expert Syst. Appl.* **41**, 3157–3167 (2014)
17. Sharifi, M., Guilani, P.P., Shahriari, M.: Using NSGA-II algorithm for a three objective redundancy allocation problem with k-out-of-n sub-systems. *J. Optim. Indl. Eng.* **19**, 87–95 (2016)
18. Srinivas, N., Deb, K.: Multi-objective optimization using non-dominated sorting in genetic algorithms. *Evol. Comput.* **2**(3), 221–248 (1994)
19. Goldberg, D.E.: *Genetic Algorithms for Search, Optimization, and Machine Learning*. Addison-Wesley, Reading (1989)

20. Wang, D., Jiang, R., Wu, Y.: A hybrid method of modified NSGA-II and TOPSIS for light weight design of parameterized passenger car sub-frame. *J. Mech. Sci. Technol.* **30**(11), 4909–4917 (2016)
21. Huang, J.: Combining entropy weight and TOPSIS method for information system selection. In: *Proceedings of IEEE International Conference Automatic Logis., Qingdao, China*, 1281–1284 (2008)

GA-Based Task Scheduling Algorithm for Efficient Utilization of Available Resources in Computational Grid



Shipra Singh, Anuradha Aggarwal, Harendera Kumar
and Pradeep Kumar Yadav

Abstract In the grid computing environment, systematic scheduling of tasks/jobs on hand resource is the important parameter for performance evaluation of computational grid. Traditional algorithms cannot produce a load balancing schedule. In the paper, a genetic approach for grid task scheduling has been considered to achieve better solutions within a reasonable period of time. The present study aims at minimizing the make-span and flow-time at the same time and also achieves equiponderant practical application of a set of “ n ” available computing agents of a grid computing to get the average load balancing. The simulation results show that the proposed approach is more efficient than the GA approach reported in the literature.

Keywords Scheduling · Computational grid · Expected computation time
Inter-task communication time · Optimization · Genetic algorithm

1 Introduction

Grid computing is a gathering of computer resources from many places to reach a similar target. Grid computing is emerging as a new and significant field that can be considered as an enhanced form of distributed computing [1]. That includes non-interactive workloads, which contain a large number of files. Users can share grid

S. Singh (✉)
Uttarakhand Technical University, Dehradun, India
e-mail: shiprasing.qst@gmail.com

A. Aggarwal · H. Kumar
Gurukul Kangri University, Haridwar, India
e-mail: annu08annu@gmail.com

H. Kumar
e-mail: balyal.kumar@gmail.com

P. K. Yadav
CSIR-Central Building Research Institute, Roorkee, India
e-mail: prd_yadav@rediffmail.com

© Springer Nature Singapore Pte Ltd. 2019
K. Deep et al. (eds.), *Decision Science in Action*, Asset Analytics,
https://doi.org/10.1007/978-981-13-0860-4_9

computing agents by assigning tasks to the computational grid. Task scheduling in grid environments poses a great challenge due to the heterogeneity of resources. Utilization of all the resources should be done in an efficient manner. Because under-utilized resources will result in worst scheduling. At present researchers are using nature-inspired meta-heuristics mechanisms to solve computational grid problems due to increasing the size of search space. Various heuristic approaches have been reported in the literature for scheduling the tasks/jobs in the computational grid. These include economic heuristic [2], population-based heuristic [3], meta-heuristic [4], simple heuristic [5], and hybrid heuristic [6, 7].

The major challenges when using GAs to solve task scheduling problems are: (a) to generate and keep the diversity in the populations, which is crucial for avoiding the premature convergence to the local optima and (b) to evolve robust solutions that are able to track the optimal [8]. A Task Duplication-Based Scheduling Algorithm on GA in Grid Computing Systems, has been reported by Lin and Wu [9]. In the present study, we consider a multipurpose scheduling problem in the computational grid in which two objectives make-span and flow-time are optimized simultaneously with the objective of optimum utilization of a set of “ n ” available modalities. We have extended our previous work [10, 11], where the genetic algorithm based scheduling did not consider the task’s computational load and computing capacity of resources. In this paper, the implementation of task’s computational load and computing capacity of resources makes the whole strategy better than previous.

Notations

The notations used throughout the paper are as follows:

- m Number of tasks
- n Number of computing agents
- M $\{1, \dots, m\}$
- N $\{1, \dots, n\}$

2 Problem Statement

The main purpose of scheduling in the computational grid is a skilled mapping for a set of “ n ” computing agents available by applications. Due to the diversity of computing agents in the grid computing environment, mapping of the tasks to computing agents is a challenging global optimization problem. In this paper, “ m ” tasks are considered for scheduling onto “ n ” computing agents and developed an algorithm for the efficient utilization of available resources to achieve the objective.

3 Description of Inter-communication Time

The tasks/jobs can be explained as massive applications with inter-task communication time of communicating tasks. In the present paper, the communication time is taken in the form of Inter-Task Communication Time Matrix

$$ITCTM(,) = [ct_{ik}]_{m*m}$$

where, ct_{ik} is the Inter-Task Communication Time between i th and k th tasks.

4 Expected Time of Computation

Execution time of a task depends on the computational agents, to which it is allocated and the work to be performed by the task. The execution time of each task on all the computational agents are given in the form of Expected Time to Computation (ETC) matrix, $ETC() = [et_{ij}]_{m*n}$, where et_{ij} represents the expected time needed for the completion of i th task on j th computational agent. All et_{ij} can be computed as the ratio of the coordinates of WL and CC vectors that is to say:

$$e_{ij} = \frac{wl_i}{cc_j} \quad (1)$$

wl_i represented the computational load parameter for every i th task, which is expressed in Millions of Instructions (MIs) [12]. On the behalf of historic data/predictions, necessities about computation need of the tasks can be known from terms provided by the user [13]. $WL = [wl_1, wl_2, \dots, wl_m]$ denotes the workload parameters and also represent the coordinates of a workload vector. Gaussian probability distribution has used to generate the values of wl_i and cc_j .

Each j th computing agent in the system is represented by the parameter cc_j , where cc_j is computing capacity of the j th computational agent, that can be represented in Millions of Instructions Per Second (MIPS), we denote by $CC = [cc_1, cc_2, cc_3 \dots, cc_n]$, a computing capacity vector.

5 Schedule Representation

Dissimilar types of scheduling representation are reported in the literature. We use one of them called direct representation. In direct representation, the schedule is encoded in the form of a vector whose size is assumed to be the task numbers. The elements of the present vector are natural numbers lies between 1 and n . In this type of representation, processing agents can appear more than once.

i.e., the schedule is encoded as

$$\text{Schedule} = \{p_1, p_2, p_3, p_4, \dots, p_n\} \quad (2)$$

6 Scheduling Criteria

Two different modes, i.e., hierarchical and simultaneous can be used to optimize the variables of multi-objective functions. All the variables of the objective function are optimized at the same time in hierarchical mode, and the priority has been defined for the optimization criteria, according to their importance in the model. In the present paper scheduling criteria is defined as a multi-objective optimization problem in which make-span and flow-time will be minimized simultaneously considering the steady utilization of all the processing agents. Make-span is usually known as a finishing time of the latest task, i.e., if $FT(i)$ denotes the finishing time of task $i \in M$, then

$$\text{Makespan} = \min_{s \in \text{schedule}} \max_{i \in M} FT(i)$$

In terms of ETC and ITCT, the make-span is expressed as

$$\text{Makespan} = \min_{s \in \text{schedule}} \max_{i \in M} T_j \quad (3)$$

where T_j is the total time on j th processing agents and it is calculated as

$$T_j = \sum_{i=1}^m et_{ij}x_{ij} + \sum_{i=1}^m \sum_{k=1}^m \sum_{l=1}^n ct_{ik}x_{ij}x_{kl} \quad (4)$$

where

$$x_{ij} = \begin{cases} 1, & \text{if } i\text{th task is assigned to } j\text{th computing agent} \\ 0, & \text{otherwise} \end{cases}$$

And

$$x_{kl} = \begin{cases} 1, & \text{if } k\text{th task is assigned to } l\text{th computing agent} \\ 0, & \text{otherwise} \end{cases}$$

The flow-time is defined as the summation of finalization times of all the tasks, i.e.,

$$\text{Flowtime} = \min_{s \in \text{schedule}} FT(i)$$

In terms of ETC and ITCT, the flow-time is defined as

$$\text{Flowtime} = \min_{s \in \text{schedule}} T$$

“ T ” is the total time on all the processors and it is calculated as

$$T = \sum_{j=1}^n T_j \tag{5}$$

7 Constraints

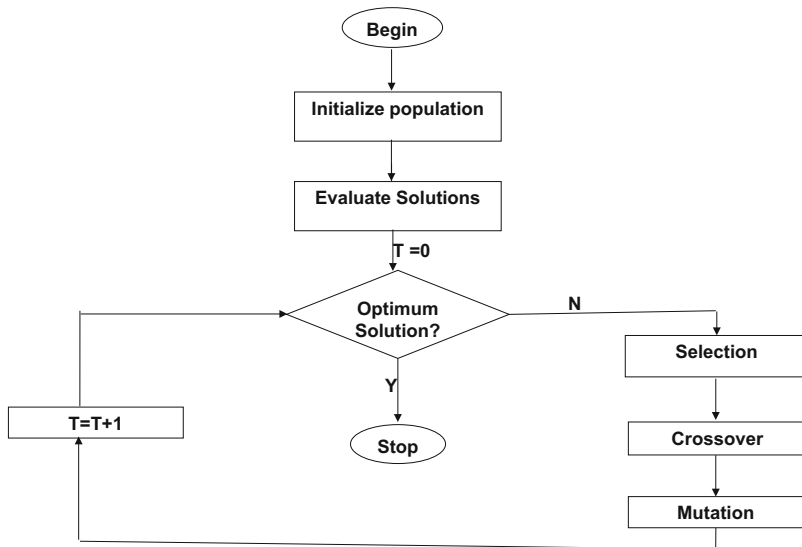
To conclude the suitable allocation, first of all average load of every computing agent must be determined.

If the finishing time of i th task on j th computing agent is et_{ij} , the average load on every computing agent is as shown in Eq. (6)

$$L_{\text{avg}}(P_j) = \frac{\sum_{i=1}^m et_{ij}}{n} \tag{6}$$

To get the balanced utilization of “ n ” available computing agents, it assumed that the number of tasks/jobs to allocated any computing agents is equal to $\frac{m}{n}$

8 Working Mechanism of GAs



9 Proposed Algorithm

The mapping of tasks to processors takes place according to the following algorithm:

- (1) Generate the initial population of random individuals and select only those which satisfy the constraint.
- (2) Evaluate the fitness of each individual.
- (3) While not termination condition do.
- (4) While not termination condition do.
- (5) Perform selection operator.
- (6) Perform crossover according to p_c ($p_c \geq 0.8$)
- (7) Perform Mutation according to p_m ($p_m \leq 0.1$).
- (8) Evaluate the fitness of the modified individual.
- (9) End while.
- (10) Choose only those childs, which suit the constraint.

10 Results and Discussions

To verify the performance of the presented algorithm (GA), the data has been taken from the literature reported by [10] and some simulation cases will be tested. Parameters used in the present study are as follows and shown in Table 1. We use the notation $N(a, b)$ for the Gaussian probability distribution.

The computation has been repeated 30 times under the same arrangement of parameters. Computational results for the make-span and flow-time values are presented in Table 2. Simulation results demonstrate that more iterations or population sizes obtain the better solution since more solutions were generated as displayed in Table 2.

The algorithm reported by [10] can minimize the only one parameter at a time either Make-span or flow-time. The algorithm in the present paper minimizes the Make-span and flow-time simultaneously and achieves equilibrated utilization of available resources by processing fruitfully tasks onto resources. The authentication of the statistical significance of the results has been conducted through a two-way

Table 1 Values of key parameters

	Small	Medium	Large
Total no. of tasks	18	24	36
Workload of tasks		$N(5000, 800)$	
Resource Cap. (in MHZ cpu)		$N(30, 10)$	
Crossover probability		$p_c = 0.8$	
Mutation probability		$P_m = 0.1$	

Table 2 Results

Task, resource	Population size							
	800				1000			
	Iterations							
	100		200		100		200	
	Make-span	Flow-time	Make-span	Flow-time	Make-span	Flow-time	Make-span	Flow-time
18, 6	14,604.29	3008.70	14,462.88	2791.11	14,368.55	2684.35	14,339.94	2684.35
24, 6	22,804.40	3999.84	22,746.92	4013.49	22,732.93	4037.30	22,775.11	4055.48
24, 8	23,947.98	3302.18	23,784.81	3186.54	23,723.58	3245.70	23,669.56	3290.45
36, 6	47,182.62	8229.78	47,314.55	8168.60	47,115.66	8227.07	46,971.52	8168.60

Table 3 Comparison of “*F*” value for make-span and flow-time results

Objectives	Source of validation	SS	df	MS	<i>F</i>	<i>P</i> -value	<i>F</i> _{crit}
Flow-time	Rows	1.10E+09	4	2.75E+08	666.1653	5.66E−14	3.25916
	Columns	2,072,441	3	690,813.6	1.672766	0.022546	3.490295
	Error	4,955,722	12	4.13E+05			
	Total	1.11E+09	19				
Make-span	Rows	77,073,912	4	19,268,478	3541.51	2.57E−18	3.259167
	Columns	26,966.43	3	8988.809	1.652126	0.22978	3.490295
	Error	65,289.03	12	5440.752			
	Total	77,166,167	19				

ANOVA test. The conclusion of this test is the acceptance or rejection of the null hypothesis (H_0) which states that any difference in the results is purely random. The null hypothesis is rejected if “*F*” value is greater than “*F*_{crit}”. Table 3 represents that “*F*” value of rows is greater than “*F*_{crit}” and therefore H_0 is rejected. This indicates that make-span and flow-time in the row of table are statistically significant. Therefore, it is proven that the proposed method most effective in both make-span and flow-time reductions with increased Iterations and population size.

The algorithm was studied with more and more runs and the reason for the size of the population and iterations was found. The present algorithm will be validated in our next study by developing the simulator for real environment.

References

1. Foster, I., Kesselman, C., Tuecke, S.: The anatomy of the grid enabling scalable virtual organizations. Int. J. Supercomput. Appl. (2001)

2. Ni, L.M., Xu, Z., Xiao, L., Zhu, Y.: Incentive based scheduling for market like computational grid. *IEEE Trans. Parallel Distrib. Syst.* **19**, 903–913 (2008)
3. Kumar Singh, P., Sahli, N.: Task scheduling in grid computing environment using compact genetic algorithm. *Int. J. Sci. Eng. Technol. Res. (IJSETR)* **3**(1) (2014)
4. Adrianto, D.: Comparison using particle swarm optimization and genetic algorithm for timetable scheduling. *Comput. Sci.* **10**(2), 341–346 (2014)
5. Alakeel, A.M.: A fuzzy dynamic load balancing algorithm for homogenous distributed systems. *World Acad. Sci. Eng. Technol.* **61** (2012)
6. Jianchun, J., et al.: Embedded static task allocation and scheduling based on simulated annealing and genetic algorithm. *J. Comput. Inf. Syst.* **10**, 4 (2014)
7. Shuqeir, S.Y.A., Al Qublan, T.A.: Hybrid algorithm based on ant and genetic algorithms for task allocation Ona network of homogeneous processors. *Int. J. Comput. Netw. Commun. (IJCNC)* **6**(1) (2014)
8. Kołodziej, J., Khan, S.U.: Multi-level hierarchic genetic-based scheduling of independent jobs in dynamic heterogeneous grid environment, *Inform. Sci.* (2012). <http://dx.doi.org/10.1016/j.ins.2012.05.016>
9. Lin, J., Wu, H.: A task duplication based scheduling algorithm on ga in grid computing systems. In: *International Conference on Natural Computation ICNC 2005: Advances in Natural Computation*, pp. 225–234 (2005)
10. Singh, M.P., Yadav, P.K., Aggarwal, A.: Response time optimization of a grid computing system using genetic approach. In: *Conference Proceeding, Dhanbad, Jharkhand*, pp. 171–179 (2013)
11. Singh, M.P., Yadav, P.K., Aggarwal, A.: Task scheduling in a distributed processing environment: a genetic approach
12. Ali, S., Siegel, H.J., Maheswaran, M., Hensgen, D.: Task execution time modelling for heterogeneous computing systems. In: *Proceedings of Heterogeneous Computing Workshop*, pp. 185–199 (2000)
13. Hotovy, S.: Workload evolution on the Cornell theory center IBM SP2. In *Job Scheduling Strategies for Parallel Processing Workshop, IPPS'96*, pp. 27–40 (1996)

Statistical Feature Analysis of Thermal Images from Electrical Equipment



Tamal Dutta, Deepjyoti Santra, Chee Peng-Lim, Jaya Sil
and Paramita Chottopadhyay

Abstract This investigation focuses on intelligent monitoring systems by assessing thermal images from electrical equipment. During modeling of any intelligent system, a variety of attributes are normally used to ensure that all the necessary information is present, which not only increases the computational complexity but also reduces classification accuracies. In this study, widely used features of thermal images like first order histogram, statistical gray level co-occurrence matrix (GLCM) and component based features are considered. The novelty of the work is that the combination of data mining techniques and clustering quality of the data in the selected feature space helps to determine the best classifier independent feature set suitable for thermal monitoring. Interestingly it is found that maximum intensity; average intensity and skewness are identified as the best feature set. Based on experimental verification, it has been demonstrated that the selected feature set gives better classification accuracies than those using all the original features. Therefore, an effective feature selection method is able to greatly improve the performance of classifiers as well as reduce the computational cost.

T. Dutta (✉)

Department of Electrical Engineering, Future Institute of Technology, Kolkata, India
e-mail: tamaldutta95@yahoo.com

D. Santra · P. Chottopadhyay

Department of Electrical Engineering,
Indian Institute of Engineering Science and Technology Shibpur, Howrah, India
e-mail: deepjyotisantra@gmail.com

P. Chottopadhyay

e-mail: paramita_chottopadhyay@yahoo.com

C. Peng-Lim

Institute for Intelligent Systems Research & Innovation,
Deakin University, Geelong, VIC 3126, Australia
e-mail: chee.lim@deakin.edu.au

J. Sil

Department of Computer Science and Technology,
Indian Institute of Engineering Science and Technology Shibpur, Howrah, India
e-mail: js@cs.iiests.ac.in

© Springer Nature Singapore Pte Ltd. 2019

K. Deep et al. (eds.), *Decision Science in Action*, Asset Analytics,
https://doi.org/10.1007/978-981-13-0860-4_10

Keywords Condition monitoring · Thermal image · Feature extraction
Feature selection · Classification

1 Introduction

Excessive heating is one of the root causes pertaining to the majority failure of electrical equipment. In this domain, Infrared Thermography (IRT) plays a significant role in fault detection at the early stage. It is a non-contact and non-invasive temperature monitoring technique. Indeed, IRT of electrical equipment is gaining importance in the area of condition monitoring and fault diagnosis [1–7]. Thermo ionic radiations can be divided into four groups according to their wavelength, as shown in Fig. 1. For monitoring and diagnosis of electrical equipment, far-infrared (FIR) radiations are utilized [5, 8].

At present, intelligent digital image processing is important in various fields including thermal monitoring and fault diagnosis of electrical equipment. In any intelligent and machine learning based approach [9], feature extraction [10–12] and feature selection play an important role. In the literature, intelligent thermal monitoring techniques [10–15] have been investigated by many researches. Various feature selection methods have also been reported [12–17]. Each method has unique criteria to identify suitable features for solving image classification problems. In many papers [10–17], the selected feature sets have been validated using the performance of different classifiers, e.g. MLP, RBF, and SVM.

In this paper a classifier independent Davies-Bouldin (DB) Index is used to measure the feature quality; obtained from various feature selection techniques. The most suitable features are then proposed for thermal monitoring of electrical equipment using infra-red images. The schematic representation of the proposed method is shown in Fig. 2.

2 Acquisition of Thermal Images

For thermal image acquisition, a special camera is used to capture infrared radiation emitted from objects. An infrared camera detects infrared energy emitted from an

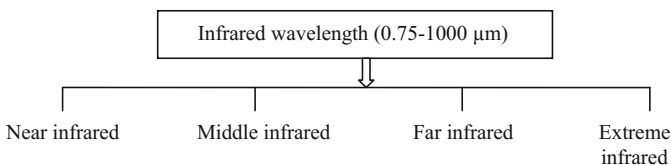


Fig. 1 Different types of infrared radiations

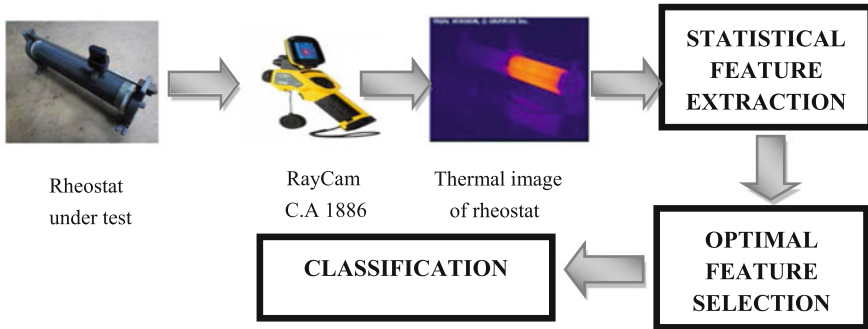


Fig. 2 Block diagram of the proposed method



Fig. 3 Different classes of thermal images of rheostat

object, converts it to temperature, and displays the image in terms of temperature distribution. There are several thermo graphic cameras which can detect radiation in the infrared range (roughly 0.1–1000 μm). In the present investigation, thermal images are taken by using RayCam C.A 1886 camera. The spectral range of the camera is 8–14 μm. Therefore, it can measure the FIR radiation of the object under study.

For thermal image acquisition purposes, electrical current through a rheostat is varied, and the images are taken at various levels of current flowing through it. Up to the rated current, the images are labeled as ‘Normal’, for a moderate value, it is labeled as ‘Incipient’, and ‘Faulty’ is for high a value of current. Some typical images of these three classes are presented in Fig. 3.

3 Statistical Features of Thermal Images

Thermal images have several features which can distinguish among various classes. Those features can be sub-divided into three categories, namely component based intensity features, first order histogram based features, and gray level co-occurrence matrix (GLCM) based features [10–12]. The image characteristic can be visualized

through its features. In the classification process of thermal images, statistical features of IR images are useful to classify the condition of electrical equipment.

3.1 Component Based Intensity Features

The following components based intensity features are extracted for the analysis of thermal images.

$$\text{Maximum intensity}(F1) = \max[I_c(i, j)]$$

$$\text{Minimum intensity}(F2) = \min[I_c(i, j)]$$

$$\text{Average intensity}(F3) = \frac{\sum_{i=1}^M \sum_{j=1}^N I_c(i, j)}{M * N | I_c \neq 0}$$

where $i = 1, 2, \dots, M$ and $j = 1, 2, \dots, N$ and $I_c(i, j)$ = segmented image with component only and the background as black.

Mean intensity is the average pixel value, which determines the brightness or darkness of the defined connected component. If the intensity values are arranged in ascending order, the middle value is defined as the median intensity value (F4).

3.2 First Order Histogram Based Intensity Features

First order histogram based intensity features are similar to component based features. The features are described as follows:

Average gray level intensity (F5)

For that image, the average gray level intensity can be represented by,

$$\text{Average gray intensity or mean} = \frac{\sum_{i=1}^M \sum_{j=1}^N I(i, j)}{M * N}$$

where $I(i, j)$; where $i = 1, \dots, M$ and $j = 1, \dots, N$, M & N are number of rows and columns respectively.

Standard deviation (F6)

It determines the deviation of pixel intensities and calculated as follows:

$$\text{Standard deviation} = \sqrt{\frac{\sum_{i=1}^M \sum_{j=1}^N [I(i, j) - \text{Mean}]^2}{M * N}}$$

Energy (F7)

Energy shows how the gray levels are distributed. When the number of gray level is low the energy is high.

$$\text{Energy} = \frac{\sum_{i=1}^M \sum_{j=1}^N I(i, j)^2}{M * N}$$

Entropy (F8)

It measures randomness of the input image. If the information content of an image is high, the image entropy becomes high. Similarly, if the information content is low, the entropy value is low.

$$\text{Entropy} = \frac{\sum_{i=1}^M \sum_{j=1}^N I(i, j) * [-\log I(i, j)]}{M * N}$$

Skewness (F9)

$$\text{Skewness} = \frac{\sum_{i=1}^M \sum_{j=1}^N [I(i, j) - \text{Mean}]^3}{M * N * \text{Standard deviation}^2}$$

Kurtosis (F10)

Kurtosis measures the peakness or flatness of the intensity distribution with respect to the normal distribution.

$$\text{Kurtosis} = \frac{\sum_{i=1}^M \sum_{j=1}^N [I(i, j) - \text{Mean}]^4}{M * N * \text{Standard deviation}^4}$$

3.3 Statistical Features Using Gray Level Co-occurrence Matrix (GLCM)

A GLCM is a square matrix which has an equal number of rows and columns as the number of gray levels in the image.

Contrast (F11)

It is a measure of the intensity contrast between a pixel and its neighbor over the whole image.

$$\text{Contrast} = \sum_{i,j} |i - j|^2 P(i, j)$$

Correlation (F12)

It is a measure of how correlated a pixel is to its neighbor over the whole image. Correlation is 1 or -1 for a perfect positively or negatively correlated image. Correlation is NaN (not a number) for a constant image.

Energy (F13)

Energy is the sum of square elements in the GLCM.

$$\text{Energy} = \sum_{i,j} P(i, j)^2$$

Homogeneity (F14)

It returns a value that measures the closeness of the distribution of elements in the GLCM with respect to the GLCM diagonal.

$$\text{Homogeneity} = \frac{P(i, j)}{1 + |i - j|}$$

where 1 = possible intensity level of the image.

4 Feature Selection

Feature selection is a one of the feature reduction technique widely used in data mining for improvement of the data quality and enhances the performance of classifiers. It works on the basis of an evaluation functions following evaluation strategies like ranking, sequential search, heuristic search. On the basis of evaluation function, the feature selection technique can be categorized into two parts like filter and wrapper based methods. In Filter methods, features are selected based on a performance measure whereas in Wrapper methods, the feature space is recognized using a particular classifier to measure the importance of a feature subset.

In literature, the quality of the data is assessed with the help of different validity indices. In this study, Davies–Bouldin index (DBI) is utilized to find the goodness of features. The DBI calculates the average of resemblance between two similar clusters. A lower DBI indicates condensed and separated clusters. In this investigation sequential feature selection (SFS), best first search (BFS), relief based feature selection (RBFS) and class based feature selection (CBFS) are used. The DB index is employed to determine the best feature space.

5 Results and Discussions

Different machine learning techniques like MLP, SVM, decision tree and KNN are applied to evaluate the quality of the original feature set and selected feature sub-set. The goodness of the feature sets are measured by the DBI. Separateness of data distribution in the selected feature space, with the DBI values in the best 3D feature spaces are shown in Fig. 4, where green denotes ‘Normal’, blue denotes ‘Incipient’, and red denotes ‘Faulty’.

A lower value of the DBI indicates the ability to identify the appropriate choice of feature selection technique, for recognition of three types of thermal images associ-

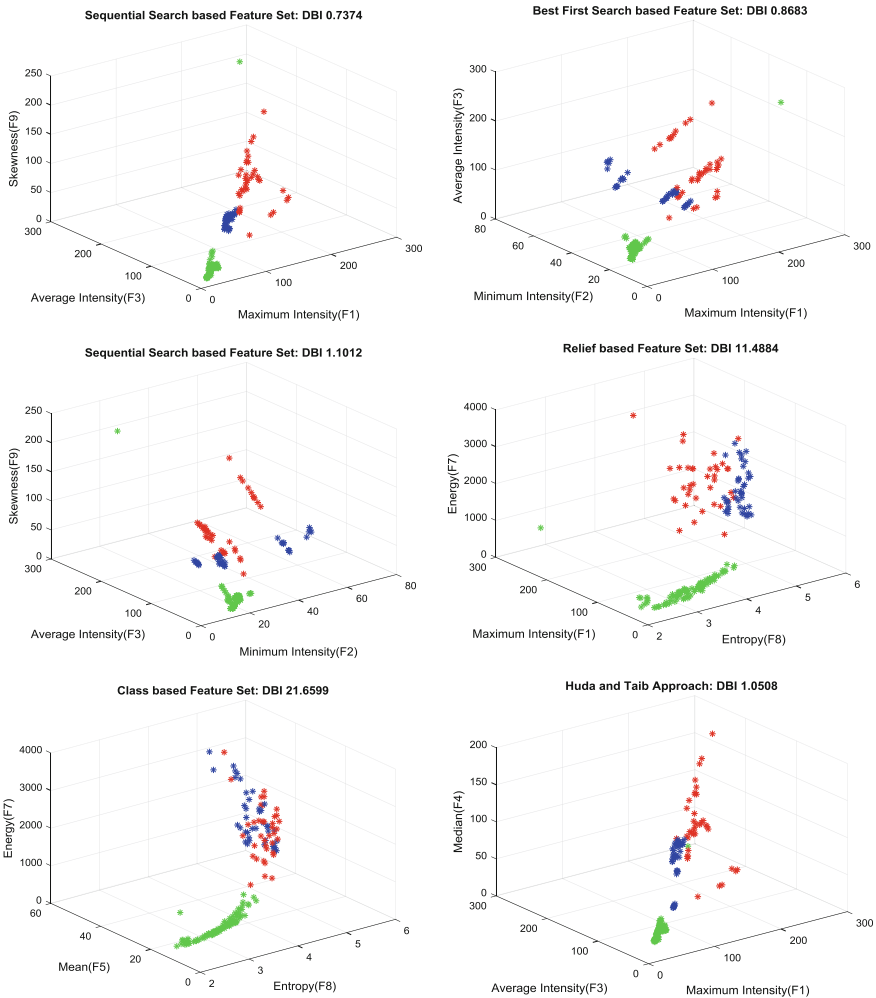


Fig. 4 Data distribution for 3D features using optimal feature subsets

Table 1 Selection of optimal feature subset using the DB index

Feature selection techniques	Feature space	DB index (DBI)
	All features	8.7662
Sequential search based feature selection (SSBFS)	F1, F3, F9	0.7374
Best first search (BFS)	F1, F2, F3	0.8683
Sequential search based feature selection (SSBFS)	F2, F3, F9	1.1012
Relief based feature selection (RBFS)	F8, F1, F7	11.4884
Class based feature selection (CBFS)	F8, F5, F7	21.6599
Huda and Taib approach [12]	F1, F3, F4	1.0508

ated with electrical equipment. Finally the sequential search based feature set (Maximum intensity (F1), Average intensity (F3) and Skewness (F9)), have the lowest DB value, therefore yielding the optimal feature subset. The optimal feature subset

Table 2 Results obtained using 10 fold cross validation using different classifiers

Feature selection techniques	Feature space	MLP	SVM	Decision tree	KNN
	All features	96.98	98.79	94.57	97.59
Sequential feature selection (SFS)	F1, F3, F9	98.19	96.38	96.98	98.19
Best first search (BFS)	F1, F2, F3	98.19	97.59	95.78	98.79
Sequential feature selection (SFS)	F2, F3, F9	98.19	98.19	96.98	98.79
Relief based feature selection (RBFS)	F8, F1, F7	97.59	96.38	95.78	97.59
Class based feature selection (CBFS)	F8, F5, F7	96.98	97.59	92.16	96.38
Huda and Taib approach [12]	F1, F3, F4	98.79	97.59	95.78	98.79

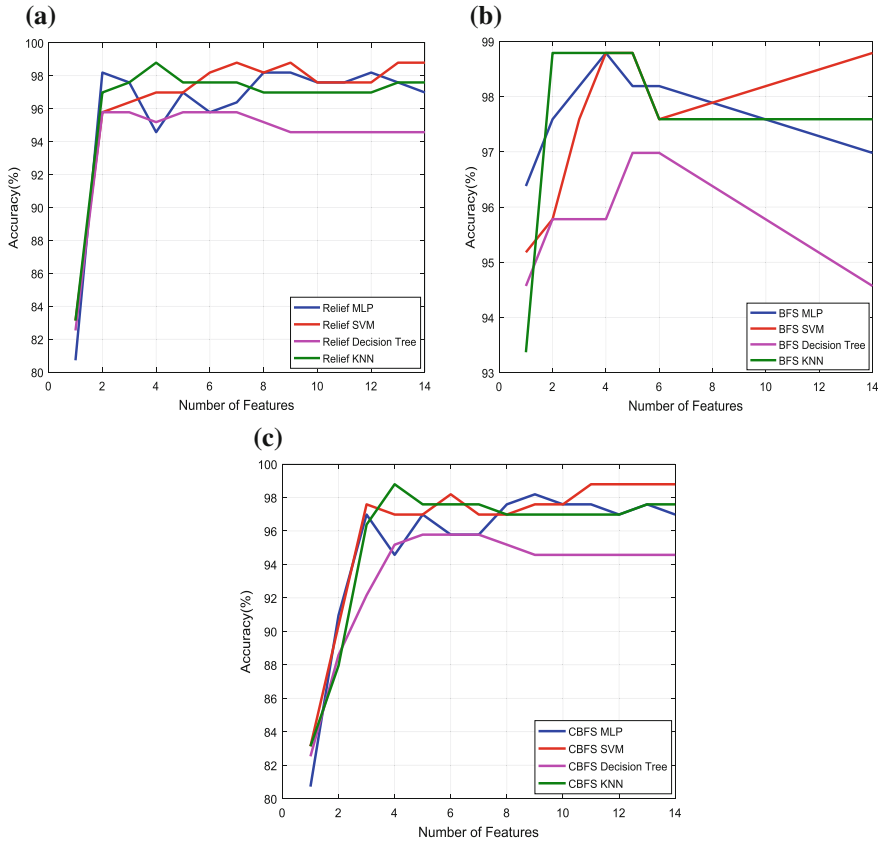


Fig. 5 Performance of different classifiers: using **a** Relief Based Feature Selection, **b** Best First Search (BFS) based Feature Selection and **c** Class Based Feature Selection (CBFS)

Table 3 Sequential feature selection: performance using different classifiers

Feature set	Classifiers			
	MLP	SVM	Decision tree	KNN
All features	96.98	98.79	94.57	97.59
F1, F3, F9	98.19	96.38	96.98	98.19
F2, F3, F9	98.19	98.19	96.98	98.79
F1, F3, F4 from Huda and Taib approach [12]	98.79	97.59	95.78	98.79

selected using the DB indexes are tabulated in Table 1. Classification accuracies obtained using MLP, SVM, Decision Tree and KNN are presented in Table 2.

The performance of each classifiers using different feature selection algorithms are depicted in Fig. 5. Accuracies obtained using different classifiers for 10 fold cross validation using the SFS method are shown in Table 3.

6 Conclusion

Statistical features of thermal images play a key role in image classification. However, irrelevant features make the system computationally exhaustive and produce poor classification accuracy. Feature selection techniques in accompanied with data mining offer an efficient solution to deal with these problems. This paper has demonstrated that the usefulness of the selected features from various feature selection techniques in the area of thermal monitoring of electrical equipment based on infrared thermal images. Among several validity indices, the DBI has been used in this work. The applicability of intelligent thermal monitoring with experimental verification has been demonstrated in this paper.

References

1. Han, Y., Song, Y.H.: Condition monitoring techniques for electrical equipment—a literature survey. *IEEE Trans. Power Deliv.* **18**(1), 4–13 (2003)
2. Siada, A.A., Islam, S.: A novel online technique to detect power transformer winding faults. *IEEE Trans. Power Deliv.* **27**(2), 849–857 (2012)
3. Nandi, S., Toliyat, H.A., Li, X.: Condition monitoring and fault diagnosis of electrical motors—a review. *IEEE Trans. Energy Convers.* **20**(4), 719–729 (2005)
4. Chan, W.L., So, A.T.P., Lai, L.L.: Three dimensional thermal imaging for power equipment monitoring. *IEE Proc. Gener. Transm. Distrib.* **147**(6), 355–360 (2000)
5. Yang, C., Xiao-ming, G., Qi, J.: Infrared technology in the fault diagnosis of substation equipment. In: *Proceedings on International Conference on Electricity Distribution*, pp. 1–6. China (2008)
6. Rodenas, M.J.P., Royo, R., Antonino-Daviu, J., Roger-Folch, J.: Use of infrared thermography for computation of heating curves and preliminary failure detection in induction motors. In: *Proceedings on International Conference on Electrical Machines*, pp. 525–531, IEEE, France (2012)
7. Eftekhari, M., Moallem, M., Sadri, S., Hsieh, M.F.: A novel indicator of stator winding inter-turn fault in induction motor using infrared thermal imaging. *J. Infrared Phys. Technol.* **61**, 330–336 (2013)
8. Cui, H., Xu, Y., Zeng, J., Tang, Z.: The methods in infrared thermal imaging diagnosis technology of power equipment. In: *Proceedings on 4th International Conference on Electronic Information and Emergency Communication*, pp. 246–251, Beijing (2013)
9. Konar, A.: *Computational Intelligence—Principals, Techniques and Applications*, 1st edn. Springer, Heidelberg (2005)
10. Albrechtsen, F.: Statistical texture measures computed from gray level co occurrence matrix. Technical Report, Department of Informatics, University of Oslo (2008)
11. Gadkari, D.: Image quality analysis using GLCM. Master of Science Thesis, College of Arts and Sciences, University of Central Florida (2004)
12. Huda, A.S.N., Taib, S.: Suitable features selection for monitoring thermal condition of electrical equipment using infrared thermography. *J. Infrared Phys. Technol.* **61**, 184–191 (2013)
13. Chaturvedi, D.K., Iqbal, M.S., Singh, M.P.: Intelligent health monitoring system for three phase induction motor using infrared thermal image. In: *Proceedings on International Conference on Energy Economics and Environment*, pp. 1–6, Portugal (2015)
14. Huda, A.S.N., Taib, S., Ghazali, K.H., Jadin, M.S.: A new thermographic NDT for condition monitoring of electrical components using ANN with confidence level analysis. *ISA Trans.* **53**(3), 717–724 (2014)

15. Huda, A.S.N., Taib, S., Jadin, M.S., Ishak, D.: A semi-automatic approach for thermographic inspection of electrical installations within buildings. *J. Energy Build.* **55**, 585–591 (2012)
16. Huda, A.S.N., Taib, S.: Application of infrared thermography for predictive/preventive maintenance of thermal defect in electrical equipment. *J. Appl. Thermal Eng.* **61**(2), 220–227 (2013)
17. Jadin, M.S., Taib, S., Ghazali, K.H.: Feature extraction and classification for detecting the thermal faults in electrical installations. *J. Measur.* **57**, 15–24 (2014)

Performance of Sine–Cosine Algorithm on Large-Scale Optimization Problems



Puneet Kumar Pal, Kusum Deep and Atulya K. Nagar

Abstract The focus of this paper is the recently proposed sine–cosine algorithm (Mirjalili, Knowl-Based Syst 96:120–1330, [23]) for nonlinear continuous function optimization. The purpose of this paper is to inspect the effect of the sine–cosine algorithm on solving large-scale optimization problems. For this purpose, the algorithm is implemented on five common scalable problems appearing in literature, namely, Ackley, Griewank, Rastrigin, Rosenbrock, and Sphere functions. The dimensions of these problems are varied from 100 to 1000, and results have been recorded for fixed 10,000 iterations. The results are presented in numerical and graphical form. These results indicate that sine–cosine algorithm is a powerful nature-inspired optimization algorithm for solving all of these problems, except Sphere and Rosenbrock functions. Furthermore, the applicability of this algorithm is demonstrated by solving a real-life problem, i.e., gear train design problem.

Keywords Sine–cosine algorithm · Large-scale problems · Gear train design problem · Optimization

1 Introduction

Numerous nature-inspired optimization techniques have emerged to solve nonlinear optimization problems. They are particularly well suited in situations where traditional computing techniques perform unsatisfactorily. The advantages of these

P. K. Pal · K. Deep (✉)

Department of Mathematics, Indian Institute of Technology Roorkee, Roorkee, Uttarakhand
247667, India
e-mail: kusumfma@iitr.ac.in

P. K. Pal

e-mail: puneetp02@gmail.com

A. K. Nagar

Liverpool Hope University, Hope Park, Liverpool L16 9JD, UK
e-mail: nagara@hope.ac.uk

methods are their ability to solve various standard or application-based problems successfully without any prior knowledge of the problem space. Moreover, these algorithms are more likely to obtain the global optima of a given problem. Continuity and/or differentiability of the objective functions and/or constraints is not needed for these algorithms. Also, they work on a randomly generated population of solutions instead of one solution. They are easy to be programmed and can be easily implemented on a computer.

Some of the most popular nature-inspired optimization techniques are genetic algorithms [1], particle swarm optimization [2], differential evolution [3], glow worm swarm optimization [4, 5], artificial bee colony optimization [6], spider monkey algorithm [7], ant colony optimization [8], bacterial foraging optimization algorithm [9], gravitational search algorithm [10], central force optimization [11, 12], harmony search algorithm [13], water drop algorithm [14], ant lion algorithm [15, 16], firework algorithms [17], teaching learning based optimization [18], water weed optimization [19], kidney-inspired optimization [20], and moth-flame optimization algorithm [15, 16]. An excellent review of nature-inspired optimization techniques is presented in [21, 22].

The focus of this paper is the Sine–Cosine Algorithm (SCA) [23]. The original paper performs an extensive experimentation on multimodal functions, unimodal functions, and fixed-size function as well as a number of engineering design problems. The problem size that has been considered is 20. The objective of this paper is to study the performance of SCA on large-scale optimization problems. With this in mind, a set of five benchmark functions have been selected and SCA is used to solve these five problems for problem size 100–1000.

The rest of the paper is organized as follows: In Sect. 2, the working of SCA algorithm is explained. Section 3 describes the five benchmark problems considered in this study. Section 4 explains the results in the form of numerical and graphical data. Section 5 describes a real-life problem. The conclusions are drawn in Sect. 6.

2 The Sine–Cosine Algorithm

This is a population-based optimization technique and starts the optimization process with a set of random solutions. This random set of solutions is evaluated repeatedly by an objective function and improved using the following mathematical model:

$$X_i^{t+1} = \begin{cases} X_i^t + r_1 \times \sin(r_2) \times |r_3 \cdot P_i^t - X_i^t|, & r_4 < 0.5 \\ X_i^t + r_1 \times \cos(r_2) \times |r_3 \cdot P_i^t - X_i^t|, & r_4 \geq 0.5 \end{cases} \quad (1)$$

where X_i^t is the position of the current solution in i th dimension at i th iteration, $r_1/r_2/r_3$ are the random numbers, P_i^t is the position of the destination point in i th dimension, and r_4 is a random number in $[0, 1]$. $| \cdot |$ indicates the absolute value. The parameter r_1 tells the movement direction. The parameter r_2 is the distance for the

Initialize a set of search agents or solutions X
Do
 Evaluate objective function value on each of the search agents
 Update the best solution obtained so far ($P = X'$)
 Update r_1, r_2, r_3 and r_4
 Update the position of search agents
 While ($t < \text{maximum number of iterations}$)
 Return the best solution obtained so far as the global optima.

Fig. 1 Pseudocode of the SCA algorithm

movement either toward or outward the destination. The parameter r_3 puts a random weight on the destination to stochastically emphasize ($r_3 > 1$) or deemphasize ($r_3 < 1$) the effect of destination in defining the distance and r_4 is to equally switch between the sine and cosine components. Due to the use of sine and cosine in this formulation, this algorithm is named Sine–Cosine Algorithm (SCA).

Due to the cyclic pattern of sine and cosine functions, a solution is repositioned around another solution and this can guarantee exploitation of the space defined between two solutions. For exploration, the solutions should be able to search outside the space between their corresponding destinations as well and this can be achieved by changing the range of the sine and cosine functions.

The random location either inside or outside is achieved by defining a random number for r_2 in $[0, 2\pi]$ in Eq. (1). Therefore, this mechanism guarantees exploration and exploitation of the search space, respectively. To balance exploration and exploitation, the range of sine and cosine in Eq. (1) is changed using the following equation:

$$r_1 = a - a \frac{t}{T}$$

where

t is the current iteration,

T is the maximum number of iterations, and

a is the constant.

The pseudocode of this algorithm is presented in Fig. 1.

3 Applications

The performance of SCA algorithm is evaluated on Ackley function, Griewank function, Rastrigin function, Rosenbrock function, and Sphere function. These functions have been selected particularly because these problems are conflicting in nature with respect to their landscapes and difficulty features and are difficult to solve by a single algorithm simultaneously. These are unimodal as well as multimodal functions.

They are scalable, and hence their complexity can be increased by increasing the dimension of the problems by the user.

i. Ackley Function

Due to exponential terms, the surface of this function has many local minima. Many search algorithms get trapped in local optima due to the dependency on gradient information, but a search strategy that analyzes a wider region will be able to escape the valley among the optima. Global minimum of Ackley test function is at $(0, 0, \dots, 0)$ with $f_{\min} = 0$. The mathematical form is given by

$$f(x) = 20 + e - 20e^{\left(-\frac{1}{5}\sqrt{\frac{1}{n}\sum_1^n x_i^2}\right)} - e^{\left(\frac{1}{n}\sum_{i=1}^n \cos(2\pi x_i)\right)}, \quad \text{for } x_i \in [-32, 32]$$

ii. Griewank Function

When dimension of Griewank function increases, its number of local minima increases. A multistart algorithm is able to determine global minimum of this function more easily when the dimension increases. The global minimum of this function is at $(0, 0, \dots, 0)$ with $f_{\min} = 0$.

$$f(x) = \sum_{i=1}^n \frac{x_i^2}{4000} - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1, \quad \text{for } x_i \in [-600, 600]$$

iii. Rastrigin Function

This test function is the extended form of sphere function with a modulator term $\alpha^* \cos(2\pi x_i)$. This has a great number of local minima solutions whose value increases with the distance to the global minima solutions. Its global minima solution is at $(0, 0, \dots, 0)$ with $f_{\min} = 0$

$$f(x) = 10n + \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i)) \quad \text{for } x_i \in [-5.12, 5.12]$$

iv. Rosenbrock Problem

Rosenbrock function or banana function is a unimodal, non-separable, and differentiable test problem. Its complexity lies due to nonlinear interaction between parameters. The global optimum of this function is inside a long narrow parabolic-shaped flat valley. The global minimum of this function is at $(1, 1, \dots, 1)$ with $f_{\min} = 0$.

$$f(x) = \sum_{i=1}^{n-1} (100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2), \quad \text{for } x_i \in [-30, 30]$$

v. **Sphere Problem**

Sphere function is a continuous, convex, and unimodal test problem. The function has numerous local minima, whereas there are only one global minima solution at $(0, 0, \dots, 0)$ with $f_{\min} = 0$.

$$f(x) = \sum_{i=1}^n x_i^2 \text{ for } x_i \in [-100, 100]$$

4 Results and Discussion on Benchmark Functions

As mentioned earlier, the main focus of presenting this paper is to calibrate the usefulness of implementing SCA on large-scale problems. Therefore, the abovementioned five well-known problems are chosen. For all the problems, 30 runs are performed. The computer code was run on an i3 processor and 4 GB RAM using MATLAB 2015a. The parameters of SCA are kept the same as proposed by Mirjalili [23]. For 30 runs, the minimum, maximum, average value, and standard deviation of the objective function value are recorded in Tables 1, 2, 3, 4, and 5 corresponding to dimension 100–1000 for fixed 10,000 iterations. These values are plotted using box plots for better visualization. The convergence curves for each dimension and for each function has been plotted to show the behavior as number of iterations increases. The tables for each function are given as follows:

Table 1 and boxplot for Ackley function represent that dimension does not affect the optimum values much. It changes a little only. This suggests that the problem is solvable using SCA. In Tables 2 and 3, the performance of SCA is observed on Rastrigin and Griewank functions. The boxplots for these functions with the tables suggest that the results are good. As the dimension increases, it would be matter of a worry a little. Table 4 represents the data of Rosenbrock function which clearly

Table 1 Average, standard deviation, maximum, minimum, and median objective function values over 30 independent runs for Ackley function

Dimension	Average	Std. dev	Max.	Min.	Median
100	1.72E+01	7.38E+00	2.06E+01	1.73E−03	2.05E+01
200	1.88E+01	4.80E+00	2.07E+01	3.34E+00	2.06E+01
300	1.78E+01	5.08E+00	2.08E+01	5.44E+00	2.07E+01
400	2.01E+01	2.33E+00	2.08E+01	1.15E+01	2.07E+01
500	1.94E+01	3.63E+00	2.08E+01	8.40E+00	2.08E+01
600	2.02E+01	2.90E+00	2.08E+01	4.87E+00	2.08E+01
700	1.96E+01	3.56E+00	2.08E+01	7.98E+00	2.08E+01
800	1.76E+01	4.92E+00	2.08E+01	8.35E+00	2.08E+01
900	1.86E+01	4.14E+00	2.08E+01	8.52E+00	2.08E+01
1000	1.92E+01	3.55E+00	2.08E+01	1.02E+01	2.08E+01

Table 2 Average, standard deviation, maximum, minimum, and median objective function values over 30 independent runs for Rastrigin function

Dimension	Average	Std. dev.	Max.	Min.	Median
100	7.54E+01	6.24E+01	1.89E+02	7.06E-07	6.68E+01
200	3.71E+02	1.55E+02	8.57E+02	1.65E+02	3.29E+02
300	5.87E+02	3.26E+02	1.30E+03	4.03E+01	4.74E+02
400	8.40E+02	3.56E+02	1.75E+03	3.42E+02	8.53E+02
500	1.16E+03	5.60E+02	2.20E+03	3.44E+02	1.09E+03
600	1.12E+03	4.73E+02	2.28E+03	3.59E+02	1.03E+03
700	1.26E+03	4.79E+02	2.43E+03	4.58E+02	1.23E+03
800	1.35E+03	5.98E+02	2.94E+03	3.62E+02	1.30E+03
900	1.60E+03	6.48E+02	3.17E+03	7.61E+02	1.44E+03
1000	1.72E+03	9.18E+02	4.31E+03	4.10E+02	1.59E+03

Table 3 Average, standard deviation, maximum, minimum, and median objective function values over 30 independent runs for Griewank function

Dimension	Average	Std. dev.	Max.	Min.	Median
100	3.80E-01	4.14E-01	1.31E+00	3.42E-09	1.70E-01
200	7.67E+01	7.11E+01	3.49E+02	5.66E+00	5.43E+01
300	2.37E+02	9.47E+01	4.58E+02	1.11E+02	2.24E+02
400	4.32E+02	1.51E+02	7.51E+02	1.14E+02	4.16E+02
500	6.75E+02	3.10E+02	1.37E+03	2.60E+02	5.94E+02
600	1.09E+03	3.43E+02	1.66E+03	5.29E+02	1.11E+03
700	1.35E+03	5.13E+02	2.28E+03	2.91E+01	1.43E+03
800	1.65E+03	5.46E+02	3.08E+03	8.08E+02	1.60E+03
900	2.00E+03	6.87E+02	3.67E+03	7.47E+02	1.91E+03
1000	2.61E+03	9.14E+02	4.23E+03	7.57E+02	2.78E+03

Table 4 Average, standard deviation, maximum, minimum, and median objective function values over 30 independent runs for Rosenbrock function

Dimension	Average	Std. dev.	Max.	Min.	Median
100	4.60E+05	8.10E+05	3.58E+06	7.95E+02	1.04E+05
200	6.81E+07	3.76E+07	2.01E+08	7.38E+05	6.33E+07
300	2.98E+08	8.82E+07	5.00E+08	1.64E+08	3.10E+08
400	4.53E+08	1.04E+08	7.08E+08	2.63E+08	4.30E+08
500	8.14E+08	1.31E+08	1.06E+09	5.65E+08	8.15E+08
600	1.15E+09	2.11E+08	1.45E+09	6.79E+08	1.18E+09
700	1.32E+09	2.32E+08	1.65E+09	7.53E+08	1.36E+09
800	1.54E+09	2.38E+08	1.98E+09	1.08E+09	1.52E+09
900	1.81E+09	3.50E+08	2.43E+09	1.09E+09	1.79E+09
1000	2.18E+09	4.61E+08	2.95E+09	1.31E+09	2.25E+09

Table 5 Average, standard deviation, maximum, minimum, and median objective function values over 30 independent runs for Sphere function

Dimension	Average	Std. dev.	Max.	Min.	Median
200	6.94E+03	5.74E+03	2.48E+04	3.81E+02	4.93E+03
300	3.00E+04	1.40E+04	6.21E+04	7.69E+03	2.81E+04
400	5.71E+04	2.51E+04	1.07E+05	2.98E+04	4.83E+04
500	7.59E+04	3.53E+04	1.52E+05	2.25E+04	6.60E+04
600	1.06E+05	4.00E+04	1.80E+05	4.13E+04	1.02E+05
700	1.50E+05	7.29E+04	3.68E+05	4.62E+04	1.38E+05
800	2.25E+05	6.32E+04	3.84E+05	1.32E+05	2.23E+05
900	2.37E+05	8.17E+04	3.79E+05	1.09E+05	2.35E+05
1000	2.76E+05	8.80E+04	4.69E+05	8.85E+04	2.62E+05

says that SCA cannot perform good for this function. So, this problem is not solvable using SCA. Table 5 is for Sphere function. For this case, as the dimension increases the results get poor.

Figure 2 represents the boxplots for different functions. Figures 3, 4, 5, 6, and 7 represent the convergence curves for the functions. These figures give a better insight into the performance of SCA.

5 Gear Train Design Problem

The term gear ratio is used interchangeably with velocity ratio. For a pair of matching gears, the velocity or gear ratio N is given by

$$N = \left| \frac{\omega_o}{\omega_i} \right| = \frac{t_i}{t_o}$$

where ω_o is the angular velocity of the output shaft and ω_i is the angular velocity of the input shaft. The ratio is, thus, inversely proportional to the number of teeth on the input and output gears.

Now consider the gear train and it is desired to produce a gear ratio as close as possible to 1/6.931. The gear ratio for the gear train may be written as

$$n = 1/6.931 = \frac{T_d T_b}{T_a T_f}$$

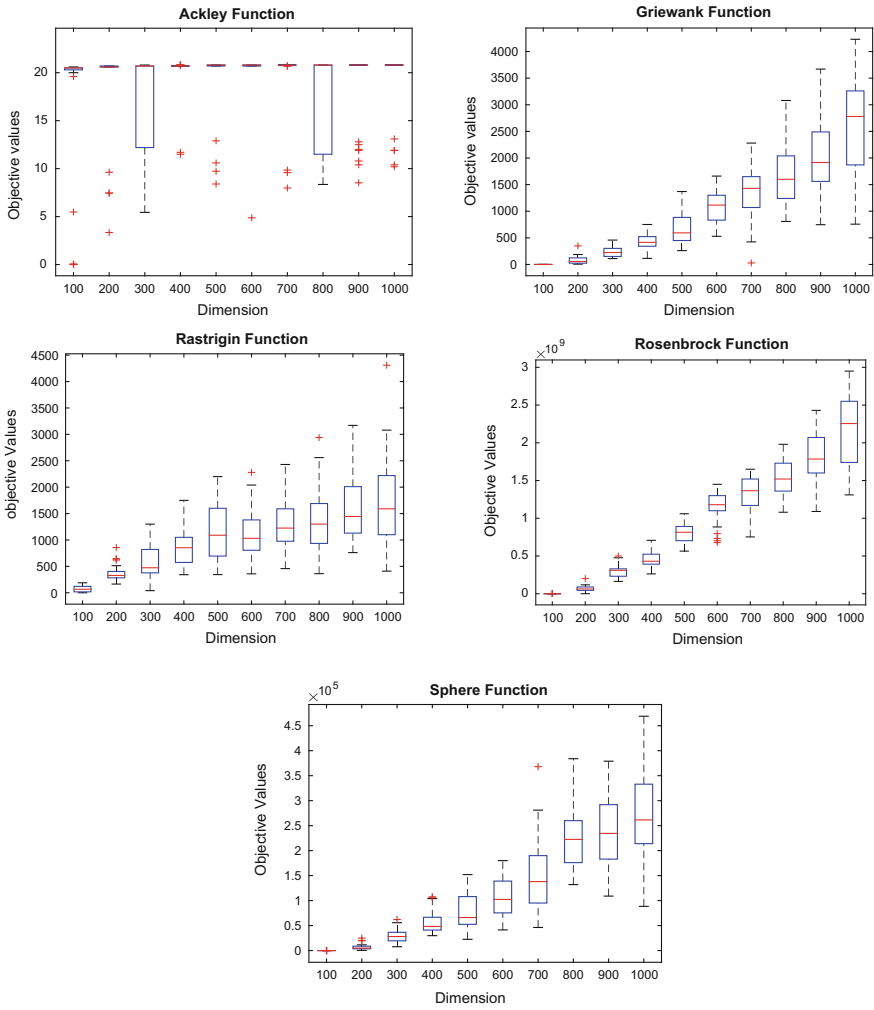


Fig. 2 Boxplots showing average, standard deviation, maximum, minimum, and median objective function values over 30 independent runs for all five functions

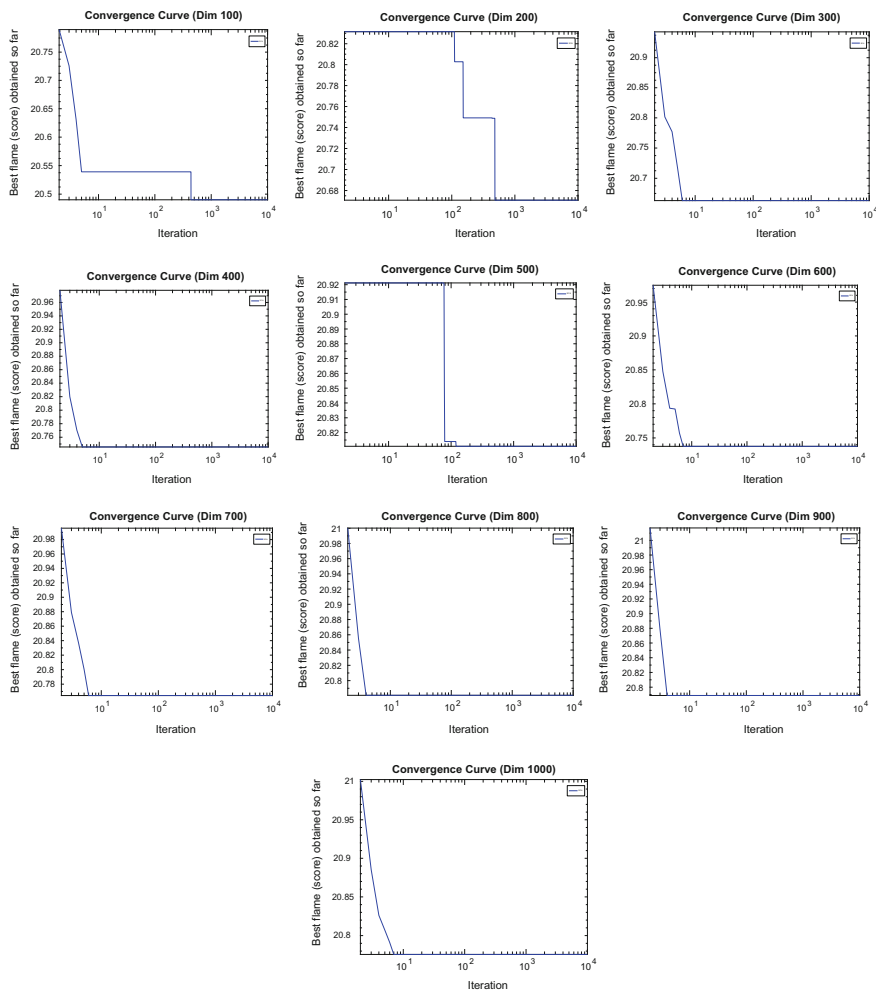


Fig. 3 Convergence curves for Ackley function over dimension 100–1000

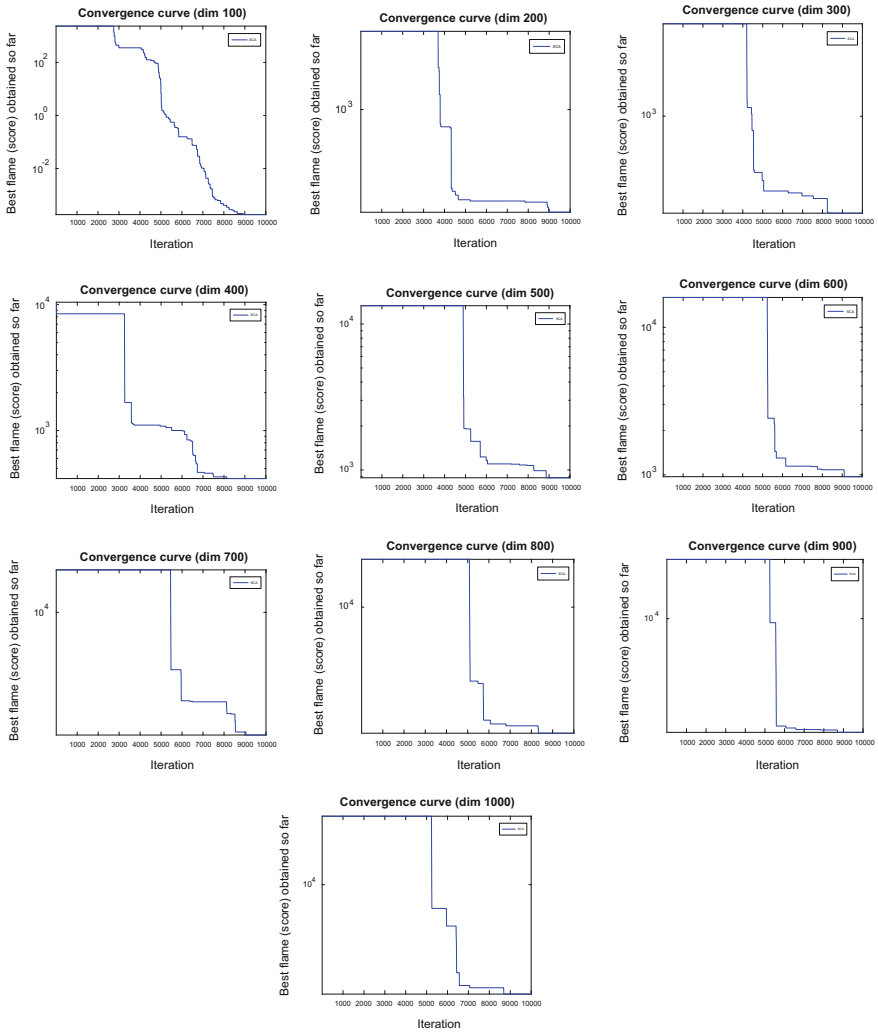


Fig. 4 Convergence curves for Griewank function over dimension 100–1000

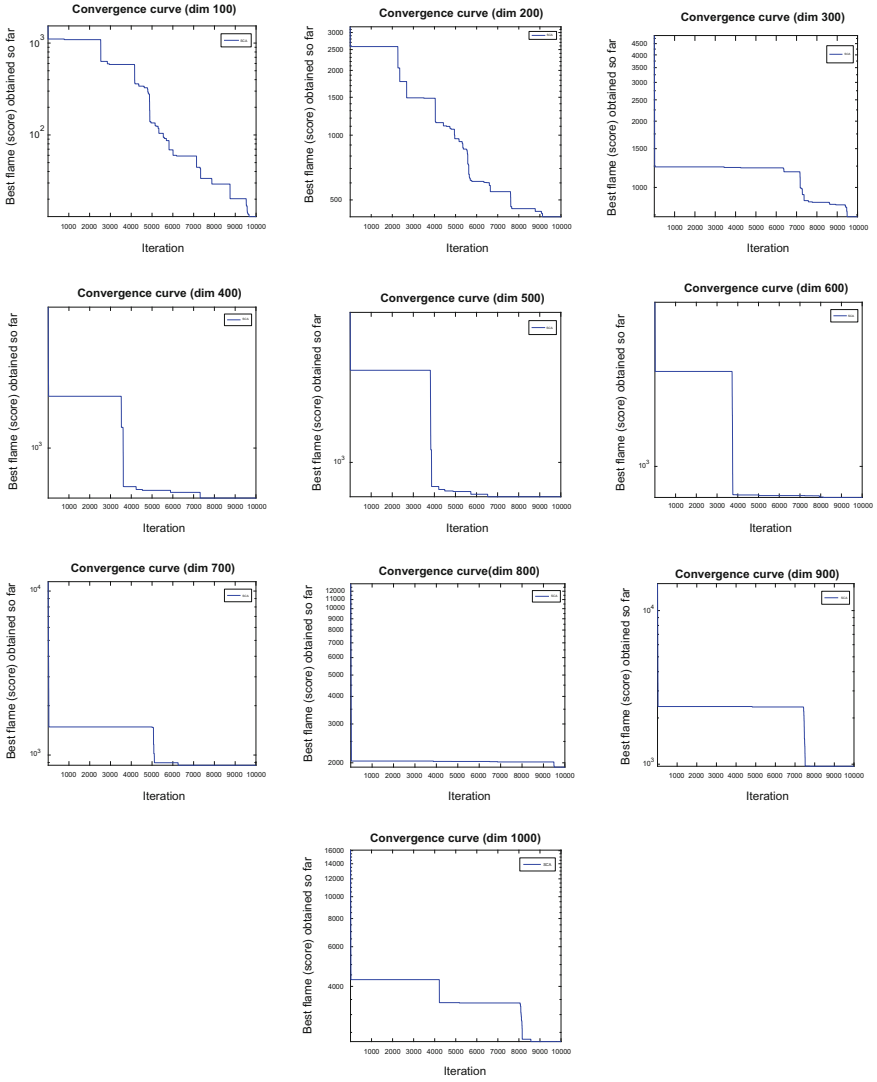


Fig. 5 Convergence curves for Rastrigin problem over dimension 100–1000

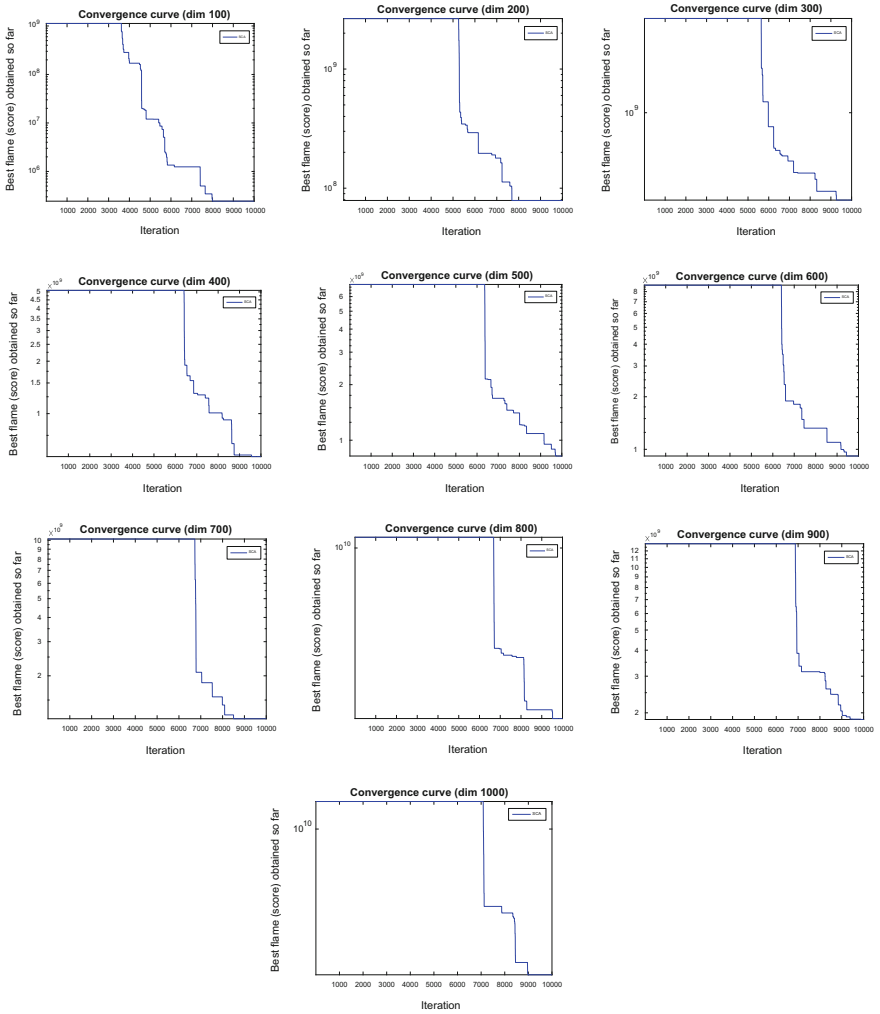


Fig. 6 Convergence curve for Rosenbrock problem over dimension 100–1000

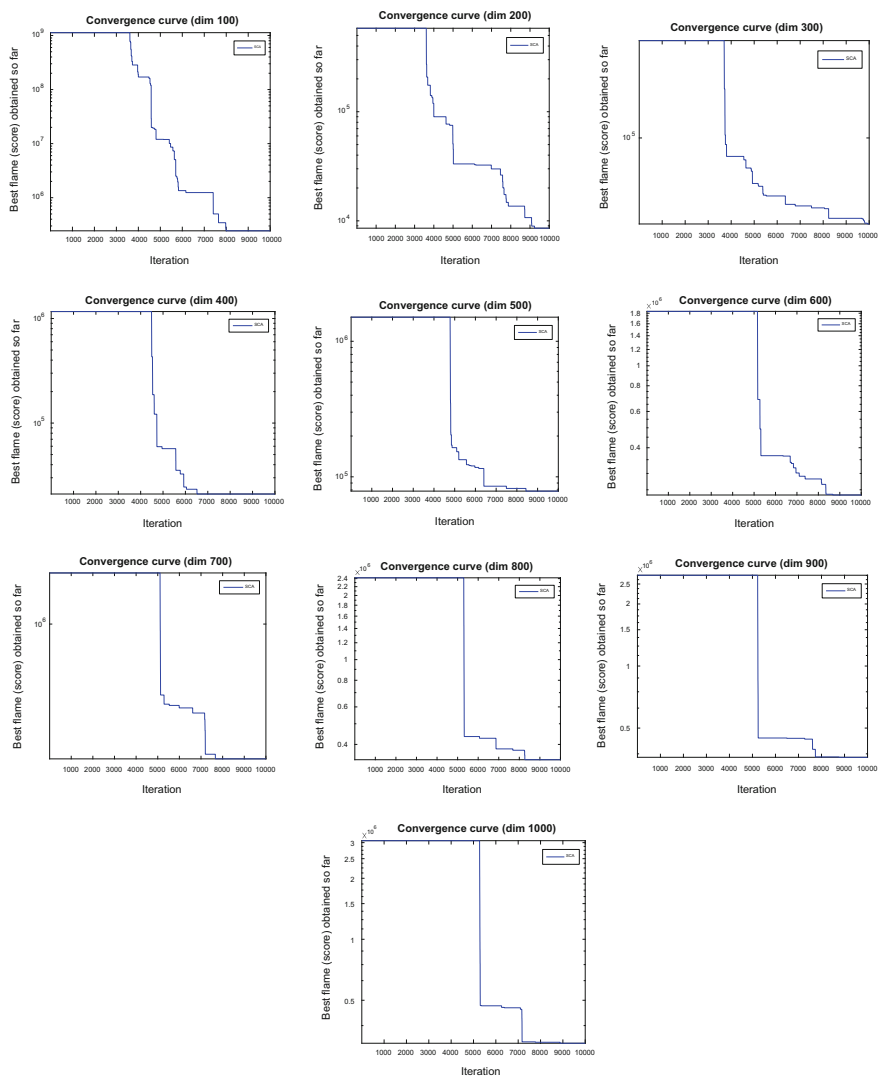


Fig. 7 Convergence curves for Sphere function over dimension 100–1000

Table 6 Objective function values of gear train design problem

Dimension	4
Average	3.57E-13
Std dev.	4.71617E-13
Max.	1.79E-12
Min.	4.59E-16

where T_d, T_b, T_a, T_f are the number of teeth on gears $a, b, d, \text{ and } f$, respectively. If we let the design variables be the number of teeth on each gear, we have the variables in standard form and

$$\vec{X} = [X_1, X_2, X_3, X_4]^T = [T_d, T_b, T_a, T_f]^T \text{ where } 12 \leq X_1, X_2, X_3, X_4 \leq 60$$

If we minimize the square of the difference between the desired gear ratio and the current design gear ratio, the objective function can be expressed as

$$\text{Min}(X) = \left(\frac{1}{6.931} - \frac{T_d T_b}{T_a T_f} \right)^2 = \left(\frac{1}{6.931} - \frac{X_1 X_2}{X_3 X_4} \right)^2.$$

For this particular case, the problem is unconstrained in nature. The solution generated is listed in Table 6.

X_1	X_2	X_3	X_4
12	12	35.1342	28.4071

From the table of gear train design problem, it is clear that the problem is solvable using the SCA (Fig. 8).

6 Conclusion

In this paper, the sine-cosine algorithm is studied for the problems of large dimensions. Any algorithm cannot solve all the problems. In this work, it is shown that not all five benchmark problems are solvable using this sine-cosine algorithm. A set of five well-known benchmark functions is taken, and SCA code is run 30 times for problem size 100-1000. The maximum number of iterations is 10,000. Based on the presentation of numerical and graphical results, it is concluded that SCA could easily solve all problems up to problem size 1000, except Rosenbrock and Sphere functions for large dimensions. Further, the gear ratio of a real challenging problem is optimized using SCA. It is therefore recommended to think of means to modify or hybridize SCA so that it can solve Rosenbrock and Sphere functions for large dimensions.

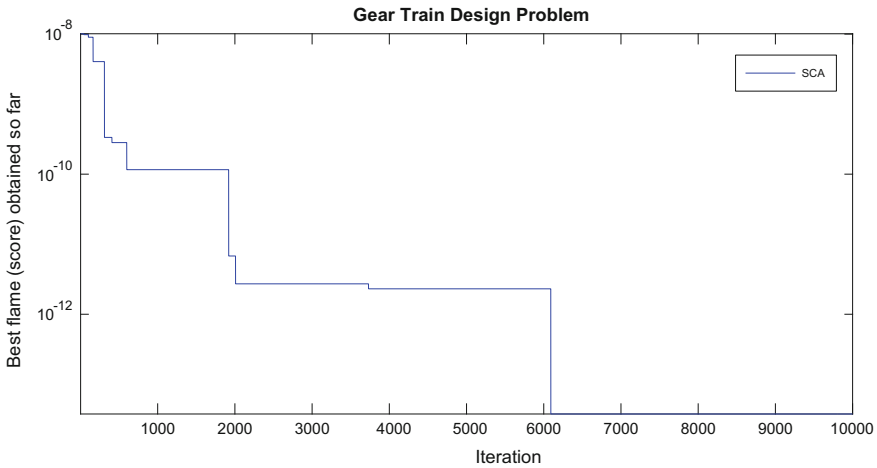


Fig. 8 Convergence curve for gear train design problem

References

1. Holland, J.H.: *Adaptation in Natural and Artificial System*. University of Michigan Press, Ann Arbor (1975)
2. Kennedy, J., Eberhart, R.C.: Particle swarm optimization. In: *Proceedings of IEEE International Conference Neural Networks*, vol. 4, 1942–1948
3. Storn, R., Price, K.: Differential evolution—a simple and efficient adaptive scheme for global optimization over continuous spaces. Technical Report TR-95-012, Berkeley, CA (1995)
4. Krishnanand, K.N., Ghose, D.: Glowworm swarm based optimization algorithm for multimodal functions with collective robotics applications. *Multiagent Grid Syst.* **2**(3), 209–222 (2006)
5. Krishnanand, K.N., Ghose, D.: Glowworm swarm optimization: a new method for optimizing multi-modal functions. *Int. J. Comput. Intell. Stud.* **1**(1), 93–119 (2009)
6. Karaboga, D.: An idea based on honey bee swarm for numerical optimization. Technical Report-TR06, Erciyes University, Engineering Faculty, Computer Engineering Department (2005)
7. Bansal, J.C., Sharma, H., Clerc, M.: Spider monkey optimization algorithm for numerical optimization. *Memetic Comput.* **6**(1), 31–47 (2014)
8. Dorigo, M., Maniezzo, V., Colomni, A.: The ant system: an autocatalytic optimizing process. Technical Report TR91-016, Politecnico di Milano (1991)
9. Passino, K.M.: Biomimicry of bacterial foraging for distributed optimization and control. *IEEE Control Syst. Mag.* **22**, 52–67 (2002)
10. Rashedi, E., Nezam Abadi-Pour, H., Saryazdi, S.: GSA: a gravitational search algorithm. *Inf. Sci.* **179**(13), 2232–2248 (2009)
11. Formato, R.A.: Central force optimization: a new nature inspired computational framework for multidimensional search and optimization. In: *Proceedings of Nature Inspired Cooperative Strategies for Optimization*, 8–10 Nov 2007, vol. 129, pp. 221–238. Acireale, Sicily, Italy (2008)
12. Formato, R.A.: Central force optimization: a new deterministic gradient-like optimization meta-heuristic. *Opsearch* **46**(1), 25–51 (2009)
13. Geem, Z.W., Kim, J.H., Loganathan, G.V.: A new heuristic optimization algorithm: harmony search. *Simulation* **76**, 60–68 (2001)
14. Shah-Hosseini, H.: Optimization with the nature-inspired intelligent water drops algorithm. *Int. J. Intell. Comput. Cybern.* **1**(2), 193–212 (2008)

15. Mirjalili, S.: Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm. *Knowl.-Based Syst.* **89**, 228–249 (2015)
16. Mirjalili, S.: The ant lion optimizer. *Adv. Eng. Softw.* **83**, 80–98 (2015)
17. Tan, Y., Zhu, Y.L.: Fireworks algorithm for optimization. In: *Proceeding of International Conference in Swarm Intelligence*, pp. 355–364. Springer, Berlin Heidelberg (2010)
18. Rao, R.V., Savsani, V.J., Vakharia, D.P.: Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems. *Comput. Aided Des.* **43**(3), 303–315 (2011)
19. Mehrabian, A.R., Lucas, C.: A novel numerical optimization algorithm inspired from weed colonization. *Ecol. Inf.* **1**(4), 355–366 (2006)
20. Jaddi, N.S., Alvankarian, J., Abdullah, S.: Kidney-inspired algorithm for optimization problems. *Commun. Nonlinear Sci. Numer. Simul.* **42**, 358–369 (2016)
21. Kar, A.K.: Bio inspired computing—A review of algorithms and scope of applications. *Expert Syst. Appl.* **59**, 20–32 (2016)
22. Yang, X.S.: *Nature-Inspired Optimization Algorithms*, 1st edn., p. 263. Elsevier (2014)
23. Mirjalili, S.: SCA: a sine cosine algorithm for solving optimization problems. *Knowl.-Based Syst.* **96**, 120–1330 (2016)
24. Geem, Z.W., Kim, J.H., Loganathan, G.V.: Harmony search optimization: application to pipe network design. *Int. J. Model. Simul.* **22**(2), 125–133 (2002)
25. Mirjalili, S., Mirjalili, S.M., Lewis, A.: Grey wolf optimizer. *Adv. Eng. Softw.* **69**, 46–61 (2014)
26. Bäck, T.: *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms*, p. 328. Oxford University Press, Oxford (1995). ISBN 0-19-509971-0
27. Dixon, L.C.W., Szego, G.P.: The global optimization problem: an introduction towards global optimization **2**, 1–15 (1978)
28. Eberhart, R.C., Kennedy, J.: A new optimizer using particle swarm theory. In: *Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, pp. 39–43 (1995)
29. Gandomi, A.H.: Interior search algorithm (ISA): a novel approach for global optimization. *ISA Trans.* **53**(4), 1168–1183 (2014)
30. Holland, J.H., Reitman, J.S.: Cognitive systems based on adaptive algorithms. *ACM SIGART Bull.* **63**, 49 (1977)
31. Crepinsek, M., Liu, S.-H., Mernik, M.: Exploration and exploitation in evolutionary algorithms a survey. *ACM Comput. Surv.* **45**, 35 (2013)
32. Picheny, V., Wagner, T., Ginsbourger, D.: A benchmark of kriging-based infill criteria for noisy optimization (2012)
33. Rosenbrock, H.H.: An automatic method for finding the greatest or least value of a function. *Comput. J.* **3**, 175–184 (1960)
34. Sandgren, E.: Nonlinear integer and discrete programming in mechanical design optimization. *J. Mech. Des.* **112**(2), 223–229 (1990)

Necessary and Sufficient Optimality Conditions for Fractional Interval-Valued Optimization Problems



Indira P. Debnath and S. K. Gupta

Abstract In this paper, we consider the class of fractional interval-valued programming problems. Utilizing the concept of LU optimal solution, the solution concepts of such type of problems have been discussed. Further, the Fritz John and KKT optimality conditions for the nondifferentiable fractional interval-valued functions have also been established.

Keywords Fractional problem · Interval-valued problem · LU optimal solution
KKT conditions · Fritz John conditions

1 Introduction

In recent years, several researchers have contributed in the development of interval optimization problems. In general optimization problems, the parameters involved in the objective function and the constraints functions are real numbers. However, in real-world problems, the data involves much indistinctness, which makes the parameters involved in the objectives and the constraints uncertain. Interval optimization problems deal with such problems where the uncertain parameters are represented in the form of closed intervals both in the objective and the constraint set of functions.

In dealing with a multiobjective fractional programming problem, the parametric approach or some kind of transformation is utilized ending up with an equivalent multiobjective programming problem [1]. Motivated by this fact, the class of nondifferentiable fractional interval optimization problem has been studied in this paper, which is further reduced to equivalent bi-objective nondifferentiable fractional

I. P. Debnath (✉)

Faculty of Applied Sciences, The Northcap University, Gurugram 122017,
Haryana, India
e-mail: idmath26@gmail.com

S. K. Gupta

Department of Mathematics, Indian Institute of Technology Roorkee,
Roorkee 247667, India
e-mail: skgiitr@gmail.com

interval optimization problems. In literature, numerous work is available which develops the optimality conditions and duality relations in interval optimization problems, few of them are [2–5]. In the meantime, differentiable interval-valued programming problems have been an interesting topic of research. Taking the objective functions as interval- and real-valued constraints, both differentiable, Jayswal et al. [6] developed the conditions for optimality and duality relations for such problems. By constructing the interval-valued Lagrangian function, Wu [7] established Lagrangian interval-valued duality results. The duality relations between an interval optimization problem and its Wolfe’s dual has been explored by Wu [8]. The KKT conditions for single-objective differentiable interval-valued problem has been obtained by Wu [9] which further has been extended to the multiobjective case in Wu [10]. Further, Osuna-Gmez et al. [11] developed new necessary and sufficient efficiency conditions for multiobjective interval-valued problems under new generalized convexity notions. On the other hand, Antczak [12] derived the Fritz John and KKT necessary optimality conditions for a nonsmooth multiobjective interval-valued optimization problem and established duality results for a multiobjective Mond–Weir dual problem under convexity assumptions. Very recently, Ghosh [13] proposed a Newton method to obtain efficient solutions for the optimization problems with interval-valued objective functions using the generalized Hukuhara differentiability of multivariable interval-valued functions.

If any of the interval objective function or the constraint function or both are not differentiable, the problem is called nondifferentiable interval optimization problems. Recently, researchers have started exploring in the field of nondifferentiable interval optimization problems. Using the concept of LU optimality, Sun and Wang [14] studied the nondifferentiable interval optimization problems having real-valued constraints. Recently, sufficient optimality conditions for a feasible point to be LU optimal for a nonsmooth interval optimization problems under invexity assumptions have been studied by Jayswal et al. [15]. Further, Wolfe and Mond–Weir type duality and the saddle point optimality conditions have also been discussed. Using the concept of convexifactors in interval optimization problems, sufficient optimality conditions and duality results have been developed in Jayswal et al. [16] under generalized δ^* -convexity assumptions. Eventually, Bhurjee and Panda [17] investigated the necessary and sufficient optimality conditions and the duality results for an interval optimization problem having both the objectives and the constraints nondifferentiable. To the authors’s knowledge, no results of sufficient optimality conditions have yet been available in the literature for the nondifferentiable fractional interval optimization problems. Henceforth, the main focus of our work is to explore the conditions for optimality for a nondifferentiable fractional interval-valued problem.

2 Problem Formulation and Preliminaries

Let X be a real vector space which is also locally convex having dual space X' and let the open convex set $S \subseteq X$.

Consider the nondifferentiable fractional interval-valued problem as

$$\text{Min} \frac{[f^L(x), f^U(x)]}{[g^L(x), g^U(x)]}$$

s. t.

$$h_i(x) \leq 0, i = 1, 2, \dots, m,$$

$$x \in S,$$

which further reduces to the problem

$$\text{Min} \left[\frac{f^L(x)}{g^U(x)}, \frac{f^U(x)}{g^L(x)} \right]$$

s. t.

$$h_i(x) \leq 0, i = 1, 2, \dots, m,$$

$$x \in S,$$

where

- (i) $\frac{f^L}{g^U}, \frac{f^U}{g^L} : X \rightarrow R, f^L(x), f^U(x) \geq 0$ are continuous functions and convex, $g^L(x), g^U(x) > 0$ are continuous functions and concave and $x \in S$,
- (ii) $h_i : X \rightarrow R, h_i(x), x \in X$ are continuous functions, also convex for $i = 1, 2 \dots m$.

Set $f^L = p^L, g^U = q^L, f^U = p^U$, and $g^L = q^U$. The above problem reduces to

$$\text{(NIVP) Min} \left[\frac{p^L}{q^L}(x), \frac{p^U}{q^U}(x) \right]$$

s. t.

$$h_i(x) \leq 0, i = 1, 2, \dots, m,$$

$$x \in S.$$

Let χ be the feasible set for the problem (NIVP).

Before we proceed further, some preliminary concepts of the operations on intervals need to be discussed:

Let $\frac{A}{B} = \left[\frac{a^L}{b^L}, \frac{a^U}{b^U} \right]$ and $\frac{C}{D} = \left[\frac{c^L}{d^L}, \frac{c^U}{d^U} \right]$ be two fractional closed intervals with $\frac{a^L}{b^L} \leq \frac{a^U}{b^U}$, and $\frac{c^L}{d^L} \leq \frac{c^U}{d^U}$, $b^L, b^U, d^L, d^U \neq 0$.

$$(i) \frac{A}{B} + \frac{C}{D} = \left[\frac{a^L}{b^L} + \frac{c^L}{d^L}, \frac{a^U}{b^U} + \frac{c^U}{d^U} \right],$$

$$(ii) \quad -\frac{A}{B} = \left[-\frac{a^U}{b^U}, -\frac{a^L}{b^L} \right],$$

$$(iii) \quad \frac{A}{B} - \frac{C}{D} = \frac{A}{B} + \left(-\frac{C}{D} \right) = \left[\frac{a^L}{b^L} - \frac{c^U}{d^U}, \frac{a^U}{b^U} - \frac{c^L}{d^L} \right]$$

$$(iv) \quad \beta \left(\frac{A}{B} \right) = \begin{cases} \left[\frac{a^L}{b^L}, \frac{a^U}{b^U} \right] & \text{if } \beta \geq 0, \\ \left[\frac{a^U}{b^U}, \frac{a^L}{b^L} \right] & \text{if } \beta < 0. \end{cases}$$

The ordering relation between two intervals $\frac{A}{B}$ and $\frac{C}{D}$ are defined as

$$(i) \quad \frac{A}{B} \leq_{LU} \frac{C}{D} \text{ iff } \frac{a^L}{b^L} \leq \frac{c^L}{d^L} \text{ and } \frac{a^U}{b^U} \leq \frac{c^U}{d^U}.$$

$$(ii) \quad \frac{A}{B} <_{LU} \frac{C}{D} \text{ iff } \frac{A}{B} \leq_{LU} \frac{C}{D} \text{ and } \frac{A}{B} \neq \frac{C}{D}, \text{ equivalently,}$$

$$\begin{cases} \frac{a^L}{b^L} < \frac{c^L}{d^L} \\ \frac{a^U}{b^U} \leq \frac{c^U}{d^U}, \end{cases} \quad \text{or} \quad \begin{cases} \frac{a^L}{b^L} \leq \frac{c^L}{d^L} \\ \frac{a^U}{b^U} < \frac{c^U}{d^U}, \end{cases} \quad \text{or} \quad \begin{cases} \frac{a^L}{b^L} < \frac{c^L}{d^L} \\ \frac{a^U}{b^U} < \frac{c^U}{d^U}, \end{cases}$$

Now, we will be using the following definitions throughout the paper.

Definition 2.1 [14] For the set S , the normal cone at $x_0 \in S$ is defined as

$$N_S(x_0) = \{z \in X' | (x - x_0)^T z \leq 0, \forall x \in S\}.$$

Definition 2.2 [14] $\eta \in X'$ will be called as the subgradient of a convex function g at a point $x_0 \in X$, if

$$g(x) - g(x_0) \geq (x - x_0)^T \eta, \forall x \in X.$$

Definition 2.3 $\eta \in X'$ will be called as the subgradient of a strictly convex function g at $x_0 \in X$, if $\forall x \in X$,

$$g(x) - g(x_0) > (x - x_0)^T \eta, x \neq x_0.$$

The set of all the subgradients of ψ at x_0 is said to be the subdifferential of ψ at x_0 and denoted by $\partial\psi(x_0)$.

Definition 2.4 [18] At $x_0 \in X$, a functional ψ will be called as quasidifferentiable, if there exists $c^+\psi(x_0; h)$ and some weak* convex set $P(x_0) \subseteq X'$, which is also closed, such that

$$c^+\psi(x_0; h) = \max_{x^* \in P(x_0)} h^T x^*, h \in X.$$

The set $P(x_0)$ is said to be quasidifferential.

Remark 2.1 If ψ is a continuous and a convex function at x_0 , then the above equation gives $P(x_0) = \partial\psi(x_0)$.

The following proposition for a single-objective function, given by Borwein [19], leads an efficient role in our further discussion.

Proposition 2.1 *Assume that $0 \leq \varphi_1 : X \rightarrow R$ is convex and $0 < \varphi_2 : X \rightarrow R$ is concave at x_0 , then $\delta(x) = \frac{\varphi_1}{\varphi_2}$ is quasidifferential at x_0 and*

$$P(x_0) = \frac{1}{\varphi_2(x_0)} \left[\partial\varphi_1(x_0) - \delta(x_0)\partial\varphi_2(x_0) \right],$$

where $\partial\varphi_2(x_0)$ is the subdifferential of φ_2 at x_0 .

Definition 2.5 A feasible point \bar{x} is said to be an LU optimal solution for the problem (NIVP) if there does not exist any $x \in \chi$ such that

$$\left[\frac{p^L}{q^L}(x), \frac{p^U}{q^U}(x) \right] <_{LU} \left[\frac{p^L}{q^L}(\bar{x}), \frac{p^U}{q^U}(\bar{x}) \right].$$

Consider two independent fractional problems as given below:

$$(FP_1) \text{ Min } \frac{p^L}{q^L}(x)$$

subject to

$$p^U(x) \leq \omega^U(\hat{x})q^U(x)$$

$$h_i(x) \leq 0, i = 1, 2, \dots, m$$

$$x \in S$$

and

$$(FP_2) \text{ Min } \frac{p^U}{q^U}(x)$$

subject to

$$p^L(x) \leq \omega^L(\hat{x})q^L(x)$$

$$h_i(x) \leq 0, i = 1, 2, \dots, m$$

$$x \in S,$$

where

$$\omega^L(\hat{x}) = \frac{p^L}{q^L}(\hat{x}) \text{ and } \omega^U(\hat{x}) = \frac{p^U}{q^U}(\hat{x}).$$

The following lemma connects the problem (NIVP) and the problems (FP_1) and (FP_2) .

Lemma 2.1 \hat{x} is an LU optimal solution of the problem (NIVP) iff \hat{x} is the optimal solution for the problems (FP_1) and (FP_2) .

Proof Let \hat{x} be optimal for the problems (FP_1) and (FP_2) . On the contrary, suppose that \hat{x} is not LU optimal for the problem (NIVP).

Therefore, \exists an $x \in \chi$ in such a way that

$$\left[\frac{p^L}{q^L}(x), \frac{p^U}{q^U}(x) \right] <_{LU} \left[\frac{p^L}{q^L}(\hat{x}), \frac{p^U}{q^U}(\hat{x}) \right].$$

Thus, we have
$$\begin{cases} \frac{p^L}{q^L}(x) \leq \frac{p^L}{q^L}(\hat{x}) \\ \frac{p^U}{q^U} < \frac{p^U}{q^U}(\hat{x}), \end{cases} \quad \text{or} \quad \begin{cases} \frac{p^L}{q^L}(x) < \frac{p^L}{q^L}(\hat{x}) \\ \frac{p^U}{q^U} \leq \frac{p^U}{q^U}(\hat{x}), \end{cases} \quad \text{or} \quad \begin{cases} \frac{p^L}{q^L}(x) < \frac{p^L}{q^L}(\hat{x}) \\ \frac{p^U}{q^U} < \frac{p^U}{q^U}(\hat{x}), \end{cases}$$

which is a contradiction to the fact that \hat{x} is the optimal solution for the problems (FP_1) and (FP_2) .

Conversely, suppose \hat{x} is an LU optimal solution for the problem (NIVP) and does not solve (FP_1) .

Therefore, there exists $x \in \chi$ such that

$$\frac{p^L}{q^L}(x) < \frac{p^L}{q^L}(\hat{x})$$

and

$$p^U(x) \leq \omega(\hat{x})q^U(x),$$

which, therefore, contradicts that \hat{x} is LU optimal for (NIVP). Hence, \hat{x} is optimal for (FP_1) .

Using the similar arguments as above, it can also be shown that \hat{x} is an optimal solution for (FP_2) . \square

Following is an example illustrating Lemma 2.1.

Example 2.1 Consider the problem

$$(FPP) \text{ Min } \frac{F}{G}(x) = \frac{[f^L, f^U]}{[g^L, g^U]} = \frac{[x^2 + 1, 2x^2]}{[x + 1, x + 2]} = \left[\frac{x^2 + 1}{x + 2}, \frac{2x^2}{x + 1} \right]$$

s. t.

$$-x + 1 \leq 0.$$

Therefore, the feasible set is $\chi = \{x \mid -x + 1 \leq 0, x \in C\}$. We see that the point $x = 1$ is feasible. So, we have

$$\frac{p^L}{q^L}(x = 1) = \frac{2}{3} \text{ and } \frac{p^U}{q^U}(x = 1) = 1.$$

Next, we show that $x = 1$ is LU optimal for (FPP).

We have

$$\begin{aligned} \frac{p^L}{q^L}(x) - \frac{p^L}{q^L}(1) &= \frac{3x^2 - 2x - 1}{3(x + 2)} \\ &= \frac{1}{3(x + 2)} \left[\left(\sqrt{3}x - \frac{1}{\sqrt{3}} \right)^2 - \left(\frac{2}{\sqrt{3}} \right)^2 \right] \\ &= \frac{1}{3(x + 2)} \left[\left(\sqrt{3}x + \frac{1}{\sqrt{3}} \right) (\sqrt{3}x - \sqrt{3}) \right] \\ &> 0, \forall 1 \neq x \in \chi \end{aligned}$$

and

$$\begin{aligned} \frac{p^U}{q^U}(x) - \frac{p^U}{q^U}(1) &= \frac{2x^2 - x - 1}{x + 1} \\ &= \frac{1}{x + 1} \left[\left(\sqrt{2}x - \frac{1}{2\sqrt{2}} \right)^2 - \left(\frac{3}{2\sqrt{2}} \right)^2 \right] \\ &= \frac{1}{x + 1} \left[\left(\sqrt{2}x + \frac{1}{\sqrt{2}} \right) (\sqrt{2}x - \sqrt{2}) \right] \\ &> 0, \forall 1 \neq x \in \chi. \end{aligned}$$

Thus, we see that there exists no $x \in \chi$ such that $\begin{cases} \frac{p^L}{q^L}(x) \leq \frac{p^L}{q^L}(x = 1) \\ \frac{p^U}{q^U} < \frac{p^U}{q^U}(x = 1), \end{cases}$ or

$$\begin{cases} \frac{p^L}{q^L}(x) < \frac{p^L}{q^L}(x = 1) \\ \frac{p^U}{q^U} \leq \frac{p^U}{q^U}(x = 1), \end{cases} \quad \text{or} \quad \begin{cases} \frac{p^L}{q^L}(x) < \frac{p^L}{q^L}(x = 1) \\ \frac{p^U}{q^U} < \frac{p^U}{q^U}(x = 1), \end{cases}$$

Hence, the point $x = 1$ is LU optimal for (FPP).

We now construct the problem (P1) and (P2) as follows:

$$(P1)' \text{ Min } \frac{x^2 + 1}{x + 2}$$

subject to

$$2x^2 - x - 1 \leq 0$$

$$-x + 1 \leq 0$$

and

$$(P2)' \text{ Min } \frac{2x^2}{x + 1}$$

subject to

$$3x^2 - 2x - 1 \leq 0$$

$$-x + 1 \leq 0$$

The optimal solutions for both the problems (P1)' and (P2)' are at $x = 1$.

Consider another example verifying Lemma 2.1.

Example 2.2 Consider the problem

$$\begin{aligned} (FPP)_1 \text{ Min } \frac{F}{G}(x) &= \frac{[f^L, f^U]}{[g^L, g^U]} = \frac{[x^2 + y^2, (x^2 + y^2)e^{(x+y)}]}{[x + 1, x + y + 1]} \\ &= \left[\frac{x^2 + y^2}{x + y + 1}, \frac{(x^2 + y^2)e^{(x+y)}}{x + 1} \right] \end{aligned}$$

subject to

$$x - 2 \leq 0$$

$$y - 2 \leq 0$$

$$x \geq 0, y \geq 0.$$

Therefore, the feasible set is

$$\chi = \{(x, y) | x - 2 \leq 0, y - 2 \leq 0, x, y \geq 0, (x, y) \in C\}.$$

We see that $(x, y) = (0, 0)$ is a feasible solution. So, we have

$$\frac{f^L}{f^U}((x, y) = (0, 0)) = 0 \text{ and } \frac{g^L}{g^U}((x, y) = (0, 0)) = 0.$$

Now, we show the point $(x, y) = (0, 0)$ is LU optimal solution for $(FPP)_1$.

We have

$$\frac{p^L}{q^L}(x, y) - \frac{p^L}{q^L}(0, 0) = \frac{x^2 + y^2}{x + y + 1} > 0, \forall x \in \chi.$$

See Fig. 1.
and

$$\frac{p^U}{q^U}(x, y) - \frac{p^U}{q^U}(0, 0) = \frac{(x^2 + y^2)e^{(x+y)}}{x + 1} > 0, \forall x \in \chi.$$

See Fig. 2.

Fig. 1 Graph of over

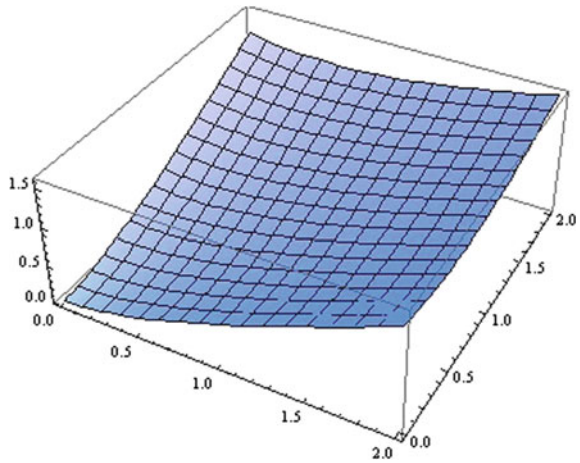
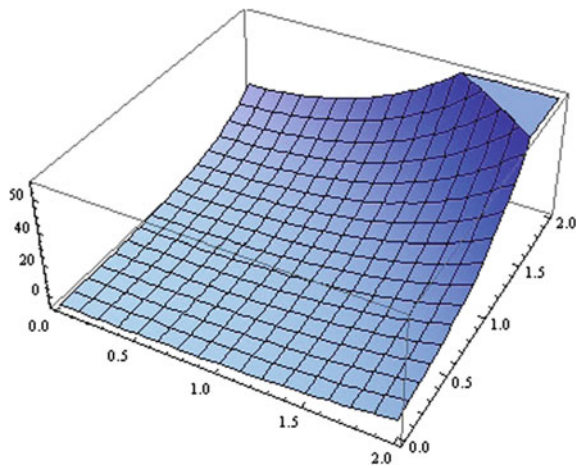


Fig. 2 Graph of over



Thus, we see that there exists no $x \in \chi$ such that

$$\begin{cases} \frac{p^L}{q^L}(x, y) \leq \frac{p^L}{q^L}(0, 0) \\ \frac{p^U}{q^U}(x, y) < \frac{p^U}{q^U}(0, 0), \end{cases} \quad \text{or}$$

$$\begin{cases} \frac{p^L}{q^L}(x, y) < \frac{p^L}{q^L}(0, 0) \\ \frac{p^U}{q^U}(x, y) \leq \frac{p^U}{q^U}(0, 0), \end{cases} \quad \text{or} \quad \begin{cases} \frac{p^L}{q^L}(x, y) < \frac{p^L}{q^L}(0, 0) \\ \frac{p^U}{q^U}(x, y) < \frac{p^U}{q^U}(0, 0). \end{cases}$$

Hence, $(x, y) = (0, 0)$ is LU optimal for $(FPP)_1$.

We now construct the problem (P1) and (P2) as follows:

$$(PP)_1 \text{ Min } \frac{x^2 + y^2}{x + y + 1}$$

subject to

$$(x^2 + y^2)e^{(x+y)} \leq 0$$

$$x - 2 \leq 0$$

$$y - 2 \leq 0$$

$$x, y \geq 0$$

and

$$(PP)_2 \text{ Min } \frac{(x^2 + y^2)e^{(x+y)}}{x + 1}$$

subject to

$$(x^2 + y^2) \leq 0$$

$$x - 2 \leq 0$$

$$y - 2 \leq 0$$

$$x, y \geq 0.$$

The optimal solutions for both the problems $(PP)_1$ and $(PP)_2$ are at $(x, y) = (0, 0)$.

Hence, Examples 2.1 and 2.2 verify Lemma 2.1. \square

3 Necessary Conditions for Optimality

In this section, utilizing a deterministic approach for optimization problem, we establish the necessary conditions for optimality for the nondifferentiable fractional interval-valued problem (NIVP). For this, we state the following lemma which will be needed in the sequel [18].

Lemma 3.1 \hat{x} is LU optimal solution for (NIVP) iff \hat{x} minimizes $\frac{p^L}{q^L}(x)$ the constraint set

$$N_i = \{x \in X \mid \frac{p^U}{q^U} \leq \frac{p^U}{q^U}(\hat{x}), h_i(x) \leq 0\}.$$

The deterministic optimization problem is considered as follows:

$$(D) \text{ Min } \varphi(x) = \frac{u(x)}{v(x)}$$

s. t.

$$\nu_i(x) \leq 0, i = 1, 2, \dots, m$$

$$x \in S,$$

where the functions involved are assumed to be quasidifferential.

We state the theorem for the problem (D) as given in [18]:

Theorem 3.1 If \hat{x} solves the problem (D), then there exist $\hat{\mu}_0 \geq 0$, and $\hat{\mu} = (\hat{\mu}_1, \hat{\mu}_2, \dots, \hat{\mu}_m) \geq 0, \hat{\mu} \neq 0$ such that

$$0 \in \hat{\mu}_0 P_0(\hat{x}) + \sum_{i=1}^m \hat{\mu}_i \partial \nu_i(\hat{x}) + N_S(\hat{x})$$

$$\mu_i \nu_i(\hat{x}) = 0, i = 1, 2, \dots, m,$$

where

$$P_0(\hat{x}) = \frac{1}{v(\hat{x})} [\partial u(\hat{x}) - \varphi(\hat{x}) \partial v(\hat{x})].$$

Here, we see that \hat{x} is LU optimal for $\varphi(x)$, having the set of constraints as

$$H = \{x \in X \mid z \in S \nu_i(x) \leq 0, i = 1, 2, \dots, m\}.$$

Therefore, following Lemma 3.1, at \hat{x} the minimum value of $\frac{p^L}{q^L}(x)$ is achieved with the constraint

$$\begin{aligned}
 F &= \{x \in H \mid \frac{p^U}{q^U} \leq \frac{p^U}{q^U}(\hat{x})\} \\
 &= \{x \in H \mid p^U(x) - \left[\frac{p^U}{q^U}(\hat{x}) \right] q^U(x) \leq 0\}.
 \end{aligned}$$

In the following, we establish the Fritz John necessary conditions for optimality for the nondifferentiable fractional interval-valued problem (NIVP).

Theorem 3.2 Suppose \hat{x} is an LU optimal solution for the problem (NIVP), then there exists $\eta = (\eta^L, \eta^U) \geq 0$ and $\hat{\xi} = (\hat{\xi}_1, \hat{\xi}_2, \dots, \hat{\xi}_m) \geq 0$, $(\hat{\xi}, \eta^L, \eta^U) \neq 0$, such that, for $i = 1, 2, \dots, m$,

$$\begin{aligned}
 0 \in \eta^L P_1(\hat{x}) + \eta^U P_2(\hat{x}) + \sum_{i=1}^m \hat{\xi}_i \partial h_i(\hat{x}) + N_S(\hat{x}) \\
 \hat{\xi}_i h_i(\hat{x}) = 0,
 \end{aligned}$$

where

$$\begin{aligned}
 P_1(\hat{x}) &= \frac{1}{q^L(\hat{x})} [\partial p^L(\hat{x}) - \omega^L(\hat{x}) \partial q^L(\hat{x})], \\
 P_2(\hat{x}) &= \frac{1}{q^U(\hat{x})} [\partial p^U(\hat{x}) - \omega^U(\hat{x}) \partial q^U(\hat{x})], \\
 \omega^L(\hat{x}) &= \frac{p^L(\hat{x})}{q^L(\hat{x})} \text{ and } \omega^U(\hat{x}) = \frac{p^U(\hat{x})}{q^U(\hat{x})}.
 \end{aligned}$$

Proof Since \hat{x} is LU optimal for (NIVP), therefore, by the Lemma 3.1, \hat{x} is optimal for the fractional scalar objective nondifferentiable problem

$$(PP_1) \text{ Min } \omega^L(x) = \frac{p^L}{q^L}(x)$$

subject to

$$\begin{aligned}
 p^U(x) - \omega^U(\hat{x})q^U &\leq 0 \\
 h_i(x) &\leq 0, i = 1, 2, \dots, m \\
 x &\in S.
 \end{aligned}$$

Thus, using Theorem 3.1, there exist $\eta = (\eta^L, \eta^U) \geq 0$ and $\hat{\mu} = (\hat{\mu}_1, \hat{\mu}_2, \dots, \hat{\mu}_m) \geq 0$, $\hat{\mu} \neq 0$ such that

$$0 \in \eta^{\hat{L}} P_1(\hat{x}) + \eta_1^{\hat{U}} \partial[p^U(\hat{x}) - \omega^U(\hat{x})q^U(\hat{x})] + \sum_{i=1}^m \hat{\xi}_i \partial h_i(\hat{x}) + N_S(\hat{x}), \quad (1)$$

$$\eta_1^{\hat{U}} (p^U(\hat{x}) - \omega^U(\hat{x})q^U(\hat{x})) = 0,$$

$$\xi_i h_i(\hat{x}) = 0, i = 1, 2, \dots, m.$$

Now,

$$\partial[p^U(x) - \omega^U(\hat{x})q^U(x)] = q^U(x)P_2(x),$$

where

$$P_2(x) = \frac{1}{q^U(x)} [\partial p^U(x) - \omega^U(\hat{x})\partial q^U(x)].$$

Therefore, when $x = \hat{x}$, we have

$$\partial[p^U(\hat{x}) - \omega^U(\hat{x})q^U(\hat{x})] = q^U(\hat{x})P_2(\hat{x}).$$

From (1), we have

$$0 \in \eta^{\hat{L}} P_1(\hat{x}) + \eta_1^{\hat{U}} q^U(\hat{x})P_2(\hat{x}) + \sum_{i=1}^m \hat{\xi}_i \partial h_i(\hat{x}) + N_S(\hat{x}).$$

Let $\eta_1^{\hat{U}} q^U(\hat{x}) = \eta^{\hat{U}}$. Therefore, we have

$$0 \in \eta^{\hat{L}} P_1(\hat{x}) + \eta^{\hat{U}} P_2(\hat{x}) + \sum_{i=1}^m \hat{\xi}_i \partial h_i(\hat{x}) + N_S(\hat{x}),$$

$$\xi_i h_i(\hat{x}) = 0, i = 1, 2, \dots, m.$$

Hence, the required result. □

Before proceeding to obtain the KKT conditions for the problem (NIVP), it is essentially important to discuss the Slater's constraint qualification for the constraints of (FP_1) and (FP_2) .

Let $\hat{x} \in X$ be an LU optimal solution of (NIVP).

Slater's constraints qualification for (FP_1) Assume $\exists x^* \in X$ in such a way that $x^* \in S, h_i(x^*) < 0, i = 1, 2, \dots, m$ and $p^U(x^*) - \omega^U(\hat{x})q^U(x^*) < 0$.

Slater's constraint qualification for (FP_2) Assume $\exists x^* \in X$ in such a way that $x^* \in S, h_i(x^*) < 0, i = 1, 2, \dots, m$ and $p^L(x^*) - \omega^L(\hat{x})q^L(x^*) < 0$. We shall use the above constraint qualification, in the sequel, in order to derive the KKT conditions for the nondifferentiable fractional interval optimization problem (NIVP), following the one given by Borwein [19].

The KKT optimality condition due to Borwein [19] is as follows:

Theorem 3.3 Let \hat{x} solves the problem (D) and also $h_i(x)$, $i = 1, 2, \dots, m$ satisfies some constraint qualification. Then there exist $\hat{\xi} = (\hat{\xi}_1, \hat{\xi}_2, \dots, \hat{\xi}_m) \geq 0$, $\hat{\xi} \neq 0$ such that for $i = 1, 2, \dots, m$,

$$0 \in P_0(\hat{x}) + \sum_{i=1}^m \hat{\xi}_i \partial h_i(\hat{x}) + N_S(\hat{x})$$

$$\xi_i h_i(\hat{x}) = 0,$$

where

$$P_0(\hat{x}) = \frac{1}{v(\hat{x})} [\partial u(\hat{x}) - \varphi(\hat{x}) \partial v(\hat{x})].$$

Theorem 3.4 Let \hat{x} be an LU optimal solution for (NIVP) and both the Slater's constraint qualification for (FP₁) and (FP₂) are satisfied, then there exist $\eta = (\eta^L, \eta^U) > 0$ and $\hat{\xi} = (\hat{\xi}_1, \hat{\xi}_2, \dots, \hat{\xi}_m) \geq 0$, such that $i = 1, 2, \dots, m$,

$$0 \in \eta^L P_1(\hat{x}) + \eta^U P_2(\hat{x}) + \sum_{i=1}^m \hat{\xi}_i \partial h_i(\hat{x}) + N_S(\hat{x})$$

$$\hat{\xi}_i h_i(\hat{x}) = 0,$$

where

$$P_1(\hat{x}) = \frac{1}{q^L(\hat{x})} [\partial p^L(\hat{x}) - \omega^L(\hat{x}) \partial q^L(\hat{x})],$$

$$P_2(\hat{x}) = \frac{1}{q^U(\hat{x})} [\partial p^U(\hat{x}) - \omega^U(\hat{x}) \partial q^U(\hat{x})],$$

$$\omega^L(\hat{x}) = \frac{p^L(\hat{x})}{q^L(\hat{x})} \text{ and } \omega^U(\hat{x}) = \frac{p^U(\hat{x})}{q^U(\hat{x})}.$$

Proof Let \hat{x} be LU optimal for (NIVP). Therefore, \hat{x} is optimal for the problems (FP₁) and (FP₂). Hence, using Lemma 3.1, at \hat{x} the minimum value of $\frac{p^L}{q^L}(x)$ is obtained with the constraint

$$N_L = \{x \in S | p^U(x) - \omega^U(\hat{x}) q^U(x) \leq 0, h_i(x) \leq 0, i = 1, 2, \dots, m\},$$

and $\frac{p^U}{q^U}(x)$ with the constraint set

$$N_U = \{x \in S | p^L(x) - \omega^L(\hat{x}) q^L(x) \leq 0, h_i(x) \leq 0, i = 1, 2, \dots, m\}.$$

By Theorem 3.3, we have nonnegative constraints $\rho_{Lk}, \rho_{Uk}, k = L, U$ with $\rho_{LL} = \rho_{UU} = 1$ and $\nu_{1k}, \nu_{2k}, \dots, \nu_{mk}, k = L, U$ such that

$$0 \in \rho_{Lk}P_1(\hat{x}) + \rho_{Uk}P_2(\hat{x}) + \sum_{i=1}^m \nu_{ik}\partial h_i(\hat{x}) + N_S(\hat{x}) \quad (2)$$

and

$$\xi_{ik}h_i(\hat{x}) = 0, i = 1, 2, \dots, m \quad (3)$$

for every $k = L, U$.

Summing (2) and (3) over $k = L, U$, we get

$$0 \in (1 + \rho_{LU})P_1(\hat{x}) + (1 + \rho_{UL})P_2(\hat{x}) + 2N_S(\hat{x}) + \sum_{i=1}^m (\nu_{iL} + \nu_{iU})\partial h_i(\hat{x}) = 0, \quad (4)$$

$$(\nu_{iL} + \nu_{iU})h_i(\hat{x}) = 0, i = 1, 2, \dots, m. \quad (5)$$

Setting $\hat{\eta}^L = 1 + \rho_{LU} > 0, \hat{\eta}^U = 1 + \rho_{UL} > 0$ and $\hat{\xi}_i = (\nu_{iL} + \nu_{iU}) \geq 0$.

Thus, from (4) and (5), it yields

$$0 \in \hat{\eta}^L P_1(\hat{x}) + \hat{\eta}^U P_2(\hat{x}) + \sum_{i=1}^m \hat{\xi}_i \partial h_i(\hat{x}) + N_S(\hat{x}) = 0,$$

$$\hat{\xi}_i h_i(\hat{x}) = 0, i = 1, 2, \dots, m.$$

Hence, the KKT conditions for the problem (NIVP) have been proved. \square

4 Sufficient Conditions for Optimality

In this section, we derive the sufficient conditions for optimality for the problem (NIVP).

First, we establish the Fritz John conditions for sufficiency for the problem (NIVP).

Theorem 4.1 *Suppose that there exist $\eta = (\eta^L, \eta^U) \geq 0, \hat{\xi} = (\hat{\xi}_1, \hat{\xi}_2, \dots, \hat{\xi}_m) \geq 0$, and $(\hat{\eta}, \hat{\xi}) \neq 0$, such that*

$$0 \in \eta^L P_1(\hat{x}) + \eta^U P_2(\hat{x}) + \sum_{i=1}^m \hat{\xi}_i \partial h_i(\hat{x}) + N_S(\hat{x}), \quad (6)$$

$$\hat{\xi}_i h_i(\hat{x}) = 0, i = 1, 2, \dots, m, \quad (7)$$

where $P_1(\hat{x})$ and $P_2(\hat{x})$ are defined as in Theorem 3.2. Further, assume that

- (i) $p^L, p^U, -q^L, -q^U$ are convex.
- (ii) h_i are convex functions for all $i = 1, 2, \dots, m, i \neq k$, also for $i = k, \hat{\xi}_k > 0$, and h_k is strict convex.

Then, \hat{x} is LU optimal for (NIVP).

Proof From the fact that

$$P_1(\hat{x}) = \frac{1}{q^L(\hat{x})}[\partial p^L(\hat{x}) - \omega^L(\hat{x})\partial q^L(\hat{x})],$$

$$P_2(\hat{x}) = \frac{1}{q^U(\hat{x})}[\partial p^U(\hat{x}) - \omega^U(\hat{x})\partial q^U(\hat{x})],$$

we have from (6) $0 \in \eta^{\hat{L}} \frac{1}{q^L(\hat{x})}[\partial p^L(\hat{x}) - \omega^L(\hat{x})\partial q^L(\hat{x})] + \eta^{\hat{U}} \frac{1}{q^U(\hat{x})}[\partial p^U(\hat{x}) - \omega^U(\hat{x})\partial q^U(\hat{x})]$

$$+ \sum_{i=1}^m \hat{\xi}_i \partial h_i(\hat{x}) + N_S(\hat{x}).$$

Now, there exists $u_1^{\hat{L}} \in \partial p^L(\hat{x}), u_2^{\hat{U}} \in \partial p^U(\hat{x}), v_1^{\hat{L}} \in \partial q^L(\hat{x}), v_2^{\hat{U}} \in \partial q^U(\hat{x})$ and $\hat{w}_i \in \partial h_i(\hat{x}), i = 1, 2, \dots, m$ and $\hat{z} \in N_S(\hat{x})$ in such a way that

$$\eta^{\hat{L}} \frac{1}{q^L(\hat{x})}[u_1^{\hat{L}} - \omega^L(\hat{x})v_1^{\hat{L}}] + \eta^{\hat{U}} \frac{1}{q^U(\hat{x})}[u_2^{\hat{U}} - \omega^U(\hat{x})v_2^{\hat{U}}] + \sum_{i=1}^m \hat{\xi}_i \hat{w}_i + \hat{z} = 0,$$

which, in turn, yields

$$(x - \hat{x})^T \left[\eta^{\hat{L}} \frac{1}{q^L(\hat{x})}[u_1^{\hat{L}} - \omega^L(\hat{x})v_1^{\hat{L}}] + \eta^{\hat{U}} \frac{1}{q^U(\hat{x})}[u_2^{\hat{U}} - \omega^U(\hat{x})v_2^{\hat{U}}] + \sum_{i=1}^m \hat{\xi}_i \hat{w}_i + \hat{z} \right] = 0. \tag{8}$$

We now claim that \hat{x} is LU optimal for the problem (NIVP). On the contrary, suppose that \hat{x} is not a LU optimal solution for the problem (NIVP). Then there exists $x \in X_0$

such that
$$\begin{cases} \frac{p^L}{q^L}(x) \leq \frac{p^L}{q^L}(\hat{x}) \\ \frac{p^U}{q^U} < \frac{p^U}{q^U}(\hat{x}), \end{cases} \quad \text{or} \quad \begin{cases} \frac{p^L}{q^L}(x) < \frac{p^L}{q^L}(\hat{x}) \\ \frac{p^U}{q^U} \leq \frac{p^U}{q^U}(\hat{x}), \end{cases} \quad \text{or} \quad \begin{cases} \frac{p^L}{q^L}(x) < \frac{p^L}{q^L}(\hat{x}) \\ \frac{p^U}{q^U} < \frac{p^U}{q^U}(\hat{x}), \end{cases}$$

which imply

$$p^L(x) - \omega^L(\hat{x})q^L(x) \leq p^L(\hat{x}) - \omega^L(\hat{x})q^L(\hat{x})$$

$$p^U(x) - \omega^U(\hat{x})q^U(x) < p^U(\hat{x}) - \omega^U(\hat{x})q^U(\hat{x})$$

or

$$\begin{aligned} p^L(x) - \omega^L(\hat{x})q^L(x) &< p^L(\hat{x}) - \omega^L(\hat{x})q^L(\hat{x}) \\ p^U(x) - \omega^U(\hat{x})q^U(x) &\leq p^U(\hat{x}) - \omega^U(\hat{x})q^U(\hat{x}) \end{aligned}$$

or

$$\begin{aligned} p^L(x) - \omega^L(\hat{x})q^L(x) &< p^L(\hat{x}) - \omega^L(\hat{x})q^L(\hat{x}) \\ p^U(x) - \omega^U(\hat{x})q^U(x) &< p^U(\hat{x}) - \omega^U(\hat{x})q^U(\hat{x}). \end{aligned}$$

From hypothesis (i), it implies that $p^L - \omega^L(\hat{x})q^L$ and $p^U - \omega^U(\hat{x})q^U$ are convex, and therefore, we have

$$\begin{aligned} (x - \hat{x})^T [u_1^{\hat{L}} - \omega^L(\hat{x})v_1^{\hat{L}}] &\leq 0 \\ (x - \hat{x})^T [u_2^{\hat{U}} - \omega^U(\hat{x})v_2^{\hat{U}}] &< 0, \end{aligned}$$

or,

$$\begin{aligned} (x - \hat{x})^T [u_1^{\hat{L}} - \omega^L(\hat{x})v_1^{\hat{L}}] &< 0 \\ (x - \hat{x})^T [u_2^{\hat{U}} - \omega^U(\hat{x})v_2^{\hat{U}}] &\leq 0, \end{aligned}$$

or,

$$\begin{aligned} (x - \hat{x})^T [u_1^{\hat{L}} - \omega^L(\hat{x})v_1^{\hat{L}}] &< 0 \\ (x - \hat{x})^T [u_2^{\hat{U}} - \omega^U(\hat{x})v_2^{\hat{U}}] &< 0. \end{aligned}$$

From the fact that $\eta^{\hat{L}} \frac{1}{q^L(\hat{x})} \geq 0$ and $\eta^{\hat{U}} \frac{1}{q^U(\hat{x})} \geq 0$, we have

$$(x - \hat{x})^T \left[\eta^{\hat{L}} \frac{1}{q^L(\hat{x})} (u_1^{\hat{L}} - \omega^L(\hat{x})v_1^{\hat{L}}) + \eta^{\hat{U}} \frac{1}{q^U(\hat{x})} (u_2^{\hat{U}} - \omega^U(\hat{x})v_2^{\hat{U}}) \right] \leq 0. \quad (9)$$

Since $\hat{\xi}_i \geq 0$, $h_i(\hat{x}) \leq 0$, so $\hat{\xi}_i h_i(\hat{x}) = 0$, $i = 1, 2, \dots, m$, therefore, we obtain

$$\hat{\xi}_i h_i(x) \leq \hat{\xi}_i h_i(\hat{x}). \quad (10)$$

Hypothesis (ii) and (9) gives

$$(x - \hat{x})^T \sum_{i=1}^m \hat{\xi}_i \hat{w}_i < 0. \quad (11)$$

Again,

$$\hat{z} \in N_S(\hat{x}) \Rightarrow (x - \hat{x})^T \hat{z} \leq 0. \quad (12)$$

Summing up (9), (11), and (12), we have

$$(x - \hat{x})^T \left[\eta^{\hat{L}} \frac{1}{q^L(\hat{x})} [u_1^{\hat{L}} - \omega^L(\hat{x})v_1^{\hat{L}}] + \eta^{\hat{U}} \frac{1}{q^U(\hat{x})} [u_2^{\hat{U}} - \omega^U(\hat{x})v_2^{\hat{U}}] + \sum_{i=1}^m \hat{\xi}_i \hat{w}_i + \hat{z} \right] < 0,$$

which contradicts (8).

Hence, \hat{x} is an LU optimal solution for the problem (NIVP). \square

Now, we derive the KKT sufficient conditions for the problem (NIVP).

Theorem 4.2 *Suppose that for a feasible point \hat{x} , there exists $(\eta^{\hat{L}}, \eta^{\hat{U}}) > 0$, $\hat{\xi} = (\hat{\xi}_1, \hat{\xi}_2, \dots, \hat{\xi}_m) \geq 0$, not all $\hat{\xi}_i = 0$, $i = 1, 2, \dots, m$ such that for $i = 1, 2, \dots, m$,*

$$0 \in \eta^{\hat{L}} P_1(\hat{x}) + \eta^{\hat{U}} P_2(\hat{x}) + \sum_{i=1}^m \hat{\xi}_i \partial h_i(\hat{x}) + N_S(\hat{x}), \quad (13)$$

$$\hat{\xi}_i h_i(\hat{x}) = 0, \quad (14)$$

where $P_1(\hat{x})$ and $P_2(\hat{x})$ are defined as in Theorem 3.2. Further, assume that

(i) $p^L, p^U, -q^L, -q^U, h_i, i = 1, 2, \dots, m$ are convex.

Then, \hat{x} is an LU optimal solution of (NIVP).

Proof Since $\eta^{\hat{L}} \frac{1}{q^L(\hat{x})} > 0$ and $\eta^{\hat{U}} \frac{1}{q^U(\hat{x})} > 0$, we have from (8)

$$(x - \hat{x})^T \left[\eta^{\hat{L}} \frac{1}{q^L(\hat{x})} (u_1^{\hat{L}} - \omega^L(\hat{x})v_1^{\hat{L}}) + \eta^{\hat{U}} \frac{1}{q^U(\hat{x})} (u_2^{\hat{U}} - \omega^U(\hat{x})v_2^{\hat{U}}) \right] < 0. \quad (15)$$

From hypothesis (ii), $i = 1, 2, \dots, m$ gives from (15)

$$(x - \hat{x})^T \sum_{i=1}^m \hat{\xi}_i \hat{w}_i \leq 0. \quad (16)$$

Adding (15) and (16), we have

$$(x - \hat{x})^T \left[\eta^{\hat{L}} \frac{1}{q^L(\hat{x})} [u_1^{\hat{L}} - \omega^L(\hat{x})v_1^{\hat{L}}] + \eta^{\hat{U}} \frac{1}{q^U(\hat{x})} [u_2^{\hat{U}} - \omega^U(\hat{x})v_2^{\hat{U}}] + \sum_{i=1}^m \hat{\xi}_i \hat{w}_i + \hat{z} \right] < 0,$$

which contradicts (8).

Hence, \hat{x} is an LU optimal solution for the problem (NIVP). \square

References

1. Kannappan, P.: Necessary conditions for optimality of nondifferentiable convex multiobjective programming. *J. Optim. Theory Appl.* **40**, 167–174 (1983)
2. Bhurjee, A.K., Panda, G.: Sufficient optimality conditions and duality theory for interval optimization problem. *Ann. Oper. Res.* **243**, 335–348 (2014)
3. Sun, Y., Xu, X., Wang, L.: Duality and saddle-point type optimality for interval-valued programming. *Optim. Lett.* **8**, 1077–1091 (2014)
4. Li, L., Liu, S., Zhang, J.: On interval-valued invex mappings and optimality conditions for interval-valued optimization problems. *J. Inequalities Appl.* **179**, 2–19 (2015)
5. Ahmad, I., Jayswal, A., Banerjee, J.: On interval-valued optimization problems with generalized invex functions. *J. Inequalities Appl.* **313**, 2–14 (2013)
6. Jayswal, A., Stancu-Minasian, I., Ahmad, I.: On sufficiency and duality for a class of interval-valued programming problems. *Appl. Math. Comput.* **218**, 4119–4127 (2011)
7. Wu, H.C.: Duality theory for optimization problems with interval-valued objective functions. *J. Optim. Theory Appl.* **144**, 615–628 (2010)
8. Wu, H.C.: Wolfe duality for interval-valued optimization. *J. Optim. Theory Appl.* **138**, 497–509 (2008)
9. Wu, H.C.: The Karush-Kuhn-Tucker optimality conditions in an optimization problem with interval-valued objective function. *Eur. J. Oper. Res.* **176**, 46–59 (2007)
10. Wu, H.C.: The Karush-Kuhn-Tucker optimality conditions in multiobjective programming problems with interval-valued objective functions. *Eur. J. Oper. Res.* **196**, 49–60 (2009)
11. Osuna-Gomez, R., Hernandez-Jimenez, B., Chalco-Cano, Y., Ruiz-Garzon, G.: New efficiency conditions for multiobjective interval-valued programming problems. *Inf. Sci.* **420**, 235–248 (2017)
12. Antczak, T.: Optimality conditions and duality results for nonsmooth vector optimization problems with the multiple interval-valued objective function. *Acta Math. Sci.* **37B**(4), 1133–1150 (2017)
13. Ghosh, D.: Newton method to obtain efficient solutions of the optimization problems with interval-valued objective functions. *J. Appl. Math. Comput.* **53**, 709–731 (2017)
14. Sun, Y., Wang, L.: Optimality conditions and duality in nondifferentiable interval-valued programming. *J. Ind. Manage. Optim.* **9**, 131–142 (2013)
15. Jayswal, A., Ahmad, I., Banerjee, J.: Nonsmooth interval-valued optimization and saddle-point optimality criteria. *Bull. Malays. Math. Sci. Soc.* **39**, 1391–1411 (2016)
16. Jayswal, A., Stancu-Minasian, I., Banerjee, J.: Optimality conditions and duality for interval-valued optimization problems using convexifactors. *Rend. Circ. Mat. Palermo* **65**, 17–32 (2016)
17. Bhurjee, A.K., Padhan, S.K.: Optimality conditions and duality results for non-differentiable interval optimization problems. *J. Appl. Math. Comput.* **50**, 59–71 (2016)
18. Bector, C.R., Chandra, S., Husain, I.: Optimality conditions and duality in subdifferentiable multiobjective fractional programming. *J. Optim. Theory Appl.* **79**, 105–125 (1993)
19. Borwein, J.M.: Fractional programming without differentiability. *Math. Program.* **11**, 190–283 (1976)

Application of Constrained Spider Monkey Optimization to Solve Portfolio Optimization Problem



Kavita Gupta, Kusum Deep and Atulya K. Nagar

Abstract Portfolio optimization problem has attracted the attention of researchers since ages because of its practical application. This problem is constrained in nature and deals with answering the question what amount of wealth should be invested in a particular asset. In this paper, portfolio optimization problem has been solved using Constrained Spider Monkey Optimization (CSMO) algorithm. The objective behind this work is the application of CSMO for solving a real-world optimization problem. For the experiment purpose, basic mean-variance optimization model is considered.

Keywords Constrained spider monkey optimization · Portfolio optimization
Metaheuristics · Markowitz model

1 Introduction

A portfolio is a collection of two or more risky/riskless assets held by an institution or an individual. Suppose a user wants to invest money in n assets. Then, its portfolio is represented by n -tuple (x_1, x_2, \dots, x_n) , where x_i denotes the amount of fund to be invested in the i th asset. Each of the assets in a portfolio has a return and risk associated with them. Portfolio optimization problem deals with maximizing the profitable returns while minimizing the associated risk of the portfolio. Markowitz [7] was the first to develop an optimization model based on this idea. Since then,

K. Gupta (✉) · K. Deep
Department of Mathematics, Indian Institute of Technology Roorkee, Roorkee,
Uttarakhand 247667, India
e-mail: gupta.kavita3043@gmail.com

K. Deep
e-mail: kusumdeep@gmail.com

A. K. Nagar
Mathematical Sciences and Faculty of Science, Liverpool Hope University,
Hope Park, Liverpool L16 9JD, UK
e-mail: nagara@hope.ac.uk

various optimization models which are variations of basic Markowitz's model have been developed. Various metaheuristics like genetic algorithm [2], particle swarm optimization [3, 10], artificial bee colony [9], bacterial foraging optimization [8], etc. have been applied to solve different models of portfolio optimization problem.

The rest of the paper is organized as follows: In Sect. 2, mean-variance optimization model has been discussed. In Sect. 3, a brief introduction of constrained spider monkey optimization has been provided. In Sect. 4, experimental setup is provided. In Sect. 5, the experimental results have been discussed. The chapter is concluded in Sect. 6.

2 Mean-Variance Model

Markowitz mean-variance model [6] is the basic model for solving portfolio optimization problem. Mathematical formulation for the mean-variance model is given below:

Let return of the i th asset is denoted by a random variable say R_i , x_i is the amount of fund to be invested in i th asset.

Asset return is the amount of return which can be calculated for a given period of time. Mathematically, it may be defined as

Return = (closing price of current period – closing price of previous period + dividend collect during the period)/(closing price of previous period)

$$r_{it} = \frac{(p_{it}) - (p_{it-1}) + (d_{it})}{(p_{it-1})},$$

where p_{it} is the closing price of the asset during the period t , d_{it} is the dividend collected during the period.

The aim is to minimize the expected return on the portfolio and maximize the risk.

$$r(x_1, x_2, \dots, x_n) = E \left[\sum_{i=1}^n R_i x_i \right] = \sum_{i=1}^n E[R_i] x_i = \sum_{i=1}^n r_i x_i, \quad (1)$$

where r_i is the expected return on the i th asset and $r_i = E[R_i]$.

$$r_i = E[R_i] = \frac{1}{T} \sum_{t=1}^T r_{it} \quad (2)$$

The covariance σ_{ij} between the asset returns R_i and R_j can be expressed as follows:

$$\sigma_{ij} = E[(R_i - E[R_i])(R_j - E[R_j])] = \frac{1}{T} \sum_{t=1}^T (r_{it} - r_i)(r_{jt} - r_j). \quad (3)$$

The portfolio risk is characterized by the variance of returns on that portfolio. The variance of return on a portfolio is then expressed as follows:

$$v(x_1, x_2, \dots, x_n) = \sum_{i=1}^n \sum_{j=1}^n \sigma_{ij} x_i x_j. \quad (4)$$

The mathematical formulation of the Markowitz's mean-variance optimization model is given in model $M(1)$.

$$M(1) \quad \min f(x) = \sum_{i=1}^n \sum_{j=1}^n \sigma_{ij} x_i x_j,$$

subject to

$$\sum_{i=1}^n r_i x_i = r_0 \quad (5)$$

$$\sum_{i=1}^n x_i = 1 \quad (6)$$

$$x_i \geq 0, \quad i = 1, 2, \dots, n. \quad (7)$$

From the model of the problem $M(1)$, it can be seen that it is a constrained optimization problem with equality constraints only. Objective function f is actually the risk $v(x_1, x_2, \dots, x_n)$. r_0 in Eq. (5) denotes the amount of return desired by the investor. Equation (5) makes sure that the expected portfolio return should be equal to the amount of return desired by the investor. Equation (6) represents the capital budget constraint on the assets. Equation (7) makes sure that the value of proportion to be invested in an asset should be nonnegative. From these constraints, it can be concluded that the value of r_0 cannot be chosen arbitrarily. Though a high portfolio return is always desirable, aspiring it to be very high can make the problem infeasible. The value of r_0 lies between r_{\min} and r_{\max} . Here, r_{\min} is the portfolio return corresponding to the minimum risk. This value can be obtained by solving the problem described in model $M(1)$ after removing the constraint represented by Eq. (5). r_{\max} is the maximum feasible value of r_0 and it is given by the maximum mean return among the mean return of all the assets.

3 Constrained Spider Monkey Optimization (CSMO)

Spider monkey optimization [1] is a new member of the swarm intelligent algorithms. The basic SMO can solve unconstrained optimization problems only. Deb's technique [4] has been used for constraint handling in Constrained Spider Monkey Optimization (CSMO) [5]. Deb's technique follows the three feasibility rules:

- Between a feasible solution and an infeasible solution, a feasible solution will be chosen.
- Between two feasible solutions, the one with higher fitness value will be chosen.
- Between two infeasible solutions, the one with less constraint violation will be chosen.

The main steps of CSMO are given below:

Initialization: In CSMO, the initial swarm is randomly generated between lower and upper bounds of the decision variables using uniform distribution. Since there is no assumption about the feasibility of the initial swarm, both feasible and infeasible solutions appear in the initial swarm.

Algorithm 1: Calculation of fitness value of solutions

```

For i = 1 to SwarmSize Do
  If (violationi = 0) Then
    fitnessi = f(SMi)
  Else
    fitnessi = fworst +  $\sum_{j=1}^m$  violationj
  End If
End For

```

Local Leader Phase: This phase allows the spider monkeys to update their positions based on the perturbation rate. A new position for a spider monkey is generated using its current position, position of the local leader and position of a randomly selected member of the group. The fitness value of newly generated position of a spider monkey is calculated and compared with its old position. This new position is adopted only if it is better than the old one. Algorithm 2 provides the update equation and execution steps of this phase.

Algorithm 2: Local Leader Phase

```

For k = 1 to NumberOfGroups Do
  For i = Index[k][0] to Index[k][1] Do
    For j = 1 to Dim Do
      If Rand (0, 1) ≥ Pr Then
        
$$sm_{newj} = sm_{ij} + \text{Rand}(0,1) \times (ll_{kj} - sm_{ij}) + \text{Rand}(-1,1) \times (sm_{rj} - sm_{ij})$$

      Else
        
$$sm_{newj} = sm_{ij}$$

      End If
    End For
    End For
    Apply Deb's Three Feasibility Rules on  $SM_{new}$  and  $SM_i$  to select the better solution
  End For
End For

```

Global Leader Phase: This phase allows the spider monkeys to update their position based on their probability. This probability is fitness proportionate which indicates that the probability of a highly fit spider monkey will be higher as compared to low fit spider monkeys. In this phase, a new position for the selected spider monkey is generated based on its current position, position of the global leader and position of the randomly selected member of the group. The update equation and execution steps of this phase are provided in Algorithm 3.

Algorithm 3: Global Leader Phase

```

For k = 1 to NumberOfGroups Do
  GroupSize = size of kth group
  counter = 0, i = 1
  While (counter < SswarmSize) Do
    For i = 1 to GroupSize Do
      If (Rand(0,1) < probabilityi) Then
        counter = counter + 1
        Randomly select j from {1, 2, ..., Dim}
        Randomly select SMr from kth group

         $sm_{newj} = sm_{ij} + \text{Rand}(0,1) \times (gl_j - sm_{ij}) + \text{Rand}(-1,1) \times (sm_{rj} - sm_{ij})$ 

      End If

      Apply Deb's Three Feasibility Rules on SMnew and SMi to select the better solution
    End For
    i = i + 1
  If (i = SwarmSize) Then
    i = 1
  End If
End While
End For

```

Global Leader Learning Phase: This phase is meant for the selection of global leader of the swarm in each iteration. The spider monkey with best fitness value is selected as the global leader of the swarm. Algorithm 4 provides the steps for selecting the global leader.

Algorithm 4: Global Leader Learning Phase

```

//Apply Deb's Three Feasibility Rules to update position of the global leader of the swarm
If (position of global leader is updated from previous position) Then
  GlobalLimitCount = 0
Else
  GlobalLimitCount = GlobalLimitCount + 1
End If

```

Local Leader Learning Phase: This phase is meant for selecting the local leader of every group. The spider monkey with best fitness value in every group is selected as the local leader of that group. The execution procedure of this phase is explained in Algorithm 5.

Algorithm 5: Local Leader Learning Phase

```

For k = 1 to NumberOfGroups do
    // Apply Deb's Three Feasibility Rules to update position of the leader of the group
    If (position of local leader is updated from previous position) Then
        LocalLimitCountk = 0
    Else
        LocalLimitCountk = LLCk + 1
    End If
End For

```

Local Leader Decision Phase: In this phase, the groups are re-initialized if their local leaders are not making progress for the specified local leader limit. The update equation and execution steps of this phase are explained in Algorithm 6.

Algorithm 6: Local Leader Decision phase

```

For k = 1 to NumberOfGroups Do
    If (LocalLimitCountk > LocalLeaderLimit) Then
        LocalLimitCountk = 0
        For i = Index[k][0] to Index[k][1] Do
            For j = 1 to Dim Do
                If (Rand(0, 1) ≥ Pr) Then
                     $sm_{ij} = sm_{minj} + Rand(0,1) \times (sm_{maxj} - sm_{minj})$ 
                Else
                     $sm_{ij} = sm_{ij} + Rand(0,1) \times (gl_j - sm_{ij}) + Rand(0,1) \times (sm_{ij} - ll_{kj})$ 
                End If
            End For
        End For
    End If
End For

```

Global Leader Decision Phase: This phase is meant to check if there is stagnation in the swarm based on the specified global leader limit. Algorithm 7 explains the procedure for executing this phase.

Algorithm 7: Global Leader Decision phase

```

If (GlobalLimitCount > GlobalLeaderLimit) Then
    GlobalLimitCount = 0
    If (NumberOfGroups < MaximumGroups) Then
        NumberOfGroups = NumberOfGroups + 1
    Else
        NumberOfGroups = 1
    End If
    Apply Local Leader Learning Phase
End If

```

Pseudocode of CSMO is provided in Algorithm 8.

Algorithm 8: Pseudocode for CSMO

```

Generate the initial swarm using uniform distribution
Initialize LocalLeaderLimit, GlobalLeaderLimit, Pr, MaximumGroups
Set Iteration = 0
Apply Algorithm 1 to Calculate fitness value of each spider monkey in the swarm
Apply Deb's three feasibility rules to select global leader and local leaders
While (termination criterion is not fulfilled) do
    //Apply Algorithm 2
    //Calculate Probability of each spider monkey
    //Apply Algorithm 3
    //Apply Algorithm 4
    //Apply Algorithm 5
    // Apply Algorithm 6
    // Apply Algorithm 7
    Iteration = iteration + 1
End While

```

4 Experimental Setup

4.1 Parameter Setting and Termination Criteria

The setting of control parameters has been adopted from Gupta et al. [5].

SwarmSize = 50
 Perturbation rate (Pr) = linearly increasing ([0.1, 0.4])
 Maximum number of groups (MaxGroups) = 5
 LocalLeaderLimit = 1500
 GlobalLeaderLimit = 50
 Total number of runs = 25
 Stopping criterion = 20,000 function evaluations

4.2 Optimization Model and Input Data

The mean-variance model described in Sect. 2 has been taken for the experiment. In order to understand the working of this portfolio optimization model, the data of a real-world problem has been taken for illustration purpose. The retail industry has been chosen for experiment because it contributes a big percentage of the gross domestic income. The 11 retail companies, listed on National Stock Exchange (NSE), have been selected as assets to construct portfolios. The list of these companies has been provided in Table 1. Our sample data includes the closing prices of these 11 assets from 1 April 2015 to 31 March 2016. The reason behind choosing these particular 11 companies is that our sample data has been extracted from Capitaline and these were the only companies listed on NSE during the financial year 2015–16 whose data was available.

Average monthly returns of these 11 assets are provided in Table 2. The expected return, variance and covariance for these assets have been provided in Tables 3 and 4, respectively.

Using the optimization model $M(1)$ and entries in Tables 2, 3 and 4 as input data, the optimization model $M(2)$ is formulated:

$$\begin{aligned}
 M(2) \min f(x) = & 1.28733x_1x_1 + 0.32512x_1x_2 + 0.96732x_1x_3 \\
 & + 0.30506x_1x_4 + 0.44927x_1x_5 + 0.39730x_1x_6 \\
 & + 0.33890x_1x_7 + 0.20332x_1x_8 + +0.22915x_1x_9 \\
 & + 0.42335x_1x_{10} + 0.31637 * x_1x_{11} + 0.84620 * x_2x_2 \\
 & + 0.08426 * x_2x_3 + 0.04178x_2x_4 + 0.16766x_2x_5 \\
 & + 0.32539x_2x_6 + 0.02563x_2x_7 + 0.27929x_2x_8 - 0.20748x_2x_9 \\
 & - 0.03959x_2x_{10} + 0.07751x_2x_{11} + 1.24506x_3x_3 + 0.53418x_3x_4 \\
 & + 0.51028x_3x_5 + 0.09637x_3x_6 + 0.27569x_3x_7
 \end{aligned}$$

Table 1 List of retail companies (assets)

Company	NSE name	Allocation of funds
Aditya Birla Fashion & Retail Ltd.	ABFRL	x_1
Cantabil Retail India Ltd.	CANTABIL	x_2
Future Enterprises-DVR	FELDVR	x_3
Future Enterprises Ltd.	FEL	x_4
Future Lifestyle Fashions Ltd.	FLFL	x_5
Provogue (India) Ltd.	PROVOGE	x_6
Shoppers Stop Ltd.	SHOPERSTOP	x_7
Store One Retail India Ltd	SORILINFRA	x_8
Trent Ltd.	TRENT	x_9
V2 Retail Ltd.	V2RETAIL	x_{10}
VMart Retail Ltd.	VMART	x_{11}

$$\begin{aligned}
& + 0.11821x_3x_8 + 0.23926x_3x_9 + 0.19023x_3x_{10} - 0.04089x_3x_{11} \\
& + 0.51375x_4x_4 + 0.41437x_4x_5 + 0.13632x_4x_6 \\
& + 0.03128x_4x_7 + 0.10967x_4x_8 + 0.09818x_4x_9 + 0.24445x_4x_{10} \\
& + 0.00358x_4x_{11} + 0.45578x_5x_5 + 0.24846x_5x_6 \\
& + 0.07455x_5x_7 + 0.26494x_5x_8 + 0.07977x_5x_9 + 0.28799x_5x_{10} \\
& + 0.11858x_5x_{11} + 0.64313x_6x_6 + 0.04808x_6x_7 \\
& + 0.02623x_6x_8 + 0.00280x_6x_9 + 0.35585x_6x_{10} + 0.42780x_6x_{11} \\
& + 0.14967x_7x_7 + 0.07379x_7x_8 + 0.15905x_7x_9 \\
& + 0.03754x_7x_{10} + 0.09046x_7x_{11} + 0.84003x_8x_8 + 0.00414 * x_8x_9 \\
& + 0.40828x_8x_{10} + 0.05099x_8x_{11} + 0.31191x_9x_9 \\
& + 0.06612x_9x_{10} + 0.12249x_9x_{11} + 0.97290x_{10}x_{10} \\
& + 0.30167x_{10}x_{11} + 0.44687x_{11}x_{11},
\end{aligned}$$

Such that

$$x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9 + x_{10} + x_{11} = 1 \quad (8)$$

$$\begin{aligned}
& 0.1617x_1 + 0.2972x_2 + 0.4546x_3 + 0.1723x_4 + 0.1189x_5 + 0.0486x_6 \\
& - 0.0329x_7 + 0.3958x_8 + 0.0515x_9 + 0.2553x_{10} - 0.0561x_{11} = r_0 \quad (9)
\end{aligned}$$

$$x_i \geq 0, \quad i = 1, 2, \dots, n. \quad (10)$$

Table 2 Monthly return of assets for the period 1 April 2015 to 31 March 2016

	1	2	3	4	5	6	7	8	9	10	11	12
ABFRL	-0.00895	2.59500	-0.03682	1.02261	-0.77381	0.73950	-0.06800	0.07895	0.27727	-0.14550	-2.48429	0.74450
CANTABIL	1.46316	0.66400	-0.60455	0.52739	0.12667	-0.29100	1.12400	-0.62474	0.07773	2.20850	-1.16619	0.06400
FELDVR	0.43105	3.17950	-0.33682	1.04043	-0.19762	0.00000	-0.07500	1.91947	-0.46091	-0.26800	-0.86095	1.08400
FEL	0.87053	0.34900	-0.71182	0.74696	-0.22476	0.34000	-0.10600	1.89842	-0.48773	-0.23600	-0.65095	0.28000
FLFL	0.74316	0.45600	-0.67682	0.85826	-1.06905	0.20200	-0.11800	1.18526	-0.07545	0.11800	-0.82048	0.62350
PROVOGE	1.07947	-0.10900	-0.46591	0.97478	-0.44667	-0.32400	0.20850	-0.18632	1.66682	-0.20700	-1.47667	-0.13100
SHOPERSTOP	-0.78000	0.72700	-0.06591	0.07522	-0.18857	-0.10150	0.16700	0.08895	0.15409	-0.08700	-0.67952	0.29550
SORILINFRA	0.05158	-0.14600	1.38955	1.10696	-1.29905	-0.21300	0.65100	0.94158	-0.41909	1.99850	-0.47190	1.15950
TRENT	-1.31632	0.27200	-0.23000	0.36609	0.11190	0.28400	0.34650	0.79263	0.38591	-0.56400	-0.39381	0.56350
V2RETAIL	0.74737	-0.55200	1.73182	2.21478	-0.36762	0.27400	-0.45200	0.60053	0.10727	-0.52050	-1.44905	0.72950
VMART	-0.12263	-0.49450	-0.02455	0.26043	-0.40476	0.19500	-0.12600	-0.24947	1.61273	-0.27700	-1.40762	0.36500

Table 3 Expected return of assets

Company	Average monthly return
ABFRL	0.16171
CANTABIL	0.29716
FELDVR	0.45460
FEL	0.17230
FLFL	0.11887
PROVOGE	0.04858
SHOPERSTOP	-0.03290
SORILINFRA	0.39580
TRENT	0.05153
V2RETAIL	0.25534
VMART	-0.05611

5 Discussion of Experimental Results

In order to solve the model $M(2)$, the expected value of return, i.e. r_0 , should be assigned. In Sect. 2, it has been mentioned that value of r_0 lies between r_{\min} and r_{\max} . So, the above problem has been solved in two parts.

In solution phase I, the range of r_0 is determined. r_{\min} is calculated by omitting the constraint represented by Eq. (5) from model $M(2)$ and solving the remaining model using CSMO by using the parameter setting and termination criterion described in Sect. 4.1 of Sect. 4. The computational result has been provided in Table 5 and based on this result, the value of r_{\min} can be calculated using Eq. (5). r_{\min} is 0.114457. From Table 3, it can be seen that r_{\max} is 0.4546. Thus, we have obtained the range in which r_0 lies.

In solution phase II, the objective function value, i.e. risk, has been minimized for different values of r_0 between r_{\min} and r_{\max} . Ten uniform random numbers have been generated in $[r_{\min}, r_{\max}]$. These ten random numbers give ten different values of r_0 . By using these ten values of r_0 in Eq. (5) one by one, ten different portfolios have been generated by solving the model $M(2)$ using CSMO and the results have been summarized in Table 6. This table contains the expected portfolio return, the proportion of fund to be invested in a particular asset and the associated risk. It can be observed that the portfolio risk level increases with an increase in the expected portfolio return. This relationship always holds in portfolio optimization problem. The average execution time taken by CSMO per run (in seconds) is given in Table 7. From the table, it can be seen that the execution time for generating these 10 portfolios is very small.

The efficient frontier of the obtained portfolios has been shown in Fig. 1. X-axis represents the different values of expected returns, i.e. r_0 , and Y-axis represents the associated risk presented in Table 7. It can also be seen that as the level of expected return increases, the level of risk also increases.

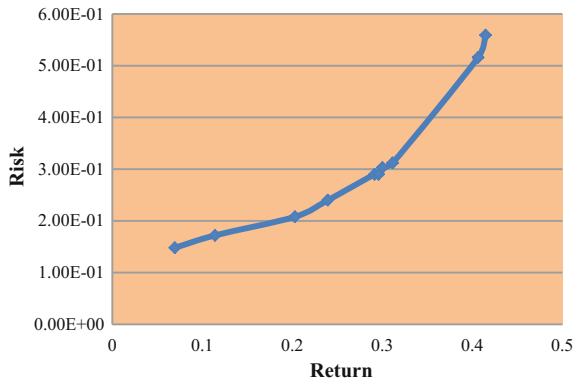
Table 4 Variance and covariance matrix

	ABFRL	CANTABIL	FELDVR	FEL	FLFL	PROVOGE	SHOPERSTOP	SORILINFRA	TRENT	V2RETAIL	VMART
ABFRL	1.28733	0.32512	0.96732	0.30506	0.44927	0.39730	0.33890	0.20332	0.22915	0.42335	0.31637
CANTABIL	0.32512	0.84620	0.08426	0.04178	0.16766	0.32540	0.02563	0.27929	-0.20748	-0.03959	0.07751
FELDVR	0.96732	0.08426	1.24506	0.53418	0.51028	0.09637	0.27569	0.11821	0.23926	0.19023	-0.04089
FEL	0.30506	0.04178	0.53418	0.51375	0.41437	0.13632	0.03128	0.10967	0.09818	0.24445	0.00358
FLFL	0.44927	0.16766	0.51028	0.41437	0.45578	0.24846	0.07455	0.26494	0.07977	0.28799	0.11858
PROVOGE	0.39730	0.32540	0.09637	0.13632	0.24846	0.64313	0.04808	0.02623	0.00280	0.35585	0.42780
SHOPERSTOP	0.33890	0.02563	0.27569	0.03128	0.07455	0.04808	0.14967	0.07379	0.15905	0.03754	0.09046
SORILINFRA	0.20332	0.27929	0.11821	0.10967	0.26494	0.02623	0.07379	0.84003	0.00414	0.40828	0.05100
TRENT	0.22915	-0.20748	0.23926	0.09818	0.07977	0.00280	0.15905	0.00414	0.31191	0.06612	0.12249
V2RETAIL	0.42335	-0.03959	0.19023	0.24445	0.28799	0.35585	0.03754	0.40828	0.06612	0.97290	0.30167
VMART	0.31637	0.07751	-0.04089	0.00358	0.11858	0.42780	0.09046	0.05100	0.12249	0.30167	0.44687

Table 5 Result of portfolio selection using variance

Risk	Allocation	Value
0.12251	x_1	0
	x_2	0.03842
	x_3	0
	x_4	0.22077
	x_5	0.02689
	x_6	0.03302
	x_7	0.28191
	x_8	0.09185
	x_9	0.03031
	x_{10}	0.1512
	x_{11}	0.12555

Fig. 1 Efficient frontier



6 Conclusions

CSMO has been applied to solve Markowitz’s mean-variance model. The above-explained procedure to solve the portfolio optimization problem can be helpful in two ways. First, if the investor has a particular choice for the expected return in advance without having the concern for the associated risk, then the optimal portfolios can be generated directly using the solution phase II. If the investor does not have a particular choice and want to see different possible portfolio returns with associated risks, then the problem can be solved using the above method in which various portfolios can be generated and the investor can choose any portfolio according to his/her choice. In Markowitz’s mean-variance model, variance has been taken as a measure for risk. In future, other optimization models based on different risk measures can be considered for experiment.

In this paper, the results of CSMO have not been compared with any other meta-heuristic algorithms. The reason which is explained in the forthcoming lines depends upon the case when the investor does not have a predefined choice of portfolio return.

Table 7 Average execution time taken by CSMO per run (in seconds)

Portfolio	Execution time
Portfolio 1	0.14
Portfolio 2	0.13624
Portfolio 3	0.13688
Portfolio 4	0.14124
Portfolio 5	0.13688
Portfolio 6	0.13816
Portfolio 7	0.13812
Portfolio 8	0.14008
Portfolio 9	0.14064
Portfolio 10	0.14376

It can be seen from the solution procedure explained in Sect. 5 that the final solution is obtained after solution phase II and the input for the solution phase II is generated from the results of solution phase I. So, we cannot compare different algorithms here because the results generated by different algorithms after solution phase I will be different. Consequently, different algorithms will have different input values for solution phase II and it is not fair to compare the results if the input values are different. But if an investor has a particular expected return in mind, then comparison can be made among different algorithms. But this case has been avoided here because it can be seen as biasness towards the selection of input value.

Also, the model considered for solving portfolio optimization model is the Markowitz's mean-variance model which is the most basic model which has limitations also [6]. There are various other advanced models which overcome the limitations of this basic portfolio optimization model with better risk measures for solving portfolio optimization problems [2, 3, 8–10]. But there are few reasons for choosing this optimization model for experiment in comparison to various other advanced versions of portfolio optimization models. Portfolio optimization problem is one of the most prominent optimization problems with varying complexities depending upon the portfolio optimization model under consideration. So, application of SMO for solving it will introduce it to the researchers working in the field of finance for using this algorithm for different types of financial optimization problems. Moreover, in order to develop SMO as a powerful tool for solving portfolio optimization problems, it is necessary to study its behaviour on the basic portfolio optimization models. This study will help in recognizing the strengths and limitations of SMO in solving these problems. These limitations can be overcome, and better versions of SMO can be designed for solving different types of financial problems.

References

1. Bansal, J.C., Sharma, H., Jadon, S.S., Clerc, M.: Spider monkey optimization algorithm for numerical optimization. *Memetic Comput.* **6**(1), 31–47 (2014)
2. Chang, T.J., Yang, S.C., Chang, K.J.: Portfolio optimization problems in different risk measures using genetic algorithm. *Expert Syst. Appl.* **36**(7), 10529–10537 (2009)
3. Cura, T.: Particle swarm optimization approach to portfolio optimization. *Nonlinear Anal. Real World Appl.* **10**(4), 2396–2406 (2009)
4. Deb, K.: An efficient constraint handling method for genetic algorithms. *Comput. Methods Appl. Mech. Eng.* **186**(2), 311–338 (2000)
5. Gupta, K., Deep, K., Bansal, J.C.: Spider monkey optimization algorithm for constrained optimization problems. In: *Soft Computing*, pp. 1–30. Springer (2016)
6. Gupta, P., Mehlawat, M. K., Inuiguchi, M., Chandra, S.: Fuzzy portfolio optimization. In: *Studies in Fuzziness and Soft Computing*, vol. 316 (2014)
7. Markowitz, H.: Portfolio selection. *J. Fin.* **7**(1), 77–91 (1952)
8. Niu, B., Fan, Y., Xiao, H., Xue, B.: Bacterial foraging based approaches to portfolio optimization with liquidity risk. *Neurocomputing* **98**, 90–100 (2012)
9. Wang, Z., Liu, S., Kong, X.: Artificial bee colony algorithm for portfolio optimization problems. *Int. J. Adv. Comput. Technol.* **4**(4), 8–16 (2012)
10. Zhu, H., Wang, Y., Wang, K., Chen, Y.: Particle swarm optimization (PSO) for the constrained portfolio optimization problem. *Expert Syst. Appl.* **38**(8), 10161–10169 (2011)

Optimal Configuration Selection in Reconfigurable Manufacturing System



Kamal Kumar Mittal, Pramod Kumar Jain and Dinesh Kumar

Abstract Reconfigurable manufacturing system (RMS) is considered as a major resource of providing variable production capacities and capabilities by different manufacturing companies. For different products needed in small quantities and with short delivery lead time, this is achieved through reconfiguring the system elements over the time. In the present work, various characteristics of RMS have been discussed and formulated. Weighted sum theory has been used for the selection of best manufacturing system. An illustration is given to analyze the applicability of the proposed methodology on a given system.

Keywords Reconfigurable manufacturing system (RMS) · Products Reconfigurable machine tool (RMT) · Convertibility · Diagnosability

1 Introduction

For quick production of products with high gain and low cost, the conventional manufacturing systems are not sufficient in a volatile and competitive market. To overcome these issues and become more responsive, a new type of manufacturing system, i.e., reconfigurable manufacturing system (RMS), has been proposed. Koren et al. [10] have described a reconfigurable manufacturing system as ideally consisting of the following characteristics: modularity, convertibility, diagnosability, scalability, customization, and integrability.

K. K. Mittal (✉)

Ajay Kumar Garg Engineering College, Ghaziabad 201001, Uttar Pradesh, India
e-mail: Kml900@rediffmail.com

P. K. Jain

IIITDM, Jabalpur 482005, Madhya Pradesh, India

D. Kumar

Mechanical and Industrial Engineering Department, Indian Institute of Technology, Roorkee 247667, Uttarakhand, India

© Springer Nature Singapore Pte Ltd. 2019

K. Deep et al. (eds.), *Decision Science in Action*, Asset Analytics,

https://doi.org/10.1007/978-981-13-0860-4_14

Since global market mostly focuses on outsource or insource production, it can rapidly and cost-efficiently react to random changes in the factory management. Each part of product needed different operations to give it final shape. These operations are done on single, double or multiple machines. The arrangement of basic modules and auxiliary modules of single, double or multiple machines in the system will develop the different configurations.

For global competition, the reconfigurability can be achieved at the system level and also at the machine level. At the system level, it can be achieved by changing the serial configuration into parallel configuration. Gupta et al. [6] have defined a specific system configuration for producing parts. At the machine level, it can be achieved by changing or readjusting the auxiliary modules. The index for combining the RMS characteristics decides which parameter needs more attention for increasing the reconfigurability of the RMS. A weighted sum theory is used to combine the various RMS characteristics.

2 Literature Review

The literature review has been carried out for the existing published work on modeling and design of reconfigurable manufacturing systems. A vast literature is available on the machine-level design issues but a very little work has been discussed to comprehensively model an RMS. The major difference between the design of RMS system and the design of other manufacturing system is that system configuration of RMS progress rapidly over the time period. The capacity and functionality in case of dedicated manufacturing system (DMS) and FMS are designed for the projected future requirements. But for global manufacturing systems, responsiveness is a crucial attribute that can be achieved by developing RMS that has a production capacity that is highly adaptable to market demand [4, 5, 16]. RMS is considered as the vibrant system, while DMS and FMS are said to be static systems because RMS offers the exact functionality and capacity, exactly when it is needed [14, 9, 15].

Scheduling in RMS also suggests the selection of best configuration during reconfiguration. According to Yu et al. [21], "Scheduling problems in a reconfigurable manufacturing system, a state-of-the-art manufacturing system designed at the beginning for fast alterations in its software and hardware components." He has suggested that total problem can be divided into subproblems. The subproblems are input sequencing and operation/machine selection. Erschler et al. [3], Hiltz [7], Smith and Steck [17], and Steck [19] have suggested various methodologies for input sequencing in manufacturing systems.

Koren et al. [11] have developed reconfigurable machine tools as modular machines comprising different modules. The RMTs have various combinations of basic modules and auxiliary modules. The RMTs can be reconfigured into many other configurations by keeping its base modules and just adding/removing or adjusting the auxiliary modules.

Abdi and Labib [1] used an analytical hierarchy process (AHP) for choosing the best manufacturing system among feasible alternative solutions based on an RMS study.

Son et al. [18] described a genetic algorithm approach for automatically generating machining system configurations. He suggested a capacity scalability approach for the homogeneous paralleling flow lines. He developed some alternative system configurations that pleased the demand for all demand periods, and then he developed many configuration paths selecting single-period configuration as starting point.

Clark and Paasch [2] presented a methodology based on diagnosability which detects and diagnoses the major reason for final product defects. It also rectified the defects rapidly.

Koren et al. [12] have studied system performance with respect to productivity and convertibility for different system configurations. This will give a quantitative measurement in terms of responsiveness.

RMS involves mainly the selection of configuration at machine level. RMS involving selection of configuration considering multiple machines simultaneously is very few and research on these systems has not explained the RMS characteristics. The detail of RMS characteristics is required for better understanding and obtaining the best configuration.

3 Methodology

In this section, different characteristics of RMS and responsiveness of machine have been quantified. Finally, an index is given to measure the reconfigurability both at system level and at machine level. The objective of this methodology is to find best manufacturing system.

3.1 RMS Characteristics

In this section, modularity, convertibility, and diagnosability characteristics of RMS have been discussed.

Modularity. According to Tanaka et al. [20], “In a reconfigurable manufacturing system, all major components are modular (e.g., structural elements, axes, controls, software, and tooling). When necessary, the components can be replaced or upgraded to better suit new applications.”

Holta et al. [8] have described modularity which depends upon the connectivity of different machines. The modularity is related to singular values of design system matrix (DSM). Design system matrix contains binary values. The value “1” is assigned to the machines which are connected to each other and vice versa value of “0” is assigned. From design system, matrix singular values are achieved after

applying singular value decomposition method. To measure modularity, singular value modularity index (SVMi) is used as

$$SVMi = 1 - \frac{1}{N * \sigma_1} \sum_{i=1}^{N-1} \sigma_i (\sigma_i - \sigma_{i+1}) \quad (1)$$

where N is the number of machines used in a particular manufacturing system configuration and σ_i are the singular values of the design system matrix.

Convertibility. Koren et al. [12] have explained convertibility in terms of ability of a system to regulate production functionality when a new product is introduced.

Convertibility Cc' can be measured as

$$Cc' = \frac{R * X}{I} \quad (2)$$

where R is the number of routing connections in each manufacturing system configuration of machines; X is the number of machine at a particular stage; and I is the least increment of conversion. The value of I decreases from serial configuration to parallel configuration.

The normalized value of Cc' is given by using equation

$$Cc'_{\text{normalised}} = 1 + \left[\frac{\log\left(\frac{Cc'_{\text{serial}}}{Cc'_{\text{parallel}}}\right)}{\log\left(\frac{Cc'_{\text{parallel}}}{Cc'_{\text{serial}}}\right) * 1/9} \right] \quad (3)$$

where Cc'_{serial} is convertibility values of a serial configuration and Cc'_{parallel} is convertibility values of a parallel configuration.

Diagnosability. Diagnosability is nothing but a property of a manufacturing system for checking and finding the reasons for product defects and therefore rectifies operational defects rapidly.

According to Kukushkin et al. [13], "As production systems are made more reconfigurable, and their layouts are modified more frequently, it becomes essential to rapidly tune the newly reconfigured system so that it produces quality parts."

Diagnosability (D) can be obtained using the following equation:

$$D = \frac{\sum_{i=1}^n \left[PI_i \left(\frac{1}{C_i} - \frac{1}{C_{\text{total}}} \right) \right]}{\left(1 - \frac{1}{C_{\text{total}}} \right) \sum_{i=1}^n PI_i} \quad (4)$$

where PI_i is the probability at each stage, C_i is the machine at each stage, and C_{total} is the total number of machines.

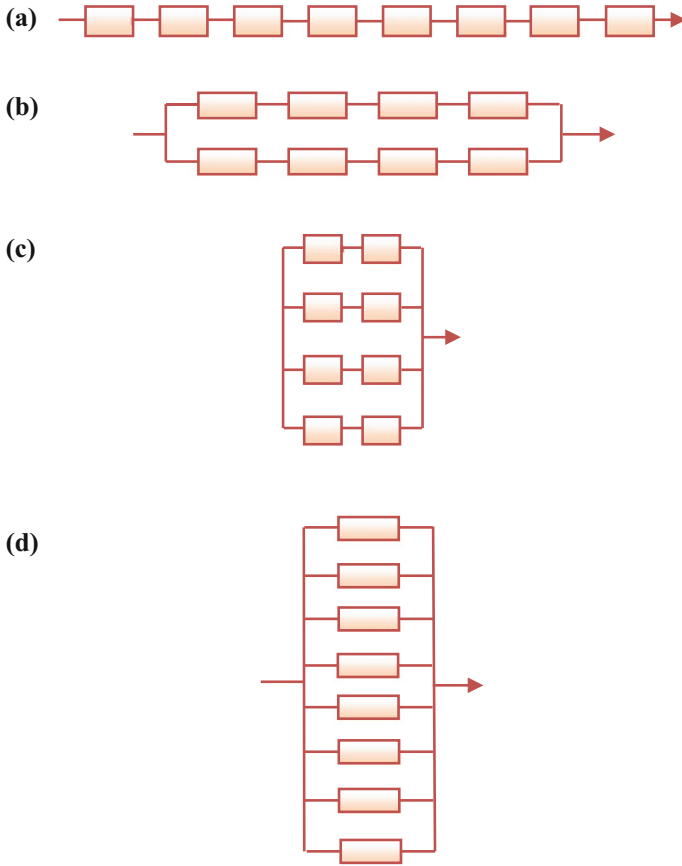


Fig. 1 Manufacturing system with eight machines

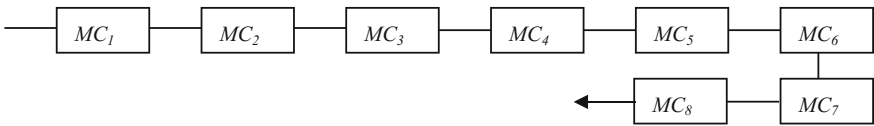


Fig. 2 Manufacturing system (a) with eight machines

Now to find the singular values applying singular value decomposition method on design system matrix using MATLAB.

S=	σ_1	0	0	0	0	0	0	0
	0	σ_2	0	0	0	0	0	0
	0	0	σ_3	0	0	0	0	0
	0	0	0	σ_4	0	0	0	0
	0	0	0	0	σ_5	0	0	0
	0	0	0	0	0	σ_6	0	0
	0	0	0	0	0	0	σ_7	0
	0	0	0	0	0	0	0	σ_8

Table 1 Modularity values for manufacturing systems (a) to (d)

System	σ_1	σ_2	σ_3	σ_4	σ_5	σ_6	σ_7	σ_8	Modularity
(a)	1.879	1.879	1.532	1.532	1.000	1.000	0.347	0.347	0.8589
(b)	1.618	1.618	1.618	1.618	0.618	0.618	0.618	0.618	0.8750
(c)	1.618	1.000	1.000	1.000	1.000	1.000	1.000	0.618	0.8932
(d)	0.000	0.000	0.000	0.000	0.000	0.00	0.00	0.00	1.0000

The different singular values are $\sigma_1 = 1.8794$, $\sigma_2 = 1.8794$, $\sigma_3 = 1.5321$, $\sigma_4 = 1.5321$, $\sigma_5 = 1.0000$, $\sigma_6 = 1.0000$, $\sigma_7 = 0.3473$, and $\sigma_8 = 0.3473$, respectively. Using Eq. 1, modularity is obtained as 0.8150. Table 1 shows the modularity of the different systems.

Step 2: Now, we have to find convertibility (Cc') of different manufacturing systems. For manufacturing system (a), it can easily be examined that $R=9$, $X=1$, and $I=1$. Using Eq. 2, convertibility is obtained as 9. Normalizing Cc' using the equation is given below:

$$\begin{aligned}
 Cc_{\text{normalised}} &= 1 + \left[\frac{\log\left(\frac{Cc'}{Cc'_{\text{serial}}}\right)}{\log\left(\frac{Cc'_{\text{parallel}}}{Cc'_{\text{serial}}}\right) * 1/9} \right] \\
 &= 1 + 0(Cc' = Cc'_{\text{serial}}) \\
 &= 1
 \end{aligned}$$

Table 2 shows the convertibility of the different systems.

Table 2 Convertibility values for manufacturing systems (a) to (d)

Systems	I	R	X	Cc'	Cc
(a)	1.00	9	1	9	1.000
(b)	0.50	10	2	40	3.836
(c)	0.25	12	4	192	6.818
(d)	0.125	16	8	1024	10.000

Table 3 Diagnosability values for manufacturing systems (a) to (d)

Systems	Types of machines	Probability at each stage	Diagnosability
(a)	8	1/8	1.00
(b)	4	1/4	0.4286
(c)	2	1/2	0.1429
(d)	1	1	0.00

Table 4 Reconfigurability values for manufacturing systems (a) to (d)

Systems	Modularity (<i>M</i>)	Convertibility (<i>C</i>)	Diagnosability (<i>D</i>)	Reconfigurability of systems (RS)
(a)	0.8589	0.100	1.00	0.6530
(b)	0.8750	0.3836	0.4286	0.5624
(c)	0.8932	0.6818	0.1429	0.5726
(d)	1.0000	1.0000	0.00	0.6666

Step 3: Next, we have to find diagnosability (*D*) of different manufacturing systems. The value of C_i for a serial manufacturing system is 1 and the total number of machine (C_{total}) is 8. It is assumed that probability at each stage is equal. Now using Eq. 4

$$D = \frac{\sum_{i=1}^8 [\frac{1}{8} (\frac{1}{1} - \frac{1}{8})]}{(\frac{1}{1} - \frac{1}{8}) \times 1} = 1$$

Table 3 shows the diagnosability of the different systems.

Step 4: Using Eq. 5 and taking equal weights of RMS characteristics, for a serial system

$$RS = (1/3) \times 0.8589 + (1/3) \times 0.1 (1/3) \times 1 = 0.6530$$

Table 4 shows the reconfigurability of the different systems.

The values of convertibility are divided by 10 to make it on a scale of 0–1.

From Fig. 3, reconfigurability of a pure parallel manufacturing system is highest among all the manufacturing system. Hence, system (d) is given priority for manufacturing of products in comparison to other systems.

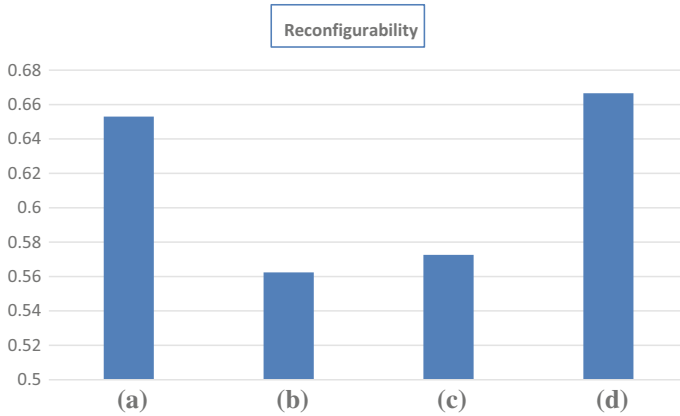


Fig. 3 Reconfigurability values for manufacturing systems (a) to (d)

5 Conclusion and Recommendation for Future Scope

It has been observed that reconfigurability depends upon both machine level and system level. The weightage of different characteristics also decides the reconfigurability values. If quality is given the priority, then diagnosability should be assigned with higher weightage than others.

All the attributes are first found out and then combined together to give reconfigurability index. It can be interpreted that system (d) is having the higher values.

This research is a base for future research on measuring reconfigurability of manufacturing systems. By quantifying other RMS characteristics, reconfigurability values can be improved. In the present work, six machines are considered to find the reconfigurability. This methodology can be applied by considering more than eight machines and the behavior of reconfigurability index can be compared. Besides that, effects of material handling system, tooling systems, etc. can also be added in evaluation of reconfigurability of the system.

References

1. Abdi, M.A., Labib, A.W.: A design strategy for reconfigurable manufacturing systems (RMSs) using analytical hierarchical process (AHP): a case study. *Int. J. Prod. Res.* **41**(10), 2273–2299 (2003)
2. Clark, G.E., Paasch, R.K.: Diagnostic modeling and diagnosability evaluation of mechanical systems. *J. Mech. Des.* **118**(3), 425–431 (1996)
3. Erschler, J., Lévêque, D., Roubellat, F.: Periodic loading of flexible manufacturing systems, pp. 327–339. IFIP Congress APMS, Bordeaux (1982)
4. Goyal, K.K., Jain, P.K., Jain, M.: A novel methodology to measure the responsiveness of RMTs in reconfigurable manufacturing system. *J. Manuf. Syst.* **32**(4), 724–730 (2013)

5. Gumasta, K., Gupta, S.K., Benyoucef, L., Tiwari, M.K.: Developing a reconfigurability index using multi-attribute utility theory. *Int. J. Prod. Res.* **49**(6), 1669–1683 (2011)
6. Gupta, A., Jain, P.K., Kumar, D.: A novel approach for part family formation using *K*-means algorithm. *Adv. Manuf.* **1**(3), 241–250 (2013)
7. Hiltz, K.L.: Scheduling of flexible flowshops. Technical Report LIDS-R-879, Massachusetts Institute of Technology (1979)
8. Holtta, K., Suh, E.S., De Weck, O.L.: Trade-off between modularity and performance for engineered systems and products. In: Proceedings of the 15th International Conference on Engineering Design, Melbourne, Australia, 15–18 Aug 2005
9. Kahloul, L., Bourekkache, S., Djouani, K.: Designing reconfigurable manufacturing systems using reconfigurable object Petri nets. *Int. J. Comput. Integr. Manuf.* **29**(8), 889–906 (2016)
10. Koren, Y., Hu, S.J., Weber, T.W.: Impact of manufacturing system configuration on Performance. *CIRP Ann.* **47**(1), 369–372 (1998)
11. Koren, Y., Hiesel, U., Jovane, F., Moriwaki, T., Pritschow, G., Ulsoy, G., Van, B.H.: Reconfigurable manufacturing systems. *CIRP Ann.* **48**(2), 527–540 (1999)
12. Koren, Y., Maler-Speredelozzi, V., Hu, S.J.: Convertibility measures for manufacturing systems. *CIRP Ann. Manuf. Technol.* **52**(1), 367–370 (2003)
13. Kukushkin, I.K., Katalinic, B., Cesarec, P., Kettler, R.: Reconfiguration in self-organizing systems. In: Proceedings of the 22nd International DAAAM Symposium, vol. 22, no. 1. Vienna, Austria, EU (2011)
14. Mehrabi, M.G., Ulsoy, K.: Reconfigurable manufacturing systems: key to future manufacturing. *J. Intell. Manuf.* **11**, 403–419 (2000)
15. Mehrabi, M.G., Ulsoy, A.G., Koren, Y., Heytler, P.: Trends and perspectives in flexible and reconfigurable manufacturing systems. *J. Intell. Manuf.* **13**(2), 135–146 (2002)
16. Mittal, K.K., Jain, P.K.: Impact of reconfiguration effort on reconfigurable manufacturing system. In: 5th International and 26th All India Manufacturing Technology, Design and Research Conference (AIMTDR), IIT Guwahati, Assam, India, 12–14 Dec 2014
17. Smith, T.M., Steck, K.E.: On the robustness of using balanced part mix ratios to determine cyclic part input sequence into flexible flow systems. *Int. J. Prod. Res.* **34**, 2925–2941 (1996)
18. Son, S.Y., Olsen, T.L., Hoi, D.Y.: An approach to scalability and line balancing for reconfigurable manufacturing systems. *Integr. Manuf. Syst.* **12**(7), 500–511 (2011)
19. Steck, K.E.: Procedures to determine part mix ratios for independent demands in flexible manufacturing systems. *IEEE Trans. Eng. Manage.* **39**, 359–369 (1992)
20. Tanaka, K., Kurahasi, M., Hayasi, M., Inao, S., Hibino, H., Fukuda, Y.: A study of the despatching order system to support module structured production system for the demand synchronized production. *J. Adv. Mech. Des. Syst. Manuf.* **4**(2), 504–515 (2010)
21. Yu, J.M., Doh, H.H., Kim, J.S., Kwon, J.Y., Lee, D.H., Nam, S.H.: Input sequencing and scheduling for a reconfigurable manufacturing system with a limited number of fixtures. *Int. J. Adv. Manuf. Technol.* **67**, 157–169 (2013)

A Comparative Study of Regularized Long Wave Equations (RLW) Using Collocation Method with Cubic B-Spline



Nini Maharana, A. K. Nayak and Pravakar Jena

Abstract A collocation technique is successfully formulated for regularized long wave equations (RLW). This method is based on the cubic B-spline finite element. The stability analysis has been discussed by using the Fourier method and shown to be marginally stable. The accuracy, efficiency, and the invariants of motion related to conservation of mass, momentum, and energy are investigated. We have also studied the propagation of single solitary wave motion and two solitary waves interaction. It has been observed that the obtained numerical results are acceptable and more accurate.

Keywords Collocation method · Cubic B-spline · Solitary wave equations

1 Introduction

The motions of the solitary wave were first presented in 1834 by John Scott Russel [1]. After a long year, the significance of this invention played an important role in the stable state of nonlinear system and then it has become a wide area of research in the field of numerical analysis.

Actually, this type of nonlinear system represents hump-shaped wave packets or pulses called as solitary wave and it appears in many areas such as physical phenomena, plasma physics, laser physics [2], optical fibers, and solid-state physics.

N. Maharana

Department of Mathematics, Ravenshaw University, Cuttack 753003, Odisha, India

e-mail: Ninimaharan01@gmail.com

A. K. Nayak (✉)

Department of Mathematics, Indian Institute of Technology, IIT Roorkee, Roorkee 247667, Uttarakhnad, India

e-mail: ameeya.iitkgp@gmail.com; ameeyakumar@gmail.com

P. Jena

Department of Mathematics, KIIT University, Bhubaneswar 75003, Odisha, India

e-mail: pravakar76@gmail.com

© Springer Nature Singapore Pte Ltd. 2019

K. Deep et al. (eds.), *Decision Science in Action*, Asset Analytics,

https://doi.org/10.1007/978-981-13-0860-4_15

A soliton is a solitary wave packet that retains its shape and velocity properties after a collision with other solitons.

The RLW equation can be represented as

$$\frac{\partial U(x, t)}{\partial t} + \frac{\partial U(x, t)}{\partial x} + \delta U \frac{\partial U(x, t)}{\partial x} - \mu \frac{\partial}{\partial t} \left(\frac{\partial^2 U(x, t)}{\partial x^2} \right) = 0 \quad (1)$$

where μ and δ are positive constants, is a class of partial differential equation, which has soliton solution. This equation was originally introduced by Peregrine [3]. Equation (1) has been solved by different types of finite element method methods like Galerkin method, least square method, homotopy perturbation method, differential transformation method, subdomain method, Petrov Galerkin method, collocation method.

Consider the following Generalized RLW equation:

$$\frac{\partial U(x, t)}{\partial t} + \frac{\partial U(x, t)}{\partial x} + \delta U^P(x, t) \frac{\partial U(x, t)}{\partial x} - \mu \frac{\partial}{\partial t} \left(\frac{\partial^2 U(x, t)}{\partial x^2} \right) = 0 \quad (2)$$

where P is a positive integer.

The GRLW equation is numerically evaluated by Zhang [4] and Kaya [5] using finite difference method for a Cauchy problem and the Adomian decomposition method (ADM), respectively. Gardner et al. [6] discussed a numerical computation scheme for modified regularized long wave (MRLW) equation using the collocation method with quintic B-spline.

The RLW equation, the KdV (Korteweg de Vries) equation, the EWE (Equal Width Wave) equation are mostly in nonlinear dispersive form in which they have solitary wave solution. So, in the present work, we will set up the general forms of the GRLW equations giving solitary wave solution.

In fact, finding the analytical solutions for the nonlinear equations generally is difficult and probably impossible for the propagation of more than one solitary wave. So, evaluating the accurate approximate solutions for these equations are the main aims for many researchers in order to study solitary waves on a wide range and to investigate their properties.

There are numerous techniques to evaluate the approximate solution and its application of the nonlinear differential equations by collocation method with a cubic spline and cubic B-spline. So, some properties of the cubic spline collocation method are summarized as follows:

- (1) The resulting framework leads to a diagonal structure to provide an easy implementation of algorithms.
- (2) Cubic spline method is formulated in a very easy way and the computational cost is also very low.

Saka and Dag [7] investigated modified cubic B-spline and splitting method for the numerical solution of RLW equation. Solutions based on collocation method using cubic B-spline are discussed by Raslan [8].

Jain et al. [9] provide an approximate solution for the propagation of single solitary wave based on collocation algorithm. The motion of interactions of solitary waves and development of an undular bore is studied by Bhardwaj et al. [10]. Mittal and Jain [11] continued the same method to obtain the solution of nonlinear Fisher’s reaction–diffusion equation and Zaid et al. [12] have used a numerical method based on the simplification of Laplace ADM.

All of the above-mentioned works deals with collocation method with spline to develop the numerical solution for RLW, GRLW, or MRLW equation. In this paper, we have tried to implement the method of collocation together with B-spline approximation in cubic sense for the solution of **GRLW** equation and the solution procedure is tested by implementing some test functions.

2 Exact Solution of the GRLW Equation

Let us take the trial function

$$U(x, t) = f(\eta) \tag{3}$$

where $\eta = x - vt$ and v represents the constant velocity of a wave.

Then from Eq. (3), we get

$$\frac{\partial U}{\partial t} = -v \frac{\partial f}{\partial \eta}, \quad \frac{\partial U}{\partial x} = \frac{\partial f}{\partial \eta}, \quad \frac{\partial^2 U}{\partial x^2} = \frac{\partial^2 f}{\partial \eta^2}, \quad \frac{\partial}{\partial t} \left(\frac{\partial^2 U}{\partial x^2} \right) = -v \frac{\partial^3 f}{\partial \eta^3} \tag{4}$$

Substituting Eq. (4) into Eq. (2), we get

$$(1 - v) \frac{\partial f}{\partial \eta} + \delta f^P \frac{\partial f}{\partial \eta} + \mu v \frac{\partial^3 f}{\partial \eta^3} = 0 \tag{5}$$

Integrating Eq. (5) w.r.t η , it follows that

$$(1 - v)f + \frac{\delta}{(P + 1)} f^{P+1} + \mu v \frac{\partial^2 f}{\partial \eta^2} = g_1 \tag{6}$$

where g_1 is a constant of integration. Now, we multiply Eq. (6) by $2 \frac{\partial f}{\partial \eta}$

$$2(1 - v)f \frac{\partial f}{\partial \eta} + \frac{2\delta}{(P + 1)} f^{P+1} \frac{\partial f}{\partial \eta} + 2\mu v \frac{\partial f}{\partial \eta} \frac{\partial^2 f}{\partial \eta^2} = 2g_1 \frac{\partial f}{\partial \eta} \tag{7}$$

Integrating both sides w.r.t η , we obtain

$$(1 - v)f^2 + \frac{2\delta}{(P + 1)(P + 2)} f^{P+2} + \mu v \left(\frac{\partial f}{\partial \eta} \right)^2 = 2g_1 f + g_2 \tag{8}$$

where g_2 is another constant of integration.

Suppose that $f \rightarrow 0 \frac{\partial f}{\partial \eta} \rightarrow 0$ as $|\eta| \rightarrow \infty$, so the constants of integration g_1 and g_2 are zero, then from Eq. (8),

$$\left(\frac{\partial f}{\partial \eta}\right)^2 = \frac{f^2}{\nu\mu}(\nu - 1 - mf^P), \quad \text{where } m = \frac{2\delta}{(P+1)(P+2)} \tag{9}$$

Taking positive square root on both sides, we have

$$\frac{\partial f}{\partial \eta} = \frac{f}{\sqrt{\mu(\nu+1)}}(\nu - mf^P)^{1/2} \text{ (replacing } \nu \text{ into } \nu + 1) \tag{10}$$

Integrating Eq. (10), we get

$$\int \frac{\partial f}{f(\nu - mf^P)^{1/2}} = \frac{1}{\sqrt{\mu(\nu+1)}} \int \partial \eta \tag{11}$$

Using the transformation, $mf^P = \nu \sec h^2\theta$, so that $mPf^{P-1}\partial f = 2\nu \sec h^2\theta \tanh\theta \partial\theta$, we get

$$\frac{2}{P\sqrt{\nu}} \int \partial\theta = \frac{1}{\sqrt{\mu(\nu+1)}} \int \partial\eta \tag{12}$$

$\Rightarrow \theta = \frac{P}{2} \sqrt{\frac{\nu}{\mu(\nu+1)}}(\eta - g_3)$, where g_3 is another constant of integration.

$$f^P = \frac{\nu}{m} \sec h^2 \left[\frac{P}{2} \sqrt{\frac{\nu}{\mu(\nu+1)}}(x - (\nu+1)t - g_3) \right]$$

$$U(x, t) = \left(\frac{\nu(P+1)(P+2)}{2\delta} \sec h^2 \left[\frac{P}{2} \sqrt{\frac{\nu}{\mu(\nu+1)}}(x - (\nu+1)t - g_3) \right] \right)^{1/P} \tag{13}$$

which is the exact solution of Eq. (2).

Taking $P = 1$, the exact solution for RLW equation is given by

$$U(x, t) = \frac{3\nu}{\delta} \sec h^2 \left[\frac{1}{2} \sqrt{\frac{\nu}{\mu(\nu+1)}}(x - (\nu+1)t - g_3) \right] \tag{14}$$

3 Implementation of the Proposed Method

Substituting $\delta = 1$ and $P = 1$ in Eq. (2), we obtained the RLW equation as follows:

$$\frac{\partial U(x, t)}{\partial t} + \frac{\partial U(x, t)}{\partial x} + U(x, t) \frac{\partial U(x, t)}{\partial x} - \mu \frac{\partial}{\partial t} \left(\frac{\partial^2 U(x, t)}{\partial x^2} \right) = 0 \tag{15}$$

Table 1 The cubic B-splines $\varphi_j(x)$ and its derivatives $\varphi'_j(x), \varphi''_j(x)$ at different node points

x	x_{j-2}	x_{j-1}	x_j	x_{j+1}	x_{j+2}
φ_j	0.0	1.0	4.0	1.0	0.0
φ'_j	0.0	$\frac{3}{h}$	0.0	$-\frac{3}{h}$	0.0
φ''_j	0.0	$\frac{6}{h^2}$	$-\frac{12}{h^2}$	$\frac{6}{h^2}$	0.0

subject to the initial and boundary conditions

$$U(x, 0) = f(x), \quad J_1 < x < J_2 \tag{16}$$

$$U(J_1, t) = \psi_0(t), \quad U(J_2, t) = \psi_1(t), \quad 0 \leq t \leq T \tag{17}$$

and

$$U \rightarrow 0 \text{ as } x \rightarrow \pm\infty, \quad t > 0 \tag{18}$$

Let us consider the domain $[J_1, J_2]$, which is uniformly partitioned at the knots x_j , such that

$$J_1 = x_0 < x_1 < \dots < x_N = J_2, \quad h = x_{j+1} - x_j = \frac{J_2 - J_1}{N}, \quad j = 0, \dots, N.$$

Let $\{\varphi_j\}_{j=-1}^{N+1}$ be the cubic B-splines defined at the knots x_j ; and the group of splines formulates a basis function over $[J_1, J_2]$. The solution in global form, $U^N(x, t)$ is expressed with cubic B-splines as

$$U^N(x, t) = \sum_{j=-1}^{N+1} \xi_j(t) \varphi_j(x) \tag{19}$$

where ξ_j are transient parameters and determined using boundary, initial, and the collocation points. The cubic B-splines $\varphi_j(x)$ and its derivatives $\varphi'_j(x), \varphi''_j(x)$ at node points are shown in Table 1.

Using Eq. (19) and the above-tabulated values, we can calculate the nodal values U_j and its first and second derivatives U'_j and U''_j , respectively, at the mesh x_j in the form of ξ_j as follows:

$$\begin{aligned} U_j &= \xi_{j-1} + 4\xi_j + \xi_{j+1} \\ U'_j &= \frac{3}{h}(\xi_{j+1} - \xi_{j-1}) \\ U''_j &= \frac{6}{h^2}(\xi_{j-1} - 2\xi_j + \xi_{j+1}) \end{aligned} \tag{20}$$

Now, Eq. (15) can be represented as

$$\frac{\partial}{\partial t} \left(U - \mu \frac{\partial^2 U}{\partial x^2} \right) + \frac{\partial U}{\partial x} + u \frac{\partial U}{\partial x} = 0 \tag{21}$$

By using a finite difference scheme, we can approximate the time derivative by

$$\frac{\partial}{\partial t} U = \frac{U^{n+1} - U^n}{k}, \text{ where } k = \Delta t = t_{n+1} - t_n$$

Then, let us consider $U = \frac{U^{n+1} + U^n}{2}$. So, Eq. (21) yields

$$\begin{aligned} & U^{n+1} - \mu \left(\frac{\partial^2 U}{\partial x^2} \right)^{n+1} - U^n + \mu \left(\frac{\partial^2 U}{\partial x^2} \right)^n \\ & + \frac{k}{2} \left(\left(\frac{\partial U}{\partial x} \right)^{n+1} + \left(U \frac{\partial U}{\partial x} \right)^{n+1} + \left(\frac{\partial U}{\partial x} \right)^n + \left(U \frac{\partial U}{\partial x} \right)^n \right) = 0 \end{aligned} \tag{22}$$

Now, linearizing the nonlinear term

$$\left(U \frac{\partial U}{\partial x} \right)_j^{n+1} = U_j^{n+1} \left(\frac{\partial U}{\partial x} \right)_j^n + U_j^n \left(\frac{\partial U}{\partial x} \right)_j^{n+1} - U_j^n \left(\frac{\partial U}{\partial x} \right)_j^n \tag{23}$$

Then Eq. (22) becomes

$$\begin{aligned} & U_j^{n+1} - \mu \left(\frac{\partial^2 U}{\partial x^2} \right)_j^{n+1} - U_j^n + \mu \left(\frac{\partial^2 U}{\partial x^2} \right)_j^n + \frac{k}{2} \left(\left(\frac{\partial U}{\partial x} \right)_j^{n+1} + U_j^{n+1} \left(\frac{\partial U}{\partial x} \right)_j^n + U_j^n \left(\frac{\partial U}{\partial x} \right)_j^{n+1} \right) \\ & - \frac{k}{2} \left(U_j^n \left(\frac{\partial U}{\partial x} \right)_j^n + U_j^n \left(\frac{\partial U}{\partial x} \right)_j^{n-1} + U_j^{n-1} \left(\frac{\partial U}{\partial x} \right)_j^n - U_j^{n-1} \left(\frac{\partial U}{\partial x} \right)_j^{n-1} + \left(\frac{\partial U}{\partial x} \right)_j^n \right) = 0 \end{aligned} \tag{24}$$

Substituting Eq. (20) into Eq. (24) yields

$$\begin{aligned} & \left(\xi_{j-1}^{n+1} + 4\xi_j^{n+1} + \xi_{j+1}^{n+1} \right) - \frac{6\mu}{h^2} \left(\xi_{j-1}^{n+1} - 2\xi_j^{n+1} + \xi_{j+1}^{n+1} \right) - \left(\xi_{j-1}^n + 4\xi_j^n + \xi_{j+1}^n \right) \\ & + \frac{6\mu}{h^2} \left(\xi_{j-1}^n - 2\xi_j^n + \xi_{j+1}^n \right) + \frac{k}{2} \left[\frac{3}{h} \left(\xi_{j+1}^{n+1} - \xi_{j-1}^{n+1} \right) + \left(\xi_{j-1}^{n+1} + 4\xi_j^{n+1} + \xi_{j+1}^{n+1} \right) \left(\frac{3}{h} \left(\xi_{j+1}^n - \xi_{j-1}^n \right) \right) \right. \\ & + \left(\xi_{j-1}^n + 4\xi_j^n + \xi_{j+1}^n \right) \left(\frac{3}{h} \left(\xi_{j+1}^{n+1} - \xi_{j-1}^{n+1} \right) \right) - \left. \left(\xi_{j-1}^n + 4\xi_j^n + \xi_{j+1}^n \right) \left(\frac{3}{h} \left(\xi_{j+1}^n - \xi_{j-1}^n \right) \right) \right] \\ & + \left(\xi_{j-1}^n + 4\xi_j^n + \xi_{j+1}^n \right) \left(\frac{3}{h} \left(\xi_{j+1}^{n-1} - \xi_{j-1}^{n-1} \right) \right) + \left(\xi_{j-1}^{n-1} + 4\xi_j^{n-1} + \xi_{j+1}^{n-1} \right) \left(\frac{3}{h} \left(\xi_{j+1}^n - \xi_{j-1}^n \right) \right) \\ & - \left. \left(\xi_{j-1}^{n-1} + 4\xi_j^{n-1} + \xi_{j+1}^{n-1} \right) \left(\frac{3}{h} \left(\xi_{j+1}^{n-1} - \xi_{j-1}^{n-1} \right) \right) + \frac{3}{h} \left(\xi_{j+1}^n - \xi_{j-1}^n \right) \right] = 0 \end{aligned} \tag{25}$$

Equation (25) can be re-written as

$$\begin{aligned}
 & \left(1 - \frac{6\mu}{h^2} - \frac{3k}{2h} + \frac{k}{2}L_2^j - \frac{3k}{2h}L_1^j\right)\xi_{j-1}^{n+1} + \left(4 + \frac{12\mu}{h^2} + 2kL_2^j\right)\xi_j^{n+1} \\
 & + \left(1 - \frac{6\mu}{h^2} + \frac{3k}{2h} + \frac{k}{2}L_2^j + \frac{3k}{2h}L_1^j\right)\xi_{j+1}^{n+1} \\
 & = L_1^j - \frac{6\mu}{h^2}L_5^j + \frac{k}{2}L_1^jL_2^j - \frac{k}{2}L_2^j - \frac{k}{2}L_1^jL_4^j - \frac{k}{2}L_3^jL_2^j + \frac{k}{2}L_3^jL_4^j, \quad 0 \leq j \leq N
 \end{aligned}
 \tag{26}$$

where we have

$$L_1^j = \xi_{j-1}^n + 4\xi_j^n + \xi_{j+1}^n, \quad L_2^j = \frac{3}{h}(\xi_{j+1}^n - \xi_{j-1}^n), \quad L_3^j = \xi_{j-1}^{n-1} + 4\xi_j^{n-1} + \xi_{j+1}^{n-1}$$

$$L_4^j = \frac{3}{h}(\xi_{j+1}^{n-1} - \xi_{j-1}^{n-1}), \quad L_5^j = \xi_{j-1}^n - 2\xi_j^n + \xi_{j+1}^n, \quad j = 0, 1, \dots, N$$

Also, if we put

$$b_j = 1 - \frac{6\mu}{h^2} - \frac{3k}{2h} + \frac{k}{2}L_2^j - \frac{3k}{2h}L_1^j, \quad c_j = 4 + \frac{12\mu}{h^2} + 2kL_2^j$$

$$d_j = 1 - \frac{6\mu}{h^2} + \frac{3k}{2h} + \frac{k}{2}L_2^j + \frac{3k}{2h}L_1^j,$$

$$G_j = L_1^j - \frac{6\mu}{h^2}L_5^j + \frac{k}{2}L_1^jL_2^j - \frac{k}{2}L_2^j - \frac{k}{2}L_1^jL_4^j - \frac{k}{2}L_3^jL_2^j + \frac{k}{2}L_3^jL_4^j,$$

Then, the system (26) can be written as

$$b_j\xi_{j-1}^{n+1} + c_j\xi_j^{n+1} + d_j\xi_{j+1}^{n+1} = G_j, \quad j = 0, 1, \dots, N \tag{27}$$

These equations establish a recurrence relation with parametric values and can be expressed in a vector form as $d^n = (\xi_{-1}, \xi_0, \xi_1, \dots, \xi_N)$, where $(N + 1)$ equations involve $(N + 3)$ unknowns at n th time level, satisfying $U(J_1, t) = \psi_0(t), U(J_2, t) = \psi_1(t)$

$$\begin{aligned}
 \xi_{-1} &= \psi_0 - 4\xi_0 - \xi_1 \\
 \xi_{N+1} &= \psi_1 - 4\xi_N - \xi_{N-1}
 \end{aligned}
 \tag{28}$$

Using the Eq. (28), we can eliminate the parameters ξ_{-1} and ξ_{N+1} , and hence the system of equations represented by Eq. (27) can be represented as

$$\begin{bmatrix} c_0 - 4b_0 & d_0 - b_0 & 0 & 0 & \dots & 0 & 0 & 0 \\ b_1 & c_1 & d_1 & 0 & \dots & 0 & 0 & 0 \\ \cdot & \cdot & \cdot & \dots & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \dots & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & \dots & b_{N-1} & c_{N-1} & d_{N-1} & \\ 0 & 0 & 0 & \dots & 0 & b_N - d_N & c_N - 4d_N & \end{bmatrix} \begin{bmatrix} \xi_0^{n+1} \\ \xi_1^{n+1} \\ \cdot \\ \cdot \\ \cdot \\ \xi_{N-1}^{n+1} \\ \xi_N^{n+1} \end{bmatrix} = \begin{bmatrix} G_0 - b_0 \psi_0 \\ G_1 \\ \cdot \\ \cdot \\ G_{N-1} \\ G_N - d_N \psi_1 \end{bmatrix}$$

The above matrix represents a tridiagonal system. From Eq. (14), we can evaluate the initial conditions at two time levels ($t = 0, k$)

$$f(x) = U(x, 0) = \frac{3\nu}{\delta} \sec h^2 \left[\frac{1}{2} \sqrt{\frac{\nu}{\mu(\nu + 1)}} (x - g_3) \right] \tag{29}$$

$$\zeta(x) = U(x, k) = \frac{3\nu}{\delta} \sec h^2 \left[\frac{1}{2} \sqrt{\frac{\nu}{\mu(\nu + 1)}} (x - (\nu + 1)k - g_3) \right] \tag{30}$$

At time level $n = 0(t = 0)$

$$U^N(x_j, 0) = \sum_{j=-1}^{N+1} \xi_j^0 \varphi_j(x_j) = \xi_{j-1}^0 + 4\xi_j^0 + \xi_{j+1}^0 = f(x_j), \quad j = 0, 1, \dots, N \tag{31}$$

Also, at time level $n = 1(t = k)$

$$U^N(x_j, k) = \sum_{j=-1}^{N+1} \xi_j^1 \varphi_j(x_j) = \xi_{j-1}^1 + 4\xi_j^1 + \xi_{j+1}^1 = \zeta(x_j), \quad j = 0, 1, \dots, N \tag{32}$$

In order to eliminate the unknowns from the Eqs. (31) and (32), we apply the following boundary conditions at the initial level as

$$U_x(x_0, 0) = 0 = U_x(x_N, 0) \tag{33}$$

$$U_x(x_0, k) = 0 = U_x(x_N, k) \tag{34}$$

From the boundary conditions (33) and with Eq. (20), we get

$$\left. \begin{aligned} \frac{3}{h} \xi_1^0 - \frac{3}{h} \xi_{-1}^0 = 0 &\Rightarrow \xi_{-1}^0 = \xi_1^0 \\ \frac{3}{h} \xi_{N+1}^0 - \frac{3}{h} \xi_{N-1}^0 = 0 &\Rightarrow \xi_{N+1}^0 = \xi_{N-1}^0 \end{aligned} \right\} \tag{35}$$

From the boundary conditions (34) and with Eq. (20), we get

$$\left. \begin{aligned} \frac{3}{h}\xi_1^1 - \frac{3}{h}\xi_{-1}^1 = 0 &\Rightarrow \xi_{-1}^1 = \xi_1^1 \\ \frac{3}{h}\xi_{N+1}^1 - \frac{3}{h}\xi_{N-1}^1 = 0 &\Rightarrow \xi_{N+1}^1 = \xi_{N-1}^1 \end{aligned} \right\} \tag{36}$$

From Eqs. (31) and (35), we get the following tridiagonal matrix system $(N + 1) \times (N + 1)$ as

$$\begin{bmatrix} 4 & 2 & 0 & 0 & \dots & 0 & 0 & 0 \\ 1 & 4 & 1 & 0 & \dots & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & \dots & 1 & 4 & 1 \\ 0 & 0 & 0 & 0 & \dots & 0 & 2 & 4 \end{bmatrix} \begin{bmatrix} \xi_0^0 \\ \xi_1^0 \\ \cdot \\ \cdot \\ \cdot \\ \xi_{N-1}^0 \\ \xi_N^0 \end{bmatrix} = \begin{bmatrix} f(x_0) + \frac{h}{3}f'(x_0) \\ f(x_1) \\ \cdot \\ \cdot \\ \cdot \\ f(x_{N-1}) \\ f(x_N) - \frac{h}{3}f'(x_N) \end{bmatrix} \tag{37}$$

By simple implementation of Thomas algorithms, we can obtain the solution for the above-mentioned tridiagonal system.

Similarly, from Eqs. (32) and (36), we also get the following tridiagonal matrix system $(N + 1) \times (N + 1)$, as

$$\begin{bmatrix} 4 & 2 & 0 & 0 & \dots & 0 & 0 & 0 \\ 1 & 4 & 1 & 0 & \dots & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & \dots & 1 & 4 & 1 \\ 0 & 0 & 0 & 0 & \dots & 0 & 2 & 4 \end{bmatrix} \begin{bmatrix} \xi_0^1 \\ \xi_1^1 \\ \cdot \\ \cdot \\ \cdot \\ \xi_{N-1}^1 \\ \xi_N^1 \end{bmatrix} = \begin{bmatrix} \zeta(x_0) + \frac{h}{3}\zeta'(x_0) \\ \zeta(x_1) \\ \cdot \\ \cdot \\ \cdot \\ \zeta(x_{N-1}) \\ \zeta(x_N) - \frac{h}{3}\zeta'(x_N) \end{bmatrix} \tag{38}$$

Hence, we can easily determine the initial time parameters ξ_j^0 and ξ_j^1 by solving the above tridiagonal system.

4 Stability Analysis

The stability analysis of our scheme is presented by applying the Fourier method. First, linearized the nonlinear term in Eq. (1) and rewrite the Eq. (22) as

$$\begin{aligned} &\xi_{j-1}^{n+1} + 4\xi_j^{n+1} + \xi_{j+1}^{n+1} - \frac{6\mu}{h^2}(\xi_{j-1}^{n+1} - 2\xi_j^{n+1} + \xi_{j+1}^{n+1}) - (\xi_{j-1}^n + 4\xi_j^n + \xi_{j+1}^n) \\ &+ \frac{6\mu}{h^2}(\xi_{j-1}^n - 2\xi_j^n + \xi_{j+1}^n) \\ &+ \frac{\Delta t}{2} \left[(1 + U) \left(\frac{3}{h}(\xi_{j+1}^{n+1} - \xi_{j-1}^{n+1}) \right) + (1 + U) \left(\frac{3}{h}(\xi_{j+1}^n - \xi_{j-1}^n) \right) \right] = 0 \end{aligned} \tag{39}$$

Or

$$\begin{aligned} &\left(\left(1 - \frac{6\mu}{h^2} \right) - \left(\frac{3\Delta t}{2h} + \frac{3U\Delta t}{2h} \right) \right) \xi_{j-1}^{n+1} + \left(4 + \frac{12\mu}{h^2} \right) \xi_j^{n+1} \\ &+ \left(\left(1 - \frac{6\mu}{h^2} \right) + \left(\frac{3\Delta t}{2h} + \frac{3U\Delta t}{2h} \right) \right) \xi_{j+1}^{n+1} \\ &= \left(\left(1 - \frac{6\mu}{h^2} \right) + \left(\frac{3\Delta t}{2h} + \frac{3U\Delta t}{2h} \right) \right) \xi_{j-1}^n + \left(4 + \frac{12\mu}{h^2} \right) \xi_j^n \\ &+ \left(\left(1 - \frac{6\mu}{h^2} \right) - \left(\frac{3\Delta t}{2h} + \frac{3U\Delta t}{2h} \right) \right) \xi_{j+1}^n \end{aligned} \tag{40}$$

$$j = 0, 1, \dots, N$$

Now, using the Fourier method

$$\xi_j^n = \hat{\chi}^n \alpha^{ikjh} \tag{41}$$

with k as mode number and h as element length, into the Eq. (41) yields

$$\begin{aligned} &(X - Z)\hat{\chi}^{n+1} \alpha^{ik(j-1)h} + Y\hat{\chi}^{n+1} \alpha^{ikjh} + (X + Z)\hat{\chi}^{n+1} \alpha^{ik(j+1)h} \\ &= (X + Z)\hat{\chi}^n \alpha^{ik(j-1)h} + Y\hat{\chi}^n \alpha^{ikjh} + (X - Z)\hat{\chi}^n \alpha^{ik(j+1)h} \end{aligned} \tag{42}$$

where

$$X = \left(1 - \frac{6\mu}{h^2} \right), \quad Y = \left(4 + \frac{12\mu}{h^2} \right), \quad Z = \left(\frac{3\Delta t}{2h} + \frac{3U\Delta t}{2h} \right)$$

Using the von Neumann stability theory, we will get the growth of Fourier mode as

$$\hat{\chi}^{n+1} = g\hat{\chi}^n \tag{43}$$

where g is the growth factor.

Substituting Eqs. (43) into (42), we get

$$g[2X \cos \beta + Y + i2Z \sin \beta] = 2X \cos \beta + Y - i2Z \sin \beta \tag{44}$$

where $\beta = kh, i = \sqrt{-1}$

So, we have

$$g = \frac{l - iq}{l + iq} \tag{45}$$

$$\text{where } l = 2X \cos \beta + Y, q = 2Z \sin \beta, \text{ then we get } |g| = 1 \tag{46}$$

Hence, marginal stability is justified for this cubic scheme.

5 Numerical Experiments

The RLW equation has the following three invariances of motion given by

$$I_1 = \int_{-\infty}^{\infty} U dx, I_2 = \int_{-\infty}^{\infty} \left(U^2 + \mu \left(\frac{\partial U}{\partial x} \right)^2 \right) dx, I_3 = \int_{-\infty}^{\infty} (U^3 + 3U^2) dx \tag{47}$$

relating to mass, momentum, and energy conservation equations.

Accuracy and efficiency are tested by using the following L_2 and L_∞ error norms:

$$L_2 = \|U^{\text{exact}} - U^N\|_2 \simeq h \sum_{j=0}^N |U_j^{\text{exact}} - U_j^N|^2 \tag{48}$$

$$L_\infty = \|U^{\text{exact}} - U^N\|_\infty \simeq \max_j |U_j^{\text{exact}} - U_j^N| \tag{49}$$

6 Motion of Single Solitary Wave

RLW equation subject to the initial condition

$$U(x, 0) = \frac{3v}{\delta} \sec h^2 \left[\frac{1}{2} \sqrt{\frac{v}{\mu(v+1)}} (x - g_3) \right] \tag{50}$$

and boundary conditions as $\psi_0 = 0$ and $\psi_1 = 0$.

We validated our scheme in a numerical and analytical sense by using the error norms L_∞ and L_2 . The conservation properties are determined by using the quantities I_1, I_2 and I_3 from Eq. (47).

We choose $v = 0.1, \mu = \delta = 1, \Delta t = 0.1, h = 0.5,$ and $g_3 = 0$. We computed the results up to $t = 20$ with range $[-40, 80]$. In this computation, it is found that the conserved quantities I_1 and I_2 are changed fewer by 5×10^{-4} and 1×10^{-4} . While the changes of invariant I_3 tends to zero. Table 2 represents invariants with error norms at different time levels for single solitary wave at different time levels,

Table 2 Invariants and error norms for single solitary wave

T	I_1	I_2	I_3	L_2	L_∞
0	3.97993	0.81046	2.57901	0	0
4	3.97993	0.81046	2.57901	0.20890	0.13013
8	3.97993	0.81046	2.57900	0.66604	0.22277
12	3.97993	0.81046	2.57901	1.08953	0.27056
16	3.97994	0.81046	2.57901	1.35798	0.29033
20	3.97994	0.81046	2.57901	1.49386	0.29695

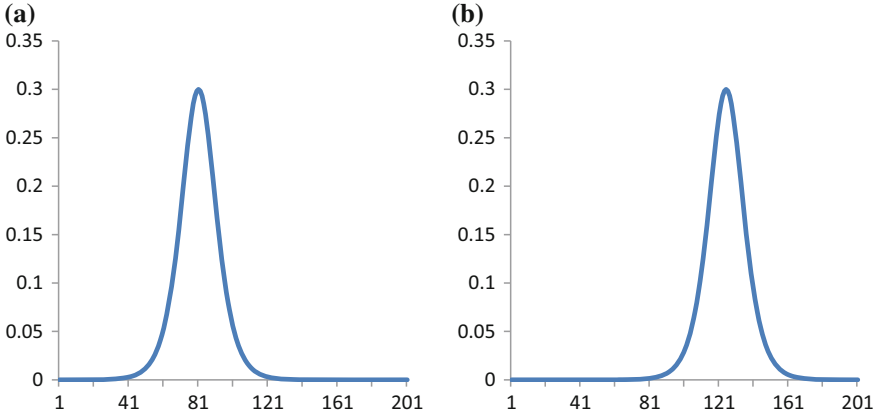


Fig. 1 Single solitary wave for $\nu = 0.1$ at time level **a** $t=0$, **b** $t=20$

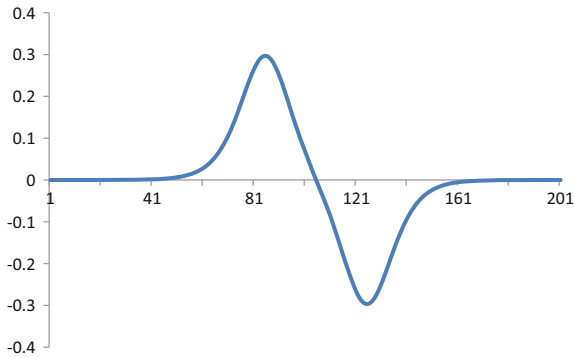


Fig. 2 Error at time level $t=20$, $\nu=0.1$

$t \leq 20$ with amplitude=0.3, $\Delta t = 0.1$, $h = 0.5$, in the region $-40 \leq x \leq 60$ (Figs. 1 and 2).

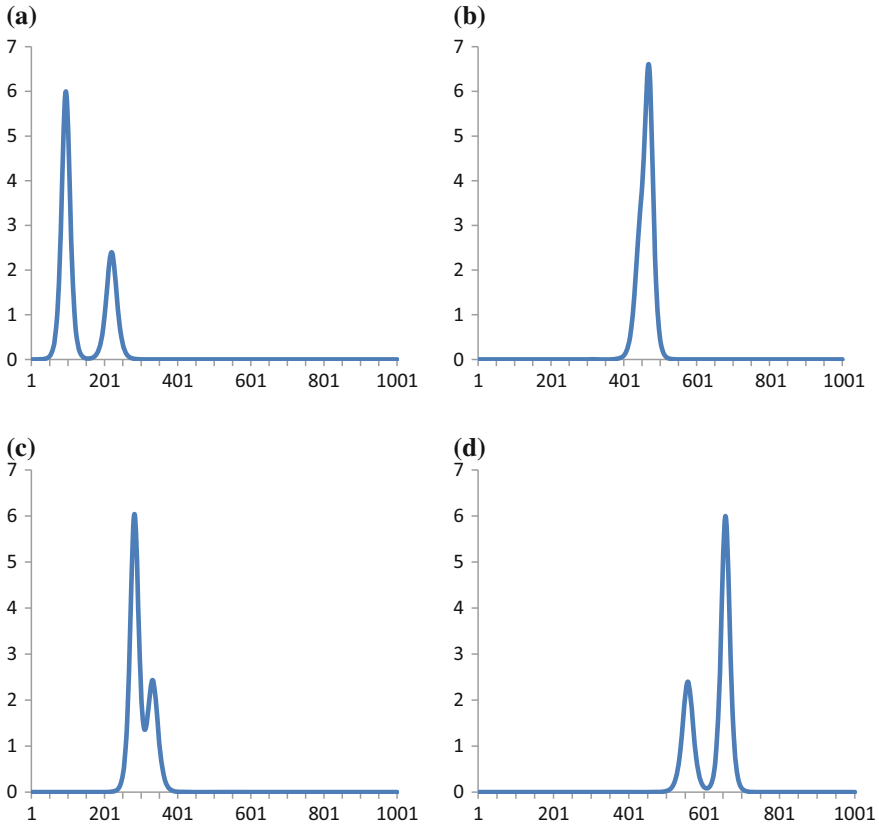


Fig. 3 Motion of two solitary waves at **a** $t=0$, **b** $t=10$, **c** $t=20$, **d** $t=30$

7 Motion of Two Solitary Waves

The cooperation between unidirectional solitary waves in well-separated form with a variation of amplitude is studied in this section.

The RLW equation with initial conditions:

$$U(x, 0) = 3v_1 \operatorname{sech}^2(r_1(x - x_1)) + 3v_2 \operatorname{sech}^2(r_2(x - x_2)), \tag{51}$$

where $r_j = \sqrt{\frac{v_j}{4\mu(v_j+1)}}$, $j = 1, 2$, x_j and v_j are arbitrary constants.

For a numerical solution, we take the parameter values: $v_1 = 2.0$, $v_2 = 0.8$, $x_1 = 15$, $x_2 = 35$, $\mu = \delta = 1$, $h = \Delta t = 0.1$ with domain $[0, 160]$. The motion of two solitary waves and its invariants at different time levels is shown in Fig. 3 and Table 3.

Table 3 Motion of invariants for two solitary waves with $v_1 = 2.0$, $v_2 = 0.8$, $x_1 = 15.0$, $x_2 = 35.0$, $h = \Delta t = 0.1$, $0 \leq x \leq 160$

T	I_1	I_2	I_3
0	27.37111	89.02151	644.04212
10	27.37119	89.02434	644.05806
20	27.37120	89.03049	644.09313
30	27.37120	89.04379	644.17033

8 Conclusion

This paper presents a collocation technique for the numerical solution of the RLW equation based on cubic B-spline. The proposed method successfully implemented the propagation of solitary waves. A linear stability analysis of the proposed method is made and found to be unconditionally stable. The present scheme is verified by considering the single solitary waves where the analytical solution is known. Furthermore, our technique is extended to find the solution for two solitary waves interaction, for which no analytical solution exists.

References

1. Russel, J.S.: Report on waves, Report of the fourteenth meeting of British Association for Advancement of Science, John Murray, London, pp. 311–390 (1844)
2. Crighton, D.G.: Applications of KdV. *Acta Applicandae Math.* **39**, 39–67 (1995)
3. Rayleigh, L.: On waves. *Phil. Mag.* **1**, 257–279, *Sci. Papers* **1**, 251–271 (1876)
4. Zhang, L.: A finite difference scheme for generalized long wave equation. *Appl. Math. Comput.* **168**, 962–972 (2005)
5. Kaya, D., El-Sayed, S.M.: An application of the decomposition method for the generalized KdV and RLW equations. *Chaos, Solitons Fractals* **17**, 869–877 (2003)
6. Gardner, L.R.T., Gardner, G.A., Ayoub, F.A., Ameen, N.K.: Approximations of solitary waves of the MRLW equation by B-spline finite element. *Arab. J. Sci. Eng.* **22**, 183–193 (1997)
7. Saka, B., Dag, I.: A collocation method for the numerical solution of the RLW equation using cubic B-spline basis. *Arab. J. Sci. Eng.* **30**(1A) (2005)
8. Raslan, K.R.: A computational method for the regularized long wave (RLW) equation. *Appl. Math. Comput.* **167**, 1101–1118 (2005)
9. Jain, P.C., Shankar, R., Singh, T.V.: Numerical solution of regularized long wave equation. *Commun. Numer. Methods Eng.* **9**, 579–586 (1993)
10. Bhardwaj, D., Shanker, R.: A Computational method for regularized long wave equation. *Comp. Math. Appl.* **40**, 1397–1404 (2000)
11. Mittal, R.C., Jain, R.K.: Numerical solutions of nonlinear Burger's equation with modified cubic B-splines collocation method. *Appl. Math. Comput.* (2012)
12. Keskin, P., Irk, D.: Numerical solution of the MRLW equation using finite difference method. *Int. J. Nonlinear Sci.* **14**(3), 355–361 (2012)

An Enhanced Fractal Dimension Based Feature Extraction for Thermal Face Recognition



Sandip Joardar, Arnab Sanyal, Dwaipayan Sen, Diparnab Sen and Amitava Chatterjee

Abstract Variance in pose during data acquisition poses a serious challenge for any biometric system which uses the human face as a physiological biometric feature. In this paper, we present an enhanced patchwise fractal dimension based feature extraction technique for the purpose of pose-invariant face recognition. We have presented an improved version of the Differential Box Counting (DBC) based fractal dimension computation technique which is used for feature extraction of thermal images of the human face. A Far-Infrared (FIR) imaging based human face database, called the JU-FIR-F1: FIR Face Database, was developed in the Electrical Instrumentation and Measurement Laboratory, Electrical Engineering Department, Jadavpur University, Kolkata, India for testing the accuracy, stability, and robustness of our proposed feature extraction methodology. We have included the results obtained through extensive experimentation to elaborate the superiority of our proposed algorithm over its other well-known counterparts.

Keywords FIR imaging · Face recognition · Fractal dimension · DBC

1 Introduction

Some of the most well-known and extensively used biometric physiological features include iris [1, 18], palmprint [3], and palm dorsal vein pattern [9–11], however, the human face [4, 6–9] has been proved over time to be one of the most reliable and robust physiological features that has been put to use in a biometric identification system. However, utilizing the human face as a physiological biometric feature has its own share of tough challenges like age variance [15], emotion variance [17], and pose variance [5], which are certainly very difficult to address in a real-time biometric system. This paper is organized as follows: Sect. 2 presents the methodology of database acquisition and creation, an elaborate mathematical analysis of our proposed

S. Joardar (✉) · A. Sanyal · D. Sen · D. Sen · A. Chatterjee
Electrical Engineering Department, Jadavpur University, Kolkata 700032, India
e-mail: sandipjoardar@gmail.com

© Springer Nature Singapore Pte Ltd. 2019
K. Deep et al. (eds.), *Decision Science in Action*, Asset Analytics,
https://doi.org/10.1007/978-981-13-0860-4_16

217

feature extraction algorithm and a brief discussion about our biometric recognition strategy are discussed in Sect. 3 and, finally, in Sect. 4, we provide the results obtained through extensive experimentation and conclude this paper with the inference of the experimental results and analysis.

1.1 Thermal Imaging of the Human

Although, the human face is one of the most extensively used physiological feature of a biometric person identification system it, however, can be posed with stiff challenges when we use its visual spectrum image. Therefore, in this research work, we have used the thermal images of real human subjects captured using the FIR imaging technology. Visual spectrum images of the human face are extremely sensitive towards lighting conditions, it is very difficult for the visual imaging technology to differentiate between the face of a real human and that of the picture of a human face, the visual images can have drastic variations depending upon pose and expression variances [2], however, thermal images remain largely insensitive and unaffected by the aforementioned challenges of face recognition using the visual images of the human face [2]. Consequently, in this research, we have opted for the thermal images of the human face as a biometric feature for identification of real human subjects.

1.2 Fractal Dimension Based Feature Extraction

Fractal Dimension (FD) is a measure of the roughness of the image [13, 16] and has been widely and extensively used for image compression, classification, recognition, and segmentation. FD is also a statistical index of the self-similarity of an image [12, 16]. In this paper, we elaborately discuss how we have formulated an enhanced FD computation approach using an improved version of DBC [16]. Here, instead of computing the FD of the whole image, we have computed FDs of patches into which the whole image is segregated and, subsequently, we achieve to compute matrices which are patchwise FDs of the original image. This is done to preserve the self-similarity and the roughness indices of the whole image. Thereafter, an inter-grid similarity measure is proposed which is used to compute the final feature vector.

1.3 Biometric Identification

We have developed a database of thermal images of the human face called the JU-FIR-F1: FIR Face Database at the Electrical Instrumentation and Measurement Laboratory, Electrical Engineering Department, Jadavpur University, Kolkata, India on which our proposed feature extraction algorithm was tested. We have also tested

some of the most well-known and widely used feature extraction algorithms on this database and presented a comparative study of all the results obtained through extensive experimentation.

2 JU-FIR-F1: FIR Face Database

In the aforementioned section, we discussed how the visual image of the human face can be sensitive to and affected by many factors like lighting conditions, synthetic images rather than real human subjects and expression variance. Consequently, we have chosen to implement the thermal images of real human subjects as the physiological feature of our biometric identification system. Now, FIR imaging is to a large extent insensitive towards lighting condition and images can be acquired even in the complete absence of visual spectrum light. It is invariant towards expression changes during data acquisition and one of its most significant advantages is that it can quite easily differentiate between real human subjects and synthetic images of the human face. In this section, we present an elaborate discussion on data acquisition and the subsequent database formation.

2.1 Thermal Image Acquisition

We have utilized the FIR imaging technology for data acquisition. The KT-384 [14], shown in the following Fig. 1, thermal imager (Manufacturer: Sonel®, Poland), a fully radiometric camera, was used for thermal image acquisition of the human face.

It provides nine different palettes for data acquisition which are shown in the following Fig. 2. We have used the grayscale palette to avoid overloading the mem-

Fig. 1 Sonel® KT-384 [14]



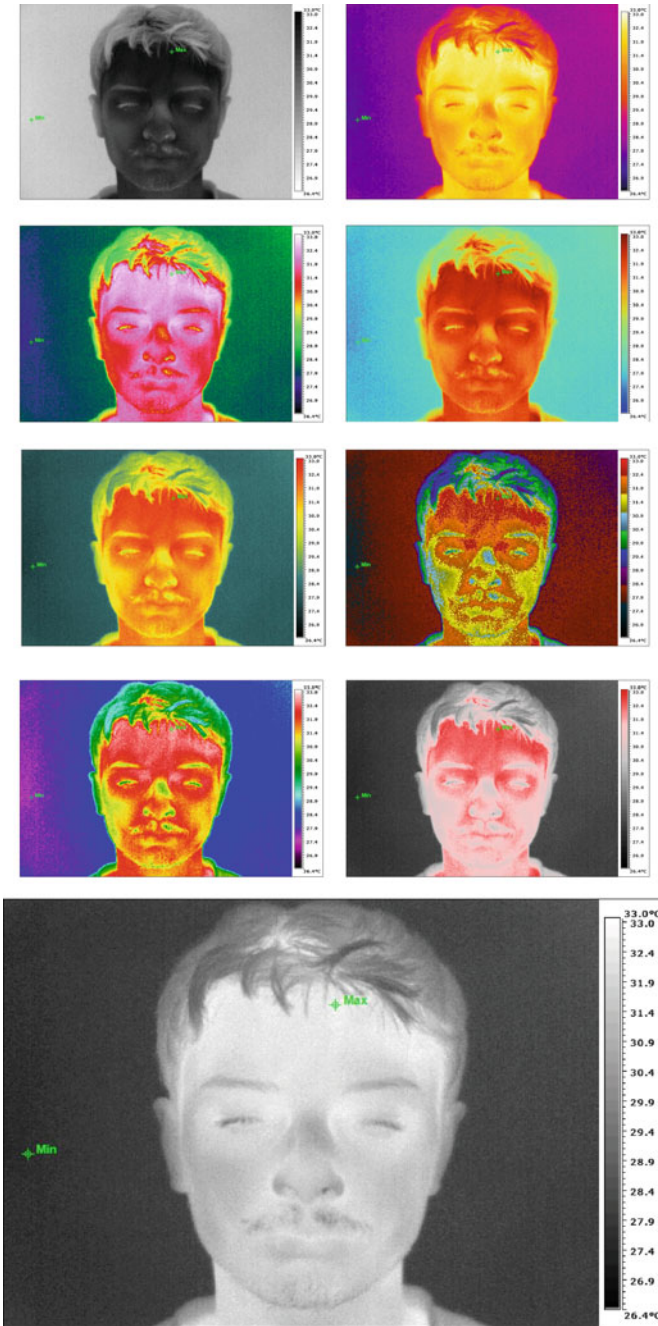


Fig. 2 Sonel[®] KT-384 [14] image acquisition palettes

ory of the system and, hence, enhancing the computational complexity of it. The images were acquired at the Electrical Instrumentation and Measurement Laboratory, Electrical Engineering Department, Jadavpur University, Kolkata, India. They were acquired under normal laboratory environmental condition of temperature about 22–24 °C and humidity about 95%. The dimension of each image acquired is (100 × 64).

One of the significant aspects of our data acquisition and database creation is that we have not implemented any form of image processing before the raw images are incorporated into the database. This highlights the fact that all forms of noises and artifacts that get included during the data acquisition phase remain intact. This was done with the sole intention to develop and test the robustness of our feature extraction and biometric identification algorithms.

2.2 Database Creation

Subsequently, after data acquisition, the raw images were incorporated into the JU-FIR-F1: FIR Face Database created in the Electrical Instrumentation and Measurement Laboratory, Electrical Engineering Department, Jadavpur University, Kolkata, India. There are in total 13 images acquired from 17 distinct subjects with a consistent gap of 5 min between two consecutive image acquisition phases. In this research, we have tried to address the challenge posed by pose variance during image acquisition. Consequently, we have incorporated deterministic pose variance during our data acquisition phase along the pitch and yaw axes. The subject was asked to undergo $\pm 15^\circ$, 30° , and 45° pitch and yaw displacements resulting in 12 poses and the 13th pose is the frontal face image. Therefore, these are the 13 images of a particular subject with prespecified pose variations.

3 Feature Extraction and Biometric Identification

Through the discussion in the previous sections, we have highlighted the implementation of an enhanced fractal dimension based feature extraction methodology. In this section, we elaborately present the detail mathematical analysis of the methodology behind the feature extraction and biometric identification algorithms.

3.1 Feature Extraction

With an image $img \in \mathfrak{R}^{m \times n}$, we can compute the FD of the image using DBC [16] by considering the size $m \times n$ of the image to be the two coordinates (x, y) and the pixelwise intensity value to be the third coordinate (z) of a three-dimensional

space (x, y, z) [16]. Consider, now that the image is scaled down to a size $m_s \times n_s$ where $m/2 \geq m_s > 1$ and $n/2 \geq n_s > 1$. Moreover, m_s and n_s must be integers. Subsequently, we have an estimate $r = 2s/(M + N)$ for which the FD is given by the following expression (1) [16]:

$$\text{FD}_{\text{img}} = \frac{\log(N_r)}{\log(1/r)} \quad (1)$$

Now, the (x, y) space of the image $\mathbf{img} \in \mathfrak{R}^{m \times n}$ is broken down into windows of size $m_s \times n_s$ and the third coordinate (z) of a three-dimensional space (x, y, z) is divided into sections of size s' such that on each grid, we have three-dimensional boxes of size $m_s \times n_s \times s'$ [16]. The s' is given by the following expression (2) [16]:

$$s' = \frac{G(m_s + n_s)}{(m + n)} \quad (2)$$

The G in the expression (2) is the total number of distinct intensity levels present in the image $\mathbf{img} \in \mathfrak{R}^{m \times n}$. Now, the N_r in expression (1) is given by the following expression (3) [16]:

$$N_r = \sum_{i,j} n_r(i, j) \quad (3)$$

In the expression (3), $n_r(i, j)$ is the contribution of the (i, j) grid towards the fractal dimension, N_r , and is given by the following expression (4) [16]:

$$n_r(i, j) = [\mathbf{b}_{\max}] - [\mathbf{b}_{\min}] + 1 \quad (4)$$

In the aforementioned expression (4), $\mathbf{b}_{\max} \in \mathfrak{R}^{m_s \times n_s \times s'}$ and $\mathbf{b}_{\min} \in \mathfrak{R}^{m_s \times n_s \times s'}$ are the three-dimensional boxes in which the maximum and minimum intensity values of the image $\mathbf{img} \in \mathfrak{R}^{m \times n}$ fall respectively and $[\bullet]$ indicates the integral box number.

However, in this paper, we have utilized FD of image patches rather than the FD computed of the whole image together [12]. This is because utilizing the FD of the whole image for face recognition has two very significant shortcomings [12]. First, two local image regions with different patterns or textural information may be having the same FD and, second, pixels located near the corners and boundaries tend to become smaller than regions representing pixels which are located well within the image. Consequently, we have opted for computation of patchwise FD with an improved version of DBC. Our proposed feature extraction algorithm using enhanced FD is given in Table 1.

The enhanced FD proposed by us is computed using expression (5) and it is quite clear from the expression that it takes into account the median intensity value of each patch and, therefore, enhances the local intensity information of each patch in FD computation which increases the distinctiveness of each patch from the others. The feature vectors of all the thermal images are then incorporated into a dic-

Table 1 Feature extraction of thermal images using patchwise enhanced FD

Step	Description
1	The image $\mathbf{img} \in \mathfrak{R}^{m \times n}$ is extended using circular padding [16] giving us a new image $\mathbf{img}_{\text{pad}} \in \mathfrak{R}^{M \times N}$
2	The following operations are then carried out for each of the $M \times N$ pixels of the $\mathbf{img}_{\text{pad}}$
2.1	The patch size is determined, $\mathbf{img}_{\text{patch}} \in \mathfrak{R}^{p \times q}$ into which the image $\mathbf{img}_{\text{pad}} \in \mathfrak{R}^{M \times N}$ is segregated
2.2	The FD for $\mathbf{img}_{\text{patch}}$ is computed with the expression (1), however, the $n_r(i, j)$ computation is modified and done with the following expression (5): $n_r(i, j) = \left\lceil \frac{(b_{\text{max}} - b_{\text{min}} + b_{\text{med}})}{(M+N)/2} \right\rceil \quad (5)$ where, $b_{\text{med}} \in \mathfrak{R}^{p \times q \times s'}$ is the box in which the median of the intensity values of $\mathbf{img}_{\text{patch}}$ fall
3	After, computation of enhanced FD of each patch we arrange the FD values in a matrix according to the central pixel location of the image patch and, thereby, obtaining the matrix $\mathbf{FDmat}_{\text{img}} \in \mathfrak{R}^{m \times n}$ consisting of FDs of local patches
4	Next, we compute the inter-grid similarity between the grids of \mathbf{FDmat} by dividing it into grids of size $(a \times b)$. The inter-grid similarity measure between two grids is computed by the following expression (6): $\sqrt{\frac{\sum_{i=j=1}^{l=a, j=b} (\mathbf{FDmat}_1(i, j) - \mathbf{FDmat}_2(i, j))^2}{a \cdot b}} \quad (6)$
5	Finally, the inter-grid similarity scores are stored in a vector $\mathbf{Fvect}_{\text{img}} \in \mathfrak{R}(mn(mn + ab) / 2a^2b^2) \times 1$ sequentially

tionary $\mathbf{D} \in \mathfrak{R}^{V \times T}$, where T is the total number of images in the database and $V = (mn + ab/2a^2b^2)$, which is then put to use for the eventual biometric identification.

3.2 Biometric Identification

We have used the *Collaborative Representation based Classification* (CRC) algorithm [19, 20] which has the Standard Tikhonov Regularization [7] for ill-posed problems at its heart. The biometric identification algorithm is given in Table 2.

4 Experimental Results

Finally, the feature extraction algorithm, Table 1, followed by the biometric identification algorithm, Table 2, is tested on the JU-FIR-F1: FIR Face Database. First, the

Table 2 Biometric identification algorithm

Step	Description
1	T_r number of training samples per class is chosen and a separate training dictionary $\mathbf{D}_{\text{TRAIN}} \in \mathfrak{R}^{V \times (T_r \bullet TNS)}$ is formed. The training samples for each of the TNS subjects are chosen randomly. Therefore, the testing dictionary is given by $\mathbf{D}_{\text{TEST}} \in \mathfrak{R}^{V \times (T - T_r \bullet TNS)}$
2	The columns of both the training and testing dictionaries are normalized such that they have unit Euclidean norm [18, 19]
3	The following operations are then carried out for each of the column vectors, $\mathbf{d}_{\text{TEST}} \in \mathfrak{R}^{V \times 1}$ which are actually reshaped test images post feature extraction, of the testing dictionary $\mathbf{D}_{\text{TEST}} \in \mathfrak{R}^{V \times (T - T_r \bullet TNS)}$
3.1	The test vector $\mathbf{d}_{\text{TEST}} \in \mathfrak{R}^{V \times 1}$ is coded over the training dictionary $\mathbf{D}_{\text{TRAIN}} \in \mathfrak{R}^{V \times (T_r \bullet TNS)}$ as a collaborative linear combination using the optimized reconstructed vector given by the following expression (7): $\hat{\boldsymbol{\alpha}} = \arg \min_{\boldsymbol{\alpha}} \left[\begin{array}{l} \ \mathbf{d}_{\text{TEST}} - \mathbf{D}_{\text{TRAIN}}\boldsymbol{\alpha}\ _2^2 + \\ \lambda \ \boldsymbol{\alpha}\ _2^2 \end{array} \right] \quad (7)$
3.2	The $\hat{\boldsymbol{\alpha}}$ vector is computed using the following expression (8): $\hat{\boldsymbol{\alpha}} = (\mathbf{D}_{\text{TRAIN}}^T \mathbf{D}_{\text{TRAIN}} + \lambda \mathbf{I})^{-1} \mathbf{D}_{\text{TRAIN}}^T \mathbf{d}_{\text{TEST}} \quad (8)$
3.3	The reconstruction residual for each class, which are physically human subjects, is computed using the following expression (9): $r_i = \left(\frac{\ \mathbf{d}_{\text{TEST}} - [\mathbf{D}_{\text{TRAIN}}]_i \hat{\boldsymbol{\alpha}}_i\ _2}{\ \hat{\boldsymbol{\alpha}}_i\ _2} \right) \quad (9)$ where $[\mathbf{D}_{\text{TRAIN}}]_i$ and $\hat{\boldsymbol{\alpha}}_i$ are the local training dictionary and the local reconstruction vector of class i , respectively
3.4	Finally, the test sample is classified to that class i which has the least reconstruction residual

optimal patch size was determined from results obtained through extensive experimentation. For this phase of experimentation, a total of 5 training samples per class were selected for each class and the patch size was varied from 3 to 23 pixels where each patch is of square shape. The results so obtained are tabulated in Table 3.

The results tabulated in Table 3 were obtained with grid size (20 × 16). The experiments for the results provided in Table 3 were run for a total of 200 times and then the mean recognition rate and the standard deviation were reported.

Finally, we compared our feature extraction algorithm with some of its state of the art and extensively used counterparts and the results so obtained are tabulated in the following Table 4. It should be noted that for raw images and Kouzani et al. [12], Step 5 of Table 2 was computed so that the feature dimension remains constant during the comparative study and, then, the classification is done using the biometric identification algorithm given in Table 2.

The results given in Table 4 confirm that the feature extraction algorithm proposed by us has shown higher accuracy, with a major leap in the mean recognition rate, and stability compared to its other well-known counterparts.

Table 3 Comparative study with different patch sizes

Patch size	Recognition rate (Mean \pm Standard deviation) (in %)
3	85.55 \pm 3.09
5	86.16 \pm 3.02
7	88.64 \pm 3.18
9	90.51 \pm 3.22
11	91.53 \pm 2.82
13	91.78 \pm 2.79
15	93.18 \pm 2.68
17	92.08 \pm 2.85
19	91.25 \pm 2.71
21	90.15 \pm 2.99
23	89.82 \pm 3.31

The best result among all has been emboldened

Table 4 Comparative study with different feature extraction algorithms

Feature extraction algorithm	Recognition rate (Mean \pm standard deviation) (in %)
Raw images	86.20 \pm 2.92
Eigen face	82.61 \pm 3.14
LDA	84.40 \pm 3.42
Kouzani et al. [16]	90.71 \pm 3.82
This Paper	93.18 \pm 2.68

The best result among all has been emboldened

5 Conclusion

This paper discusses a novel method of feature extraction using enhanced Fractal Dimension computed using improved Differential Box Counting approach. This paper aims to address the pose variance that occurs during any real-time data acquisition of a human subject for biometric identification. We have developed a thermal face database wherein predetermined pose variations were incorporated during the data acquisition phase. Subsequently, on application of our proposed algorithm on the developed face database, we found that our algorithm shows higher accuracy and stability compared to its other widely used counterparts. Moreover, our proposed algorithm has shown considerable robustness as there is no image processing involved into the database creation phase. This was an intentional decision to keep the noises and artifacts of the data acquisition phase intact and test the robustness of the subsequent algorithms.

Acknowledgements This work was supported by University Grants Commission (UGC) India under University with Potential for Excellence (UPE)—Phase II Scheme awarded to Jadavpur University, Kolkata, India.

References

1. Bowyer, K.W., Hollingsworth, K.P., Flynn, P.J.: A survey of iris biometric research: 2008–2010. In: Bowyer, K.W., Burge, M.J. (eds.) *Handbook of Iris Recognition*. Advances in Computer Vision and Pattern Recognition, pp. 23–61. Springer London, Springer-Verlag London (2016)
2. Bhowmik, M.K., Saha, K., Majumder, S., Majumder, G., Saha, A., Sarma, A.N., Bhattacharjee, D., Basu, D.K., Nasipuri, M.: Thermal infrared face recognition—a biometric identification technique for robust security system. In: *Reviews, Refinements and New Ideas in Face Recognition*. InTech (2011)
3. Chakraborty, S., Bhattacharya, I., Chatterjee, A.: A palmprint based biometric authentication system using dual tree complex wavelet transform. *Measurement* **46**(10), 4179–4188 (2013)
4. Chakraborty, A., Jain, H., Chatterjee, A.: Volterra kernel based face recognition using artificial bee colony optimization. *Eng. Appl. Artif. Intell.* **26**(3), 1107–1114 (2013)
5. Ding, C., Tao, D.: A comprehensive survey on pose-invariant face recognition. *ACM Trans. Intell. Syst. Technol.* **7**(3), 37 (2016)
6. Galbally, J., Marcel, S., Fierrez, J.: Biometric anti-spoofing methods: a survey in face recognition. *IEEE Access* **2**, 1530–1552 (2014)
7. Golub, G.H., von Matt, U.: Tikhonov regularization for large scale problems. In: Golub, G.H., Lui, S.H., Luk, F., Plemmons, R. (eds.) *Workshop on Scientific Computing*, pp. 3–26. Springer (1997)
8. Joardar, S., Chatterjee, A.: Collaborative representation based face recognition using a hybrid similarity measure with single training sample per person. In: *Proceedings of IEEE International Conference on Control, Instrumentation, Energy and Communication (CIEC)*, pp. 631–635 (2014)
9. Joardar, S., Chatterjee, A., Rakshit, A.: A real-time palm dorsa subcutaneous vein pattern recognition system using collaborative representation based classification. *IEEE Trans. Instrum. Meas.* **64**(4), 959–966 (2014)
10. Joardar, S., Chatterjee, A., Rakshit, A.: Real-time NIR imaging of palm dorsa subcutaneous vein pattern based biometrics: an src based approach. *IEEE Instrum. Measur. Mag.* **19**(2), 13–19 (2016)
11. Joardar, S., Chatterjee, A., Bandyopadhyay, S., Maulik, U.: Multi-size patch based collaborative representation for palm dorsa vein pattern recognition by enhanced ensemble learning with modified interactive artificial bee colony algorithm. *Eng. Appl. Artif. Intell.* **60**, 151–163 (2017)
12. Kouzani, A.Z., He, F., Sammut, K.: Face image matching using fractal dimension. In: *Proceedings of International Conference on Image Processing (ICIP 99)*, vol. 3, pp. 642–646 (1999)
13. Lai, K., Li, C., He, T., Chen, L., Yu, K., Zhou, W.: Study of an improved differential box-counting approach for gray-level variation of images. In: *10th International Conference on Sensing Technology (ICST)*, pp. 1–6 (2016)
14. [online] http://www.sonel.pl/sites/default/files/pl/ins/kt384_ins_104_plgp.pdf
15. Park, U., Tong, Y., Jain, A.K.: Age-invariant face recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **32**(5), 947–954 (2010)
16. Sarkar, N., Chaudhuri, B.B.: An efficient approach to estimate fractal dimension of textural images. *Pattern Recogn.* **25**(9), 1035–1041 (1992)
17. Shojaeilangari, S., Yau, W.-Y., Nandakumar, K.: Robust representation and recognition of facial emotions using extreme sparse learning. *IEEE Trans. Image Process.* **24**(7), 2140–2152 (2015)
18. Sony, S., Singh, A.K.: Survey on methods used in iris recognition system. *J. Image Process. Artif. Intell.* **3**(1) (2017)
19. Zhang, L., Yang, M., Feng, X.: Sparse representation or collaborative representation: which helps face recognition? In: *Proceedings of IEEE Conference ICCV 2011*, pp. 471–478 (2011)
20. Zhang, L., Yang, M., Feng, X., Ma, Y., Zhang, D.: Collaborative representation based classification for face recognition. arXiv preprint [arXiv:1204.2358](https://arxiv.org/abs/1204.2358) (2012)

Seismic Analysis of Multistoried Building with Optimized Damper Properties



Dipti Singh, Shilpa Pal and Abhishek Singh

Abstract In today's scenario where space is an issue, the increase in population has led to a boom in the construction industry. With the lack of land for construction, the buildings are becoming higher and more complex, so with the increase in the number of stories, it is necessary to make them safe under adverse seismic conditions. Dampers are one way to make the structure earthquake resistant and the optimization of their properties is sometimes required. In this study, the damper properties, i.e., damping and stiffness have been optimized using self-organizing migrating genetic algorithm (SOMGA) and genetic algorithm (GA) technique on a model of 10-storey building which has equal mass, stiffness, etc. on all the floors. The optimized damper properties obtained from SOMGA result in the reduction of 52% of the storey displacement while that of GA is 60% as compared to the undamped model. Both techniques provide better optimized damper properties. It is observed that the optimized damper helps in significant reduction of the seismic response of the structure, thus justifying the need of optimized parameters of dampers.

Keywords Optimization · Genetic algorithm · Self-organization migrating algorithm · Supplement damper · Structural control · SAP 2000

1 Introduction

One of the most devastating hazards of nature is earthquake which destroys the lives and homes of virtually every continent. Their effect of destruction is almost instantaneous and the damage is entirely associated with the man-made structures.

D. Singh · S. Pal (✉) · A. Singh
Gautam Buddha University, Greater Noida 201312, India
e-mail: shilpa@gbu.ac.in

D. Singh
e-mail: diptipma@rediffmail.com

A. Singh
e-mail: apro9899@gmail.com

© Springer Nature Singapore Pte Ltd. 2019
K. Deep et al. (eds.), *Decision Science in Action*, Asset Analytics,
https://doi.org/10.1007/978-981-13-0860-4_17

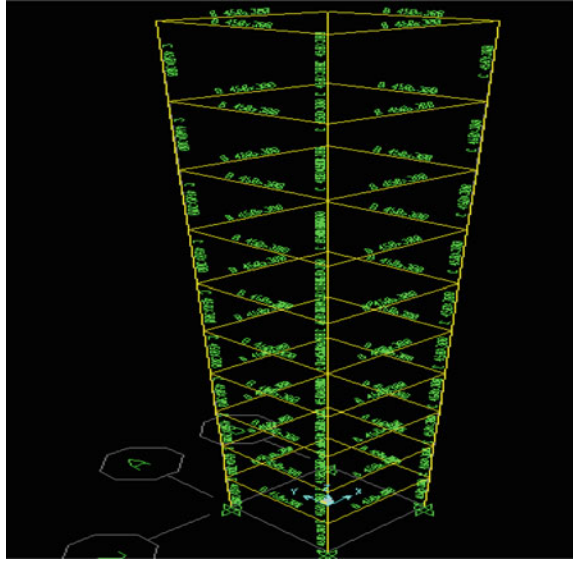
Moreover, there is no or very little warning unlike other natural disasters which makes earthquake engineering an important area of study.

There are control devices which are aimed to prevent structural damage caused by structure vibration in case of earthquake loads. These devices consist of dampers and base isolators which control and reduce the response of the structure during an earthquake. The most basic dampers are active and passive [10]. In the passive dampers, the damper modifies the structure response without external power supply while in the active damper, the response is reduced by generating the required forces to oppose the work done by earthquake forces with the help of external power supply. Now, moving to the semi-active dampers, they use both the properties of active and passive dampers, and the hybrid dampers use various combination of the above dampers [12].

For dampers to work properly, the properties of the dampers have to be designed so that the response of the structure can be reduced. Hadi and Arfaidi [6] conducted a study on a 10-storey shear building to optimize the properties of tuned mass damper on the 10th floor. Asahina et al. [2] used linear viscous damper (LVD) for the optimization. Tovar and Lopez [14] optimized the number and location of the damper of 5-storey moment resisting frame using simplified method, similar work of optimizing the number and location of actuators has been worked out by Abbasi and Markazi [1] using genetic algorithm. Murudi and Mane [11] observed that TMD was found to be the most effective damping device to get the minimum relative displacement. Islam and Ashan [8] optimized the parameters for tuned mass damper for a multi-storey building, using EVOP technique. Kaveh et al. [9] optimized the parameters for TMD to minimize the dynamic response of multistorey structure under seismic load using charged system search (CSC). Hadi and Arfaidi [5] investigated optimized the placement of the dampers using the hybrid genetic algorithm. Sebt et al. [13] applied genetic algorithm to get the optimum location and properties for TADAS dampers in a moment resisting steel structure.

When these dampers are used in multi-degree of freedom (MDOF) systems, the optimization of the properties of the damper is important from economical point. Several techniques and methods have been proposed for the optimization of these properties, i.e., damper mass, stiffness and damping coefficient in the past few decades as mentioned above. Most of these techniques take storey displacement as the minimizing criteria when optimizing the above-said parameters [2]. In this paper, optimization of the stiffness and damping properties of the damper has been done using genetic algorithm (GA) and one of its hybrid variant self-organizing migrating genetic algorithm (SOMGA). The computational steps and working methodology of these algorithms can be found in Davendra and Zelinka [3] and Deep and Dipti [4]. The paper is organized as follows: in Sect. 1 introduction and literature review is given; in Sect. 2, problem statement and analysis using SAP software are given; Sect. 3 highlights the analysis without optimization; Sect. 4 elaborates the problem formulation; Sect. 5, result and analysis with optimization has been discussed and finally conclusions are drawn in Sect. 6..

Fig. 1 RC model of building without dampers



2 Problem Statement

In this study, a 10-story RC moment resisting frame, one bay along X -direction and Y -direction with following material properties has been analyzed using SAP 2000 as shown in Fig. 1.

2.1 Material Properties

Reinforced concrete of grade M-20 and Fe-415 grade steel has been taken for concrete and steel respectively. The stress–strain relationship is as per I.S. 456-2000 [7] and the basic properties taken while modeling are as follows:

- Modulus of Elasticity of concrete, $E_C = 22,360,680 \text{ kN/m}^2$
- Density of concrete is 25 kN/m^3
- Poisson's ratio = 0.15
- Concrete Compressive Strength $f'_c = 20,000 \text{ kN/m}^2$
- Modulus of Elasticity of steel, $E_s = 1.999 \times 10^8 \text{ kN/m}^2$
- Minimum yield stress, $f_y = 415,000 \text{ kN/m}^2$
- Minimum tensile stress, $f_u = 498,000 \text{ kN/m}^2$
- Expected yield stress, $f_{ye} = 518,750 \text{ kN/m}^2$
- Expected tensile stress, $f_{ue} = 622,500 \text{ kN/m}^2$
- Poisson's ratio = 0.3.

2.2 Model Geometry

Both the columns and beams are modeled as rigid and the details of the structure are given below:

- Number of stories = 10
- Number of bays along X -direction = 1
- Number of bays along Y -direction = 1
- Storey height = 3.5 m
- Bay with along X -direction = 6.0 m
- Bay width along Y -direction = 6.0 m.

2.3 Section Dimensions

The column size is 300 mm \times 450 mm and the beam size is also 300 mm \times 450 mm are taken for all ten floors. The building has a uniform mass of 12,561.63 kg and uniform stiffness of 284,937.65 kN/m at all stories. Mass proportional coefficient is 0.7563 and stiffness proportional coefficient is 0.002448 for the building.

2.4 Loading Case

The problem has been analyzed for various load cases, i.e., dead load, seismic load, etc. and also the combination of the above said loads as per IS codes. The structure has been analyzed for the load combination which gives the maximum displacement. The major horizontal component of the El Centro earthquake has been taken to analyze the problem.

2.5 Damper Properties

A damper is placed on the top storey of the structure as shown in Fig. 2 whose properties have been calculated as per IS 1893-2002 (Part 1). It was assumed that the mass of the damper is 5% of the total mass of the structure and the stiffness and damping were calculated for the first modal frequency using the following formulae.

$$m_d = 5\% \times 12561.3 \text{ kg} = 6280.8 \text{ kg}$$

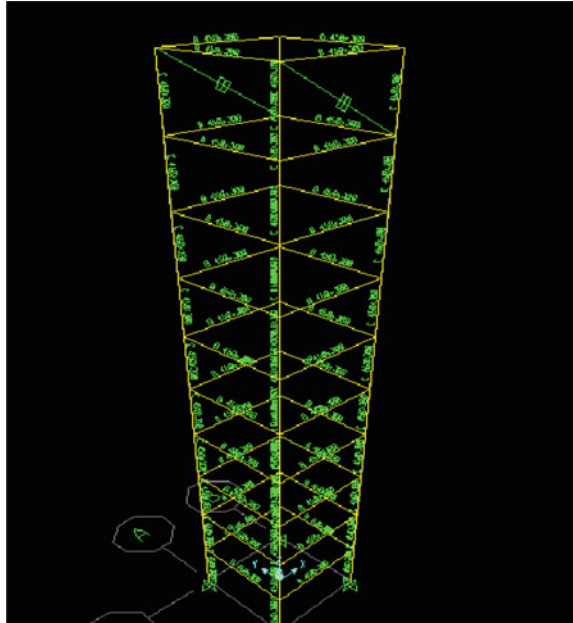
$$\text{First modal period} = 1.319 \text{ s}$$

$$f = 0.7577 \text{ s}^{-1}$$

$$\omega = 2\pi f = 2 \times 3.14 \times 0.7577$$

$$\omega = 4.762$$

Fig. 2 RC model of building with dampers



$$\omega^2 = k_d/m_d$$

$$k_d = 142427 \text{ kN/m}$$

$$c_d = 2 \zeta \omega m_d = 2 \times 0.05 \times 4.762 \times 6280.8$$

$$c_d = 2990.9 \text{ kN-s/m.}$$

3 Result and Analysis Without Optimization

Nonlinear time history analysis was done for both the undamped and damped model as discussed in Sect. 2 and the following results are obtained without optimized parameters. The damped model has a significant reduction in the storey displacement as compared to the undamped model as shown in Table 1. The top storey displacement reduces by approx. 44%.

4 Problem Formulation

The RC model of the building in SAP with damper has reduced top storey displacement. In the problem, objective is to minimize the storey displacement and the function “f” is defined in terms of displacement of each storey as given in Eq. 1.

Table 1 Results for 10-storey undamped and damped model using SAP software

Storey	Undamped (A)	5% Damping (B)	A-B (m)	% Age reduction in displacement
	Displacement (m)	Displacement (m)		
1	0.0212	0.0129	0.0083	39.15
2	0.0566	0.0342	0.0224	39.57
3	0.0934	0.0553	0.0381	40.79
4	0.1283	0.0736	0.0547	42.63
5	0.1598	0.0875	0.0723	45.24
6	0.1875	0.0965	0.091	48.53
7	0.2109	0.1099	0.101	47.88
8	0.2295	0.1236	0.1059	46.14
9	0.2432	0.1347	0.1085	44.61
10	0.2526	0.1422	0.1104	43.70

$$\text{Minimize } f = x_1^2 + x_2^2 + x_3^2 + \dots + x_{10}^2 \tag{1}$$

Subject to

$$M\ddot{X} + C\dot{X} + KX = e\ddot{x}_g \tag{2}$$

where

M $\text{diag} [m_1 \ m_2 \ m_3 \ \dots \ m_N \ m_d]$

\ddot{X} storey acceleration matrix for damped building.

\dot{X} storey velocity matrix for damped building.

X storey displacement, when damper is placed on 10th floor

e $\text{diag}[-m_1 \ -m_2 \ -m_3 \ \dots \ -m_N \ -m_d]^T$

X $[x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, 0]^T$

$$K = \begin{bmatrix} k_1 + k_2 & -k_2 & . & . & . \\ -k_2 & k_2 + k_3 & -k_3 & . & . \\ . & . & . & . & . \\ . & . & . & -k_n & k_n + k_d & -k_d \\ . & . & . & -k_d & k_d & . \end{bmatrix}$$

$$C = \begin{bmatrix} C_1 + C_2 & -C_2 & . & . & . \\ -C_2 & C_2 + C_3 & -C_3 & . & . \\ . & . & . & . & . \\ . & . & . & -C_n & C_n + C_d & -C_d \\ . & . & . & -C_d & C_d & . \end{bmatrix}$$

M is the mass matrix, C is the damping matrix, and K is the stiffness matrix along with e and \ddot{x}_g being the matrix-induced ground acceleration and ground acceleration, respectively. And X is the displacement matrix with dot indicating as derivative with respect to time.

This study considers the stiffness and damping coefficient of the damper as design parameters for optimizing performance. During the optimization process, the parameters which are to be optimized are changed continuously to get the optimal results within the desired range such as

$$\begin{aligned}
 0 < C_d < 1000 \text{ kN-s/m} \\
 0 < K_d < 4000 \text{ kN/m} \\
 x_1 < x_2 < x_3 < x_4 < x_5 < x_6 < x_7 < x_8 < x_9 < x_{10}
 \end{aligned}$$

where for the structure,

$$\begin{aligned}
 x_g &= 0.313g \text{ (PGA for El Centro 1940)} \\
 m_1 = m_2 = m_3 = \dots = m_{10} &= 12,561.63 \text{ kg} \\
 k_1 = k_2 = k_3 = \dots = k_{10} &= 284,937.65 \text{ kN/m} \\
 c_1 = c_2 = c_3 = \dots = c_{10} &= 5981.85 \text{ kN-s/m} \\
 m_d &= \text{mass of damper i.e. } 6280.8 \text{ kg (5\% of total structure mass)} \\
 \ddot{X} &= [0.865, 1.965, 3.009, 3.929, 4.608, 5.068, 5.445, 5.840, 6.095, 6.412, 0.00]^T \\
 \dot{X} &= [0.0779, 0.2047, 0.3413, 0.4869, 0.6504, 0.8185, 0.9777, 1.114, 1.2186, 1.2884, 0.00]^T
 \end{aligned}$$

5 Result and Analysis with Optimization

The objective function was to minimize the storey displacement and get the optimal values for C_d and K_d . The optimization of the damper properties has been done using optimization techniques namely genetic algorithm and SOMGA. The optimized parameters of the damper, maximum displacement of all stories and the percentage reduction of the displacement are given in Tables 2, 3, and 4, respectively, using both techniques. The percentage reduction with the optimized values of the dampers obtained from both the techniques has been compared and is shown in Table 5. The average peak displacement reduction for all the stories in the X-direction using GA is 58% while using SOMGA the reduction percentage is 52% as compared to the undamped model.

Figure 3 shows the storey displacement for all the cases. It is observed that the damped displacement results obtained from the GA and SOMGA reduce the structure response significantly. Also, it can be seen that both GA and SOMGA give approximately same results.

Table 2 Optimized parameters of the damper

Technique	Optimized parameters		
	Mass (kg)	Damping coefficient (kn-s/m)	Stiffness (kn/m)
Analytical	6280.8	2990.9	142,427.46
SOMGA	6280.8	169.997	1316.1
GA	6280.8	548.852	1382.33

Table 3 Peak displacement for the 10-storey model

Storey	Undamped (Model A)	5% Damping (Model B)	GA results (Model C)	SOMGA results (Model D)
	Displacement (m)	Displacement (m)	Displacement (m)	Displacement (m)
1	0.0212	0.0129	0.0087	0.0100
2	0.0566	0.0342	0.0236	0.0271
3	0.0934	0.0553	0.0395	0.0451
4	0.1283	0.0736	0.0552	0.0627
5	0.1598	0.0875	0.0702	0.0792
6	0.1875	0.0965	0.0840	0.0940
7	0.2109	0.1099	0.0963	0.1068
8	0.2295	0.1236	0.1068	0.1172
9	0.2432	0.1347	0.1115	0.1249
10	0.2526	0.1422	0.1207	0.1298

Table 4 Percentage reduction in displacement

Storey	A-B (m)	% Age	A-C (m)	% Age	A-D (m)	% Age
1	0.0083	39.15	0.125	58.96	0.0112	52.83
2	0.0224	39.57	0.033	58.30	0.0295	52.12
3	0.0381	40.79	0.0539	57.70	0.0483	51.71
4	0.0547	42.63	0.0731	56.97	0.0656	51.13
5	0.0723	45.24	0.0896	56.07	0.0806	50.43
6	0.091	48.53	0.1035	55.20	0.0935	49.86
7	0.101	47.88	0.1146	54.33	0.1041	49.35
8	0.1059	46.14	0.1227	53.46	0.1123	48.93
9	0.1085	44.61	0.1317	54.15	0.1183	48.64
10	0.1104	43.70	0.1319	52.21	0.1228	48.61

Table 5 Comparison between SOMGA and GA Technique

Storey	Model A–C	Model A–D
	Percentage reduction	Percentage reduction
1	58.96	52.83
2	58.30	52.12
3	57.70	51.71
4	56.97	51.13
5	56.07	50.43
6	55.20	49.86
7	54.33	49.35
8	53.46	48.93
9	54.15	48.64
10	52.21	48.61

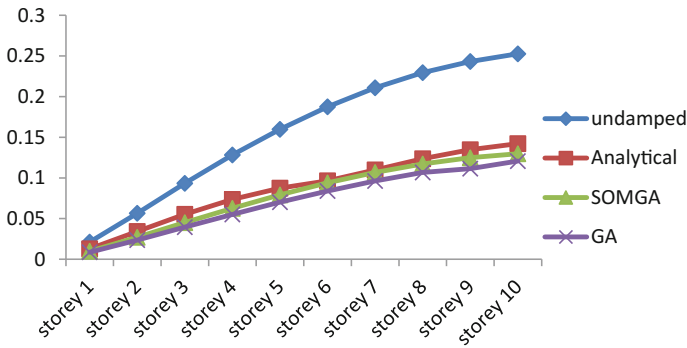


Fig. 3 Variations in displacement for all the cases

6 Conclusion

This study signifies the utilization of optimization techniques namely GA and SOMGA to optimize the properties of tuned mass damper of a 10-storey building. The building was modeled in SAP software keeping equal storey mass, stiffness and damping at all storey and a nonlinear dynamic time history analysis was done with El Centro as the base excitation earthquake. The optimized values of the damper on the top storey of the structure were obtained using optimization techniques keeping the displacement as a constraint and the analysis was done. Following are the main conclusions:

- Analytical analysis of the model with damper shows a reduction of storey displacement by 48% w.r.t. undamped model.
- The storey displacement reduces by 58 and 52%, respectively, with optimal parameters of damper obtained from GA and SOMGA w.r.t. model without damper.

- The storey displacement reduces by 10 and 6%, respectively, with the optimal parameters of damper obtained from GA and SOMGA w.r.t. analytical analysis with damper.
- Both the techniques of optimization show a close range of reduction in storey displacement with the optimized values of damper properties. Both the above techniques can be used for the optimization problem of similar cases.

References

1. Abbasi, M., Markazi, A.H.D.: Optimal assignment of seismic vibration control actuators using genetic algorithm. *Int. J. Civil Eng.* **12**(1), 24–31 (2014)
2. Asahina, D., Bolander, J.E., Berton, S.: Design optimization of passive devices in multi-degree of freedom structures. In: 13th World Conference on Earthquake Engineering, Paper no. 1600 (2004)
3. Davendra, D., Zelinka, I.: Self-organizing migrating algorithm, methodology and implementation. In: *Studies in Computational Intelligence Series*, vol. 626 (2016). <https://doi.org/10.1007/978-3-319-28161-2>
4. Deep, K.: Dipti: self organizing migrating genetic algorithm for constrained optimization. *Appl. Math. Comput.* **198**(1), 237–250 (2008)
5. Hadi, M., Arfiadi, Y.: Optimum design of absorber for MDOF structures. *J. Struct. Eng.* **124**(11), 1272–1280 (1998)
6. Hadi, M., Arfiadi, Y.: Optimum placement and properties of tuned mass damper using hybrid genetic algorithm. *Int. J. Optim. Civil Eng.* **1**(1), 167–187 (2011)
7. IS Code 456-2000: Plain and Reinforced Concrete—Code of practice—4th Revision
8. Islam, B., Ashan, R.: Optimization of tuned mass damper parameters using evolutionary operation algorithm. In: *Proceedings of the 15th World Conference on Earthquake Engineering* (2012)
9. Kaveh, A., Mohammadi, S., Hosseini, O.K., Keyhani, A., Kalatjari, V.R.: Optimum parameters of tuned mass damper for seismic application using charged system search. *Iran. J. Sci. Technol. Trans. Civil Eng.* **39**(C1), 21–40 (2015)
10. Kokil, A.S., Shrikhande, M.: Optimal placement of supplemental dampers in seismic design of structures. *J. Seismolog. Earthq. Eng.* **9**(3), 125–135 (2007)
11. Murudi, M.M., Mane, S.M.: Seismic effectiveness of tuned mass damper (TMD) for different ground motion parameters. In: *Proceedings of 13th World Conference on Earthquake Engineering*, Paper No. 2325 (2004)
12. Qu, J., Li, H.: Optimal placement of passive energy dissipation devices by genetic algorithm. *Math. Probl. Eng.* **2012**, 21 (2012)
13. Sebt, M.H., Yousefzadeh, A., Tehranizadeh, M.: The optimal TADAS damper placement in moment resisting steel structures based on a cost benefit analysis. *Int. J. Civil Eng.* **9**(1), 23–32 (2011)
14. Tovar, C., Lopez, O.A.: Effect of the position and number of dampers on the seismic response of frame structures. In: *Proceedings of 13th World Conference on Earthquake Engineering*, Paper no. 1044 (2004)

Effect of Upper Body Motion on Biped Robot Stability



Ruchi Panwar and N. Sukavanam

Abstract Achieving stability of biped robot during walking is a tough task. In this paper, we generate polynomial cubic spline for ankle joints, hip joints, and upper body so that the resulting walk is stable. Stability is assured by calculating zero momentum point with largest stability margin in Matlabs.

Keywords ZMP · Inverse kinematics · Trajectory generation · Upper body

1 Introduction

During 1970s, the study of biped robots started. Many technical and scientific efforts have been used to design and develop humanoid robots with human-like gait using artificial intelligence. Vukobratovic et al. [1] worked on dynamic stability of legged machines. McGeer [2] investigated the passive walking with knees. To realize human-like walk of a biped robot, trajectory planning is considered the most important factor. Narvez-Aroche et al. [3] have obtained a kinematic model which generated satisfactory results for the positions. Huang et al. [4] proposed an iterative computation trajectory generation method for hip and foot by specifying walking speed and step length to obtain the largest dynamic balance margin base on the ZMP. An approach presented by Erbatur and Kurt [5] improved the iterative computation trajectory generation by specifying a desired ZMP reference trajectory. Zhu Xiaoguang and Hu Ruyi [6] presented a humanoid robot gait planning. The authors in [6, 7] presented a cubic Hermitian polynomial interpolation algorithm to implement biped walking. Recently, much attention has been focused on neural-network-based inverse kinematics solutions in robotics. In [8, 9], several neural network structures used for solving the inverse kinematics problem were analyzed. Stability is a critical issue in bipedal

R. Panwar (✉) · N. Sukavanam
Department of Mathematics, IIT Roorkee, Roorkee, India
e-mail: ruchipanwar321@gmail.com

N. Sukavanam
e-mail: nsukvfma@iitr.ac.in

walking. The most widely used dynamic balance criterion is the zero-moment point (ZMP). ZMP for human walk can be either a fixed ZMP [3, 4] typically at the center of the sole in the single-support phase, or a moving ZMP that changes in a periodic fashion during locomotion as Erbatur et al. [5]. In human locomotion, the ZMP never stays at a fixed position, but moves forward in the direction of locomotion [6, 7, 9, 11]. Liu et al. [10] proposed a control, which is based on the motion of the upper body to maintain good stability of the biped and to relief from knee bending problem.

In this paper, first we generate trajectories for ankle, hip and upper body and then find the ZMP stability. To ensure stability, we generate three type of upper body motion and whichever give the best ZMP trajectory with largest stability margin will be chosen. The inverse kinematics is solved using artificial neural network. In Sect. 2, we discuss the robot model. Section 3 describes planning of leg trajectories for biped robot's walk with suitable conditions. This is followed by Sect. 4 which includes the forward kinematics and inverse kinematics of robot model. ZMP stability is calculated in Sect. 5. Upper body mass trajectories are discussed in Sects. 6 and 7 includes simulation results with graphs and discussion.

2 Robot Model

In this simple model, each leg of biped robot have 2 degrees of freedom with flat foot as in Fig. 1. All the joints are revolute which are called hip joint (H), knee joint (K) and ankle joint (A). Center of Gravity of upper body is denoted by (U). Total length of leg is $(l_1 + l_2)$ and length of foot is l_3 . It is assured that the length and mass of both legs are the same and the details of parameters are given in Table 1.

Robot walking can be considered as a repetition of one-step motion. The walking sequence can be determined by computing the trajectory of the hip, ankle and upper body. For hip trajectory, we consider stable ankle joint as a base and hip as the end effector, and for ankle trajectory, we consider swing leg's hip as base and its ankle joint as the end effector. Flat foot is attached at ankle joint.

3 Trajectory Generation

Consider the coordinates frame F with coordinates axes x - y - z . The planes xy , yz and xz respectively are called transverse plane, frontal plane and sagittal plane and biped robot walks in x -direction. The motion range of the legs in the frontal plane and the transverse plane is negligible compared to the motion in the sagittal plane. Here, we assume that the robot walk in sagittal plane (xz -plane). Total time is t_f .

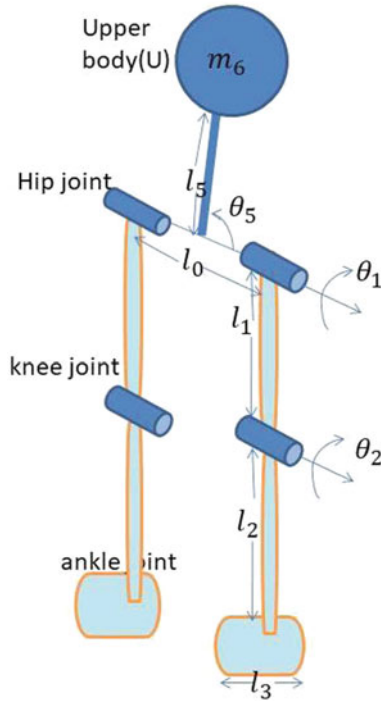


Fig. 1 Schematic 3D biped model

Table 1 Parameters

Link	Length	Value (in.)	Mass	Value (kg)
HK	l_1	14	m_1	4
KA	l_2	14	m_2	4
HU	l_5	10	m_6	60
HH	l_0	8	m_5	4

3.1 Foot Trajectory

We assume that swing leg's ankle joint follow a cubic polynomial trajectory is given by

$$x_A(t) = a_1 + b_1t + c_1t^2 + d_1t^3; \tag{1}$$

$$z_A(t) = l_1 + m_1x_A(t) + n_1x_A(t)^2 + p_1x_A(t)^3; \tag{2}$$

with boundary conditions

$$x_A(t_0) = x_i; x_A(t_f) = x_i + x_f; \dot{x}_A(t_0) = 0; \dot{x}_A(t_f) = 0.$$

$$z_A(x_0) = 0; z_A(x_f) = 0; z_A(x_m) = h_1; \dot{z}_A(x_f) = 0.$$

Here, h_1 is step height, x_i is initial position, x_f is step length and x_m is x -coordinate at which maximum height is achieved.

Considering these boundary conditions, the x and z coordinates of ankle joint at time $t \in (t_0, t_f)$ are given below:

$$x_A(t) = x_i + \left(\frac{3x_f}{t_f^2} \right) t^2 - \left(\frac{2x_f}{t_f^3} \right) t^3; \quad (3)$$

$$z_A(t) = \frac{h(-(x_f + x_i)^2 x_i)}{(x_m - x_i)(x_m - x_f - x_i)^2} + \frac{h(x_f + x_i)(x_f + 3x_i)x_A(t)}{(x_m - x_i)(x_m - x_f - x_i)^2} - \frac{h((2x_f + 3x_i)x_A(t)^2 + hx_A(t)^3)}{(x_m - x_i)(x_m - x_f - x_i)^2} \quad (4)$$

3.2 Hip Trajectory

For biped robot walking on a plane, we assume that the stable leg moves like an inverted pendulum considering its ankle joint as base and hip as end effector. During walking, humans do not fold their stable leg as the whole body weight is on this leg. The hip follows a circular path with center at ankle joint A and radius $(l_1 + l_2)$ with suitable boundary conditions. During the time interval (t_0, t_f) , the hip trajectories in x and z direction are computed by the polynomial.

$$x_H(t) = q_0 + q_1 t + q_2 t^2 + q_3 t^3; \quad (5)$$

$$z_H(t) = \sqrt{(l_1 + l_2)^2 - (x_H(t) - (x_i + x_f/2))^2}; \quad (6)$$

with boundary conditions

$$x_H(t_0) = x_i + x_f/4; x_H(t_f) = x_i + 3x_f/4; \dot{x}_H(t_0) = v_s; \dot{x}_H(t_f) = v_e.$$

$$z_H(t_0) = h; z_H(t_f) = h; \dot{z}_H(t_0) = v_{zs}; \dot{z}_H(t_f) = v_{ze}.$$

where h maximum hip height of the robot's hip at time t_2 , h_0 is hip height of the robot at starting and end position.

Hence, hip trajectory during the time (t_0, t_f) is given below:

$$x_H(t) = \frac{x_f}{4} + v_s t + \left(\frac{(v_e - v_s)}{2t_f} - r_4 \frac{3t_f}{2} \right) t^2 - 2 \left(\frac{x_f}{2t_f^3} - \frac{(v_s + v_e)}{2t_f^2} \right) t^3; \quad (7)$$

$$z_H(t) = \sqrt{(l_1 + l_2)^2 - (x_H(t) - (x_i + x_f/2))^2} \quad (8)$$

where $r_4 = -2 \left(\frac{x_f}{2t_f^3} - \frac{(v_s + v_e)}{2t_f^2} \right)$

4 Forward and Inverse Kinematics

Forward kinematics of a biped means finding the position and orientation of the end effector for given joint variables and dimensions of the links. The kinematics equations of swing leg's ankle are obtained by considering hip (H) as the base and ankle (A) as the end effector. So the forward kinematic equations of the swing leg are

$$x_A(t) - x_H(t) = l_1 \cos \theta_1(t) + l_2 \cos(\theta_1(t) + \theta_2(t)); \quad (9)$$

$$z_A(t) - z_H(t) = l_1 \sin \theta_1(t) + l_2 \sin(\theta_1(t) + \theta_2(t)); \quad (10)$$

where $(x_A(t), z_A(t))$ and $(x_H(t), z_H(t))$ are defined earlier.

Stable leg's ankle joint is fixed on the ground (x -axis) and knee joint is locked (no rotation) while hip is moving. Thus, the stable leg moves like single link manipulator with A as base and H as end effector. Its forward kinematic equations are

$$x_H(t) - \left(x_i + \frac{x_f}{2}\right) = (l_1 + l_2) \cos \theta_3(t); \quad (11)$$

$$z_H(t) = (l_1 + l_2) \sin \theta_3(t); \quad (12)$$

where $(x_i + \frac{x_f}{2}, 0)$ is the position of the stable leg's ankle joint fixed on xz -axis. $\theta_1, \theta_2, \theta_3$ are joint angles.

In this paper, we are solving the inverse kinematics problem for the robot legs to follow the hip and ankle trajectories using a feed-forward neural network which has two input neurons, one hidden layer with 10 neurons and 2 output neurons as shown in Fig. 2. The transfer function for the hidden layer is the sigmoid function given by

$$y = \frac{1}{(1 + e^{-x})} \quad (13)$$

At particular instant of time t , we find the position from the trajectory (x_{oi}, y_{oi}) and use it as a input in neural network for finding θ_1 and θ_2 (as output). After that, we put θ_1 and θ_2 in the 2 link manipulator forward kinematics equations and find the position (x_{ni}, y_{ni}) from neural network. Then, compare the original and the neural network value and find the error,

$$E = ((x_{oi} - x_{ni})^2 + (y_{oi} - y_{ni})^2) \quad (14)$$

The objective is to minimize this error using artificial neural network. This is done by updating weights by partially differentiate the error with respect to weights.

$$\delta(i, j) = \frac{\partial E}{\partial W(i, j)} \quad i = [1 : 2], \quad j = [1 : 10] \quad (15)$$

$$\delta(j, k) = \frac{\partial E}{\partial W(j, k)} \quad j = [1 : 10], \quad k = [1 : 2] \quad (16)$$

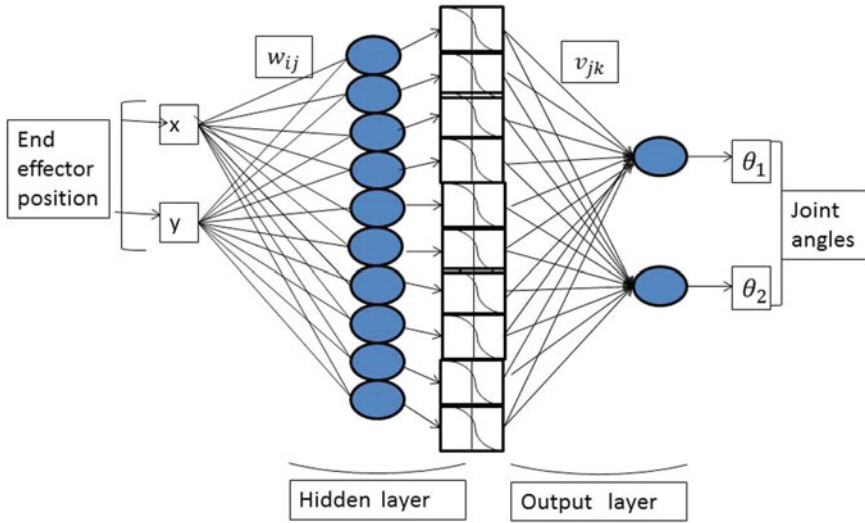


Fig. 2 NN with 2 input, 2 output, 10 neurons in hidden layer

Equation (15) calculates delta (δ) for input layer and Eq. (16) calculates delta (δ) for hidden layer. We update the weights by the following formulas by using delta and the learning rate α ,

$$W_{n+1}(t) = W_n(t) - \alpha \delta \quad 0 < \alpha < 1. \tag{17}$$

Here, n represents the iteration number and learning rate (α) is taken according to the problem. The weights are initialized to 0.1, and the maximum number of iterations is 3200 for ankle trajectory and 50 for hip trajectory. The neural network stops processing if the error reaches below a certain threshold.

5 ZMP Stability Analysis

Zero-moment point (ZMP) is defined as the point where the net moment of the inertial forces and the gravity forces along the axes parallel to the ground is equal to zero. If the ZMP is within the supported region which is the convex hull of all contact points on the floor support, the biped robot is stable and able to walk. In Huang et al. 2001 [4], the value of two scalars representing the ZMP is given as below:

$$x_{ZMP} = \frac{\sum_{i=1}^n m_i(x_i(\ddot{z}_i + g) - \ddot{x}_i z_i)}{\sum_{i=1}^n m_i(\ddot{z}_i + g)}$$

$$y_{ZMP} = \frac{\sum_{i=1}^n m_i (y_i (\ddot{z}_i + g) - \ddot{y}_i z_i)}{\sum_{i=1}^n m_i (\ddot{z}_i + g)}$$

where n is the number of links, m_i is the mass of links, and g is gravity.

In typical human locomotion, the ZMP never stays in a fixed position but moves forward in the direction of locomotion.

6 Upper Body Motion

Modern walking robots usually have heavy upper body as electronic circuits and batteries are there and this mass affect the stability. To ensure stable walking, ZMP must be within the support region. For this, we change the parameters of upper body, try to find the ZMP trajectory which moves in a desired manner. The total mass of upper body is assumed to be a single mass point for planning its trajectory.

6.1 Upper Body Motion on a Frontal Plane

On the frontal plane, upper body mass shifts from one position to another and its trajectory in y -direction highly affects the y -ZMP trajectory. In order to find a desirable ZMP trajectory, we generate three type of upper body mass trajectory in y -direction and choose the one which ensures the higher stability margin. These trajectories are determined by cubic polynomials as given below:

Case-1: As the robot start its step, upper body starts to move from middle of hips to the side of the stable leg's hip during time t_0 to t_1 , stay there during time t_1 to t_3 , then again starts moving towards the middle of legs between t_3 and t_f time in y -direction where $t_1 = t_f/4$ and $t_3 = 3t_f/4$. So the moving mass trajectory in y -direction is given below:

$$y_M(t) = \begin{cases} y_l + y_v t + \left(\frac{3(y_a - y_l)}{t_1^2} - \frac{2y_v}{t_1} \right) t^2 + \left(\frac{-2(y_a - y_l)}{t_1^3} - \frac{y_v}{t_1^2} \right) t^3 & t_0 \leq t \leq t_1 \\ y_a & t_1 \leq t \leq t_3 \\ \left(y_a + \frac{(-3t_f t_3^2 + t_3^3)(y_l - y_a)}{(t_3 - t_f)^3} + \frac{t_f t_3^2 y_v}{(t_3 - t_f)^2} \right) \\ \left(\frac{6t_f t_3 (y_l - y_a)}{(t_3 - t_f)^3} - \frac{(t_3^2 + 2t_f t_3) y_v}{(t_3 - t_f)^2} \right) t + \left(\frac{-3((y_l - y_a)(t_3 + t_f)}{(t_3 - t_f)^3} \right. \\ \left. + \frac{y_v(4t_3 + 2t_f)}{2(t_3 - t_f)^2} \right) t^2 + \left(\frac{2(y_l - y_a)}{(t_3 - t_f)^3} - \frac{y_v}{(t_3 - t_f)^2} \right) t^3 & t_3 \leq t \leq t_f \end{cases}$$

Case-2: As robot start its step, upper body mass starts to move from middle of hip to stable leg's hip during $t \in (t_0, t_f/8)$, fixed there during $t \in (t_f/8, 7t_f/8)$ and then returns back in time $t \in (7t_f/8, t_f)$. Then the moving mass trajectory can be calculated by case-1 equation by putting $t_1 = t_f/8$ and $t_3 = 7t_f/8$.

Case-3: As the robot starts its step, upper body starts to move from middle of both legs to the side of the stable foot from time t_0 to t_2 , then again starts moving towards middle of both legs between t_2 and t_f time in y -direction. So the moving mass trajectory in y -direction is given below:

$$y_M(t) = \begin{cases} y_l + y_v t + \left(\frac{3(y_a - y_l)}{t_2^2} - \frac{2y_v}{t_2}\right)t^2 + \left(\frac{-2(y_a - y_l)}{t_2^3} - \frac{y_v}{t_2^2}\right)t^3 & t_0 \leq t \leq t_2 \\ \left(y_a + \frac{(-3t_f t_2^2 + t_2^3)(y_l - y_a)}{(t_2 - t_f)^3} + \frac{t_f t_2^2 y_v}{(t_2 - t_f)^2}\right) \\ \left(\frac{6t_f t_2 (y_l - y_a)}{(t_2 - t_f)^3} - \frac{(t_2^2 + 2t_f t_2)y_v}{(t_2 - t_f)^2}\right)t + \left(\frac{-3((y_l - y_a)(t_2 + t_f))}{(t_2 - t_f)^3} \right. \\ \left. + \frac{y_v(4t_2 + 2t_f)}{2(t_2 - t_f)^2}\right)t^2 + \left(\frac{2(y_l - y_a)}{(t_2 - t_f)^3} - \frac{y_v}{(t_2 - t_f)^2}\right)t^3 & t_2 \leq t \leq t_f \end{cases}$$

where y_l middle position between both hip, y_a is final position of upper body mass, and $y_v > 0$ is initial velocity of moving mass.

7 Result

The parameters of the biped robot are total length of foot is 6 and width is 4, and initial and end velocity for ankle is 0. Ankle is attached at middle of foot so initial position of ankle is $x_i = 3$. The ankle joint follows a step length $x_f = 14$ from initial position to the final position $x_i + x_f$ with step height $h = 2.5$. $y_l = 4$ is middle position of hip, $y_a = 8.5$ in y -direction. Swing foot is at the position $0 < x < 6$ and $-2 < y < 2$ and stable foot is at the position $7 < x < 13$ and $6 < y < 10$. These units are in inches. Figure 3 presents the desired trajectory graph for the ankle and hip.

The inverse kinematics of these trajectories are calculated using neural network in Matlab. Figures 4 and 5 show the swing leg and stable leg follows the desired trajectories. Joint angles of swing leg and stable leg are calculated using neural network given in Figs. 6 and 7 respectively.

Fig. 3 Ankle and hip trajectory in xz -plane

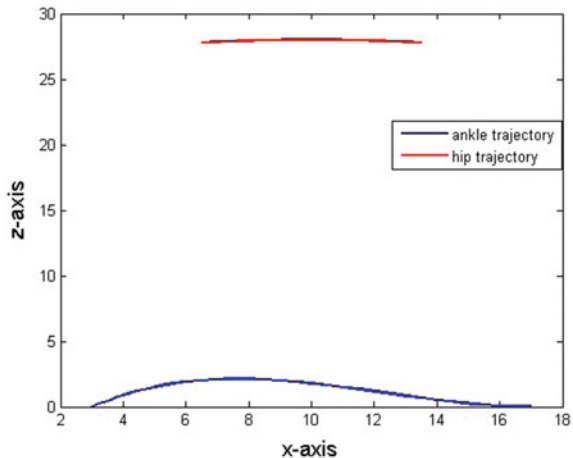


Fig. 4 Swing leg following the given trajectories

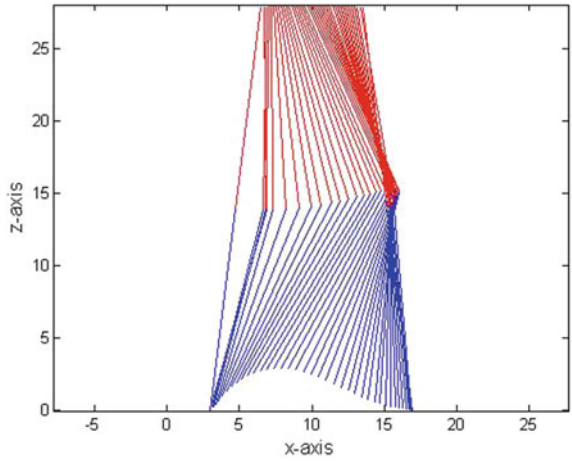


Fig. 5 Stable leg following the given trajectories

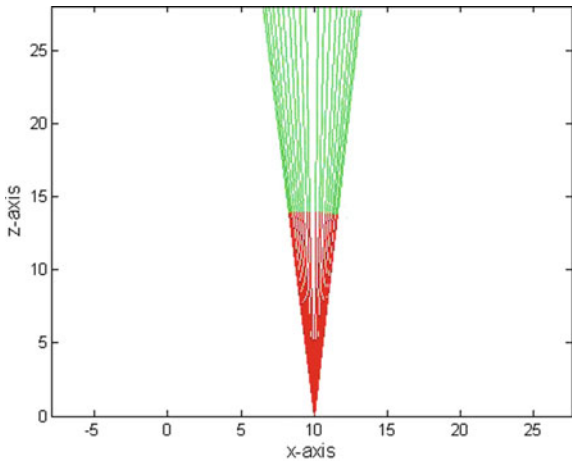


Fig. 6 Joint angles θ_1 and θ_2 of swing leg $t \in (0, t_f)$

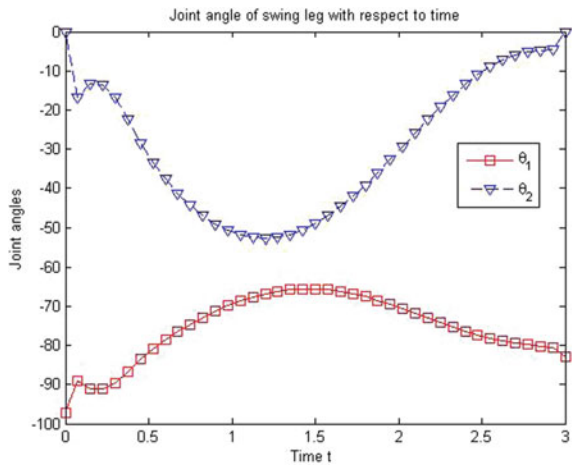


Fig. 7 Joint angles θ_3 of stable leg $t \in (0, t_f)$

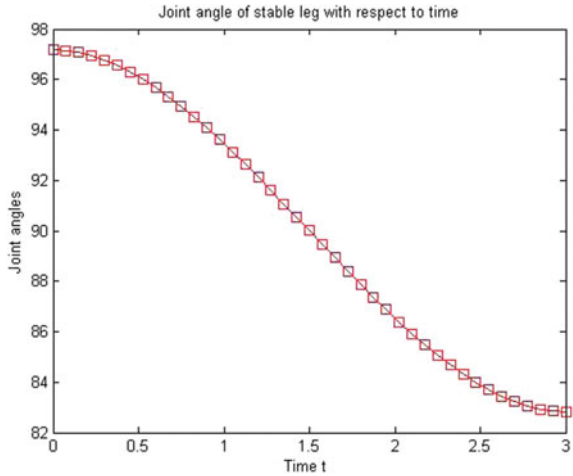
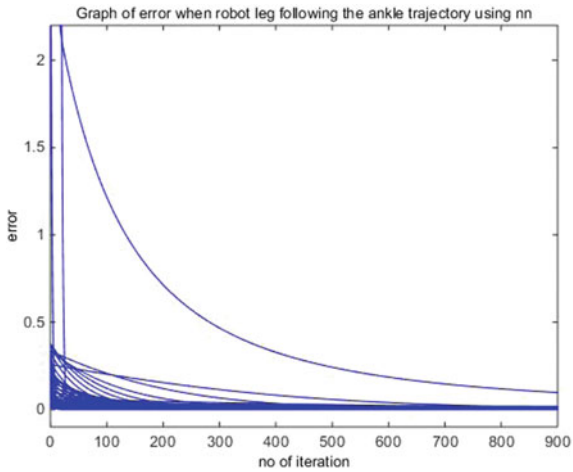


Fig. 8 Error of swing leg's ankle trajectory



Figures 8 and 9 show the error in the NN output for the position of the swing leg's ankle and stable leg's hip (vertical axis) versus the neural network iteration number (horizontal axis). Each curve in Figs. 8 and 9 represents the error graph at a time instant $t = 0.1i$, where $i = 1, 2, 3 \dots 40$.

Figures 10, 12 and 13 present the x - and y -ZMP graph of this biped robot for three step in $t_f = 3$ s.

As we can see from figure that the ZMP varying trajectory is inside the support polygon for case-1,-3 and providing the stable walking for the modeled trajectories. But when we decrease the time then only upper body trajectory in case-3 gives the most stable ZMP trajectory (see Figs. 11 and 14) as given in Table 1 because in this case, upper body moves slowly from middle of hip to stable leg's hip and then returns back in the same manner.

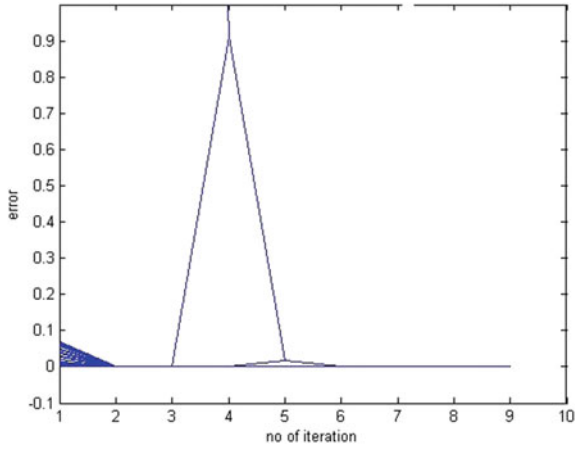


Fig. 9 Error of stable leg's hip trajectory

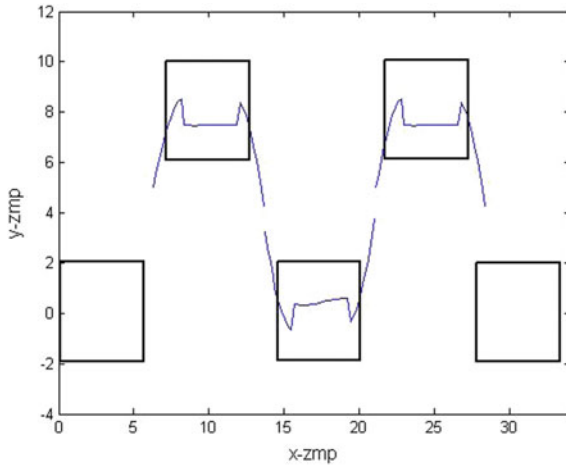


Fig. 10 Stable ZMP trajectory in 3 s of case-1

Figure 14 shows the stable ZMP trajectory for case-3 for $t_f = 1.5$ s.

Whole body motion in 3D for one step with case-3 upper body trajectory is given in Fig. 15.

Fig. 11 Unstable ZMP trajectory 1.5 s of case-1

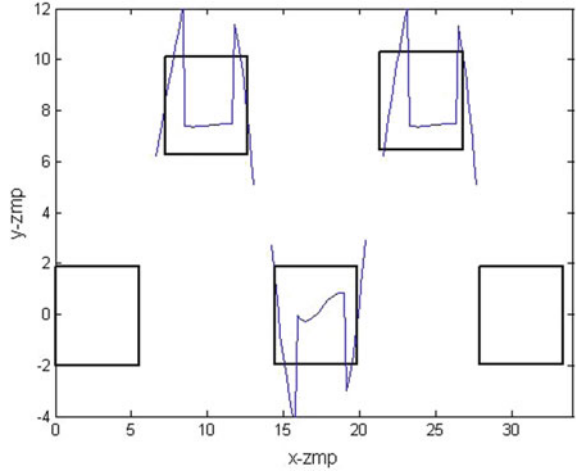


Fig. 12 Unstable ZMP trajectory in 3 s of case-2

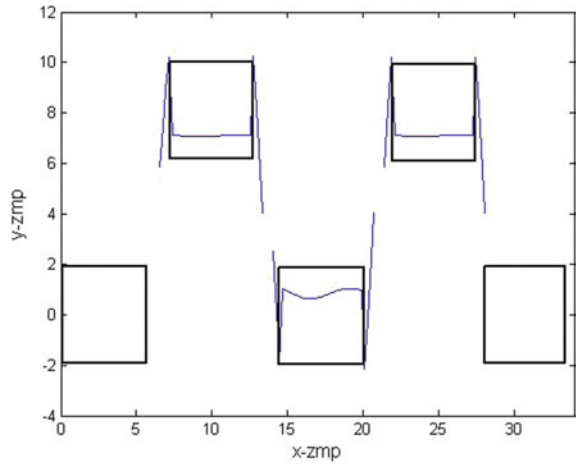
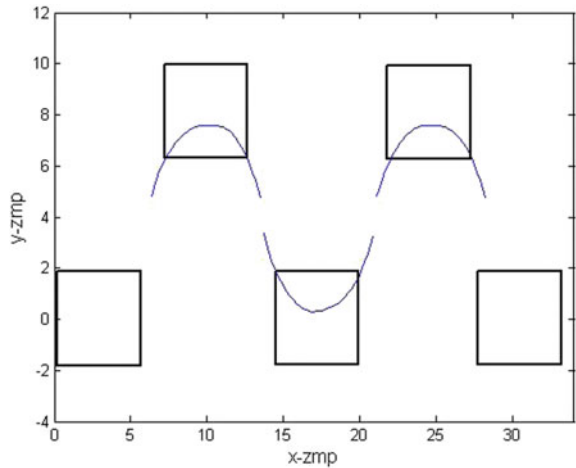


Fig. 13 Stable ZMP trajectory in 3 s for case-3



Hip		Upper body		ZMP
Velocity (in/s)	Time (s)	Trajectory	Initial velocity (in/s)	Stability
$v_s=2.3$	3	Case-1	$y_v = 10$	Stable
$v_s=3.5$	2	Case-1	$y_v = 15$	Stable but small margin
$v_s=4.7$	1.5	Case-1	$y_v = 20$	Unstable
$v_s=2.4$	3	Case-2	$y_v = 16$	Unstable
$v_s=3.5$	2	Case-2	$y_v = 20$	Unstable
$v_s=4.7$	1.5	Case-2	$y_v = 22$	Unstable
$v_s=2.3$	3	Case-3	$y_v = 7.3$	Stable
$v_s=3.5$	2	Case-3	$y_v = 10.3$	Stable
$v_s=4.7$	1.5	Case-3	$y_v = 11$	Stable

Fig. 14 Stable ZMP trajectory in 1.5 s for case-3

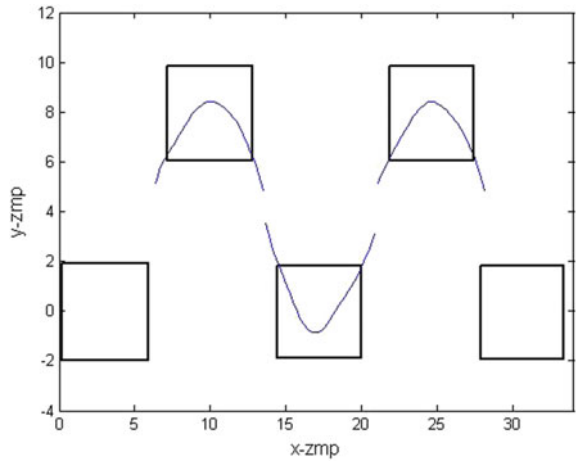
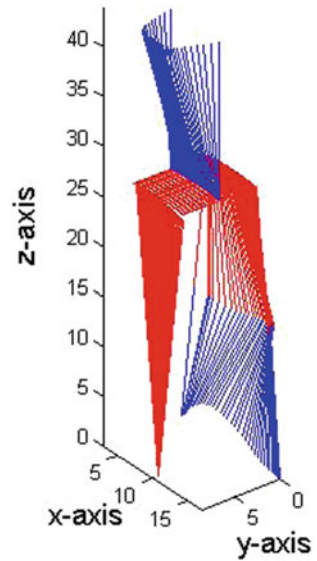


Fig. 15 Biped walk in 3D for one step



Acknowledgements The first author is thankful to the “Ministry of Human Resource and Development India” grant Number MHR-02-23-200-44 for financial support to carry out this research work.

References

1. Vukobratovic, M., Frank, A.A., Juricic, D.: On the stability of biped locomotion. *Biomed. Eng. IEEE Trans.* **1**, 25–36 (1970)
2. McGeer, T.: Passive walking with knees. In: *Proceedings of the IEEE-ICRA*, pp. 1640–1645 (1990)
3. Narvez-Aroche, O., Rocha-Cozatl, E., Cuenca-Jimenez, F.: Kinematic analysis and computation of ZMP for a 12-internal-dof biped robot. *13th World Congress in Mechanism and Machine Science*, pp. 19–25. Guanajuato, Mexico (2011)
4. Huang, Q., Yokoi, K., Kajita, S., Kaneko, K., Arai, H., Koyachi, N., Tanie, K.: Planning walking patterns for a biped robot. *IEEE Trans. Rob. Autom.* **17**(3), 280–289 (2001)
5. Erbaturo, K., Kurt, O.: Humanoid walking robot control with natural ZMP references. In: *Proceedings of the IECON*, France (2006)
6. Xiaoguang, Z., Ruyi, H.: Research on humanoid robot slope gait planning. *Open Autom. Control Syst. J.* **7**, 1002–1009 (2015)
7. Cuevas, E., Zaldivar, D., Perez-Cisneros, M., Ramirez-Ortegon, M.: Polynomial trajectory algorithm for a biped robot. *Int. J. Rob. Autom.* **25**(4), 294–303 (2010)
8. Srinivasan, A., Nigam, M.J.: Neuro-fuzzy based approach for inverse kinematics solution of industrial robot manipulators. *Int. J. Comput. Commun. Control* **3**, 224–234 (2008)
9. Duka, A.V.: Neural network based inverse kinematics solution for trajectory tracking of a robotic arm. *Procedia Techn.* **12**, 20–27 (2014)
10. Liu, J., Urbann, O.: Bipedal walking with dynamic balance that involves three-dimensional upper body motion. *Rob. Auton. Syst.* **77**, 39–54 (2016)
11. Xu, W., et al.: An improved ZMP trajectory design for the biped robot BHR. *Proceedings of the IEEE-ICRA*, Shanghai, China, pp. 569 (2011)

Ant Colony Algorithm for Routing Alternate Fuel Vehicles in Multi-depot Vehicle Routing Problem



Shuai Zhang, Weiheng Zhang, Yuvraj Gajpal and S. S. Appadoo

Abstract A Multi-depot Green Vehicle Routing Problem (MDGVRP) is considered in this paper. An Ant Colony System-based metaheuristic is proposed to find the solution to this problem. The solution for MDGVRP is useful for companies, who employ the Alternative Fuel-Powered Vehicles (AFVs) to deal with the obstacles brought by the limited number of the Alternative Fuel Stations. This paper adds an important constraint, vehicle capacity to the model, to make it more meaningful and closer to real-world case. The numerical experiment is performed on randomly generated problem instances to understand the property of MDGVRP and to bring the managerial insights of the problem.

Keywords Vehicle routing · Multi-depot · Alternative fuel-powered vehicle operations · Fuel tank capacity limitation · Capacitated vehicle

1 Introduction

Recent years, green logistics has become a high-profile research field because of the growing environmental and of the pollution concern worldwide. The current production and distribution system has triggered various environmental problems, which lead to an unsustainable environmental situation.

Under this background, more and more researchers have concentrated on the Green Vehicle Routing problem (GVRP) [1–3]. Different from the classical Vehicle Routing Problem (VRP) which only focuses on the selection of the optimal route by minimizing total transportation cost generated in the process of distribution services, the GVRP emphasizes not only on the optimal economic cost of delivery, but also on addressing sustainable issues in delivery distribution of supply chains [2]. The design of GVRP requires the use of the Alternative Fuel-powered Vehicles (AFV),

S. Zhang · W. Zhang · Y. Gajpal (✉) · S. S. Appadoo
Asper School of Business, University of Manitoba, 181 Freedman Crescent,
Winnipeg, MB R3T 5V4, Canada
e-mail: Yuvraj.Gajpal@umanitoba.ca

© Springer Nature Singapore Pte Ltd. 2019
K. Deep et al. (eds.), *Decision Science in Action*, Asset Analytics,
https://doi.org/10.1007/978-981-13-0860-4_19

251

which relies on greener fuel source such as electricity, natural gas, hydrogen, etc. [1]. However, there are two main obstacles encountered when replacing the conventional vehicles with the AFVs: (1) the limited capacity of the fuel tank or batteries of AFVs, and (2) the scarcity of Alternative Fuel Stations (AFSs). Because of these obstacles, problem formulation and algorithm design of GVRP become more complex than those of VRP [1].

At the same time, the Multi-depot Vehicle Routing Problem (MDVRP) has also attracted a lot of attention [4–6]. In the MDVRP, the fleet of vehicles serves customers from several depots and returns back to the same depot [6]. Research about the MDVRP is meaningful for companies that have a wide range of business scope and have more than one depot because the solution of MDVRP could help these companies reduce their transportation costs and improve their financial performances.

In recent years, many large-scale multinational companies such as UPS, Coca-Cola, and GM have especially paid attention to their environmental sustainable performances and update their sustainability reports every year. They are exhausting their ability to keep a balance between economic performance and environmental protection. For these companies, the solutions for GVRP or MDVRP cannot provide an optimal solution they desired. Most of the GVRP solutions methods only work in situations where there is only one depot and most of the results for the MDVRP only focus on minimizing the transportation cost and ignore the sustainable issues.

Therefore, in this paper, a new variety of problem called the Multi-depot Green Vehicle Routing Problem (MDGVVRP), is addressed. In the MDGVVRP, the AFVs departure from different depots, serve customers, and at the end come back to original depots. Due to the limited capacity of the fuel tank of AFVs and the scarcity of AFSs, each AFV needs to go back its original depot or the nearest AFS to refuel. Based on the two main constraints above, the objective of MDGVVRP is to minimize the route distance of the AFV fleets. Thus, compared with MDVRP or GVRP, MDGVVRP has more constraints and subsequently, is more different to formulate and solve.

It is widely known that VRP is an NP-hard problem, which means that increasing the size of the problem leads to exponential growth in the computational effort required to find the corresponding solution. Because the MDGVVRP is a special variant of the VRP, it can be determined that the MDGVVRP is also NP-hard. Therefore, in this paper, the ant colony algorithm is proposed to find solutions for MDGVVRP.

The structure of the rest of this paper is organized as follows. In Sect. 2, related literature review is presented. Section 3 describes the MDGVVRP problem. Section 4 presents the proposed ant colony algorithm. Numerical experiments are presented in Sect. 5 and are followed by the conclusion in Sect. 6.

2 Literature Review

Because the MDGVVRP is a quite new variety of problem, there is no literature focusing on this area. However, the MDGVVRP is based on the GVRP and MDVRP; therefore, some important previous studies are reviewed in the following sections.

2.1 GVRP

The research of the GVRP just began about 10 years before. However, the GVRP has received extensive attention from researchers because people are becoming aware of the importance of environment protection. According to the comprehensive literature survey on the GVRP of Lin et al. [2], there are mainly two categories of GVRP: Pollution-Routing Problem (PRP) and Green-VRP. Although both these two categories of GVRP focus on economic cost and environment cost simultaneously, the PRP reduces environment cost by minimizing the fuel consumption or minimizing the Green House Gas (GHG) emissions, while the Green-VRP alleviates the environmental damage by using AFVs instead of conventional vehicles. Erdoğan and Miller-Hooks [1] first addressed that the conventional vehicles can be replaced by the AFVs. They proposed a model to help companies which apply the AFVs to optimize the transportation routes in order to overcome the limited capacity of fuel capacity of the AFVs. Based on their work, Schneider et al. [3] added the customer time window constraints to the VRP for electric vehicles. The MDGVRP considered in this paper is based on the Green-VRP of Erdoğan and Miller-Hooks [1]. However, compared with their model, our model considers the demands of customers and can be used to solve the multi-depot problem instead of the single-depot problem.

2.2 MDVRP

The MDVRP was first described in the research of Cassidy and Bennett [4], and is a generalization of the standard VRP, in which there are multiple depots [5]. The MDVRP is very easy to be described. However, an NP-hard problem, the MDVRP is extremely difficult to solve. Therefore, the research of MDVRP mainly focuses on proposing and developing new methods and algorithms to solve the problem. The work of Montoya-Torres et al. [6] revealed that most researchers tend to solve the MDVRP by heuristics or meta-heuristics. For example, Vidal et al. [7] solved the MDVRP by using a hybrid genetic algorithm. In the research of Yu et al. [5], they changed the MDVRP to Single-depot VRP (SVRP) by adding a virtual depot in the first step, and then they applied an improved Ant Colony Optimization (ACO) to solve the SVRP. Therefore, the development of the research on MDVRP is followed by the continually improving the algorithms. In this paper, the ant colony algorithm is developed to solve the MDGVRP.

3 Problem Description

A standard MDGVRP can be described as the problem of designing least distance routes from the N_s ' depots to a set of geographically scattered points (customers).

AFVs start from different depots and serve customers one by one, and finally, they return their original depots. Each customer $c_i \in C$ (customer set) is associated with a non-negative demand q_i to be delivered. To ensure the efficiency of delivery, each customer is visited by the AFVs one time and the demand of customer would be satisfied after this visit. During the service process, the AFVs need to return their original depots to reload to ensure that the remaining cargos always can satisfy the demand of the next customer. Besides, if it is necessary to refuel during the service process, the AFVs have to visit the AFSs or return their original depots to refuel. It is assumed that the number of AFSs visited by an AFV in a tour can be more than one. Besides, a particular AFS can be visited more than once on a given vehicle route. The objective of the problem is to minimize the total distance traveled by all vehicles.

4 Solution of MDGVRP

The proposed algorithm first assigns a customer to its nearest depot. Then a single-depot GVRP is solved for each depot using the Ant Colony System (ACS) algorithm.

4.1 *Ant Colony System (ACS) Algorithm for Single-Depot GVRP*

We solve the single-depot GVRP by using the Ant Colony System (ACS) algorithm. The problem consists of the depot and associated customers. Ants always can find the shortest route between their nest and the food. Through simulating the food-seeking behaviors of ant colonies in nature, the Ant Colony System (ACS) algorithm was developed [8]. During the past several years, the ACS algorithm has been successfully applied to solve the VRP and its variants (e.g., Lin et al. [2], Yu et al. [5], Montoya-Torres et al. [6], Dorigo et al. [8], Bell and McMullen [9], Gajpal and Abad [10], etc.).

In the ACS algorithm, some artificial ants are created to find the feasible solutions based on constraints and trail intensity generated or accumulated during previous iterations. The paths in solutions (routes) with a higher value of the objective function (shorter route distance) accumulate a higher level of trail intensity. The paths with a higher level of trail intensity have a higher chance to be selected by artificial ants in the next iteration. In this way, after several iterations, the near-optimal solution can be found. The fundamental procedures of ACS are as follows:

- Step 1: Initialize the trail intensity matrix, create m artificial ants.
- Step 2: Repeat the following steps until the termination condition is fulfilled.
 - Generate a solution for each ant based on trail intensity.
 - Optimize the solutions by local search.

- Update elitist ants.
- Update trail intensity matrix based on the elitist ant solutions.

Step3: Record the best solution of all generated solutions so far.

4.1.1 Ant Solution Generation

Every GVRP is first simplified as a Traveling Salesman Problem (TSP) and the ACS algorithm is applied to seek the feasible solutions. The feasible solutions of each TSP are the route set which only consists of the original depot and customers. Finally, in the third phase, the TSP solutions found in the second phase are used to build the routes of GVRP. The rules to build these routes are included: (1) insert an AFS or the original depot when the remaining fuel is not enough to support the AFV to reach the next customer on the TSP route or return its original depot and (2) insert the original depot when the remaining products are not able to satisfy the demand of the next customer on the TSP route. In this way, each GVRP can be solved.

In every iteration, there are n number of artificial ants to create n number of TSP solutions (n is the number of customers in the problem). The artificial ants select the next customer mainly based on two factors: the saving value and the trail intensity between two customers.

The saving value S_{ij} represents the saved traveling distance between the customers i and j who are served by one AFV instead of two. The following function shows how to calculate S_{ij} and d_{ij} denotes the distance between the customer i and j :

$$S_{ij} = d_{0i} + d_{j0} - d_{ij}$$

The trail intensity τ_{ij} is defined as the intensity of serving customer j from the customer i and the trail intensity records the information on the visit between two customers. Therefore, at the beginning, all elements in the τ_{ij} matrix are same and are set to 0.01 in this paper.

The saving value (S_{ij}) and trail intensity (τ_{ij}) between two customers constitute the attractiveness value ξ_{ij} between these two customers. And,

$$\xi_{ij} = [S_{ij}]^\alpha [\tau_{ij}]^\beta$$

In this equation, α and β are the biases of saving value and trail intensity, respectively. These two parameters are set at the beginning of the algorithm execution and the values of them need to be altered according to different problem scenarios.

Based on the attractiveness value, the probability of selecting customer j as the next customer from customer i is calculated by the following function:

$$P_{ij} = \frac{\xi_{ij}}{\sum_{k=1}^q \xi_{ixk}}, 1 \leq k \leq q$$

In this function, X_k represents the element of unvisited customer set Ω_q . The set Ω_q contains q number of elements, which means that there are q numbers of unvisited customers. x_k represents the k th element of set Ω_q .

According to the probability calculation function, the m number of artificial ants generates m number of TSP routes in every generation. In the next step, m number of GVRP routes would be generated from TSP route based on the following rules:

- 1) Insert the depot if the remaining load of the vehicle cannot satisfy the demand of the next customer;
- 2) Insert the nearest available fuel station if the remaining fuel level is not enough to get the next customer.

However, sometimes, the quality of the solutions generated in this way is not good enough. To improve the quality of these solutions, the local search is necessary. Local search improves the quality (objective function value) of a solution (a GVRP route) by changing the visiting consequence of a customer to check whether the value objective function can decrease and local search is applied in every iteration after the artificial ants generating new solutions. In this way, the solutions of every iteration can be improved.

4.1.2 Trail Intensity Update

At the end of every iteration, the trail intensity between two customers τ_{ij} needs to be updated to ensure the artificial ants can generate high-quality solutions in the next iteration. To update trail intensity, the elitist ant set which contains λ number of ants (represent λ best solutions in the past iterations) need to be set first. Then, τ_{ij} will be updated according to the solutions of elitist ant set. The function to change τ_{ij} is as follows:

$$\tau_{ij}^{\text{new}} = \tau_{ij}^{\text{old}} \times \varphi + \sum_{\theta=1}^{\lambda} \tau_{ij}^{\theta}, i \neq j \text{ and } i, j = 1, 2, \dots, n$$

In this equation, τ_{ij}^{old} represents the old trail intensity accumulated until the last iteration and φ is the trail persistence which is between 0 and 1. The number of φ determines the decreasing speed of pheromone density, and is set as 0.95. The second term of the equation represents the pheromone increase brought by the elitist ant θ . And the value of τ_{ij}^{θ} is determined by

$$\tau_{ij}^{\theta} = \begin{cases} 0 & \text{if the edge between customer } i \text{ and } j \text{ is not in the elitist ant route.} \\ \frac{1}{l^{\theta}} & \text{otherwise.} \end{cases}$$

l^{θ} represents the route length of θ th elitist ant solution.

Table 1 Strategic location of AFS

Pattern	Number of AFSs	Details
1	2	The grid is horizontally divided into two equal sections with each AFS randomly assigned to the two sections
2	4	The grid was divided into four equal sections with each assigned an AFS
3	6	This is similar to pattern 2 except that the two additional AFSs are distributed using pattern 1
4	8	This is similar to pattern 3 with the grid vertically divided into two equal section and the two additional AFSs are randomly assigned to each section

5 Numerical Experiment and Analysis

To test the validity of the proposed algorithm, the numerical experiment is designed. Totally, 48 problem instances are created. In every instance, the different participants in the MDGVRP are set in a 330 by 300 miles grid. The first 24 instances (MDGVRP1-24) have 4 depots and other instances (MDGVRP25-48) have 6 depots. Two locating schemes of AFSs are considered. To be specific, in the instances MDGVRP1-12 and MDGVRP25-36, the AFSs are located strategically according to the principles shown in Table 1. In the instance MDGVRP 13-24 and MDGVRP37-48, the AFSs are located randomly. In addition, each instance has different numbers of customers and AFSs. The detailed characteristics of instances are given in Table 2.

In the experiment, the capacity of fuel tank is set as 60 gallons. The vehicle capacity is assumed to be 300 units of particular cargos. The fuel consumption rate is set at 0.2 gallons per mile. One of the rules for generating the data used in the experiment is that one tank of fuel is enough for a vehicle to reach to a customer from depot via an AFS.

The construction of algorithm is coded in C programming and implemented on AMD Opteron 2.3 GHz with 16 GB of RAM. The result of instances with strategic AFS location and random AFS location are shown in Tables 2 and 3 respectively. All problem instances are solved in seconds.

The results reported in Tables 2 and 3 show that the ACS can solve the MDGVRP in seconds. The solved instances vary in terms of the number of customers, AFSs, and depots and show the scalability of the proposed ACS on solving the MDGVRP. Further, the results show that the strategic location of AFSs can minimize the total route length, because the average route length of instances with the strategic AFSs location is less than that of instances with random AFSs locations. However, this observation does not hold for every instance used.

It is also worth to mention that the growth in the number of depots leads to the decrease in the route length. However, more depots can raise the maintenance costs

Table 2 Results of instances with strategic AFS location

Instance	Quantity of customers	Quantity of AFSs	Number of depots	Distance
MDGVRP1	25	2	4	958.933
MDGVRP2	50	2	4	1420.91
MDGVRP3	75	2	4	1870.26
MDGVRP4	25	4	4	1072.3
MDGVRP5	50	4	4	1499.85
MDGVRP6	75	4	4	1714.51
MDGVRP7	25	6	4	974.518
MDGVRP8	50	6	4	1418.11
MDGVRP9	75	6	4	1845.3
MDGVRP10	25	8	4	1106.49
MDGVRP11	50	8	4	1336.83
MDGVRP12	75	8	4	1817.46
MDGVRP25	25	2	6	815.249
MDGVRP26	50	2	6	1494.85
MDGVRP27	75	2	6	1876.56
MDGVRP28	25	4	6	1106.16
MDGVRP29	50	4	6	1354.45
MDGVRP30	75	4	6	1683.09
MDGVRP31	25	6	6	1022.31
MDGVRP32	50	6	6	1300.97
MDGVRP33	75	6	6	1746.58
MDGVRP34	25	8	6	871.961
MDGVRP35	50	8	6	1235.61
MDGVRP36	75	8	6	1788.41
Average				1388.81

and increase the vehicle used in delivery. Therefore, future research can focus on determining the optimal quantity of depots in the distribution network.

6 Conclusion

In this paper, the formulation of the MDGVRP is proposed and the algorithm based on the ACS is designed to solve this problem. The ACS algorithm seeks the shortest tour when considering the vehicle capacity and the fuel tank capacity.

Numerical experiments illustrate that the proposed algorithm performs well and can be used to deal with different instances. The results of numerical experiments also show some implications to the company who has employed the AFVs or intends

Table 3 Results of instances with random AFS location

Instance	Quantity of customers	Quantity of AFSs	Number of depots	Distance
MDGVRP13	25	2	4	1166.79
MDGVRP14	50	2	4	1417.87
MDGVRP15	75	2	4	2269.84
MDGVRP16	25	4	4	976.786
MDGVRP17	50	4	4	1451.22
MDGVRP18	75	4	4	1778.68
MDGVRP19	25	6	4	1151.4
MDGVRP20	50	6	4	1466.59
MDGVRP21	75	6	4	1886.86
MDGVRP22	25	8	4	1028.35
MDGVRP23	50	8	4	1497.78
MDGVRP24	75	8	4	1631.41
MDGVRP37	25	2	6	1149.69
MDGVRP38	50	2	6	1433.49
MDGVRP39	75	2	6	2193.64
MDGVRP40	25	4	6	900.919
MDGVRP41	50	4	6	1433.75
MDGVRP42	75	4	6	1846.76
MDGVRP43	25	6	6	1125.36
MDGVRP44	50	6	6	1394.92
MDGVRP45	75	6	6	1863.23
MDGVRP46	25	8	6	945.133
MDGVRP47	50	8	6	1486.94
MDGVRP48	75	8	6	1671.45
Average				1465.37

to use in the future. The first implication is that the company has to decide the number of depots based on the calculation of benefits induced by the AFSs and the additional costs induced by depots maintenance. In addition, we also find that the limited fuel tank capacity of AFVs creates more complexity to the routing problem. This situation is quite different from the classic routing problem where the traditional fuel tank capacity is large enough traveling for a fairly long distance.

References

1. Erdođan, S., Miller-Hooks, E.: A green vehicle routing problem. *Transp. Res. Part E Logist Transp. Rev.* **48**(1), 100–114 (2012)

2. Lin, C., Choy, K.L., Ho, G.T.S., Chung, S.H., Lam, H.Y.: Survey of green vehicle routing problem: past and future trends. *Expert Syst. Appl.* **41**(4 PART 1), 1118–1138 (2014)
3. Schneider, M., Stenger, A., Goeke, D.: The electric vehicle routing problem with time windows and recharging stations. *Transp. Sci.* **48**(4), 500–520 (2014)
4. Cassidy, P.J., Bennett, H.S.: TRAMP—a multi-depot vehicle scheduling system. *Oper. Res. Q.* **23**(2), 151–163 (1972)
5. Yu, B., Yang, Z.-Z., Xie, J.-X.: A parallel improved ant colony optimization for multi-depot vehicle routing problem. *J. Oper. Res. Soc.* **62**(1), 183–188 (2011)
6. Montoya-Torres, J.R., López Franco, J., Nieto Isaza, S., Felizzola Jiménez, H., Herazo-Padilla, N.: A literature review on the vehicle routing problem with multiple depots. *Comput. Ind. Eng.* **79**, 115–129 (2015)
7. Vidal, T., Crainic, T.G., Gendreau, M., Lahrichi, N., Rei, W.: A hybrid genetic algorithm for multidepot and periodic vehicle routing problems. *Oper. Res.* **60**(3), 611–624 (2012)
8. Dorigo, M., Maniezzo, V., Colomi, A.: Ant system: optimization by a colony of cooperating agents **26**(1), 1–13 (1996)
9. Bell, J.E., McMullen, P.R.: Ant colony optimization techniques for the vehicle routing problem. *Adv. Eng. Informatics.* **18**(1), 41–48 (2004)
10. Gajpal, Y., Abad, P.: An ant colony system (ACS) for vehicle routing problem with simultaneous delivery and pickup. *Comput. Oper. Res.* **36**(12), 3215–3223 (2009)

Semidefinite Approximation of Closed Convex Set



Anusuya Ghosh and Vishnu Narayanan

Abstract Approximation of convex sets takes a major role in optimization theory and practice. Approximation by semidefinite representable set draws more attention as semidefinite programming problems can be solved very efficiently using numerous existing algorithms. We contribute a technique by which a closed convex set can be approximated by a compactly semidefinite representable set. Further, we extend the technique of approximation and we prove that a closed convex set can be approximated by semidefinite representable set. These results give new techniques in semidefinite programming.

Keywords Semidefinite representation · Convex set · Approximation
Semidefinite representable set

1 Introduction

The approximation of convex sets takes an important role in modern convex optimization. Several types of approximations have been discussed. Approximating a norm in any vector space or generally approximating the Minkowski functional of a convex body in vector space by a polynomial is contributed in [1]. Approximating an Euclidean ball [2], a zonoid [3], a symmetric convex body [4], any convex body [4], a cut-norm [5], a second-order cone [6] and a p -order cone [7] have been developed.

An ellipsoidal approximation is defined in [8]. For any convex body K , there exists a unique ellipsoid E such that E has the largest volume among all ellipsoids

A. Ghosh (✉)
Production and Operations Management,
Indian Institute of Management Bangalore, Bengaluru, India
e-mail: ghosh.anusuya007@gmail.com

V. Narayanan
Industrial Engineering and Operations Research,
Indian Institute of Technology Bombay, Mumbai, India
e-mail: vishnu@iitb.ac.in

contained in K [8, Theorem 9.3]. The ellipsoid is known as the John ellipsoid of K . If the convex body K is symmetric, then the John ellipsoid is centred at the origin.

The approximation of a convex body B in real vector space V by a set X is discussed in [4] such that the membership question: “given an $x \in V$, does x belongs to X ?” can be evaluated efficiently. Several optimization problems provide various convex bodies for which the membership question is very hard to solve. So, the main aim of approximating any convex body B by an efficient computable set X lies in the fact that the membership question can be solved easily. The convex body B is called symmetric if $B = -B$. A norm is $\|\cdot\|$ associated with the convex body B such that

$$\|b\| = \inf\{\lambda > 0 : b \in \lambda B\}.$$

Approximation of the convex body B by a set X is equivalent to the problem of approximating its norm $\|\cdot\|$ by an efficient function f . Thus, the symmetric convex body B can be efficiently approximated by algebraic hyper-surfaces as given in [4, Theorem 2.2]. The set $X = \{v \in V : p(v) \leq 1\}$ where p is a homogeneous polynomial from V to \mathbb{R} , which approximates the symmetric convex body B efficiently.

Any convex body B can be arbitrarily well approximated by a polytope X [4, Sect. 3]. The size of the polytope X plays a main role in this context. The bound on the size of the set X is given by $|X| \leq (1 + \frac{2}{\epsilon})^d$ [9] for any symmetric convex body B in d -dimensional vector space V and for any $1 > \epsilon > 0$. The bound on the size of set X is given by $|X| \geq \exp\{\frac{d}{2\alpha^2}\}$ for a unit ball $B \subseteq \mathbb{R}^d$ such that $\text{conv}(X) \subseteq B \subseteq \alpha \text{conv}(X)$ [8].

It is discussed in [4] that any convex body K in vector space V can be approximated by a section of a polytope P . The main idea is to construct a vector space $W \supset V$ and a polytope $P \subseteq W$ such that $P \cap V$ approximates B . Sections and projections are inter-related. A section of a polytope with at most n vertices can be represented as a projection of a polytope with at most n facets. But the interesting feature of approximations of symmetric convex body B by sections or projections is that they break symmetry. So, we need to approximate a symmetric convex body B by a polytope P which may not be symmetric. The paper [8] shows that the unit ball cannot be approximated tightly by a polytope, although the unit ball can be efficiently approximated by a projection of a polytope or by a section of a polytope [4, Sect. 4.4]. This approach of approximating the unit ball has been generalized in Sect. 4.5 [4]. In Sect. 6, [4] the approximation of a convex body B by a section of the cone of positive semidefinite quadratic forms has been contributed. The main application of this approach is associated with the cut polytope [10].

Let K be a convex compact semi-algebraic set in \mathbb{R}^n . In [11, Lemma 5.1] the set K is being characterized as a projection of a semi-infinite semidefinite representable set S_∞ . The semi-infinite set S_∞ is defined by finitely many LMIs involving matrices of infinite dimension and countably many variables. A procedure is developed in [11, Corollary 5.2(b)] to obtain a sequence of monotone non-increasing outer convex approximations of K . Each outer convex approximation of K is semidefinite representable and its semidefinite representation can be obtained by finite truncation of the representation of S_∞ .

Let A_k be the intersection of cone of positive semidefinite quadratic forms with affine subspaces for any $k = 1, 2, \dots$. Approximation of any convex body B in a vector space by the sequence of set $\{A_k\}_{k=1}^{\infty}$ has been discussed in [12]. It is shown [12] that each A_k is contained in convex body B . This approximation can be applied for symmetric travelling salesman polytope.

We move to the problem of polyhedral approximation of Lorentz cone [6, Theorem 1.1]. This problem is equivalent to the problem of converting conic quadratic problem to a linear programming problem. To the best of our knowledge the software for conic quadratic programming problem is capable of handling problems with tens of thousands of conic quadratic constraints with severe restrictions on the design dimension of the problem (a few thousand variables). In linear programming problem, we can solve routine problems with even hundreds of thousands of variables and constraints. So, polyhedral approximations play a vital role in optimization. The construction of polyhedral approximation of the Lorentz cone is discussed in Sect. 2 of [6].

It is proved in [13] that any closed convex set K can be approximated by boundedly polyhedral set P . A technique has been developed to obtain boundedly polyhedral approximation of closed convex set K in the Theorem 6.3 [13]. We generalize ‘boundedly polyhedral set’ as ‘compactly semidefinite representable set’ as we discussed in Sect. 2. In this chapter, we contribute a technique to approximate any closed convex set K by a compactly semidefinite representable set P .

To optimize a linear function over a convex set, say K , is a hard problem. But optimizing the linear function over the semidefinite representable set which approximates the convex set K is easy to solve as there exists numerous efficient algorithms [14–16] to solve semidefinite programming problems [17]. So, our approximation technique is significant in optimization.

Contribution

This chapter presents results on approximating any closed convex set, say K , by a compactly semidefinite representable set P . We develop a technique to construct such compactly semidefinite representable set P from the closed convex set K such that P efficiently approximates set K . We show that there exists a sequence of compactly semidefinite representable sets which give tighter approximation of K gradually. We discuss the convergence of the sequence of compactly semidefinite representable sets to closed convex set K , where we show that the recession cone of K and recession cones of compactly semidefinite representable sets say $\{P_i\}_{i=1}^{\infty}$ are equal.

Notation

The set $A(x, \epsilon)$ is the union of all open ϵ -neighbourhoods of the points of X . The relative interior of a set S is $\text{relint}(S)$. The set theoretic operations union, intersection and difference are denoted by \cup, \cap and \setminus , respectively. The Hausdorff distance, $d_H(X, Y)$ between two sets X and Y in \mathbb{R}^n is defined as the greatest lower bound of numbers d such that $X \subseteq A(Y, d)$ and $Y \subseteq A(X, d)$. The Hausdorff distance may

be infinite when the sets are unbounded. For any a, b in \mathbb{R}^n , the closed interval $[a, b] = \{\lambda a + (1 - \lambda)b : 0 \leq \lambda \leq 1\}$. The transpose of a vector c is c^T . The set $N_{\mu x}$ is the neighbourhood of the point x .

2 Approximation of Convex Set by Compactly Semidefinite Representable Set

This section deals with the approximation of any closed convex set K by a compactly semidefinite representable set P . The technique to construct the compactly semidefinite representable set P from the closed convex set K is established in the Theorem 1. It is given below.

Theorem 1 *Suppose K is a closed convex subset of \mathbb{R}^n , K contains no line, and μ is a continuous function on K to $]0, \infty[$. Then, there is a compactly semidefinite representable set P such that $P \subseteq K \subseteq \cup_{x \in P} A(x, \mu x)$.*

Proof The proof is divided into several steps and the steps are given below.

- Step 1: Let p be any point in K . Let us consider C as the recession cone of K such that $C = \{x : [0, \infty[x \subseteq K - p\}$. As C contains no line, C° is not contained in any hyperplane of \mathbb{R}^n . So, C° must have interior points. It is clear that C° is a closed convex cone with vertex 0.
- Step 2: Let us consider a set F such that $F = \{f \in \mathbb{R}^n : f > 0 \text{ on } C \setminus \{0\}\}$. We get $F = -\text{int}(C^\circ)$. Thus F is a convex cone with vertex 0.
- Step 3: Let us consider a closed half-space in \mathbb{R}^n as $\{x : f^T x \leq r\}$. Any translate of the half-space is $\{x : f^T x \leq s\}$. For any $f \in F$, the set $\{x : f^T x \leq s\} \cap K$ contains no ray and is thus a bounded set.
- Step 4: Let us consider that $m_0 = \inf_{k \in K} f^T k$ and $m_0 > 0$. Let us consider the set $K_{m_0 r_1}$ to be $K_{m_0 r_1} = \{x : m_0 \leq f^T x \leq r_1\} \cap K$. The set $K_{m_0 r_1}$ is compact. Let us consider that $d_{r_1} = \mu K_{m_0 r_1}$, where $d_{r_1} > 0$. Let us consider

$$K_{m_i r_{i+1}} = \{x : m_i \leq f^T x \leq r_{i+1}\} \cap K, \tag{1}$$

where $r_{i+1} > m_i + \delta$, $m_i = r_i$ and $r_1 > m_0$ for any $i = 1, 2, 3, \dots$ and where $\delta > 0$ is a small real number. Thus there exists an increasing sequence r_α in $[m_0, \infty]$ such that

$$K \subseteq \cup_{i=1}^\infty A\left(K_{m_{i-1} r_i}, \frac{1}{2} d_{r_i}\right). \tag{2}$$

- Step 5: Let F_1 be a compact set in $\text{relint}(K_{m_0 r_1})$ such that $K_{m_0 r_1} \subseteq A\left(F_1, \frac{1}{2} d_{r_1}\right)$ and $B_1 = \text{conv}(F_1)$ where B_1 is semidefinite representable. As $F_1 \subseteq \text{relint}(K_{m_0 r_1})$ and hence, $B_1 \subseteq \text{relint}(K_{m_0 r_1})$ and $K_{m_0 r_1} \subseteq A\left(B_1, \frac{1}{2} d_{r_1}\right)$.
- Step 6: Let $\text{conv}(B_1 \cup K_{m_2 r_3}) = J_{13}$ and it is trivially compact set such that $J_{13} \setminus K_{m_2 r_3} \subseteq \text{relint}(K)$. So, $J_{13} \cap K_{m_1 r_2} \subseteq \text{relint}(K_{m_1 r_2})$. It is thus possible to produce

semidefinite representable set B_2 such that

$$J_{13} \cap K_{m_1 r_2} \subseteq B_2 \subseteq \text{relint}(K_{m_1 r_2}) \subseteq K_{m_1 r_2} \subseteq A \left(B_2, \frac{1}{2} d_{r_2} \right).$$

- Step 7: In this way we obtain a sequence B_n of semidefinite representable sets such that

$$J_{(n-1)(n+1)} \cap K_{m_{(n-1)r_n} r_n} \subseteq B_n \subseteq \text{relint}(K_{m_{(n-1)r_n} r_n}) \subseteq K_{m_{(n-1)r_n} r_n} \subseteq A \left(B_n, \frac{1}{2} d_{r_n} \right) \subseteq A(B_n, d_{r_n}) \tag{3}$$

for each $n \geq 2$.

- Step 8: Let us consider that $P = \text{conv}(\cup_{i=1}^{\infty} B_i)$.

$$\begin{aligned} x \in P &\implies x \in \text{conv}(\cup_{i=1}^{\infty} B_i) \\ &\implies x = (1 - \lambda)u + \lambda v; u, v \in B_i \text{ for some } i \\ &\implies x = (1 - \lambda)u + \lambda v; u, v \in K_{m_{(i-1)r_i} r_i} \text{ for some } i \\ &\implies x = (1 - \lambda)u + \lambda v; u, v \in K \\ &\implies x \in K \end{aligned}$$

Thus, $P \subseteq K$ and for each n it is true that

$$A \left(K_{m_{(n-1)r_n} r_n}, \frac{1}{2} d_{r_n} \right) \subseteq A(B_n, d_{r_n}) \subseteq \cup_{x \in B_n} A(x, \mu x).$$

So

$$\cup_{n=1}^{\infty} A \left(K_{m_{(n-1)r_n} r_n}, \frac{1}{2} d_{r_n} \right) \subseteq \cup_{n=1}^{\infty} [\cup_{x \in B_n} A(x, \mu x)].$$

- Step 9: We say

$$\begin{aligned} K &\subseteq \cup_{n=1}^{\infty} A \left(K_{m_{(n-1)r_n} r_n}, \frac{1}{2} d_{r_n} \right) \subseteq \cup_{n=1}^{\infty} [\cup_{x \in B_n} A(x, \mu x)] \\ &\implies K \subseteq \cup_{n=1}^{\infty} [\cup_{x \in B_n} A(x, \mu x)] \\ &\implies K \subseteq \cup_{x \in K} A(x, \mu x). \end{aligned}$$

Thus, we get $P \subseteq K \subseteq \cup_{x \in K} A(x, \mu x)$.

- Step 10: Let us consider $i < j < k$ and B_i, B_j, B_k are semidefinite representable sets. Let $p \in B_i$ and $q \in B_k$ and $j = i + 1, k = i + 2$, then the segment joining p and q intersects $K_{m_i r_{(i+1)}}$. So, it intersects B_{i+1} . Thus, every segment from B_i to B_k intersects B_j . The set $\text{conv}(B_i \cup B_j \cup B_k)$ is the union of all segments $[p, q]$ such that $q \in B_k$ and $p \in [v, w]$ for some $v \in B_i$ and $w \in B_j$. For such p, q, v, w the segment $[q, v]$ must intersect B_j at some point s and $[p, q] \subseteq \text{conv}\{w, s, q\} \cup \text{conv}\{w, s, v\}$. Thus,

$$\text{conv}(B_i \cup B_j \cup B_k) = \text{conv}(B_i \cup B_j) \cup \text{conv}(B_j \cup B_k). \tag{4}$$

An application of Eq. (4) shows that

$$P = \bigcup_{i=1}^{\infty} \text{conv}(B_i \cup B_{(i+1)}).$$

- Step 11: P is an unbounded set and is the infinite union of bounded sets of the form $\text{conv}(B_i \cup B_{i+1})$, where B_i, B_{i+1} are compact semidefinite representable sets. So, $\text{conv}(B_i \cup B_{i+1})$ is compact semidefinite representable set. Let us consider any compact semidefinite representable set B intersecting the set P . As B is bounded, B intersects only finite number of sets of the form $\text{conv}(B_i \cup B_{i+1})$, say B intersects m number of sets of the form $\text{conv}(B_i \cup B_{i+1})$. Thus, we get

$$\begin{aligned} P \cap B &= [\bigcup_{i=1}^{\infty} \text{conv}(B_i \cup B_{(i+1)})] \cap B, \\ &= [\bigcup_{i=1}^m \text{conv}(B_i \cup B_{(i+1)})] \cap B, \\ &= \text{conv}(B_1 \cup \dots \cup B_{m+1}) \cap B. \end{aligned}$$

As B_1, \dots, B_{m+1} are compact semidefinite representable sets, $\text{conv}(B_1 \cup \dots \cup B_{m+1})$ is semidefinite representable. Thus, $B \cap \text{conv}(B_1 \cup \dots \cup B_{m+1})$ is semidefinite representable. So, P is compactly semidefinite representable set.

Hence, the proof is complete. □

Corollary 1 *If K is a closed convex subset of \mathbb{R}^n and $\epsilon > 0$, there are compactly semidefinite representable sets P and Q such that $P \subseteq K \subseteq A(P, \epsilon)$ and $K \subseteq Q \subseteq A(K, \epsilon)$.*

Proof Let us assume that K contains no line. Using Theorem 1, we have

$$P \subseteq K \subseteq \bigcup_{x \in P} A(x, \mu x), \tag{5}$$

where $\mu : K \rightarrow]0, \infty[$ is a continuous function. We know that $A(x, \mu x)$ is the set of union of all μx -neighbourhoods of x where $x \in P \subseteq K$. Let us assume that $\mu x = \epsilon$. So, we have

$$\begin{aligned} A(x, \mu x) &= A(x, \epsilon), \\ \implies \bigcup_{x \in P} A(x, \mu x) &= \bigcup_{x \in P} A(x, \epsilon), \\ &= A(P, \epsilon). \end{aligned}$$

Thus, we get a compactly semidefinite representable set P such that

$$P \subseteq K \subseteq A(P, \epsilon). \tag{6}$$

Let us consider the set $\text{cl conv } A(K, \epsilon)$ which is a closed convex set in \mathbb{R}^n . We apply the above result and using Eq. (6) we get

$$Q \subseteq \text{cl conv } A(K, \epsilon) \subseteq A(Q, \epsilon), \quad (7)$$

where Q is a compactly semidefinite representable set. Let us assume that $Q \subseteq K$. We get

$$\begin{aligned} Q &\subseteq K, \\ \implies A(Q, \epsilon) &\subseteq A(K, \epsilon), \\ \implies A(Q, \epsilon) &\subseteq \text{cl conv } A(K, \epsilon). \end{aligned}$$

This is a contradiction to the fact that $\text{cl conv } A(K, \epsilon) \subseteq A(Q, \epsilon)$ Eq. (7). Thus, $Q \not\subseteq K$.

Let us assume that $K \cap Q = \phi$. Then $\text{cl conv } A(K, \epsilon) \subseteq A(Q, \epsilon)$ is not possible. So, $K \cap Q = \phi$ is not true.

We know

$$\begin{aligned} K &\subseteq \text{cl conv } A(K, \epsilon) \subseteq A(Q, \epsilon), \\ \implies A(K, \epsilon) &\subseteq A(Q, \epsilon). \end{aligned} \quad (8)$$

Let us consider that $K \not\subseteq Q$ and K intersects Q at some points. Then neither $A(K, \epsilon) \subseteq A(Q, \epsilon)$ nor $A(Q, \epsilon) \subseteq A(K, \epsilon)$ is true. This is a contradiction to Eq. (8). Thus, this case is false.

Hence, we get $K \subseteq Q$. We write

$$K \subseteq Q \subseteq \text{cl conv } A(K, \epsilon) \subseteq A(Q, \epsilon). \quad (9)$$

Now, we have

$$\begin{aligned} K &\subseteq Q, \\ \implies A(K, \epsilon) &\subseteq A(Q, \epsilon). \end{aligned}$$

Hence, either $Q \subseteq A(K, \epsilon)$ or $A(K, \epsilon) \subseteq Q$ is true. If $A(K, \epsilon) \subseteq Q$ holds true, we get $\text{cl conv } A(K, \epsilon) \subseteq Q$. This is a contradiction to Eq. (7). So, we get $Q \subseteq A(K, \epsilon)$. Thus, we get

$$K \subseteq Q \subseteq A(K, \epsilon). \quad (10)$$

We combine Eqs. (6) and (10). Hence, the proof is complete. \square

If $\epsilon > 0$ and K is a bounded convex set with boundary W , then K can be ϵ approximated in the way discussed in Corollary 1 by a set $\text{conv}(Y)$ where $Y \subseteq W$ and by semidefinite representable sets which are intersections of supporting half-spaces of W . When K is unbounded set then compactly semidefinite representable approximations of this type may not exist. The circular cone cannot be approximated by compactly semidefinite representable set. There exists another type of characteriza-

tion of all convex sets. These types of weaker uniform approximations give us future research direction.

3 Existence of a Sequence of Compactly Semidefinite Representable Sets

This section establishes the existence of a sequence of compactly semidefinite representable sets say $\{P_i\}_{i=1}^\infty$. The result is given below.

Theorem 2 *Suppose K is a closed convex subset of \mathbb{R}^n , K contains no line, and μ is a continuous function on K to $]0, \infty[$. Then there exists a sequence of compactly semidefinite representable sets $\{P_i\}_{i=1}^\infty$ such that*

$$P_1 \subseteq P_2 \subseteq \dots \subseteq P_n \subseteq P_{n+1} \subseteq \dots \subseteq K$$

$$\subseteq \cup_{x \in P_1} A(x, \mu x) \subseteq \cup_{x \in P_2} A(x, \mu x) \subseteq \dots \subseteq \cup_{x \in P_n} A(x, \mu x) \subseteq \cup_{x \in P_{n+1}} A(x, \mu x) \subseteq \dots$$

Proof From Theorem 1, Eq. (3) we have the following relation:

$$J_{(n-1)(n+1)} \cap K_{m_{(n-1)r_n}} \subseteq B_n \subseteq \text{relint}(K_{m_{(n-1)r_n}}) \subseteq K_{m_{(n-1)r_n}}$$

$$\subseteq A\left(B_n, \frac{1}{2}d_{r_n}\right) \subseteq A(B_n, d_{r_n}).$$

Let us consider another polytope or a compact semidefinite representable set A_n such that $\text{conv}(B_n \cup A_n) = B'_n$. We have

$$J_{(n-1)(n+1)} \cap K_{m_{(n-1)r_n}} \subseteq B_n \subseteq B'_n \subseteq \text{relint}(K_{m_{(n-1)r_n}}) \subseteq K_{m_{(n-1)r_n}}$$

$$\subseteq A\left(B_n, \frac{1}{2}d_{r_n}\right) \subseteq A(B_n, d_{r_n}).$$

Let us consider the set $\text{conv}(\cup_{i=1}^\infty B'_i) = P'$. Let us consider any point x in P' . Then, we get

$$x = (1 - \lambda)u + \lambda v; u, v, \in B'_i \text{ for some } i,$$

$$\implies x = (1 - \lambda)u + \lambda v; u, v, \in K,$$

$$\implies x \in K.$$

Thus, we get $P' \subseteq K$ and it is obvious from the construction of P' that $P \subseteq P' \subseteq K \subseteq \cup_{x \in P} A(x, \mu x)$. Let us consider any element y in $\cup_{x \in P} A(x, \mu x)$. So, we get

$$\begin{aligned}
& y \in \cup_{x \in P} A(x, \mu x), \\
\implies & y \in N_{\mu x} \text{ for any } x \in P, \\
\implies & y \in N_{\mu x} \text{ for any } x \in P' \text{ as } P \subseteq P', \\
\implies & y \in \cup_{x \in P'} A(x, \mu x).
\end{aligned}$$

So, we get $\cup_{x \in P} A(x, \mu x) \subseteq \cup_{x \in P'} A(x, \mu x)$. Thus, we say that

$$P \subseteq P' \subseteq K \subseteq \cup_{x \in P} A(x, \mu x) \subseteq \cup_{x \in P'} A(x, \mu x).$$

Repeating the above technique, we get a sequence of compactly semidefinite representable sets such that

$$\begin{aligned}
& P_1 \subseteq P_2 \subseteq \dots \subseteq P_n \subseteq P_{n+1} \subseteq \dots \subseteq K \\
& \subseteq \cup_{x \in P_1} A(x, \mu x) \subseteq \cup_{x \in P_2} A(x, \mu x) \subseteq \dots \subseteq \cup_{x \in P_n} A(x, \mu x) \subseteq \cup_{x \in P_{n+1}} A(x, \mu x) \subseteq \dots
\end{aligned}$$

So, the sequence of compactly semidefinite representable sets $\{P_i\}_{i=1}^{\infty}$ gradually gives tighter approximation of closed convex set K . The proof is complete. \square

3.1 Convergence

This subsection deals with the case where we show that the sequence of compactly semidefinite representable sets $\{P_i\}_{i=1}^{\infty}$ converge to the convex set K . We define strong convergence.

Definition 1 (Strong convergence) Let us consider a sequence of unbounded closed sets $\{P_i\}_{i=1}^{\infty}$ in \mathbb{R}^n . The sequence $\{P_i\}_{i=1}^{\infty}$ strongly converges to an unbounded closed set K if $\text{rec}(P_i) = \text{rec}(K)$ for any i .

We show that the recession cone of the compactly semidefinite representable set P_i is equal to the recession cone of the convex set K for any i .

- (\implies) For the sequence of compactly semidefinite representable sets $\{P_i\}_{i=1}^{\infty}$, we know

$$P_1 \subseteq P_2 \subseteq \dots \subseteq P_n \subseteq P_{n+1} \subseteq \dots \subseteq K. \quad (11)$$

Thus, it is very trivial to say that $\text{rec}(P_i) \subseteq \text{rec}(K)$ for all i .

- (\impliedby) Now we prove $\text{rec}(P_i) \supseteq \text{rec}(K)$ for any i . By definition of recession cone, we say that

$$\text{rec}(K) = \{d \in \mathbb{R}^n : x + \alpha d \in K, \alpha \geq 0, \text{ for all } x \in K\}.$$

As $x, x + \alpha d \in K$ we get

$$\begin{aligned} x + \alpha d &\in K_{m_p r_{p+1}} \text{ and } x \in K_{m_q r_{q+1}} \text{ for some } p + 1, q + 1, \\ \implies x + \alpha d &\in B_{p+1} \text{ and } x \in B_{q+1} \text{ for some } p + 1 \text{ and } q + 1, \\ \implies x + \alpha d, x &\in P_i \text{ for some } i \in [1, 2, \dots, n, n + 1, \dots]. \end{aligned}$$

So, there exists $d \in \mathbb{R}^n$ such that $x + \alpha d \in P_i$ and $\alpha \geq 0, x \in K$. Thus, we say $d \in \text{rec}(P_i)$. So, we get $\text{rec}(K) \subseteq \text{rec}(P_i)$.

We combine the above cases and we conclude that $\text{rec}(P_i) = \text{rec}(K)$ for some $i \in [1, 2, \dots, n, n + 1, \dots]$.

The recession cone of compactly semidefinite representable set, P_i and the recession cone of convex set K are equal. So, we say that the sequence of compactly semidefinite representable sets strongly converges to the closed convex set K .

4 Approximation of Convex Set by Semidefinite Representable Set

Any closed convex set, say Q can be uniformly approximable by polyhedral set in \mathbb{R}^n [13, Theorem 6.7]. First we give the definition of uniform approximation.

Definition 2 (Uniform approximation by semidefinite representable set) Let us consider a closed convex set Q in \mathbb{R}^n . The set Q is uniformly approximable by semidefinite representable set means for each arbitrary positive number ϵ , there exists a semidefinite representable set P at Hausdorff distance ϵ such that P approximates Q .

We see that parabola cannot be approximated by polyhedral set. This example motivates us to generalize the result. Our conjecture is given below. We also give the idea of the proof.

Theorem 3 *If a closed convex subset Q of \mathbb{R}^n is a finite Hausdorff distance d from some semidefinite representable set P and $Q \subseteq X + \text{rec}(Q)$, where X is a compact convex set, then Q is uniformly approximable by means of semidefinite representable sets.*

(Outline of the proof). Without loss of generality, we consider $d < 1$. We divide the proof in two cases. It is given below.

- Case 1: Let us consider that the unit ball $U \subseteq \mathbb{R}^n$ is contained in P . For each $f \in \mathbb{R}^n$, let us define μf and νf such that

$$\begin{aligned} \mu f &= \sup_{p \in P} f^T p, \\ \nu f &= \sup_{q \in Q} f^T q. \end{aligned}$$

Now, we get

$$\begin{aligned}
 |\mu f| &= |\sup f^T p|, \\
 &\geq |f^T p|, \\
 &= \|f\| \|p\| |\cos \theta^0| (\theta^0 \text{ is the angle between } f \text{ and } p), \\
 &\geq \|f\| |\cos \theta^0|, \\
 &\geq \|f\| \text{ as } 0 \leq |\cos \theta^0| \leq 1.
 \end{aligned}
 \tag{12}$$

So when $\|f\| = 1$, we say $\mu f \geq \|f\|$. The Hausdorff distance between the sets P and Q is d and $P \subseteq \{p : f^T p \leq \mu f\}$, $Q \subseteq \{q : f^T q \leq \nu f\}$, where $\{p : f^T p \leq \mu f\}$, $\{q : f^T q \leq \nu f\}$ are the supporting half-spaces of P and Q , respectively. So, we have $|\nu f - \mu f| \leq d$. As $|\nu f - \mu f| \leq d$, we have either $(\nu f - \mu f) \leq d$ or $(\mu f - \nu f) \leq d$. Thus, we get

$$\begin{aligned}
 (\nu f - \mu f) &\leq d \leq d \mu f, \\
 \implies \nu f &\leq (1 + d) \mu f.
 \end{aligned}
 \tag{13}$$

Again we have

$$\begin{aligned}
 (\mu f - \nu f) &\leq d \leq d \nu f, \\
 \implies \nu f &\geq (1 - d) \mu f.
 \end{aligned}
 \tag{14}$$

We combine (13) and (14) and we get

$$\nu f \in [1 - d, 1 + d] \mu f \text{ whenever } \|f\| = 1.
 \tag{15}$$

Let us assume that F is a set defined as

$$F = \{f \in \mathbb{R}^n : \|f\| = 1 \text{ and } \mu f < \infty\}.
 \tag{16}$$

For each convex set K in \mathbb{R}^n with $0 \in K$, let us assume that

$$\beta K = \{f \in \mathbb{R}^n : [0, 1]f \subseteq K \text{ and }]1, \infty]f \subseteq \mathbb{R}^n \setminus K\}.
 \tag{17}$$

We need to prove that

$$\beta P^\circ = \left\{ \left(\frac{1}{\mu f} \right) f : f \in F \right\}.
 \tag{18}$$

The set βP° is defined as

$$\beta P^\circ = \{x \in \mathbb{R}^n : [0, 1]x \subseteq P^\circ \text{ and }]1, \infty]x \subseteq \mathbb{R}^n \setminus P^\circ\}.
 \tag{19}$$

We prove $\left\{ \left(\frac{1}{\mu f} \right) f : f \in F \right\} \subseteq \beta P^\circ$. Let us consider any point x in $\left\{ \left(\frac{1}{\mu f} \right) f : f \in F \right\}$ such that $x = \left(\frac{1}{\mu f} \right) f$, $\|f\| = 1$ and $\mu f < \infty$. We know $0, \left(\frac{1}{\mu f} \right) f \in P^\circ$. As P° is a convex set, we say $[0, 1] \left(\frac{1}{\mu f} \right) f \subseteq P^\circ$. We know P° is a compact convex set, so there exists $\alpha \in [1, \infty[$ such that $\alpha \left(\frac{1}{\mu f} \right) f \notin P^\circ$. This implies

$$\begin{aligned}
 & [1, \infty[\left(\frac{1}{\mu f} \right) f \not\subseteq P^\circ, \\
 & [1, \infty[\left(\frac{1}{\mu f} \right) f \subseteq \mathbb{R}^n \setminus P^\circ.
 \end{aligned}$$

So there exists $\left(\frac{1}{\mu f} \right) f \in \left(\frac{1}{\mu f} \right) F$ such that

$$\begin{aligned}
 & [0, 1] \left(\frac{1}{\mu f} \right) f \subseteq P^\circ \text{ and }]1, \infty[\left(\frac{1}{\mu f} \right) f \subseteq \mathbb{R}^n \setminus P^\circ, \\
 & \implies \left(\frac{1}{\mu f} \right) f \in \beta P^\circ, \\
 & \implies \left(\frac{1}{\mu f} \right) F \subseteq \beta P^\circ.
 \end{aligned}$$

We prove $P^\circ = [0, 1] \beta P^\circ = [0, 1] \left(\frac{1}{\mu f} \right) F$. We consider the set $\alpha \left(\frac{1}{\mu f} \right) F$ for all $\alpha \in [0, 1]$. If $\alpha = 0$, $0 \in P^\circ$ and if $\alpha = 1$, $\left(\frac{1}{\mu f} \right) f \in P^\circ$. As P° is convex, we get that $[0, 1] \left(\frac{1}{\mu f} \right) F \subseteq P^\circ$. It implies $[0, 1] \beta P^\circ \subseteq P^\circ$. As $\beta P^\circ = \text{bd } P^\circ$, we get $P^\circ = [\lambda, 1 - \lambda] \beta P^\circ$ for all $\lambda \in [0, 1]$. Thus, we get $P^\circ \subseteq [0, 1] \beta P^\circ$ as P° is a compact convex set. So, we get $P^\circ = [0, 1] \beta P^\circ$. Similarly, we get $\beta Q^\circ = \left\{ \left(\frac{1}{\nu f} \right) f : f \in F \right\}$ and $Q^\circ = [0, 1] \beta Q^\circ$. Since, P° is a compact semidefinite representable set, βP° must be compact set. So, using the result in Eq. (15), we say that βQ° is also compact. We know $\beta P^\circ = \left(\frac{1}{\mu f} \right) F$ and $\beta Q^\circ = \left(\frac{1}{\nu f} \right) F$. Thus, F is a compact set. Hence, we get $\sup \nu F = s$ and s is finite. Now we prove $[0, \infty[\beta Q^\circ = [0, \infty[P^\circ$.

$$\begin{aligned}
 & x \in [0, \infty[\beta Q^\circ, \tag{20} \\
 & \implies x = 0 + t.x; t > 0 \text{ and } x \in \beta Q^\circ, \\
 & \implies x = 0 + t.x; t > 0 \text{ and } x = \left(\frac{1}{\nu f} \right) F, \\
 & \implies x = 0 + t.x; t > 0 \text{ and } x \in \left[\frac{1}{1+d}, \frac{1}{1-d} \right] \left(\frac{1}{\mu f} \right) F, \\
 & \implies x = 0 + t.x; t > 0 \text{ and } x \in \left[\frac{1}{1+d}, \frac{1}{1-d} \right] \beta P^\circ,
 \end{aligned}$$

$$\begin{aligned}
&\implies x = 0 + t.x; t > 0 \text{ and } x \in \left[\frac{1}{1+d}, \frac{1}{1-d} \right] P^\circ, \\
&\implies x = 0 + t.\alpha.x; t.\alpha > 0, y \in P^\circ, \\
&\implies x \in [0, \infty[P^\circ, \\
&\implies [0, \infty[\beta Q^\circ \subseteq [0, \infty[P^\circ.
\end{aligned}$$

Now we prove that $[0, \infty[\beta Q^\circ \supseteq [0, \infty[P^\circ$.

$$\begin{aligned}
&x \in [0, \infty[P^\circ, \tag{21} \\
&\implies x = 0 + t.y; t > 0 \text{ and } y \in P^\circ, \\
&\implies x = 0 + t.y; t > 0 \text{ and } y \in [0, 1]\beta P^\circ, \\
&\implies x = 0 + t.y; t > 0 \text{ and } y = 0 + u.w; w \in \beta P^\circ, u \in [0, 1], \\
&\implies x = 0 + t.y; t > 0 \text{ and } y = 0 + u.w; w \in \left(\frac{1}{\mu f} \right) F, u \in [0, 1], \\
&\implies x = 0 + t.y; t > 0 \text{ and } y = 0 + u.w; w \in [1-d, 1+d] \left(\frac{1}{\nu f} \right) F, u \in [0, 1], \\
&\implies x = 0 + t.y; t > 0 \text{ and } y = 0 + u.w; w \in [1-d, 1+d]\beta Q^\circ, u \in [0, 1], \\
&\implies x = 0 + t.y; t > 0 \text{ and } y = 0 + u.\gamma.z; z \in \beta Q^\circ, \gamma \in [1-d, 1+d], u \in [0, 1], \\
&\implies x = 0 + t.y; t > 0 \text{ and } y = 0 + u.\gamma.z; z \in \beta Q^\circ, u.\gamma \in [0, 1], \\
&\implies x = 0 + t.y; t > 0 \text{ and } y = 0 + v.z; z \in \beta Q^\circ, v \in [0, 1], \\
&\implies x = 0 + t.(0 + v.z); z \in \beta Q^\circ, \\
&\implies x = 0 + 0 + t.v.z; t.v > 0, z \in \beta Q^\circ, \\
&\implies x \in [0, \infty[\beta Q^\circ, \\
&\implies [0, \infty[P^\circ \subseteq [0, \infty[\beta Q^\circ.
\end{aligned}$$

We combine Eqs. (20) and (21) and we get

$$[0, \infty[P^\circ = [0, \infty[\beta Q^\circ. \tag{22}$$

Let us consider the set Y such that $Y = \beta Q^\circ \cap \text{rext}[0, \infty[P^\circ$.

Let ϵ be an arbitrary positive number. Since βQ° is compact, we get a compact semidefinite representable set Z such that $Y \subseteq \text{cl } Y \subseteq Z \subseteq \beta Q^\circ$. We get

$$\begin{aligned}
&Y \cup \{0\} \subseteq \text{cl } Y \cup \{0\} \subseteq Z \cup \{0\} \subseteq \beta Q^\circ \cup \{0\}, \\
&\implies \text{conv}(Y \cup \{0\}) \subseteq \text{conv}(\text{cl } Y \cup \{0\}) \subseteq \text{conv}(Z \cup \{0\}) \subseteq \text{conv}(\beta Q^\circ \cup \{0\}), \\
&\implies \text{conv}(Y \cup \{0\}) \subseteq \text{conv}(\text{cl } Y \cup \{0\}) \subseteq M \subseteq \text{conv}(\beta Q^\circ \cup \{0\}).
\end{aligned}$$

The set M is a compact semidefinite representable set such that $\beta M \subseteq [1 - \frac{\epsilon}{s}, 1] \beta Q^\circ$. Let us consider that $P_\epsilon = M$ and $\tau f = \sup f^T P_\epsilon$ for any point $f \in \mathbb{R}^n$. We get

$$\begin{aligned} \beta M &\subseteq \left[1 - \frac{\epsilon}{s}, 1\right] \beta Q^\circ, \\ \implies \text{conv}(\beta M) &\subseteq \left[1 - \frac{\epsilon}{s}, 1\right] \text{conv}(\beta Q^\circ), \\ \implies M &\subseteq \left[1 - \frac{\epsilon}{s}, 1\right] Q^\circ, \\ \implies Q &\subseteq \left[1 - \frac{\epsilon}{s}, 1\right] M^\circ, \\ \implies Q &\subseteq \left[1 - \frac{\epsilon}{s}, 1\right] P_\epsilon, \\ \implies Q &\subseteq \left[1 - \frac{\epsilon}{s}, 1 + \frac{\epsilon}{s}\right] P_\epsilon, \\ \implies f^T Q &\in \left[1 - \frac{\epsilon}{s}, 1 + \frac{\epsilon}{s}\right] f^T P_\epsilon, \\ \implies \sup f^T Q &\in \left[1 - \frac{\epsilon}{s}, 1 + \frac{\epsilon}{s}\right] \sup f^T P_\epsilon, \\ \implies \nu f &\in \left[1 - \frac{\epsilon}{s}, 1 + \frac{\epsilon}{s}\right] \tau f, \\ \implies |\nu f - \tau f| &\leq \frac{\epsilon}{s}, \\ \implies |\nu f - \tau f| &\leq \epsilon, \\ \implies h(P_\epsilon, Q) &\leq \epsilon. \end{aligned}$$

Thus, we get the semidefinite representable set P_ϵ which gives tighter approximation of the closed convex set Q .

- Case 2: We consider the case where the unit cell is not contained in the set P . Let us consider a closed convex set Q and a semidefinite representable set P such that $h(P, Q) = d < 1$. Let X is compact semidefinite representable set in \mathbb{R}^n such that $P + X \supset U$. Then $h(P + X, Q + X) = h(P, Q) = d$. Without loss of generality, we consider that $d < 1$. Using the proof in Case 1, we say that there exists a semidefinite representable set Y such that $h(Y, Q + X) < \epsilon$. Let us consider a set Z such that $Z = \{y \in Y : y + X \subseteq Y\}$. So, Z is a closed convex set and $Z + X = Y$. It implies that $Z + X = P + X$ and we get $Z = P$. Thus, Z is semidefinite representable. So, it follows that $d_H(Z, Q) < \epsilon$. The proof is complete.

Remark 1 We mention that $\beta P^\circ \subseteq \left(\frac{1}{\mu f}\right) F$, which we could not prove. This proof will be included in our future research topic.

5 Future Research Prospect

We proved Theorem 1, Corollary 1, Theorem 2 on approximation of any closed convex set by compactly semidefinite representable set. Further, we proved that the sequence of compactly semidefinite representable sets strongly converge to the convex set K in Sect. 3.1.

Any closed convex set can be approximated by polyhedral set [13, Theorem 6.7]. In this context, we say that the parabola in \mathbb{R}^2 cannot be approximated by polyhedral set. So, it will be challenging to extend the Theorem 6.7 from [13]. The extension of this Theorem gives a technique to approximate any closed convex set by semidefinite representable set as semidefinite representable set generalizes polyhedron.

The approximation of any closed convex set by compactly semidefinite representable set gives very tight approximation. But, there exist few sets such as circular cones which cannot be approximated by compactly semidefinite representable set, as we discussed in Sect. 2. So, we develop another approximation technique which gives an approximation of the convex set which cannot be approximated by compactly semidefinite representable set. This approximation technique provides a uniform approximation of closed convex set under some condition. The generalized version of this approximation technique gives another research direction in this approximation theory.

References

1. Barvinok, A.: Approximating a norm by a polynomial. *Geometric Aspects of Functional Analysis*, Lecture Notes in Mathematics, vol. 1807, pp. 20–26 (2003)
2. Lindenstrauss, J., Figiel, T., Milman, V.: The dimension of almost spherical sections of convex bodies. *Acta Math.* **129**, 53–94 (1977)
3. Lindenstrauss, J., Bourgain, J., Milman, V.: Approximation of zonoids by zonotopes. *Acta Math.* **162**, 73–141 (1989)
4. Voemett, E.R.: *The Computational Complexity of Convex Bodies*. Ph.D. thesis, Ann Arbor, MI, USA (2007). AAI3276318
5. Alon, N., Naor, A.: Approximating the cut-norm via grothendieck’s inequality. *SIAM J. Comput.* **35**(4), 787–803 (2006)
6. Ben-Tal, A., Nemirovski, A.: On polyhedral approximations of the second-order cone. *Math. Oper. Res.* **26**(2), 193–205 (2001b)
7. Vinel, A., Krokmal, P.A.: On polyhedral approximations in p-order cone programming. *Optim. Methods Softw.* **29**, 1210–1237 (2014)
8. Ball, K.: *An elementary introduction to modern convex geometry*. Mathematical Sciences Research Institute Publications, Cambridge (1997)
9. Pisier, G.: *The Volume of Convex Bodies and Banach Space Geometry*, vol. 94. Cambridge University Press, Cambridge (1999)
10. Deza, M., Laurent, M.: *Geometry of Cuts and Metrics*, volume 15 of *IS*. Springer, Berlin, Heidelberg (1997)
11. Laraki, R., Lasserre, J.B.: Computing uniform convex approximations for convex envelopes and convex hulls. *J. Convex Anal.* **15**, 635–654 (2008)
12. Veomett, E.: A positive semidefinite approximation of the symmetric traveling salesman polytope. *Discrete Comput. Geom.* **38**, 15–28 (2007)

13. Klee, V.: Some characterizations of convex polyhedra. *Acta Math* **102**, 79–107 (1959). ISSN 0001-5962. <http://dx.doi.org/10.1007/BF02559569>
14. Ben-Tal, A., Nemirovski, A.: Lectures on modern convex optimization: analysis, algorithms and engineering applications. MOS-SIAM Series on Optimization. SIAM, Philadelphia, PA, USA (2001a). ISBN 978-0-89871-491-3
15. Nemirovski, A.: Advances in Convex Optimization: Conic Programming, vol. I, pp. 413–444. European Mathematical Society, Zürich (2007)
16. Nesterov, Y., Nemirovskii, A.: Interior Point Polynomial Algorithms in Convex Programming, 13. SIAM (1994)
17. Tuncel, L.: Polyhedral and Semidefinite Programming Methods in Combinatorial Optimization. Institute Monographs, American Mathematical Society, Fields (2010)