IJNAO

Iranian Journal of Numerical Analysis and Optimization

Volume 7 , Number 1

Winter 2017

Ferdowsi University of Mashhad, Iran

In the Name of God

Iranian Journal of Numerical Analysis and Optimization (IJNAO)

This journal is authorized under the registration No. 174/853 dated 1386/2/26, by the Ministry of Culture and Islamic Guidance.

Volume 7, Number 1, Winter 2017

ISSN: 2423-6977

Publisher: Faculty of Mathematical Sciences, Ferdowsi University of Mashhad

Published by: Ferdowsi University of Mashhad Press

Circulation: 100

Address: Iranian Journal of Numerical Analysis and Optimization

Faculty of Mathematical Sciences, Ferdowsi University of Mashhad

P.O. Box 1159, Mashhad 91775, Iran.

Tel. : +98-51-38806222 , **Fax:** +98-51-38807358

E-mail: mjms@um.ac.ir

Website: http://ijnao.um.ac.ir

This journal is indexed by:

• Zentralblatt

• ISC

• SID

به اطلاع کلیه محققان، پژوهشگران، اساتید ارجمند، دانشجویان تحصیلات تکمیلی و نویسندگان محترم می رساند که نشریه ایرانی آنالیز عددی و بهینه سازی - IJNAO - طبق مجوز شماره ۳/۱۸/۵۴۸۹۱۳/ مورخه ۱۳۹۲/۱۰/۲۵ مدیر کل محترم سیاست گذاری و برنامه ریزی امور پژوهشی وزارت علوم، تحقیقات و فناوری، علمی – پژوهشی میباشد.
بر اساس نامه شماره ۹۲/۱۵۳۶/پ مورخه ۱۳۹۲/۱۱/۲۱ سرپرست محترم معاونت پژوهشی وفناوری پایگاه استنادی علوم جهان اسلام، نشریه ایرانی آنالیز عددی و بهینه سازی در پایگاه ISC نیز نمایه می شود.

Iranian Journal of Numerical Analysis and Optimization

Volume 7, Number 1, Winter 2017

Ferdowsi University of Mashhad - Iran

Iranian Journal of Numerical Analysis and Optimization

This journal is published under the auspices of Ferdowsi University of Mashhad

---

* Full Professor

** Associate Professor

We would like to acknowledge the help of Narjes khatoon Zohorian in the preparation of this issue. We would also like to thanks Miss Shakiba Gachpazan for designing the Journal's logo.

Letter from the Editor in Chief

I would like to welcome you to the Iranian Journal of Numerical Analysis and Optimization (IJNAO). This journal is published biannually and supported by the Faculty of Mathematical Sciences at the Ferdowsi University of Mashhad. Faculty of Mathematical Sciences with three centers of excellence and three research centers is well-known in mathematical communities in Iran.

The main aim of the journal is to facilitate discussions and collaborations between specialists in applied mathematics, especially in the fields of numerical analysis and optimization, in the region and worldwide.

Our vision is that scholars from different applied mathematical research disciplines, pool their insight, knowledge and efforts by communicating via this international journal.

In order to assure high quality of the journal, each article is reviewed by subject-qualified referees.

Our expectations for IJNAO are as high as any well-known applied mathematical journal in the world. We trust that by publishing quality research and creative work, the possibility of more collaborations between researchers would be provided. We invite all applied mathematicians especially in the fields of numerical analysis and optimization to join us by submitting their original work to the Iranian Journal of Numerical Analysis and Optimization.

Mohammad Hadi Farahi

Contents

Iranian Journal of Numerical Analysis and Optimization
Vol. 7, No. 1, (2017), pp 1-12
DOI: 10.22067/ijnao.v7i1.44788

Computing the eigenvalues of fourth order Sturm-Liouville problems with Lie Group method

H. Mirzaei*

Abstract

In this paper, we formulate the fourth order Sturm-Liouville problem (FSLP) as a Lie group matrix differential equation. By solving this matrix differential equation by Lie group Magnus expansion, we compute the eigenvalues of the FSLP. The Magnus expansion is an infinite series of multiple integrals of Lie brackets. The approximation is, in fact, the truncation of Magnus expansion and a Gaussian quadrature are used to evaluate the integrals. Finally, some numerical examples are given.

Keywords: Lie group method; Fourth order Sturm-Liouville problem; Magnus expansion.

1 Introduction

A classical fourth order Sturm-Liouville equation, is a real fourth order linear differential equation of the form

$$(p(x)y''(x))'' - (q_1(x)y'(x))' + q_2(x)y(x) = \lambda w(x)y(x), \tag{1}$$

where $p(x), q_1(x), q_2(x)$ and $w(x)$ are given continuous functions on the finite interval $[a, b]$. Usually the equation (1) isaccompanied with four boundary conditions of the form

$$A_1u(a) + A_2v(a) = 0, \quad B_1u(b) + B_2v(b) = 0, \tag{2}$$

where $u = [y, y']$, $v = [py'', (py'')' + q_1y']$, A_1, A_2, B_1 and B_2 are real matrices of order 2, such that

*Corresponding author
Received 3 March 2015; revised 9 March 2016; accepted 28 May 2016
H. Mirzaei
Faculty of Basic Sciences, Sahand University of Technology, Tabriz, Iran.
e-mail: h_ mirzaei@sut.ac.ir

$$A_1A_2^T = A_2A_1^T, \quad B_1B_2^T = B_2B_1^T,$$

and the matrices $(A_1 : A_2)$ and $(B_1 : B_2)$ have rank 2, see [8,9]. The equation (1) with boundary conditions (2) is called fourth-order Sturm-Liouville problem (FSLP). If λ is such that the FSLP has a nontrivial solution, then λ is called an eigenvalue and nontrivial solution for corresponding to λ is called an eigenfunction. The set of all eigenvalues of FSLP called the spectrum. We assume that the interval (a, b) is finite and coefficient functions $r = \frac{1}{p}, w, q_1, q_2$ are real and belong to $L^1(a, b)$. The classical results of self-adjoint Sturm-Liouville problem states that under this assumptions the eigenvalues are bounded below and can be ordered as

$$\lambda_0 < \lambda_1 < \lambda_2 < \cdots, \tag{3}$$

where $\lim_{k\to\infty} \lambda_k = \infty$, see [8–10].

FSLP arise in mathematical modeling of motion for the transversely vibrating Beam. In fact, the equation (1) is the standard form of Euler-Bernoulli Beam equation, see [7]. Moan [15] approximates the eigenvalues of second order Sturm-Liouville problems using Lie group methods. Ledoux and et al. [17] introduced a modified Magnus method for numerical solution of one dimensional Schrodinger eigenvalue problem. In [1] the Homotopy analysis method (HAM) is applied to numerically approximate the eigenvalues of the second and fourth order Sturm-Liouville problem. For further schemes to study FSLP, we refer to [2,5,10].

Let $u_1 = y,\ u_2 = y',\ u_3 = py'',\ u_4 = (py'')' - q_1y'$. Then we have the matrix representation of equation (1) and boundary conditions (2):

$$\begin{cases} U' = G(x)U \\ AU(a) + BU(b) = 0, \end{cases} \tag{4}$$

where

$$G(x) = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & r(x) & 0 \\ 0 & q_1(x) & 0 & 1 \\ \lambda w(x) - q_2(x) & 0 & 0 & 0 \end{pmatrix} \tag{5}$$

$$U = [u_1, u_2, u_3, u_4],\ A = \begin{pmatrix} A_1 & A_2 \\ \mathbf{0}_2 & \mathbf{0}_2 \end{pmatrix},\ B = \begin{pmatrix} \mathbf{0}_2 & \mathbf{0}_2 \\ B_1 & B_2 \end{pmatrix}.$$

The paper is organized as follows. In Section 2, we introduce Lie group and Magnus expansion. In Section 3, we present the numerical procedure of Magnus expansion for solving matrix differential equations. In Section 4, we apply the Magnus expansion for computing eigenvalues of FSLP. In Section 5, some numerical examples are given.

2 Magnus expansion

The concept of Lie group was first proposed by Sophus Lie in 1893 to study transformation groups [13]. A Lie group $(G, .)$ is a differentiable manifold that also has the algebraic structure of a group. For a given Lie group G, the tangent space at identity is called the Lie algebra of G and denoted by $\mathbf{g}$. A binary operation $[., .] : \mathbf{g} \times \mathbf{g} \to \mathbf{g}$ is called the Lie bracket. Let $x, y \in \mathbf{g}$, then the Lie bracket $[x, y]$ is defined as

$$[x, y] = xy - yx. \tag{6}$$

The operator (6) is bilinear, anti symmetric and satisfy the Jacobi identity,

$$[[x, y], z] + [[y, z], x] + [[z, x], y] = 0, \tag{7}$$

for all $x, y, z \in \mathbf{g}$. The Lie algebra $\mathbf{g}$ is a vector space under Lie bracket (6). An important special case of Lie group is the Special Lie group $(SL(F, n))$. Let $F^{n\times n}$ be the set of all $n \times n$ matrices with entries in F. The set

$$SL(F, n) = \{y \in F^{n\times n}, \det y = 1\}. \tag{8}$$

The usual matrix multiplication has the structure of a Lie group and is called the Special Linear group. The corresponding Lie algebra $sl(F, n)$ of this group is the set of all $n \times n$ matrices with zero trace. The Lie bracket in this case is $[A, B] = AB - BA$, for $A, B \in sl(F, n)$. For more details see [4, 11, 16, 22].

The Lie group is the differentiable manifold, therefor we can study ordinary differential equations on a Lie group. A system of linear ordinary differential equations on $SL(F, n)$ have the form [11, 22]

$$y' = G(x)y, \quad tr(G(x)) = 0 \longleftrightarrow \det y = 1. \tag{9}$$

Iserles and Nörset [11] studied the solution of the linear matrix differential equation by the Lie group. In [22] the author introduce the collocation and Runge kutta type methods for the Lie group and present algebraic formulae for coefficients of these methods. Ros et al. [3], studied the Magnus expansion and some of its application in linear system of differential equations. The Magnus expansion represents the solution of (9) in the form

$$y(x) = e^{\sigma(x)} y_0 \tag{10}$$

where y_0 is the initial data of the problem (9) and

$$\begin{aligned}\sigma(x) = &\int_a^x G(s)ds + \frac{1}{2}\int_a^x \int_a^s [G(s), G(u)]duds \\ &+ \frac{1}{4}\int_a^x \int_a^s \int_a^u [G(s), [G(u), G(v)]]dvduds\end{aligned}$$

$$+\frac{1}{12}\int_a^x\int_a^s\int_a^s[[G(s),G(u)],G(v)]dvduds+... \tag{11}$$

In practice, we have to consider only a finite number of terms in expansion. The following theorem denotes the order of this approximation:

Theorem 1. *(Adapted from Theorem 6 in [11]) Let* $\|G(h)\|=O(h^k)$*, then*

$$\|\sigma(h)-\sum_{i=1}^{p}\sigma_i(h)\|=O(h^{(p+1)(k+1)+1}), \tag{12}$$

where

$$\begin{aligned}
\sigma_1 &= \int_0^h G(s)ds,\\
\sigma_2 &= \frac{1}{2}\int_0^h\int_0^s[G(s),G(u)]duds,\\
\sigma_3 &= \frac{1}{4}\int_0^h\int_0^s\int_0^u[G(s),[G(u),G(v)]]dvduds\\
&\quad+\frac{1}{12}\int_0^h\int_0^s\int_0^s[[G(s),G(u)],G(v)]dvduds,
\end{aligned}$$

Moreover, if $\overline{\sigma}$ *is an approximation of* σ *such that* $\|\sigma-\overline{\sigma}\|\leq O(h^q)$ *then*

$$\|e^{\sigma(h)}-e^{\overline{\sigma}(h)}\|\leq O(h^q).$$

In this paper we suppose that $p=3$, and approximate $\sigma(h)$ as follows

$$\begin{aligned}
\sigma(h)\simeq\overline{\sigma}(h) &= \int_0^h G(s)ds+\frac{1}{2}\int_0^h\int_0^s[G(s),G(u)]duds\\
&\quad+\frac{1}{4}\int_0^h\int_0^s\int_0^u[G(s),[G(u),G(v)]]dvduds\\
&\quad+\frac{1}{12}\int_0^h\int_0^s\int_0^s[[G(s),G(u)],G(v)]dvduds.
\end{aligned} \tag{13}$$

We explain the error of Magnus expansion for FSLP by the following Remarks.

Remark 1. By Theorem 1, if $\|\ G(h)\ \|=O(1)$ then the numerical Magnus series scheme produces an (at least) local order $O(h^5)$ approximation of σ. A Magnus series integrator with local order p will have global order $p-1$ [12], thus Theorem 1 guarantees (at least) error of order $O(h^4)$ (if $\|\ G(h)\ \|=O(1)$).

Remark 2. In FSLP the coefficient matrix G is also a function of eigenvalue

λ and for large eigenvalues, the norm of matrix G is : $\|G\| = O(\lambda)$. Thus for large eigenvalues, the error is of order $O(\lambda h^4)$.

3 Numerical procedure

The first step for computing the solution $y(x) = e^{\sigma(x)}y_0$ by Magnus series is the truncation of the series (11). As mentioned before we use only four terms (13). Next, we apply a numerical integration method for approximating the integrals (13). Here we use the two point Gaussian integration method. Thus we have

$$\begin{aligned}
&\textstyle\int_0^h G(x)dx \simeq \frac{h}{2}(G(c_1h) + G(c_2h)),\\
&\textstyle\int_0^h \int_0^k [G(k), G(\xi)]d\xi dk \simeq -\frac{\sqrt{3}}{6}h^2[G(c_1h), G(c_2h)],\\
&\textstyle\frac{1}{4}\int_0^h \int_0^s \int_0^u [G(s), [G(u), G(v)]]dvduds \qquad (14)\\
&\textstyle+\frac{1}{12}\int_0^h \int_0^s \int_0^s [[G(s), G(u)], G(v)]dvduds\\
&\simeq \frac{h^3}{80}[[G(c_1h), G(c_2h)], G(c_1h) - G(c_2h)],
\end{aligned}$$

Substituting (14) in (13) we obtain

$$\begin{aligned}
\overline{\sigma}(h) = &\frac{h}{2}(G(c_1h) + G(c_2h)) - \frac{\sqrt{3}}{12}h^2[G(c_1h), G(c_2h)]\\
&+ \frac{h^3}{80}[[G(c_1h), G(c_2h)], G(c_1h) - G(c_2h)] \qquad (15)
\end{aligned}$$

where $c_1 = \frac{1}{2} - \frac{\sqrt{3}}{6}, c_2 = \frac{1}{2} + \frac{\sqrt{3}}{6}$ are the nods of Gaussian quadrature, see [11]. Finally by Matlab software we compute $y(h) \simeq e^{\overline{\sigma}(h)}y_0$. For computing $y(x)$ in interval $[a, b]$ we divide the interval $[a, b]$ into n subintervals with length $h = \frac{b-a}{n}$ and use the following iterative process

$$Y(x_i) = e^{\sigma(x_i)}Y(x_{i-1}), \quad Y(a) = Y_0, \quad x_i = a + ih. \qquad (16)$$

4 Eigenvalues of fourth order Sturm-Liouville problem

In this section, we formulate the fourth order Sturm-Liouville problem as the Lie group matrix differential equation and find the eigenvalues. For solving the system (4) and finding the eigenvalues of FSLP, we consider the matrix differential equation

$$Y' = G(x)Y, \quad Y(a) = I_4. \tag{17}$$

Theorem 2 [14]. *If $\Phi(x)$ is a solution of the matrix differential equation $X' = G(x)X$ on an interval J and if τ is any point of J, then*

$$\forall \tau, x \in J, \det \Phi(x) = \det \Phi(\tau) exp[\int_{\tau}^{x} tr(G(s))ds]. \tag{18}$$

From (5) it is clear that $tr(G(x)) = 0$, and by Theorem 2 for any solution $Y(x)$ of system (17) we have

$$\det Y(x) = \det Y(a) exp[\int_{a}^{x} tr(G(s))ds] = 1.$$

Thus system (17) is a system of matrix differential equations on the Lie Group $SL(\mathbb{R}^4, 4)$. Therefore we can solve the system (17) by Magnus expansion. Let $Y(x)$ be a solution of (17) for $x \geq a$, then the solution of the system (4) with initial condition $U(a)$ is

$$U(x) = Y(x)U(a). \tag{19}$$

Thus, $U(b) = Y(b)U(a)$. Following $U(b)$ in the boundary condition (4) we obtain

$$AU(a) + BY(b)U(a) = 0 \Longrightarrow [A + BY(b)]U(a) = 0.$$

The initial condition $U(a)$ is nonzero if $U(a) = 0$, then we have the trivial solution $U = 0$, thus determinant of the coefficient matrix $A + BY(b)$ must be zero. This determinant is a function of λ and the eigenvalues of the FSLP are roots of equation:

$$F(\lambda) := \det(A + BY(b)) = 0. \tag{20}$$

Equation (20) is the main equation of this paper. In fact, this equation is the characteristic equation of the fourth order Sturm-Liouville problem. For solving system (17) and computing $Y(b)$ by Magnus expansion, we need the brackets in (15). For FSLP these brackets are as follows

$$[G(s), G(u)] =$$

$$\begin{pmatrix} 0 & 0 & r(u)-r(s) & 0 \\ 0 & r(s)q_1(u)-r(u)q_1(s) & 0 & r(s)-r(u) \\ 0 & 0 & r(u)q_1(s)-r(s)q_1(u) & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$$+ \{\lambda(w(u) - w(s)) + (q_2(s) - q_2(u))\} \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{pmatrix},$$

$$[[G(s),G(u)],G(s)-G(u)]=$$

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 2(r(s)-r(u))(r(s)q_1(u)-r(u)q_1(s)) & 0 \\ 0 & 2(q_1(s)-q_1(u))(r(u)q_1(s)-r(s)q_1(u)) & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$$+(r(u)-r(s))(q_1(s)-q_1(u))\begin{pmatrix} 0&1&0&0\\0&0&0&0\\0&0&0&1\\0&0&0&0 \end{pmatrix}$$

$$+2(r(s)-r(u))\{\lambda(w(s)-w(u))+(q_2(u)-q_2(s))\}\begin{pmatrix} 0&0&0&0\\1&0&0&0\\0&0&0&0\\0&0&1&0 \end{pmatrix}.$$

For computing $Y(x_i)$ and finally $Y(b)$ for finding the characteristic equation (20), we should compute $e^{\sigma(x_i)}$ which $\sigma(x_i)$ is a matrix function of unknown parameter λ. In general, we can not obtain $e^{\sigma(x_i)}$ explicitly. But for any given real number λ, we can compute it by Matlab or Maple software. For solving this problem, we limit the parameter λ into interval $[\lambda_0,\lambda^*]$ and find the eigenvalues of the FSLP in this interval. For this purpose, we divide interval $[\lambda_0,\lambda^*]$ into m subintervals $\lambda_0<\lambda_1<...<\lambda_m=\lambda^*$ with length $h'=\frac{\lambda^*-\lambda_0}{m}$ and consider the following algorithm:

1. $i=0$,
2. Compute $Y(b)$ for $\lambda=\lambda_i$ with iterative process (8),
3. Compute $D_i=det(A+BY(b))$, if $D_i=0$, then λ_i is an eigenvalue, if $D_iD_{i+1}>0$ go to step 4. If $D_iD_{i+1}<0$, we have eigenvalue in interval $[\lambda_i,\lambda_{i+1}]$. By applying Bisection method for function $F(\lambda)=det(A+BY(b))$, we compute the eigenvalues in $[\lambda_i,\lambda_{i+1}]$.
4. $i=i+1$, if $i\leq m$ go to step 2, if $i>m$ go to step 5,
5. End.

Remark 3. We can not obtain the explicit form of characteristic function $F(\lambda)$, but we can compute the values of $F(\lambda)$ for any given real number λ_i. On the other hand by (3) the roots of $F(\lambda)$ is simple hence the convergence of Bisection method is guaranteed. Thus in step 3 we use the Bisection method. Also, we can apply other numerical methods for computing the roots of $F(\lambda)$.

Example 1. Consider the fourth-order Sturm-Liouville problem

$$y^{(4)}-(0.02x^2y')'+(0.0001x^4-.02)y(x)=\lambda y(x),\quad x\in[0,5], \qquad (21)$$

with the boundary conditions

$$y(0) = 0 = y''(0), \quad y(5) = 0 = y''(5). \tag{22}$$

It is easy to show that the eigenvalues of this problem are the square of the eigenvalues of second order Sturm-Liouville problem

$$\begin{aligned} &-u''(x) + 0.01x^2u(x) = \mu u(x), \quad x \in [0, 5], \\ &u(0) = 0 = u(5). \end{aligned} \tag{23}$$

The eigenvalues of problem (23) can be computed with Matslise [18]. We solve the problem (21) and (22) using Magnus expansion with $n = 500$ ($h = 0.01$) and approximate the eigenvalues in interval $[0, 650]$. In Table 1 we listed the first 8 eigenvalues and compared them with other methods.

In this example, for lower eigenvalues $\|G\| = O(1)$, thus by Theorem 1 and Remark 1, we have error of order (at least) $O(h^4)$. For large eigenvalues $\|G\| = O(\lambda)$ such that we must have error of order (at least) $O(\lambda h^4)$. The absolute errors in Table 1 confirm this order of error. If we compare the results with reference solution [18], it is obvious that our results is better than results of $[1, 2, 5]$. The Homotopy method [1], for eigenvalue λ_5 has large error. Also for eigenvalues λ_6 and λ_7 the errors on [2] are larger.

Table 1: Eigenvalues and absolute errors $\triangle\lambda_k$ of Example 1

λ_k Matslise [18]	$\triangle\lambda_k$ Our method	$\triangle\lambda_k$ [1]	$\triangle\lambda_k$ [2]	$\triangle\lambda_k$ [5]
0.2150508644	$2.9E-11$	$4.4E-9$	$7.0E-11$	$2.8E-13$
2.7548099347	$1.8E-10$	$1.1E-6$	$8.3E-11$	$3.0E-12$
13.2153515406	$2.1E-11$	$4.4E-5$	$5.8E-11$	$4.2E-11$
40.9508197592	$4.7E-10$	$2.6E-3$	$6.1E-11$	$4.1E-7$
99.0534780635	$5.2E-9$	$3.5E-1$	$7.5E-8$	
204.355732268	$2.4E-7$	$\underline{11.3}$	$1.2E-3$	
377.430420689	$7.6E-6$		$\underline{12.73}$	
642.590868170	$3.4E-5$		$\underline{215.5}$	

Example 2. We consider problem (21) in Example 1 with the different boundary conditions

$$y(0) = 0 = y'(0), \quad y(5) = 0 = y'(5).$$

For the Magnus expansion result with $n = 500$, see Table 2. For this problem, we don't have the exact eigenvalues, but since the equation of this example is the same equation of the Example 1, we expect the same error order of Example 1. Also there is an agreement between the results of our method and the other methods in Table 2.

Eexample 3. Consider the following Clamped-Clamped Euler-Bernoulli beam:

Table 2: Eigenvalues of Example 2

λ_k Our method	λ_k [5]	λ_k [2]	λ_k [19]	λ_k [20]
0.86690250	0.86690250	0.86690250	0.86690250	0.86690250
6.35768645	6.35768645	6.35768645	6.35768645	6.35768645
23.99274685	23.99274698	23.99274685	23.99274685	23.99274685
64.97866759	64.97863592	64.97866760	64.97866759	64.97866761
144.28062714		144.28062804	144.28062684	144.28062693
280.60096225		280.58602048	280.60096700	280.60096374
496.38280690				

$$\begin{array}{l}((1+x)^{\frac{3}{2}}u''(x))'' = \lambda(1+x)^{\frac{-1}{2}}u(x), \quad 0 \le x \le 1,\\ u(0) = 0 = u'(0), \quad u(1) = 0 = u'(1).\end{array} \tag{24}$$

Problem (24) has the same spectrum as the following uniform Euler-Bernoulli beam problem:

$$\begin{array}{l}v^{(4)}(s) = \lambda v(s), \quad 0 \le s \le L, \quad L = 2(\sqrt{2}-1),\\ v(0) = 0 = v'(0), \quad v(L) = 0 = v'(L).\end{array} \tag{25}$$

We called that the problems (24) and (25) are isospectral. For isospectral beam equation see chapter 12 in [7] and papers [6, 21]. The eigenvalues of uniform beam (25) are the roots of the equation

$$cos(\beta)cosh(\beta) - 1 = 0, \quad \lambda = (\frac{\beta}{L})^4, \quad \beta \neq 0. \tag{26}$$

We compute the eigenvalues of problem (24) by Magnus expansion with $n = 300$ and compare them with exact solution of equation (26) in Table 3. This problem has large eigenvalues such that $\|G\| = O(\lambda)$. Thus the errors are of order $O(\lambda h^4)$ and the results in Table 3 confirm the error order of Theorem 1 and Remarks 1 and 2.

Table 3: Eigenvalues and absolute errors $\triangle\lambda_k$ of Example 3

k	λ_k Our method	λ_k Exact	$\triangle\lambda_k$
1	1062.777339630	1062.777339606	$2.4E-8$
2	8075.518441201	8075.518441201	$4.5E-10$
3	31035.57009892	31035.57010021	$1.3E-6$
4	84807.08316091	84807.08315850	$2.4E-6$
5	189248.7474113	189248.7474331	$2.2E-5$

5 Conclusion

In this paper, a method based on the Magnus Lie group method developed, which can be used to compute the eigenvalues of the FSLP. Numerical examples were given to confirm the efficiency and accuracy of the method. Although the error grows for large eigenvalues, however the examples show that comparing to the other methods, our method has good accuracy for large eigenvalues. Example 3 shows that this algorithm is applicable for computing the eigenvalues of nonuniform Euler-Bernoulli beam equations. Ledoux [17], presented a modified Magnus method and obtained the large eigenvalues of second order Sturm- Liouville problem with error of order $O(\frac{1}{\lambda})$. It can be an open problem that one may modify the method of this paper in order to improve the results for higher index of eigenvalues.

Acknowledgements

The author would like to thank anonymous reviewers for careful reading of the paper and valuable comments to improve it.

References

1. Abbasbandy, S. and Shirzadi, A. *A new application of the homotopy analysis method: solving the Sturm-Liouville problems,* Commun Nonlinear Sci Numer Simulat, 16 (2011), 112-126.

2. Attili, B.S. and Lesnic, D. *An efficient method for computing eigenelements of Sturm-Liouville fourth-order boundary value problems,* Applied Mathematics and Computation, 182 (2006), 1247-1254.

3. Blanes, S., Casas, F., Oteo, J.A. and Ros, J. *The Magnus expansion and some of its applications,* Physics Reports, 470 (2009), 151-238.

4. Bluman, G.W., Kumei, S. *Symmetries and Differential Equations,* Springer, New York, 1989.

5. Chanane, B. *Accurate solutions of fourth order Sturm-Liouville problems,* Journal of Computational and Applied Mathematics, 234 (2010), 3064-3071.

6. Ghanbari, K. *On the isospectral beams,* Electronic Journal of Differential Equations, 12 (2005), 57-64.

7. Gladwell, G. M. L. *Inverse problems in vibrations,* Kluwer academic, New York, 2004.

8. Greenberg, L. and Marletta, M. *Numerical methods for higher order Sturm-Liouville problems,* Journal of Computational and Applied Mathematics, 125 (2000), 367-383.

9. Greenberg, L. and Marletta, M. *Oscillation theory and numerical solution of fourth order Sturm-Liouville problems,* IMA J. Numer.Anal., 15 (3)(1995), 319-356.

10. Greenberg, L. *A prufer method for calculating eigenvalues of selfadjoint systems of ordinary differential equations,* Parts 1 and 2, University of Maryland, Technical Report TR 91-24, 1991.

11. Iserles, A. and Norsett, S.P. *On the solution of linear differential equations in Lie groups,* Numerical Analysis Report, University of Cambridge, 1997.

12. Iserles, A. *On the global error of discretization methods for highly oscillatory ordinary differential equations,* BIT, 42 (2002), 561-599.

13. Lie, S. and Engel, F. *Theorie de Transformations gruppen,* Teubner, Leipzig, 1893.

14. Miller, K.R. and Michel, N.A. *Ordinary differential equations,* Academic Press, New York, 1982.

15. Moan, P.C. *Efficient approximation of Sturm-Liouville problems using Lie group methods,* Numerical Analysis Report, University of Cambridge, 1998.

16. Olver, P.J. *Applications of Lie Groups to Differential Equations,* Springer, New York, 1986.

17. Ledoux, V., Daele, M.V. and Berghe, G.V. *Efficient numerical solution of the 1D schrodinger problem using Magnus integrators,* IMA journal of Numerical Analysis, 30 (2010), 751-776.

18. Ledoux, V., Daele, M.V. and Berghe, G.V. *Matslise: A Matlab package for the Numerical Solution of Sturm-Liouville and Schrodinger equations,* ACM Transactions on Mathematical Software, 31 (2005), 532-554.

19. Taher, AHS., Malek, A., Masuleh SHM. *Chebychev differentiation matrices for efficient Computation of the eigenvalues of fourth order Sturm-Liouville problems,* Applied Mathematical Modeling, 37 (2013), 4634-4642.

20. Yucel, U. and Boubaker, K. *Differential Quadrature method (DQM) and Boubaker polynomials expansion scheme (BPES) for efficient computation of the eigenvalues of fourth order Sturm-Liouville problems,* Applied Mathematical Modelling, 36 (2012), 158-167.

21. Wafo Soh, C. *Isospectral Euler-Bernoulli beams via factorization and the Lie method,* International Journal of Non-Linear Mechanics, 44 (4)(2009), 396-403.

22. Zanna, A. *On the numerical solution of isospecyral flows,* Ph.D Thesis, University of Cambridge, 1998.

Iranian Journal of Numerical Analysis and Optimization
Vol. 7, No. 1, (2017), pp 13-32
DOI: 10.22067/ijnao.v7i1.46917

Stability Analysis of Conformable Fractional Systems

H. Rezazadeh, H. Aminikhah*, and A.H. Refahi Sheikhani

Abstract

In this paper, we investigate stability analysis of fractional differential systems equipped with the conformable fractional derivatives. Some stability conditions of fractional differential systems are proposed by applying the fractional exponential function and the fractional Laplace transform. Moreover, we check the stability of conformable fractional Lotka-Volterra system with the multi-step homotopy perturbation method to demonstrate the efficiency and effectiveness of the proposed procedure.

Keywords: Stability analysis; Asymptotical stability; Conformable fractional derivative; Lotka-Volterra system.

1 Introduction

Fractional differential equations are generalizations of classical differential equations of integer order that have recently proved to be valuable tools for the modelling of many physical phenomena, and have been the focus of many studies due to their frequent appearances in various applications, such as physics, biology, finance and fractional dynamics, engineering, signal processing and control theory [8, 12]. Finding solutions to fractional differential

*Corresponding author
Received 20 March 2015; revised 9 March 2016; accepted 24 May 2016
H. Rezazadeh
Department of Applied Mathematics, Faculty of Mathematical Sciences, University of Guilan, P.O. Box 1914, Rasht, Iran. e-mail: rezazadehadi1363@gmail.com
Faculty of Engineering Technology, Amol University of Special Modern Technologies, P.O. Box 46168-49767, Amol, Iran.

H. Aminikhah
Department of Applied Mathematics, Faculty of Mathematical Sciences, University of Guilan, P.O. Box 1914, Rasht, Iran. e-mail: aminikhah@guilan.ac.ir

A.H. Refahi Sheikhani
Department of Applied Mathematics, Faculty of Mathematical Sciences, Islamic Azad University, Lahijan Branch, P.O. Box 1616, Lahijan, Iran. e-mail: ah_refahi@yahoo.com

systems is rather complicated, consequently, the stability results of the fractional differential systems have been the main goal of the previous studies. For example, Matignon considers the stability of fractional differential systems in control processing [11], Deng has studied the stability of fractional differential system with multiple time delays [6] and we have investigated the stability of fractional differential systems with Hilfer derivatives [15]. More novelty, authors in [3, 4, 14] studied stability analysis of distributed order fractional differential equations with respect to the nonnegative density function. There are many definitions of fractional derivatives, such as Riemann-Liouville, Grunwald-Letnikov and Caputo's fractional derivatives [12], which these fractional derivatives do not satisfy the product rule, the quotient rule and the chain rule. In 2014, to overcome these and other difficulties, Khalil et al. [7] introduced a new simple well-behaved definition of the fractional derivative called conformable fractional derivative. This fractional derivative is theoretically very easier to handle and also obeys some conventional properties that cannot be satisfied by the existing fractional derivatives, for instance, the chain rule [1]. However this fractional derivative has a weakness, which is the fractional derivative of any differentiable function at point zero.

In this paper, we shall describe stability conditions for conformable fractional differential systems. In particular our analysis covers the linear conformable fractional differential systems with commensurate order and incommensurate order and the nonlinear conformable fractional differential system. At first, stability conditions will be established using fractional exponential function for linear conformable fractional differential systems, corresponding to this result we will derive asymptotic stability for the nonlinear conformable fractional system. Then, we produce sufficient conditions for asymptotical stability of in-commensurate linear conformable fractional differential system by of the fractional Laplace transform and the fractional final value theorem. Finally, we present the multi-step homotopy perturbation method for obtain approximate analytical solution and stability of the conformable fractional Lotka-Volterra system to illustrate the validity of the results.

2 Conformable fractional derivative

Here, some basic definitions and properties of the conformable fractional calculus theory which can be found in [1, 2, 7] are presented.

Definition 1. Let $f : (0, \infty) \to \mathbb{R}$, then, the conformable fractional derivative of f of order α is defined as [7]

$$ {}_tT_\alpha\left(f\right)(t)=\lim_{\varepsilon\to 0}\frac{f(t+\varepsilon t^{1-\alpha})-f(t)}{\varepsilon}, \tag{1}$$

for all $t>0$, $\alpha\in(0,1)$.
If f is α-differentiable in some $(0,a)$, $a>0$, and $\lim_{t\to 0^+}{}_tT_\alpha\left(f\right)(t)$ exists, then by definition

$$ {}_tT_\alpha\left(f\right)(0)=\lim_{t\to 0^+}{}_tT_\alpha\left(f\right)(t). \tag{2}$$

The new definition satisfies the properties which are given in the following theorem.

Theorem 1. *Let $\alpha\in(0,1]$, and f, g be α-differentiable at point t, then [7]*

(i) ${}_tT_\alpha(af+bg)=a\ {}_tT_\alpha(f)+b\ {}_tT_\alpha(g)$, *for all* $a,b\in\mathbb{R}$.

(ii) ${}_tT_\alpha(t^\mu)=\mu\ t^{\mu-\alpha}$, *for all* $\mu\in\mathbb{R}$.

(iii) ${}_tT_\alpha(fg)=f\ {}_tT_\alpha(g)+g\ {}_tT_\alpha(f)$.

(iv) ${}_tT_\alpha(\frac{f}{g})=\frac{g\ {}_tT_\alpha(f)-f\ {}_tT_\alpha(g)}{g^2}$. *In addition, if f is differentiable, then* ${}_tT_\alpha(f)(t)=t^{1-\alpha}\frac{df}{dt}$.

In [1] T. Abdeljawad established the chain rule for conformable fractional derivatives as following theorem.

Theorem 2. Let $f:(0,\infty)\to\mathbb{R}$ be a function such that f is differentiable and also α-differentiable. Let g be a function defined in the range of f and also differentiable; then, one has the following rule

$$ {}_tT_\alpha(fog)(t)=\left({}_tT_\alpha f\right)(g(t))\left({}_tT_\alpha g\right)(t)\ g(t)^{\alpha-1}. \tag{3}$$

If $t=0$, then

$$ {}_tT_\alpha(fog)(0)=\lim_{t\to 0^+}\left({}_tT_\alpha f\right)(g(t))\left({}_tT_\alpha g\right)(t)\ g(t)^{\alpha-1}.$$

The fractional exponential function plays a very important role in the conformable fractional differential equations. The fractional exponential function $e^{\frac{1}{\alpha}t^\alpha}$, where $0<\alpha\leq 1$, is defined by the following series representation,

$$ e^{\frac{1}{\alpha}t^\alpha}=\sum_{k=0}^{\infty}\frac{t^{\alpha k}}{\alpha^k k!}.$$

Now, we list here the fractional derivatives of certain functions [7]

(i) ${}_tT_\alpha(e^{\frac{1}{\alpha}t^\alpha})=e^{\frac{1}{\alpha}t^\alpha}$,

$(ii)\ {}_tT_\alpha(\sin\frac{1}{\alpha}t^\alpha) = \cos\frac{1}{\alpha}t^\alpha,$

$(iii)\ {}_tT_\alpha(\cos\frac{1}{\alpha}t^\alpha) = -\sin\frac{1}{\alpha}t^\alpha,$

$(iv)\ {}_tT_\alpha(\frac{1}{\alpha}t^\alpha) = 1.$

On letting $\alpha = 1$ in these derivatives, we get the corresponding ordinary derivatives.

Definition 2. *(Fractional Integral) [7]* Let $a \geq 0$ and $t \geq a$. Also, let f be a function defined on $(a, t]$ and $\alpha \in (0, 1]$. Then, the α-fractional integral of f is defined by,

$$ {}_tI_a^\alpha f(t) = \int_a^t \frac{f(x)}{x^{1-\alpha}}dx, \tag{4}$$

if the Riemann improper integral exists.

Theorem 3. *(Integration by parts) [1], Let $f, g : [0, b] \to \mathbb{R}$ be two functions such that fg is differentiable. Then*

$$\int_0^b f(t)\ {}_tT_\alpha\left(g\right)(t)\ d\alpha(t) = fg\big|_0^b - \int_0^b g(t)\ {}_tT_\alpha\left(f\right)(t)\ d\alpha(t), \tag{5}$$

where $d\alpha(t) = t^{\alpha-1}dt$.

It is interesting to observe that the α-fractional derivative and the α-fractional integral are inverse of each other as given in [7].

Theorem 4. *(Inverse property). Let $\alpha \in (0, 1]$ and f be a continuous function such that ${}_tI_0^\alpha f$ exists. Then*

$$ {}_tT_\alpha({}_tI_0^\alpha f)(t) = f(t), \qquad for\ \ t \geq 0.$$

Definition 3. *(fractional Laplace transform) [1]* Let $0 < \alpha \leq 1$ and $f : [0, \infty) \to \mathbb{R}$ be real valued function. Then the fractional Laplace transform of order α starting from a of f is defined by,

$$L_\alpha\{f(t)\} = F_\alpha(s) = \int_0^\infty e^{-s\frac{t^\alpha}{\alpha}} f(t)\ d\alpha(t), \tag{6}$$

where $d\alpha(t) = t^{\alpha-1}dt$.

Fractional Laplace transform for certain functions are presented as follows [1]

- $L_\alpha\{1\} = \frac{1}{s}, \quad s > 0.$
- $L_\alpha\{t^p\} = \frac{\alpha^{p/\alpha}}{s^{1+p/\alpha}}\Gamma(1+\frac{1}{\alpha}), \quad s > 0.$
- $L_\alpha\{e^{\frac{t^\alpha}{\alpha}}\} = \frac{1}{s-1}, \quad s > 1.$
- $L_\alpha\{sin\frac{1}{\alpha}t^\alpha\} = \frac{1}{s^2+1}, \quad s > 1.$
- $L_\alpha\{cos\frac{1}{\alpha}t^\alpha\} = \frac{s}{s^2+1}, \quad s > 1.$

Furthermore, using the properties of the fractional exponential function and integration by parts, we have

$$L_\alpha\{{}_tT_\alpha\left(f\right)(t)\} = s\ F_\alpha(s) - f(0). \tag{7}$$

Next we prove a fractional version of final value theorem which will be useful is studying stability of conformable fractional systems.

Theorem 5. *(final value theorem) Let $F_\alpha(s)$ be the fractional Laplace transform of the function $f(t)$. If all poles of $sF_\alpha(s)$ are in the open left-half plane, then,*

$$\lim_{t\to\infty} f(t) = \lim_{s\to 0} sF_\alpha(s). \tag{8}$$

Proof. By using formula (7) we arrive at the following relation

$$\lim_{s\to 0} L_\alpha\{{}_tT_\alpha\left(f\right)(t)\} = \lim_{s\to 0}\left(s\ F_\alpha(s) - f(0)\right) = \lim_{s\to 0}\left(s\ F_\alpha(s)\right) - f(0). \tag{9}$$

Consider now the first term only. Since s is independent of t, the order of integrating and taking the limit can be interchanged

$$\begin{aligned}\lim_{s\to 0} L_\alpha\{{}_tT_\alpha\left(f\right)(t)\} &= \lim_{s\to 0}\int_0^\infty e^{-s\frac{t^\alpha}{\alpha}}\ {}_tT_\alpha\left(f\right)(t)\ d\alpha(t)\\ &= \int_0^\infty\left[\lim_{s\to 0} e^{-s\frac{t^\alpha}{\alpha}}\right]\ {}_tT_\alpha\left(f\right)(t)\ d\alpha(t)\\ &= \int_0^\infty\ {}_tT_\alpha\left(f\right)(t)\ d\alpha(t).\end{aligned} \tag{10}$$

Taking the integration in the last term of (10), implies that

$$\int_0^\infty\ {}_tT_\alpha\left(f\right)(t)\ d\alpha(t) = \lim_{t\to\infty}\ \left(f(t) - f(0)\right). \tag{11}$$

Finally, using (9) and (11), we have

$$\lim_{s\to 0}\left(s\ F_\alpha(s)\right) - f(0) = \lim_{t\to\infty}\ \left(f(t) - f(0)\right).$$

Neither term on the right-hand side depends on s, so we can remove the limit and simplify, resulting in the final value theorem

$$\lim_{s\to 0} s\, F_\alpha(s) = \lim_{t\to\infty} f(t).$$

□

3 Stability analysis of linear conformable fractional differential system

In this section, we consider the stability of the following linear conformable fractional differential system

$$ {}_tT_\alpha\, x(t) = Ax(t), \quad t > 0, \quad x(0) = x_0, \tag{12}$$

where $x \in \mathbb{R}^n$, matrix $A \in \mathbb{R}^{n\times n}$, $x_0 = (x_{10}, x_{20}, \ldots, x_{n0})^T$, and $\alpha = [\alpha_1, \alpha_2, \ldots, \alpha_n]$ such that $0 < \alpha_i \leq 1$, for $i = 1, 2, \ldots, n$.

Remark 1. If $\alpha = \alpha_1 = \alpha_2 = \cdots = \alpha_n$, then system (12) is called a commensurate order system, otherwise system (12) indicates an in-commensurate order system.

Definition 4. The zero solution of linear conformable fractional differential system (12) is said to be stable if, for any initial value x_0, there exists an $\varepsilon > 0$ such that $\|x(t)\| \leq \varepsilon$ for all $t > t_0$. The zero solution is said to be asymptotically stable if, in addition to being stable, $\|x(t)\| \to 0$ as $t \to \infty$.

Now we state stability theorems from commensurate order system, next produce conditions for asymptotical stability of the in-commensurate order system.

Theorem 6. *The solution to the linear commensurate order system* (12) *is given by*

$$x(t) = x_0\, e^{\frac{1}{\alpha}At^\alpha}, \tag{13}$$

where the solution is assumed to be differentiable on $(0, \infty)$.

Proof. Taking the fractional Laplace transform of (12), by formula (7), we have that

$$sX_\alpha(s) - sx_0 = AX_\alpha(s),$$

so that

$$X_\alpha(s) = x_0(sI - A)^{-1}. \tag{14}$$

Now, by applying the fractional Laplace transform of the fractional exponential function, we obtain the claimed result. □

Definition 5. If all the eigenvalues of A satisfy

$$|\arg(\lambda(A)| > \frac{\pi}{2}, \tag{15}$$

then A is said to be Hurwitz stable. If all the eigenvalues of A satisfy

$$|\arg(\lambda(A)| \geq \pi/2, \tag{16}$$

and if $|\arg(\lambda_{j0}(A)| = \pi/2$, λ_{j0} only correspond to simple elementary divisor of A, then A is said to be quasi-stable.

Theorem 7. *The zero solution of the conformable fractional differential system* (12) *is asymptotically stable if and only if A is a Hurwitz stable, and the zero solution of system* (12) *is stable but not asymptotically stable if and only if A is quasi-stable.*

Proof. According to the relation (13), we have

$$x(t) = x_0\, e^{\frac{1}{\alpha}At^\alpha}.$$

Let $A = SJS^{-1}$, J is a Jordan canonical form, where J_i, $1 \leq i \leq r$ has the following form

$$J_i = \begin{pmatrix} \lambda_i & 1 & & \\ & \lambda_i & \ddots & \\ & & \ddots & 1 \\ & & & \lambda_i \end{pmatrix}_{n_i \times n_i}, \qquad \lambda_i \in \mathbb{C},$$

and $\sum_{i=1}^{r} n_i = n$. Then,

$$e^{\frac{1}{\alpha}At^\alpha} = e^{\frac{1}{\alpha}SJS^{-1}t^\alpha} = S\, e^{\frac{1}{\alpha}Jt^\alpha}\, S^{-1},$$

$$e^{\frac{1}{\alpha}Jt^\alpha} = diag\left(e^{\frac{1}{\alpha}J_1t^\alpha}, e^{\frac{1}{\alpha}J_2t^\alpha} \ldots, e^{\frac{1}{\alpha}J_rt^\alpha}\right),$$

$$e^{\frac{1}{\alpha}J_jt^\alpha} = \begin{pmatrix} 1 & \frac{1}{\alpha}t^\alpha & \frac{1}{\alpha^2}\frac{t^{2\alpha}}{2!} & \cdots & \frac{1}{\alpha^{n_j-1}}\frac{t^{(n_j-1)\alpha}}{(n_j-1)!} \\ 0 & 1 & \frac{1}{\alpha}t^\alpha & \cdots & \frac{1}{\alpha^{n_j-2}}\frac{t^{(n_j-2)\alpha}}{(n_j-2)!} \\ 0 & 0 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \cdots & 1 & \frac{1}{\alpha}t^\alpha \\ 0 & 0 & \cdots & 0 & 1 \end{pmatrix}_{n_j \times n_j} e^{\frac{1}{\alpha}\lambda_jt^\alpha}.$$

One can easily show that the stability of zero solution of system (12) is determined by the boundedness of $e^{\frac{1}{\alpha}At^\alpha}$, or the boundedness of $e^{\frac{1}{\alpha}Jt^\alpha}$ or the

boundedness of all $e^{\frac{1}{\alpha}J_j t^\alpha}$ $j = 1, 2, \ldots, r$ i.e., $|\arg(\lambda_j(A)| \geq \pi/2$, and when $|\arg(\lambda_j(A)| = \pi/2$, $n_j = 1$, i.e., A is quasi-stable. Asymptotical stability of the zero solution of system (12) is given by

$$\lim_{t\to+\infty} e^{\frac{1}{\alpha}At^\alpha} = 0,$$

or

$$\lim_{t\to+\infty} e^{\frac{1}{\alpha}Jt^\alpha} = 0,$$

which is equivalent to

$$\lim_{t\to+\infty} e^{\frac{1}{\alpha}J_j t^\alpha} = 0, \qquad (j = 1, 2, \ldots, r).$$

That is,

$$\lim_{t\to+\infty} \begin{pmatrix} 1 & \frac{1}{\alpha}t^\alpha & \frac{1}{\alpha^2}\frac{t^{2\alpha}}{2!} & \cdots & \frac{1}{\alpha^{n_j-1}}\frac{t^{(n_j-1)\alpha}}{(n_j-1)!} \\ 0 & 1 & \frac{1}{\alpha}t^\alpha & \cdots & \frac{1}{\alpha^{n_j-2}}\frac{t^{(n_j-2)\alpha}}{(n_j-2)!} \\ 0 & 0 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \cdots & 1 & \frac{1}{\alpha}t^\alpha \\ 0 & 0 & \cdots & 0 & 1 \end{pmatrix}_{n_j\times n_j} \quad e^{\frac{1}{\alpha}\lambda_j t^\alpha} = 0, \ \forall j = 1, 2, \ldots, r,$$

and thus $|\arg(\lambda_j(A)| > \pi/2$ $(j = 1, 2, \ldots, n)$ implying that A is a Hurwitz matrix. The proof is complete. □

Remark 2. If there is a λ_0 such that $|\arg(\lambda_0)| < \pi/2$, then zero solution of system (12) is unstable.

Remark 3. If A has zero eigenvalue, then zero solution of system (12) is unstable.

Proof. If $\lambda = 0$, then from the proof of Theorem 7, we have

$$\frac{1}{(j-1)!}\left\{\left(\frac{\partial}{\partial\lambda}\right)^{j-1} e^{\frac{1}{\alpha}\lambda t^\alpha}\right\}\Big|_{\lambda=0} = \frac{t^{(j-1)\alpha}}{(j-1)!\alpha^{(j-1)}}, \ j = 1, 2, \ldots, n_i, \ 1 \leq i \leq r.$$

It is obvious that $\lim_{t\to\infty} \frac{t^{(j-1)\alpha}}{(j-1)!\alpha^{(j-1)}} = \infty$ for $j \geq 1$. Thus, $\lim_{t\to\infty} \|x(t)\| = \infty$. □

Now, we consider an in-commensurate linear conformable fractional differential system

$$\begin{cases} {}_tT_{\alpha_1}\, x_1(t) = a_{11}x_1(t) + a_{12}x_2(t) + \cdots + a_{1n}x_n(t), \\ {}_tT_{\alpha_2}\, x_2(t) = a_{21}x_1(t) + a_{22}x_2(t) + \cdots + a_{2n}x_n(t), \\ \qquad\vdots \qquad\qquad\qquad\qquad \vdots \\ {}_tT_{\alpha_n}\, x_n(t) = a_{n1}x_1(t) + a_{n2}x_2(t) + \cdots + a_{nn}x_n(t), \end{cases} \tag{17}$$

where $x_i(0) = x_{i0}$ and $0 < \alpha_i \leq 1$ for $i = 1, 2, \ldots, n$.

We study the stability of system (17) by applying the fractional Laplace transforms on both sides of this system, we have

$$sX_{\alpha_i}(s) - x_{i0} = \sum_{j=1}^{n} a_{ij} X_{\alpha_i}(s), \tag{18}$$

for $i = 1, \ldots, n$, where $X_{\alpha_i}(s)$ is the fractional Laplace transform of $x_i(t)$. We can rewrite (18) as follows

$$\Delta(s). \begin{pmatrix} X_{\alpha_1}(s) \\ X_{\alpha_2}(s) \\ \vdots \\ X_{\alpha_n}(s) \end{pmatrix} = x_0. \tag{19}$$

in which

$$\Delta(s) = \begin{pmatrix} \Delta_{11}(s) & \Delta_{12}(s) & \ldots & \Delta_{1n}(s) \\ \Delta_{21}(s) & \Delta_{22}(s) & \ldots & \Delta_{2n}(s) \\ \vdots & \vdots & \ddots & \vdots \\ \Delta_{n1}(s) & \Delta_{n2}(s) & \cdots & \Delta_{nn}(s) \end{pmatrix},$$

where

$$\Delta_{ij}(s) = \begin{cases} s - a_{ii} & \text{if } i = j, \\ -a_{ij} & otherwise. \end{cases}$$

and $x_0 = (x_{10}, x_{20}, \ldots, x_{n0})^T$. For simplicity, we call $\Delta(s)$ a characteristic matrix of (17), moreover $\det(\Delta(s)) = 0$ is the characteristic equation of system (17). Now, we express the main theorem for checking the stability of system (17).

Theorem 8. *If all roots of* $\det(\Delta(s)) = 0$ *have negative real parts, then zero solution system of* (17) *is asymptotically stable.*

Proof. Multiplying s on both sides of (19) gives, we have

$$\Delta(s). \begin{pmatrix} sX_{\alpha_1}(s) \\ sX_{\alpha_2}(s) \\ \vdots \\ sX_{\alpha_n}(s) \end{pmatrix} = sx_0. \tag{20}$$

if all roots of the $\det(\Delta(s)) = 0$ lie in open left half complex plane (i.e. $\Re(s) < 0$), then, we consider (20) in $\Re(s) \geq 0$. In this restricted area, the relation (20) has a unique solution $sX(s) = (sX_{\alpha_1}(s), sX_{\alpha_2}(s), \ldots, sX_{\alpha_n}(s))$. Since $\lim_{s \to 0} s = 0$, so we have

$$\lim_{s\to 0,\Re(s)\geq 0} sX_{\alpha_i}(s) = 0, \qquad i = 1, 2, \ldots, n$$

which from the Theorem 5, we get

$$\lim_{t\to\infty} x(t) = \lim_{t\to\infty} (x_1(t), x_2(t), \ldots, x_n(t)) = \lim_{s\to 0} (sX_{\alpha_1}(s), sX_{\alpha_2}(s), \ldots, sX_{\alpha_n}(s)) = 0.$$

The above result shows that system (17) is asymptotically stable. □

The inertia of a matrix is the triplet of the numbers of eigenvalues of A with positive, negative, and zero real parts. Now, we generalize the inertia concept for analyzing the stability of fractional linear system.

Definition 6. The inertia of the system (17) is the triple

$$I_{n(\alpha)}(A) = (\pi_{n(\alpha)}(A), \nu_{n(\alpha)}(A), \delta_{n(\alpha)}(A)),$$

where $\pi_{n(\alpha)}(A), \nu_{n(\alpha)}(A)$ and $\delta_{n(\alpha)}(A)$ are, respectively, the number of roots of $\det(\Delta(s)) = 0$ with positive, negative, and zero real parts.

Theorem 9. *The linear conformable fractional differential system* (17) *is asymptotically stable if any of the following equivalent conditions holds.*
(i) The matrix A is a Hurwitz matrix.
(ii) $\pi_{n(\alpha)}(A) = \delta_{n(\alpha)}(A) = 0$.
(iii) All roots of the characteristic equation of system (17) *satisfy* $|\arg(s)| > \pi/2$.

Proof. According to Theorem 8 and Definition 5, proof can be easily obtained. □

4 Stability of non-linear conformable fractional differential systems

In this section, we will mainly discuss the stability of a nonlinear conformable fractional differential system, which can be described by

$$_tT_\alpha\, X(t) = F(X(t)), \qquad t > 0, \;\; 0 < \alpha \leq 1, \tag{21}$$

with the initial value $X(0) = X_0$, where

$$F(X(t)) = \begin{pmatrix} f_1(x_1(t), x_2(t), \ldots, x_n(t)) \\ f_2(x_1(t), x_2(t), \ldots, x_n(t)) \\ \vdots \\ f_n(x_1(t), x_2(t), \ldots, x_n(t)) \end{pmatrix},$$

and $X(t) = (x_1(t), x_2(t), \ldots, x_n(t))^T \in \mathbb{R}^n$.

Theorem 10. *Let $\hat{X}(t) = (\hat{x}_1(t), \hat{x}_2(t), \ldots, \hat{x}_n(t))^T$ is the equilibrium of system (21), i.e. ${}_tT_\alpha \hat{X} = F(\hat{X}) = 0$ and $J = \left(\frac{\partial F}{\partial X}\right)\Big|_{X=\hat{X}}$ is the Jacobian matrix at the point $\hat{X}$, then the point $\hat{X}$ is asymptotically stable if and only if J is a Hurwitz matrix.*

Proof. Let $\zeta(t) = X(t) - \hat{X}$, where $\zeta(t) = (\zeta_1(t), \zeta_2(t), \ldots, \zeta_n(t))$ is a small disturbance from a fixed point. Therefore

$$ {}_tT_\alpha\, \zeta(t) = {}_tT_\alpha \left(X(t) - \hat{X}\right), \tag{22}$$

since ${}_tT_\alpha \left(X(t) - \hat{X}\right) = {}_tT_\alpha X(t) - {}_tT_\alpha \hat{X}$, and ${}_tT_\alpha \hat{X} = 0$, thus, we have

$$\begin{aligned} {}_tT_\alpha \zeta(t) = {}_tT_\alpha X(t) &= F(X(t)) = F(\zeta(t) + \hat{X}) \\ &= F(\hat{X}) + J\zeta(t) + \textit{higher order terms} \\ &\approx J\zeta(t). \end{aligned}$$

System (22) can be written as

$$ {}_tT_\alpha \zeta(t) \approx J\zeta(t), \tag{23}$$

with the initial value $\zeta(0) = X_0 - \hat{X}$. The analytical procedure of linearization is based on the fact that if the matrix J has no purely imaginary eigenvalues, then the trajectories of the nonlinear system in the neighborhood of the equilibrium point have the same form as the trajectories of the linear system [17]. Hence, by applying Theorem 7 the linear system (23) is asymptotically stable if and only if all roots of the characteristic function of J satisfy $|\arg(\lambda(J)| > \pi/2$, which implies that the equilibrium $\hat{X}$ of the nonlinear conformable fractional system (21) is as asymptotically stable. □

Remark 4. The nonlinear conformable fractional system (22) in the point $\hat{X}$ is asymptotically stable if and only if $\pi_{n(\alpha)}(J) = \delta_{n(\alpha)}(J) = 0$.

5 The multi-step homotopy perturbation method

The homotopy perturbation method (HPM) was proposed by He [10] in 1999. This method has been used by many mathematicians and engineers to solve various functional equations. Although the HPM yields a solution series which converges very rapidly in most linear and nonlinear equations, in the case of a large time interval t it may produce a large error. To overcome this

shortcoming, the multi-step homotopy perturbation method (MHPM) was presented in [5], to solve nonlinear ordinary differential equations. For the convenience of the reader, we will first present a brief account of HPM. Let us consider the following differential equation:

$$A(u) - f(r) = 0, \quad r \in \Omega, \tag{24}$$

with boundary conditions

$$B(u, \frac{\partial u}{\partial n}) = 0, \quad r \in \Gamma, \tag{25}$$

where A is a general differential operator, B is a boundary operator, $f(r)$ is a known analytic function, and Γ is the boundary of the domain Ω.
The operator A can be generally divided into two parts L and N, where L is linear, while N is nonlinear. Therefore Equation (24) can be written as follows:

$$L(u) + N(u) - f(r) = 0. \tag{26}$$

By using homotopy technique, one can construct a homotopy $y(r,p) : \Omega \times [0,1] \longrightarrow R$ which satisfies

$$H(y,p) = (1-p)[L(y)+L(u_0)]+p[A(y)-f(r)] = 0, \quad p \in [0,1], \quad r \in \Omega, \tag{27}$$

which is equivalent to

$$H(y,p) = L(y) - L(u_0) + pL(u_0) + p[N(y) - f(r)] = 0, \tag{28}$$

where $p \in [0,1]$ is an embedding parameter, and u_0 is an initial guess approximation of Equation (24) which satisfies the boundary conditions. Clearly, we have

$$H(y,0) = L(y) - L(u_0) = 0, \tag{29}$$

$$H(y,1) = A(y) - f(r)] = 0. \tag{30}$$

Thus, the changing process of p from 0 to 1 is just that of $y(r,p)$ from $u_0(r)$ to $y(r)$. In topology this is called deformation and $L(y) - L(u_0)$ and $A(y) - f(r)$ are called homotopic. If, the embedding parameter p, $(0 \leq p \leq 1)$ is considered as a small parameter, applying the classical perturbation technique, we can naturally assume that the solution of Equations (27) and (28) can be given as a power series in p, i.e.,

$$y = y_0 + py_1 + p^2 y_2 + \cdots . \tag{31}$$

According to HPM, the approximation solution of Equation (24) can be expressed as a series of the power of p, i.e.

$$u = \lim_{p \to 1} y = y_0 + y_1 + y_2 + \cdots . \tag{32}$$

Now, consider a general systems of conformable fractional ordinary differential equations

$$\begin{aligned} {}_tT_{\alpha_1}u_1 + g_1(t,u_1,u_2,\ldots,u_m) &= f_1(t),\\ {}_tT_{\alpha_2}u_2 + g_2(t,u_1,u_2,\ldots,u_m) &= f_2(t),\\ &\vdots\\ {}_tT_{\alpha_m}u_m + g_m(t,u_1,u_2,\ldots,u_m) &= f_m(t), \end{aligned} \tag{33}$$

subject to the initial conditions

$$u_1(t_0) = c_1, \quad u_2(t_0) = c_2, \quad \ldots \quad u_m(t_0) = c_m. \tag{34}$$

First, we write the system (33) in the operator form

$$\begin{aligned} L(u_1) + N_1(t,u_1,u_2,\ldots,u_m) - f_1(t) &= 0,\\ L(u_2) + N_2(t,u_1,u_2,\ldots,u_m) - f_2(t) &= 0,\\ &\vdots\\ L(u_m) + N_m(t,u_1,u_2,\ldots,u_m) - f_m(t) &= 0, \end{aligned} \tag{35}$$

subject to the initial conditions (34), where $L\,(u_l) =_t T_\alpha\,(u_l)$ is a linear operator and $N_1, N_2, \ldots, N_m$ are the nonlinear operators. To apply the MHPM, we first construct a homotopy for system (35) as follows

$$\begin{aligned} L(u_1) - L(v_1) + pL(v_1) + p[N_1(u_1,u_2,\ldots,u_m) - f_1(t)] &= 0,\\ L(u_2) - L(v_2) + pL(v_2) + p[N_2(u_1,u_2,\ldots,u_m) - f_2(t)] &= 0,\\ &\vdots\\ L(u_m) - L(v_m) + pL(v_m) + p[N_m(u_1,u_2,\ldots,u_m) - f_m(t)] &= 0, \end{aligned} \tag{36}$$

where $v_1, v_2, \ldots, v_m$ are initial approximations which satisfying the given conditions. Let us take the initial approximations as follows

$$\begin{aligned} u_{1,0}(t) = v_1(t) = u_1(t_0) &= c_1,\\ u_{2,0}(t) = v_2(t) = u_2(t_0) &= c_2,\\ &\vdots\\ u_{m,0}(t) = v_m(t) = u_m(t_0) &= c_m, \end{aligned} \tag{37}$$

and

$$\begin{aligned} u_1(t) &= u_{1,0}(t) + pu_{1,1}(t) + p^2u_{1,2}(t) + p^3u_{1,3}(t)\ldots, \\ u_1(t) &= u_{2,0}(t) + pu_{2,1}(t) + p^2u_{2,2}(t) + p^3u_{2,3}(t)\ldots, \\ &\vdots \\ u_m(t) &= u_{m,0}(t) + pu_{m,1}(t) + p^2u_{m,2}(t) + p^3u_{m,3}(t)\ldots, \end{aligned} \tag{38}$$

where $u_{i,j}$, $(i = 1, 2, \ldots, m; j = 1, 2, \ldots)$ are functions yet to be determined. Substituting (38) into (36) and arranging the coefficients of the same powers of p, we get

$$\begin{aligned} &L(u_{1,1}) + L(v_1) + N_1(u_{1,0}, u_{2,0}, \ldots, u_{m,0}) - f_1 = 0, \quad u_{1,1}(t_0) = 0, \\ &L(u_{2,1}) + L(v_2) + N_2(u_{1,0}, u_{2,0}, \ldots, u_{m,0}) - f_2 = 0, \quad u_{2,1}(t_0) = 0, \\ &\qquad\vdots \\ &L(u_{m,1}) + L(v_m) + N_m(u_{1,0}, u_{2,0}, \ldots, u_{m,0}) - f_m = 0, \quad u_{m,1}(t_0) = 0, \\ &L(u_{1,2}) + N_1(u_{1,1}, u_{2,1}, \ldots, u_{m,1}) = 0, \quad u_{1,2}(t_0) = 0, \\ &L(u_{2,2}) + N_1(u_{1,1}, u_{2,1}, \ldots, u_{m,1}) = 0, \quad u_{2,2}(t_0) = 0, \\ &\qquad\vdots \\ &L(u_{m,2}) + N_m(u_{1,1}, u_{2,1}, \ldots, u_{m,1}) = 0, \quad u_{m,2}(t_0) = 0, \end{aligned} \tag{39}$$

etc. We solve the above systems of equations for the unknowns $u_{i,j}$, $(i = 1, 2, \ldots, m; j = 1, 2, \ldots)$ by applying the inverse operator

$$L^{-1}(.) = \int_0^t (.)\, d\alpha(t). \tag{40}$$

Therefore, according to HPM the n-term approximations for the solutions of (33) can be expressed as

$$\begin{aligned} \phi_{1,n}(t) &= u_1(t) = \lim_{p\to 1} u_1(t) = \textstyle\sum_{k=0}^{n-1} u_{1,k}(t), \\ \phi_{2,n}(t) &= u_2(t) = \lim_{p\to 1} u_2(t) = \textstyle\sum_{k=0}^{n-1} u_{2,k}(t), \\ &\vdots \\ \phi_{m,n}(t) &= u_m(t) = \lim_{p\to 1} u_m(t) = \textstyle\sum_{k=0}^{n-1} u_{m,k}(t), \end{aligned} \tag{41}$$

The solution obtained by HPM is not valid for large t. A simple way of ensuring validity of the approximations for large t is to treat the algorithm of HPM in a sequence of intervals choosing the initial approximations as

$$\begin{aligned} u_{1,0}(t) &= v_1(t) = u_1(t^*) = c_1^*, \\ u_{2,0}(t) &= v_2(t) = u_2(t^*) = c_2^*, \\ &\vdots \\ u_{m,0}(t) &= v_m(t) = u_m(t^*) = c_m^*, \end{aligned} \tag{42}$$

where t^* is the left-end point of each subinterval. Now we solve (42) for the unknowns $u_{i,j}$, $(i = 1, 2, \ldots, m; j = 1, 2, \ldots)$ by applying the inverse linear operator

$$L^{-1}(.) = \int_{t^*}^{t} (.)\, d\alpha(t). \tag{43}$$

In order to carry out the iterations in every subinterval of equal length $\triangle t$, $[0, t_1)$, $[t_1, t_2)$,
$[t_2, t_3), \ldots,$ $[t_{j-1}, t)$, we would need to know the values of the following

$$u_{1,0}^*(t) = u_1(t^*), \quad u_{2,0}^*(t) = u_2(t^*), \ldots, u_{m,0}^*(t) = u_m(t^*). \tag{44}$$

But, in general, we do not have these information at our clearance except at the initial point $t^* = t_0$. A simple way for obtaining the necessary values could be by means of the previous n-term approximations $\phi_{1,n}$, $\phi_{2,n}, \ldots, \phi_{m,n}$ of the preceding subinterval given by (41), i.e.

$$u_{1,0}^* \simeq \phi_{1,n}(t^*), \quad u_{2,0}^* \simeq \phi_{2,n}(t^*), \ldots, u_{m,0}^* \simeq \phi_{m,n}(t^*). \tag{45}$$

6 An illustrative example

The following example is presented to illustrate the effectiveness and applicability of the proposed stability criteria.

The Lotka-Volterra equations, also known as the predator-prey (or parasite-host) equations, are a pair of first order, non-linear, differential equations frequently used to describe the dynamics of biological systems in which two species interact on each other, one is a predator and the other is its prey. They were proposed independently by Alfred J. Lotka in 1925 and Vito Volterra in 1926 [9, 16]. The conformable fractional Lotka-Volterra system is described by the following nonlinear fractional conformable differential equations [13]

$$\begin{aligned} {}_tT_{\alpha_1} x_1(t) &= x_1(t)\left(r - a\, x_1(t) - b\, x_2(t)\right), \\ {}_tT_{\alpha_2} x_2(t) &= x_2(t)\left(-d + c\, x_1(t)\right), \end{aligned} \qquad 0 < \alpha_1, \alpha_2 \leq 1, \tag{46}$$

with the initial values $x_1(0) = x_{10}$ and $x_2(0) = x_{20}$, where $x_1, x_2 \geq 0$ are prey and predator densities, respectively, and all constants r, a, b, c and d are positive. This system has three equilibrium $E_1 = (0, 0)$, $E_2 = (\frac{r}{a}, 0)$ and

$E_3 = (\frac{d}{c}, \frac{cr-ad}{cb})$. The Jacobian matrix for equilibria $E^* = (x^*, y^*)$ is defined as

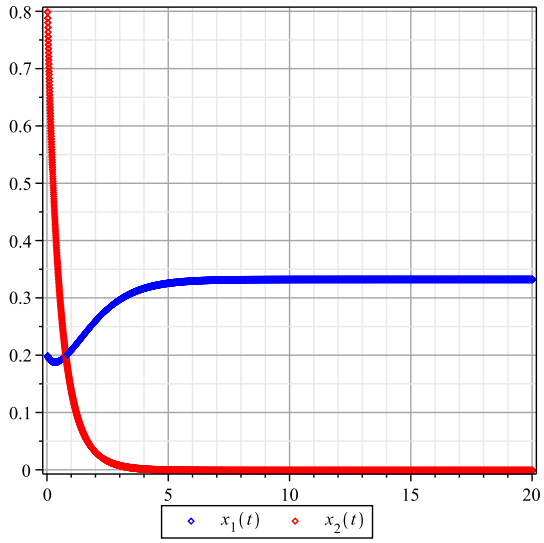


Figure 1: The equilibrium point E_2 of the system (46) with $(\alpha_1, \alpha_2) = (0.95, 0.9)$ and $(r, a, b, c, d) = (1, 3, 1, 2, 2)$ is asymptotically stable

$$J = \begin{bmatrix} r - 2ax^* - by^* & -bx^* \\ cy^* & -d + cx^* \end{bmatrix}. \tag{47}$$

When $(r, a, b, c, d) = (1, 3, 1, 2, 2)$ the system equilibrium points are $E_1 = (0, 0)$, $E_2 = (\frac{1}{3}, 0)$ and $E_3 = (1, -2)$. Corresponding eigenvalues for equilibrium point E_1 are $\lambda_1 = 1$, $\lambda_2 = -2$, since $I_{n(\alpha)}(J) = (1, 1, 0)$, hence the equilibrium point E_1 is unstable. For equilibrium point E_2 the eigenvalues are $\lambda_1 = -1$, $\lambda_2 = \frac{-4}{3}$, because $I_{n(\alpha)}(J) = (0, 2, 0)$, thus the equilibrium point is asymptotically stable and the equilibrium point E_3 is unstable, because $I_{n(\alpha)}(J) = (1, 1, 0)$. Let us consider the following parameters of the system (46), $(r, a, b, c, d) = (3, 0.5, 2, 2, 2.5)$, for these parameters the system (46) has three equilibrium points $E_1 = (0, 0)$, $E_2 = (6, 0)$ and $E_3 = (1.25, 1.1875)$ and their corresponding inertias are $I_{n(\alpha)}(J) = (1, 1, 0)$ for E_1, E_2 and $I_{n(\alpha)}(J) = (0, 2, 0)$ for E_3, thus the equilibrium point E_3 is asymptotically stable.

Figure 1 shows that system (46) with parameters $(r, a, b, c, d) = (1, 3, 1, 2, 2)$ and $(\alpha_1, \alpha_2) = (0.95, 0.9)$ is asymptotically stable in the equilibrium point E_2. From Figure 2 we can see that the system (46) is asymptotically stable in the equilibrium point E_2, with parameters $(r, a, b, c, d) = (3, 0.5, 2, 2, 2.5)$ and $(\alpha_1, \alpha_2) = (0.95, 0.85)$.

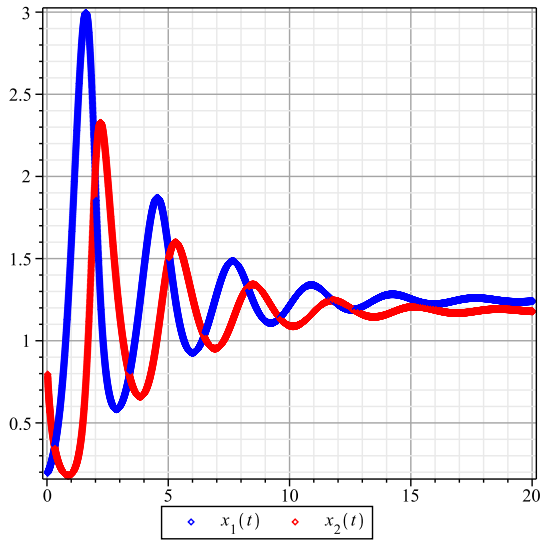


Figure 2: The equilibrium point E_3 of the system (46) with $(\alpha_1, \alpha_2) = (0.95, 0.85)$ and $(r, a, b, c, d) = (3, 0.5, 2, 2, 2.5)$ is asymptotically stable

In system (46) when we take $a = 0$, we obtain modified the fractional conformable Lotka-Volterra system

$$\begin{aligned} {}_tT_{\alpha_1} x_1(t) &= r x_1(t) - b\, x_1(t) x_2(t), \\ {}_tT_{\alpha_2} x_2(t) &= x_2(t)\left(-d + c\, x_1(t)\right). \end{aligned} \tag{48}$$

System (48) has two equilibrium points: $E_1 = (0, 0)$, $E_2 = (\frac{d}{c}, \frac{r}{b})$.

The Jacobian matrix of the system (48), evaluated at the equilibrium $E^* = (x^*, y^*)$, is given by

$$J = \begin{bmatrix} r - by^* & -bx^* \\ cy^* & -d + cx^* \end{bmatrix}. \tag{49}$$

The eigenvalues of the Jacobian matrix (48) evaluated at all equilibrium points show that all equilibria are unstable. Since, for the equilibrium point E_1 we obtain $\lambda_1 = r$ and $\lambda_2 = -d$, for the equilibrium point E_2 we get $\lambda_{1,2} = \pm\, i\sqrt{rd}$. All these eigenvalues satisfy the condition for the system to be unstable $\left(I_{n(\alpha)}(J) = (1, 1, 0)\right)$. Figure 3 shows that the system (48) with parameters $(r, b, c, d) = (1, 1, 4, 2)$ and $(\alpha_1, \alpha_2) = (0.97, 0.97)$ is unstable.

All the results are calculated by using the computer algebra package Maple. The term-number of MHPM series solutions is fixed $N = 4$ and the time step size $h = 0.1$, with the initial conditions $(x_1(0), x_2(0)) = (0.2, 0.8)$.

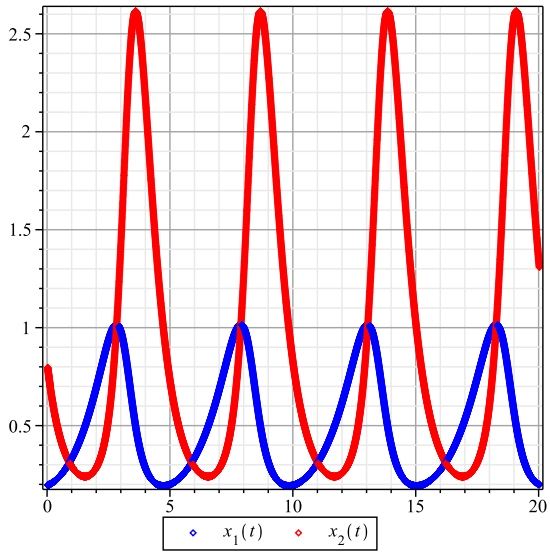


Figure 3: System (48) with $(\alpha_1, \alpha_2) = (0.97, 0.97)$ and $(r, b, c, d) = (1, 1, 4, 2)$ is unstable

7 Conclusion

In this paper, we have studied the stability analysis of fractional differential systems. Fractional derivatives are described by conformable fractional derivatives. At first, the stability conditions established by fractional exponential function for commensurate linear fractional differential systems. Then, we proved a fractional version of final value theorem and by using this theorem we proposed sufficient conditions on the asymptotical stability for in-commensurate linear fractional differential system. The numerical simulations of conformable fractional Lotka-Volterra system are used to illustrate our main result. Although this paper just focuses on the linear systems and non-linear systems but, the problems remain open for the multi-order fractional differential systems, the time-delayed fractional differential systems and distributed order fractional systems. This will be the investigation goal of future works.

Acknowledgement

We are very grateful to the anonymous referees for their careful reading and valuable comments which led to the improvement of this paper.

References

1. Abdeljawad, T. *On conformable fractional calculus,* J. Comput. Appl. Math., 279 (2015) 57-66.

2. Abdeljawad, T., Al Horani, M., and Khalil, R. *Conformable fractional semigroups of operators,* J. Semigroup Theory Appl., 2015 (2015) 1-9.

3. Aminikhah, H., Refahi Sheikhani, A.H. and Rezazadeh, H. *Stability analysis of distributed order fractional Chen system*, The Sci. World J., 2013 (2013) 1-13.

4. Aminikhah, H., Refahi Sheikhani, A.H. and Rezazadeh, H. *Stability analysis of linear distributed order system with multiple time delays*, U.P.B. Scientific Bulletin-Series A-Applied Mathematics and Physics, 77 (2015) 207-218.

5. Chowdhury, M.S. and Hashim, I. *Application of multistage homotopy perturbation method for the solutions of the Chen system,* Nonlinear Anal. Real World Appl., 10 (2009) 381-391.

6. Deng, W., Li, C. and Lu, J. *Stability analysis of linear fractional differential system with multiple time delays*, Nonlinear Dynam., 48 (2007) 409-416.

7. Khalil, R., Al Horani, M., Yousef, A. and Sababheh, M. *A new definition of fractional derivative,* J. Comput. Appl. Math., 264 (2014) 65-70.

8. Kilbas, A.A., Srivastava, H.M., Trujillo, J.J. *Theory and Application of Fractional Differential Equations,* Elsevier, New York, 2006.

9. Lotka, A.J. *Elements of Physical Biology,* Williams and Wilkins Company, Baltimore, 1925.

10. He, J.H. *Homotopy perturbation technique,* Comput. Methods Appl. Mech. Eng., 178 (1999) 257-262.

11. Matignon, D. *Stability results for fractional differential equations with applications to control processing*, Comput. Engrg. Syst. Appl., 2 (1996) 963-968.

12. Oldham, K.B., Spanier, J. *The Fractional Calculus,* Academic Press, New York, 2006.

13. Pielou, E. *An Introduction to Mathematical Ecology,* Wiley-Interscience, New York, 1969.

14. Saberi Najafi, H., Refahi Sheikhani, A.H. and Ansari, A. *Stability Analysis of Distributed Order Fractional Differential Equations*, Abstr. Appl. Anal., 2011 (2011) 1-12.

15. Rezazadeh, H., Aminikhah, H., and Refahi Sheikhani, A.H. *Stability analysis of Hilfer fractional differential systems*, Mathematical Communications, 21 (2016) 45-64.

16. Volterra, V. *Fluctuations in the abundance of a species considered mathematically,* Nature, 118 (1926) 558-560.

17. Vukic, Z., Kuljaca, L. *Nonlinear Control Systems*, CRC Press, 2003.

Iranian Journal of Numerical Analysis and Optimization
Vol. 7, No. 1, (2017), pp 33-45
DOI: 10.22067/ijnao.v7i1.39519

A parametric iteration method for solving Lane-Emden type equations

R. Chaharpashlou

Abstract

In this paper, an analytical method called the parametric iteration method (PIM) is presented for solving the second-order singular IVPs of Lane-Emden type, and its local convergence is discussed. Since it is often useful to have an approximate analytical solution to describe the Lane-Emden type equations, especially for ones where the closed-form solutions do not exist at all, therefore, an effective improvement of the PIM is further proposed that is capable of obtaining an approximate analytical solution. The improved PIM is finally treated as an algorithm in a sequence of intervals for finding accurate approximate solutions of the nonlinear Lane-Emden type equations. Also, we show how to identify an approximate optimal value of the convergence accelerating parameter within the frame of the method. Some examples are given to demonstrate the efficiency and accuracy of the proposed method.

Keywords: Piecewise-truncated parametric iteration method; Truncated parametric iteration method; Parametric iteration method; Nonlinear Lane-Emden type equations.

1 Introduction

Recently, a lot of attention has been focused on the study of singular initial value problems (IVPs) in the second-order ordinary differential equations (ODEs). Many problems in mathematical physics and astrophysics can be modelled by the so-called IVPs of the Lane-Emden type equation [2, 4, 14]:

$$\begin{cases} y^{''} + \dfrac{2}{x} y^{'} + f(x,y) = g(x), \quad x > 0, \\ y(0) = a, \quad y^{'}(0) = b, \end{cases} \tag{1}$$

*Corresponding author
Received 17 September 2014; revised 19 November 2015; accepted 23 July 2016
R. Chaharpashlou
Department of Basic Sciences, Jundi-Shapur University of Technology, Dezful, Iran. e-mail: chaharpashlou@jsu.ac.ir

where a and b are constants, $f(x, y)$ is a continuous real valued function, and $g(x) \in C[0, \infty]$. When $f(x, y) = K(y)$, $g(x) = 0$, Equation(1) reduces to the classical Lane-Emden equation which, with specified $K(y)$, was used to model several phenomena in mathematical physics and astrophysics such as the theory of stellar structure, the thermal behavior of a spherical cloud of gas, isothermal gas sphere and theory of thermionic currents [2, 4, 14].

Since, the Lane-Emden type equations have significant applications in many fields of scientific and technical world, a variety of forms of $f(x, y)$ and $g(x)$ have been investigated by many researchers (e.g., [3, 16, 17]). A discussion of the formulation of these models and the physical structure of the solutions can be found in the literature. Though the numerical solution of the Lane-Emden Equation (1), as well as other various linear and nonlinear singular IVPs in quantum mechanics and astrophysics [9], is numerically challenging because of the singularity behavior at the origin $x = 0$, but analytical solutions are much needed for physical understanding. Recently, many analytical methods were used to solve the Lane-Emden equation [7, 8, 10, 18]. Those methods are based on either series solutions or perturbation techniques [1, 11–13]. However, the convergence region of the corresponding results is rather small.

The strategy that will be pursued in this work rests mainly on establishing useful algorithms based on the parametric iteration method (PIM) [5, 6, 15] for finding highly accurate solution of the Lane-Emden type equations that they

- Overcome the main difficulty arising in the singularity of the equation at $x = 0$.
- Provide us with a convenient way to modify the convergence region and rate of the solution.
- Are simple to implement, accurate when applied to the Lane-Emden type equations and avoid tedious computational works.

The examples analyzed in the present paper reveal that the newly developed algorithms are easy, effective and accurate to solve the singular IVPs of Lane-Emden type equation.

2 Analysis of methods

In this section, the PIM is described for solving Equation(1). This method provides the solution of Equation(1) as a sequence of iterations. The method gives rapidly convergent successive approximations of the exact solution if such a solution exists, otherwise approximations can be used for numerical purposes. The idea of the PIM is very simple and straightforward. To explain the basic idea of the PIM, we first consider Equation(1) as follows:

$$L[y(x)] + N[y(x)] = g(x), \tag{2}$$

with

$$L[y(x)] = y^{''}(x) + \frac{2}{x}y^{'}(x), \quad N[y(x)] = f(x, y), \tag{3}$$

where L denotes the so-called auxiliary linear operator with respect to y and N is a nonlinear operator with respect to y. The basic character of the PIM is to construct a family of iterative processes for Equation(1) as follows [15]:

$$y_{n+1}(x) = y_n(x) + h\int_0^x \left(t - \frac{t^2}{x}\right)\left\{y_n^{''}(t) + \frac{2}{t}y_n^{'}(t) + f(t, y_n(t)) - g(t)\right\}dt, \tag{4}$$

where $y_0(x)$ is the initial guess and the subscript n denotes the n-th iteration, and $h \neq 0$ denotes the so-called auxiliary parameter which can be identified easily and efficiently by the technique proposed in this paper. Accordingly, the successive approximations $y_n(x)$, $n \geq 0$ of the PIM in the auxiliary parameter will be readily obtained by selecting the initial approximation. Consequently, the exact solution can be obtained by using

$$y(x) = \lim_{n\to\infty} y_n(x). \tag{5}$$

It is interesting to note that for the linear Lane-Emden type equations, its exact solution can be obtained easily by only one iteration step due to the fact that the multiplier can be suitably identified, as will be shown in this paper later.
Now we will have the following proposition for the iteration formula (4).

Proposition 1. *If $y(x) \in C^2[0, T]$, then, for $x \leq T$,*

$$\int_0^x \left(t - \frac{t^2}{x}\right)\left\{y^{''}(t) + \frac{2}{t}y^{'}(t)\right\}dt = y(x) - y(0). \tag{6}$$

Proof. Simple integration by parts. □

In the light of (4) and (6), we will have the following simple iteration formula:

$$y_{n+1}(x) = (1 + h)y_n(x) - hy_0 + h\int_0^x \left(t - \frac{t^2}{x}\right)[f(t, y_n(t)) - g(t)]dt, \tag{7}$$

where $y_0 = y(0)$.

Note that the expressions (4) and (7) demonstrate that the variational iteration method (VIM) [7] is a special case of the PIM when $h = -1$. The fact that the PIM solves the Lane-Emden type equations without correction functional and restricted variation can be considered as an advantage of this method over the VIM.

The PIM (7) makes a recurrence sequence $\{y_n(x)\}$. Obviously, the limit of this sequence will be the solution of Equation (1) if this sequence is convergent.
In order to prove the sequence $\{y_n(x)\}$ is convergent, we construct a series

$$y_0(x) + [y_1(x) - y_0(x)] + \cdots + [y_n(x) - y_{n-1}(x)] + \cdots \tag{8}$$

Noticing that

$$S_{n+1} = y_0(x) + [y_1(x) - y_0(x)] + \cdots + [y_n(x) - y_{n-1}(x)] = y_n(x), \tag{9}$$

the sequence $\{y_n(x)\}$ will be convergent if the series is convergent.

Theorem 1. *If $N[y] = f(x, y)$ is Lipschitz-continuous in $[0, T]$ and $g(x) \in C[0, T]$ then the series of (8) is convergent, i.e., the sequence $\{y_n(x)\}$ is convergent for $x \in [0, T]$.*

Proof. According to (7), note that

$$|y_1(x) - y_0(x)| = \left| h \int_0^x \left(t - \frac{t^2}{x} \right) \{N[y_0(t)] - g(t)\} dt \right| \leq |h| M N x, \tag{10}$$

where

$$M = \max_{0 \leq t \leq x \leq T} \left| t - \frac{t^2}{x} \right| = \frac{T}{2}, \quad N = \max_{0 \leq t \leq x \leq T} |N[y_0(t)] - g(t)| \tag{11}$$

From (7) and (10), and the assumption that $|N[y_n] - N[y_{n-1}]| \leq L|y_n - y_{n-1}|$ where L denotes the Lipschitz constant of $N[y(x)]$, it follows that

$$\begin{aligned} |y_2(x) - y_1(x)| &= |1 + h||y_1(x) - y_0(x)| + |h| M L \int_0^x |y_1(t) - y_0(t)| dt \\ &\leq \frac{N}{L} \left[|1 + h| \frac{(|h| M L x)}{1!} + \frac{(|h| M L x)^2}{2!} \right] \\ &= \frac{N}{L} \sum_{k=0}^{1} \binom{1}{k} |1 + h|^{1-k} \frac{(|h| M L x)^{k+1}}{(k+1)!}, \end{aligned} \tag{12}$$

$$|y_3(x) - y_2(x)| \leq \frac{N}{L} \sum_{k=0}^{2} \binom{2}{k} |1+h|^{2-k} \frac{(|h|MLx)^{k+1}}{(k+1)!}, \tag{13}$$

$$|y_{n+1}(x) - y_n(x)| \leq \frac{N}{L} \sum_{k=0}^{n} \binom{n}{k} |1+h|^{n-k} \frac{[|h|MLx]^{k+1}}{(k+1)!}, \tag{14}$$

In view of (14), the convergence of the series (8) can be concluded for the solution domain $x < T$ and $|1+h| < 1$ with the help of the mathematical software such as Maple. Therefore the series of (8) is absolute convergence, i.e., the sequence $\{y_n(x)\}$ is convergent for $x \in [0, T]$. □

3 A piecewise-truncated PIM

The successive iterations of the PIM may be very complex, so that the resulting integrals in the relation (4) may not be performed analytically. Also, the implementation of the PIM generally leads to calculation of unneeded terms, which more time is consumed in repeated calculations for series solutions. Here, an effective modification of the PIM is introduced to eliminate these repeated calculations. To completely stop these repeats in each step, provided that the integrand of (4) in each of iterations is expanded in multivariate Taylor series around $x = 0$. We propose the following improvement of the PIM (4), which is called the truncated PIM (TP):

$$y_{n+1}(x) = y_n(x) + h \int_0^x F_n(x,t)dt, \quad n \geq 0, \tag{15}$$

where

$$\left(t - \frac{t^2}{x}\right)\left\{y_n''(t) + \frac{2}{t}y_n'(t) + f(t, y_n(t)) - g(t)\right\} = F_n(x,t) + O(x^{n+1}) + O(t^{n+1}). \tag{16}$$

It is noteworthy to point out that the TP formula (15) can cancel all the repeated calculations and terms that are not needed as will be shown below. Furthermore, it can reduce the size of calculations. Most importantly, however it is the fact that the TP algorithm (15) solves a Lane-Emden differential equation exactly if its solution is an algebraic polynomial up to some degree.

In general, by using the TP formula (15), we obtain a series solution, which in practice is a truncated series solution. This series solution gives a good approximation to the exact solution in a small region of x. An easy and reliable way of ensuring validity of the approximations (15) for large x, i.e. $I_i = [x_i, x_{i+1}]$ where $\Delta x = x_{i+1} - x_i,\ i = 0, 1, 2, \cdots, N-1$, with $x_0 = 0$ and $x_N = T$. According to the relation (15), therefore, we can construct the following piecewise TP approximations (PTP) in the subintervals I_i. On

$[x_0, x_1]$, let

$$\begin{cases} y_{1,m+1}(x) = y_{1,m}(x) + h\int_{x_0}^{x} F_{1,m}(x,t)dt, \quad m = 0, 1, \cdots, n_1 - 1, \quad x \in [x_0, x_1], \\ y_{1,0}(x) = y(0) + y^{'}(0)(x - x_0) = c_0 + c_0^{'}(x - x_0), \\ \left(t - \frac{t^2}{x}\right)\left\{y_{1,m}^{''}(t) + \frac{2}{t}y_{1,m}^{'}(t) + f(t, y_{1,m}(t)) - g(t)\right\} = \\ \qquad F_{1,m}(x,t) + O[(x - x_0)^{m+1}] + O[(t - x_0)^{m+1}]. \end{cases} \tag{17}$$

Then one can obtain the n_1-order approximation $y_{1,n_1}(x)$ on $[x_0, x_1]$.
On $[x_1, x_2]$, let

$$\begin{cases} y_{2,m+1}(x) = y_{2,m}(x) + h\int_{x_1}^{x} F_{2,m}(x,t)dt, \quad m = 0, 1, \cdots, n_2 - 1, \quad x \in [x_1, x_2], \\ y_{2,0}(x) = y_{1,n_1}(x_1) + y_{1,n_1}^{'}(x_1)(x - x_1) = c_1 + c_1^{'}(x - x_1), \\ \left(t - \frac{t^2}{x}\right)\left\{y_{2,m}^{''}(t) + \frac{2}{t}y_{2,m}^{'}(t) + f(t, y_{2,m}(t)) - g(t)\right\} = F_{2,m}(x,t) \\ \qquad +O[(x - x_1)^{m+1}] + O[(t - x_1)^{m+1}]. \end{cases} \tag{18}$$

Also the n_2-order approximation $y_{2,n_2}(x)$ on $[x_1, x_2]$ can be obtained
In a similar way, on $[x_i, x_{i+1}]$, $i = 2, \cdots, N - 1$, let

$$\begin{cases} y_{i+1,m+1}(x) = y_{i+1,m}(x) + h\int_{x_i}^{x} F_{i+1,m}(x,t)dt, \quad m = 0, 1, \cdots, n_{i+1} - 1, \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad x \in [x_i, x_{i+1}], \\ y_{i+1,0}(x) = y_{i,n_i}(x_i) + y_{i,n_i}^{'}(x - x_i) = c_i + c_0^{'}(x - x_i), \\ \left(t - \frac{t^2}{x}\right)\left\{y_{i+1,m}^{''}(t) + \frac{2}{t}y_{i+1,m}^{'}(t) + f(t, y_{i+1,m}(t)) - g(t)\right\} = F_{i+1,m}(x,t) \\ \qquad +O[(x - x_i)^{m+1}] + O[(t - x_i)^{m+1}]. \end{cases} \tag{19}$$

Hence this implies the n_{i+1}-order approximation $y_{i+1,n_{i+1}}(x)$ on $[x_i, x_{i+1}]$.

Therefore, according to (17)-(19), the approximation of Equation (1) on the entire interval $[0,T]$ will be calculated. It should be emphasized that the PIM and TF algorithms provide analytical solutions on $[0,T]$, while the PTP technique provides analytical solutions in $[x_i, x_{i+1}]$, which are continuous at the end points of each interval, i.e., $y_{i,n_i}(x_i) = c_i = y_{i+1,n_{i+1}}(x_i)$ and $y'_{i,n_i}(x_i) = c'_i = y'_{i+1,n_{i+1}}(x_i)$, $i = 1, 2, \cdots, N-1$.

Remark 1. In the case of failure of convergence of the PIM, the presence of the parameter could play a very important role in the frame of the PIM. Although, we can find a valid region for every physical problem by plotting the solution or its derivatives versus the parameter in some points [6,15], but an approximate optimal value of the convergence accelerating parameter h can be determined at the order of approximation by residual error:

$$Re\ s(h) = \int_{x_0}^{x_f} \{L[y_n(t)] + N[y_n(t)] - g(t)\}^2 dt, \tag{20}$$

One can easily minimize (20) by imposing the requirement $\dfrac{dRes(h)}{dh} = 0$.

4 Implementations

To give a clear overview of the content of this study, the several Lane-Emden type equations will be studied. These equations will be tested by the above-mentioned algorithms, which will ultimately show the usefulness and accuracy of these methods. Moreover, the numerical results indicate that the approach is easy to implement. All the results here are calculated by using the symbolic calculus software Maple 17.

Eexample 1. As a first example, let consider the following linear, the non-homogeneous Lane-Emden equation, i.e., Equation (1) with $f(x,y) = y$ and $g(x) = x^2 e^{-x}$:

$$y'' + \frac{2}{x} y' + y = x^2 e^{-x}, \tag{21}$$

subject to conditions

$$y(0) = 1, \quad y'(0) = 0. \tag{22}$$

The PIM has a very simple approach. Its concepts begin with dividing the left hand (21) into two parts, i.e., the auxiliary linear operator L and the nonlinear operator N as:

$$L[y(x)] = y^{''}(x) + \frac{2}{x}y^{'}(x) + y(x) \quad \text{and} \quad N[y(x)] \equiv 0. \tag{23}$$

This will allow us to construct a family of iterative processes for Eq.(21) as follows [14]:

$$y_{n+1}(x) = y_n(x) + h\int_0^x \left[\frac{t}{x}\sin(x-t)\right]\left\{y_n^{''}(t) + \frac{2}{t}y_n^{'}(t) - t^2e^{-t}\right\}dt. \tag{24}$$

Using simple integration by parts, similar to Proposition 1, implies that

$$\int_0^x \left[\frac{t}{x}\sin(t-x)\right]\{y^{''}(t) + \frac{2}{t}y^{'}(t) + y(t)dt = y(x) - \frac{\sin(x)}{x}y(0), \tag{25}$$

In the light of (24) and (25), the following PIM is:

$$y_{n+1}(x) = (1+h)y_n(x) - hy_0\frac{\sin(x)}{x} - h\int_0^x \left[\frac{t}{x}\sin(t-x)\right](t^2e^{-t})dt, \tag{26}$$

where $y_0 = y(0)$ and $y_0(x) = y(0) + y^{'}(0)(x)$. According to (26), therefore the following approximations with starting the initial guess $y_0(x) = 1$ are:

$$y_1(x) = \frac{(2+2h)x + h\sin(x) - hx(3+3x+x^2)e^{-x}}{2x}, \tag{27}$$

$$y_2(x) = \frac{(2+4h+2h^2)x + (2h+h^2)\sin(x) - hx(6x+2x^2+6+3hx+hx^2+3h)e^{-x}}{2x}, \tag{28}$$

$\vdots$

which the exact solution of Equation(21) yields for $h = -1$, i.e.,

$$y(x) = \frac{x(3+3x+x^2)e^{-x} - \sin(x)}{2x}. \tag{29}$$

Example 2. As other example, we consider the nonlinear and non-homogeneous Lane-Emden equation, i.e., Equation (1) with $f(x,y) = y^3$ and $g(x) = 6 + x^6$ [10]:

$$y^{''} + \frac{2}{x}y^{'} + y^3 = 6 + x^6, \tag{30}$$

subject to conditions

$$y(0) = 0, \quad y^{'}(0) = 0. \tag{31}$$

Now, our aim is to solve the equation (30) by means of the TP algorithm (15). According to (15), it is easily to obtain the following approximations of the TP with starting the initial approximation $y_0(x) = 0$:

$$y_1(x) = 0, \tag{32}$$
$$y_2(x) = -hx^2, \tag{33}$$
$$y_n(x) = [1 - (1+h)^{n-1}]x^2, \quad n \geq 3. \tag{34}$$

If we choose $h = -1$, then the TP algorithm yields the exact solution

$$y(x) = x^2. \tag{35}$$

It is interesting to note that the TP algorithm (15) can solve a Lane-Emden type equation exactly if its solution is an algebraic polynomial up to some degree.

Example 3. As final example, we consider the nonlinear, homogeneous Lane-Emden-type equation, i.e., Equation(1) with $f(x, y) = 4(2e^y + e^{\frac{y}{2}})$ and $g(x) = 0$:

$$y^{''} + \frac{2}{x}y^{'} + 4(2e^y + e^{\frac{y}{2}}) = 0, \tag{36}$$

subject to conditions

$$y(0) = 0, \quad y^{'}(0) = 0. \tag{37}$$

Here, the aim to solve the equation (36) by means of the above-proposed methods. Since the integration of the nonlinear term $4(2e^y + e^{\frac{y}{2}})$ in Equation (36) is not easily evaluated, thus the PIM requires a large amount of computational work to obtain few iterations of the solution (we can replace the nonlinear term with a series of finite components). However, one can used the modified PIM method, i.e., the TP algorithm (15). According to (15), it is easily to obtain the following approximations of (36) with starting the initial approximation $y_0(x) = 0$:

$$y_1(x) = 0, \tag{38}$$
$$y_2(x) = 2hx^2, \tag{39}$$
$$y_3(x) = (4h + 2h^2)x^2, \tag{40}$$
$$y_4(x) = (6h + 6h^2 + 2h^3)x^2 + (2h^2 + h^3)x^4, \tag{41}$$

and so on. To investigate the influence of h on the convergence of the solution obtained via the truncated PIM, here we plot the curves of $y_{20}^{''}(0)$ and $y_{20}^{(4)}$, as shown in Figure 1. According to these curves, it is easy to discover the

valid region h , which corresponds to the horizontal line segments. Now, in the light of (20), an approximate optimal value of h can be determined by the following residual error:

$$Res(h) = \int_0^1 \{y_{20}^{''}(t) + \frac{2}{t} y_{20}^{'}(t) + 4(2e^{y_{20}(t)} + e^{\frac{y_{20}(t)}{2}})\}^2 dt. \qquad (42)$$

By imposing the requirement $dRes(h)/dh = 0$ and solving the resulting equation, we can obtain the approximate optimal value for $h = -1$. Fig. 2 shows a comparison of approximation obtained using the 20th-order TP algorithm for $h = -1$ with the exact solution of Eq.(36), i.e., $y(x) = -2\ln(1 + x^2)$.

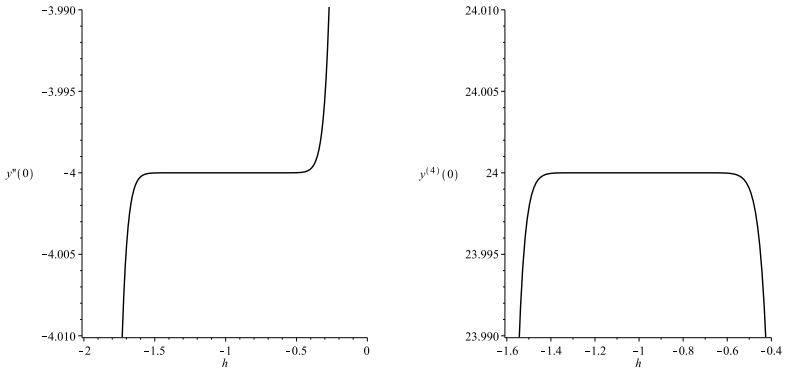


Figure 1: The valid region of h for Example 3 by using the 20th-order TP algorithm (15)

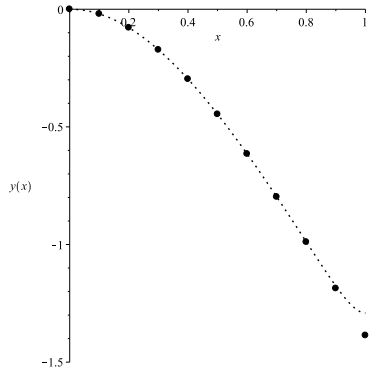


Figure 2: Approximate solution for Example 3 using the TP algorithm where the dotted-line: the 20th-order TP algorithm when $h = -1$ and symbol: the exact solution

As observed, the TP algorithm (15) in solving Equation (36) gives good approximations to the exact solution in a small region of x. In order to

enlarge the convergence region and rate of the series solution, here we implement the PTP (19) proposed in the section 3. According to (19), taking $N = 1000$ and $n_{i+1} = 3, n_{i+1} = 5, i = 0, 1, \cdots N-1$, we can obtain the approximations of (36) on $[0, 100$. Figure 3 shows the absolute errors (the differences between the approximate values and the exact values) of the PTP solution for $n_{i+1} = 3$, $n_{i+1} = 4$, $\Delta x = 0.1$ and the approximate optimal value $h = -1$. From Figure 3, it is easily to found that the present approximations are efficient for a larger interval.

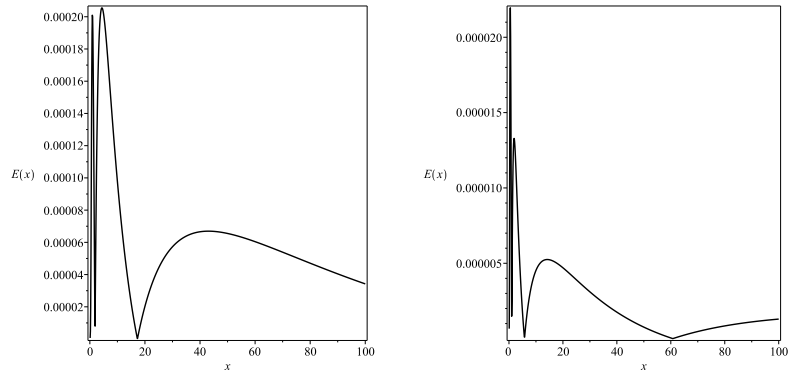


Figure 3: Shows the absolute errors ($E_k(x) = |\mu_{Exact}(x) - \mu_k(x)|$, $k = 4, 5$) of the PTP solution with $\Delta x = 0$ for Example 3 where left: the Abs. Err. of the 4th-order PTP solution for the approximate optimal value $h = -1$ and right: the Abs. Err. of the 5th-order PTP solution for the approximate optimal value $h = -1$

In closing our analysis, we point out that three concreted modeling equations of second-order singular IVPs of the Lane-Emden type equation were investigated by using the algorithms proposed. The obtained results have shown noteworthy performance.

5 Conclusions

Application of the methods based on the PIM presented in this paper to the three Lane-Emden type equations indicates that for the linear Lane-Emden type equations, its exact solution, if such a solution exists, can be obtained easily by only one iteration step due to the fact that the multiplier can be suitably identified, that the TP algorithm can solve a nonlinear Lane-Emden differential equation exactly if its solution is an algebraic polynomial up to some degree, and that for nonlinear Lane-Emden type equations can be useful in general. The numerical results demonstrate that the PIM is a useful analytic tool for solving the Lane-Emden type equations.

Acknowledgement

This research was supported by a grant from Jundi-Shapur University of Technology of Dezful (No. 92-396-02). Further, the author is very grateful to the editor and referees for their comments and suggestions.

References

1. Bender, C.M., Milton, K.A., Pinsky, S.S., and Simmons L.M. *A new perturbative approach to nonlinear problems,* J Math Phys. 30 (1989) 1447-1455.

2. Chandrasekhar, S. *Introduction to the Study of Stellar Structure,* Dover, New York, 1967.

3. Chowdhury, M.SH., and Hashim, I. *Solutions of a class of singular second-order IVPs by homotopy-perturbation method,* Phys. Lett. A 365 (2007) 439-447.

4. Davis, H.T. *Introduction to Nonlinar Differential and Integral Equations,* Dover, New York, 1962.

5. Ghorbani, A. *Toward a new analytical method for solving nonlinear fractional differential equations,* Comput. Methods Appl. Mech. Engrg. 197 (2008) 4173-4179.

6. Ghorbani, A., and Saberi-Nadjafi, J. *A piecewise-spectral parametric iteration method for solving the nonlinear chaotic Genesio system,* Math. Comput. Modelling 54 (2011) 131-139.

7. He, J.H. *Variational approach to the Lane-Emden equation,* Appl Math Comput 143 (2003) 539-541.

8. Hosseini, S. Gh., and Abbasbandy S. *Solution of Lane-Emden Type Equations by Combination of the Spectral Method and Adomian Decomposition Method,* Mathematical Problems in Engineering, vol. 2015, Article ID 534754, 10 pages, 2015. doi:10.1155/2015/534754.

9. Krivec, R., and Mandelzweig V.B. *Numerical investigation of quasilinearization method in quantum mechanics,* Comput Phys Commun 138 (2001) 69-79.

10. Liao, SH.J. *A new analytic algorithm of Lane-Emden type equations,* Appl Math Comput 142 (2003) 1-16.

11. Mandelzweig, V.B., and Tabakin, F. *Quasilinearization approach to nonlinear problems in physics with application to nonlinear ODEs,* Comput Phys Commun. 141 (2001) 268-281.

12. Ramos, J.I. *Linearization techniques for singular initial-value problems of ordinary differential equations,* Appl. Math. Comput. 161 (2005) 525-542.

13. Ramos, J.I. *Series approach to the Lane-Emden equation and comparison with the homotopy perturbation method,* Chaos Solitons Fractals 38 (2008) 400-408.

14. Richardson, O.U. *The Emission of Electricity from Hot Bodies,* London, 1921.

15. Saberi-Nadjafi, J., and Ghorbani, A. *Piecewise-truncated parametric iteration method: a promising analytical method for solving Abel differential equations,* Z. Naturforsch. 65a (2010) 529-539.

16. Shawagfeh, N.T. *Nonperturbative approximate solution for Lane-Emden equation,* J.Math. Phys. 34 (1993) 4364-4369.

17. Wazwaz, A.M. *A new algorithm for solving differential equations of Lane-Emden type,* Appl. Math. Comput. 118 (2001) 287-310.

18. Yildirim, A., and Özis, T. *Solutions of singular IVPs of Lane-Emden type by homotopy perturbation method,* Physics Letters A 369 (2007) 70-76.

Iranian Journal of Numerical Analysis and Optimization
Vol. 7, No. 1, (2017), pp 47-63
DOI: 10.22067/ijnao.v7i1.43960

The concept of B-efficient solution in fair multiobjective optimization problems

D. Foroutannia* and A. Mahmodinejad

Abstract

A problem that sometimes occurs in multiobjective optimization is the existence of a large set of fairly efficient solutions. Hence, the decision making based on selecting a unique preferred solution is difficult. Considering models with fair B-efficiency relieves some of the burden from the decision maker by shrinking the solution set, since the set of fairly B-efficient solutions is contained within the set of fairly efficient solutions for the same problem. In this paper, first some theoretical and practical aspects of fairly B- efficient solutions are discussed. Then, some scalarization techniques are developed to generate fairly B-efficient solutions.

Keywords: Fair optimization; Nondominated; Equitability; B-efficiency; Scalarization.

1 Introduction

The Multiobjective programming has been studied for many years and multiobjective methods have found applications in diverse areas of human life. It is well-known that any multiobjective optimization problem starts usually with an assumption that the criteria are incomparable, i.e., different criteria may have different units and physical interpretations. Many applications, however, arise from situations which present equitable criteria. Equitability is based on the assumption that the criteria are not only comparable (measured on a common scale) but also anonymous (impartial). The latter makes

*Corresponding author
Received 31 January 2015; revised 18 June 2016; accepted 23 July 2016
D. Foroutannia
Department of Mathematics, Vali-e-Asr University of Rafsanjan, Rafsanjan, Iran. e-mail: foroutan@vru.ac.ir

A. Mahmodinejad
Department of Mathematics, Sirjan University of Technology, Sirjan, Iran. e-mail: amahmodi@sirjantech.ac.ir

the distribution of outcomes among the criteria more important than the assignment of outcomes to specific criteria, and therefore models fair allocation of resources.

The fair preference was first known as the generalized Lorenz dominance [5, 7]. Kostreva and Ogryczak [3] are the first ones who introduced the concept of equitability into multiobjective programming. They have shown fair efficiency to be a refinement of Pareto efficiency by adding, to the reflexivity, strict monotonicity and transitivity of the Pareto preference order, the requirements of impartiality and satisfaction of the principle of transfers. Then, Kostreva et al. [4] presented the theory of equitable efficiency in greater generality. They have developed scalarization approaches to generate equitably efficient solutions of linear and nonlinear multiobjective programs. Ogryczak applied equitability to portfolio optimization [8], location problems [9, 12], and telecommunications [11]. Moreover Ogryczak et al. [10], applied fair (equitable) optimization methods for the resource allocation problems in communication networks.

It should be noted that some authors have used the term "equitable" rather than "fair". In this paper, we investigate some theoretical and practical aspects of the fairly B-efficient solutions and propose some approaches to generate equitably B-efficient solutions. The results are generalizations of the results of Kostreva and Ogryczak [3] and Kostreva et al. [4].

2 Terminology

Throughout this article, the following notation is used. Let R^m be the Euclidean vector space and $y', y'' \in R^m$. $y' \leqq y''$ denotes $y'_i \leq y''_i$ for all $i = 1, \cdots, m$. $y' < y''$ denotes $y'_i < y''_i$ for all $i = 1, \cdots, m$. $y' \leq y''$ denotes $y' \leqq y''$ but $y' \neq y''$.

Consider a decision problem defined as an optimization problem with m objective functions. For simplification we assume, without loss of generality, that the objective functions are to be minimized. The problem can be formulated as follows:

$$\begin{cases} \min(f_1(x), f_2(x), \cdots, f_m(x)), \\ \textit{subject to } x \in X \end{cases} \tag{1}$$

where x denotes a vector of decision variables selected from the feasible set X and $f(x) = (f_1(x), f_2(x), \cdots, f_m(x))$ is a vector function that maps the feasible set X into the objective (criterion) space R^m. We refer to the elements of the objective space as outcome vectors. An outcome vector y is attainable if it expresses outcomes of a feasible solution, i.e., $y = f(x)$ for some $x \in X$. The set of all attainable outcome vectors will be denoted by $Y = f(X)$.

In single objective minimization problems, we compare the objective values at different feasible decisions to select the best decision. Decisions are ranked according to the objective values at those decisions and any decision with smallest objective value is called an optimal solution. Similarly, to make the multiobjective optimization model operational, one needs to assume some solution concept specifying what it means to minimize multiobjective functions. The solution concepts are defined by the properties of the corresponding preference model. We assume that solution concepts depend only on the evaluation of the outcome vectors while not taking into account any other solution properties not represented within the outcome vectors. Thus, we can limit our considerations to the preference model in the objective space Y.

In the following, some basic concepts and definitions of preference relations are reviewed from [3]. Preferences are represented by a weak preference relation by $\preceq$, which allows us to compare pairs of outcome vectors y', y'' in the objective space Y. We say $y' \preceq y''$ if and only if "y' is at least as good as y''" or "y' is weakly preferred to y''". In words, $y' \preceq y''$ means that the decision maker thinks that the outcome vector y' is at least as good as the outcome vector y''. It should be noticed that the weak preference relation, the " at least as good as" relation, is given by the decision maker. From $\preceq$, we can derive two other important relations on Y.

Definition 1. Let $y', y'' \in R^m$ and let $\preceq$ be a relation of weak preference defined on $R^m \times R^m$. The strict preference relation, $\prec$, defined by

$$y' \prec y'' \Leftrightarrow (y' \preceq y'' \textit{and not } y'' \preceq y'), \tag{2}$$

and read y' is strictly preferred to y''. Also the indifference relation, $\simeq$, defined by

$$y' \simeq y'' \Leftrightarrow (y' \preceq y'' \textit{and } y'' \preceq y'), \tag{3}$$

and read y' is indifferent to y''.

Hence "y' is at least as good as y''" means that "y' is strictly preferred to y''" or "y' is indifferent to y''".

Definition 2. Preference relations satisfying the following axioms are called rational preference relations:

1. Reflexivity: for all $y \in R^m$, $y \preceq y$.
2. Transitivity: for all $y', y'', y''' \in R^m$, $y' \preceq y''$ and $y'' \preceq y''' \Rightarrow y' \preceq y'''$.
3. Strict monotonicity: for all $y \in R^m$, $y - \epsilon e_i \prec y$ for all $\epsilon > 0$ where e_i denotes the i^{th} unit vector in R^m, for all $i \in \{1, 2, \cdots, m\}$.

The rational preference relations allow us to formalize the Pareto-optimal solution concept with the following definitions.

Definition 3. The outcome vector $y' \in Y$ rationally dominates $y'' \in Y$ iff $y' \prec y''$ for all rational preference relations $\preceq$.

An outcome vector y is rationally nondominated if and only if there does not exist another outcome vector y' such that y' rationally dominates y. Analogously, a feasible solution $x \in X$ is an efficient or Pareto-optimal solution of the multiobjective problem (1) if and only if $y = f(x)$ is rationally nondominated.

It has been shown in [3] that the outcome vector $y' \in Y$ rationally dominates $y'' \in Y$ if and only if $y' \leq y''$. As a consequence, we can state that a feasible solution $x \in X$ is a Pareto-optimal solution of the multiobjective problem (1), if and only if, there does not exist $x' \in X$ such that $f_i(x') \leq f_i(x)$ for $i = 1, 2, \cdots, m$, where at least one strict inequality holds.

Let $\preceq$ be a preference relation defined on $\mathbb{R}^m$.

Definition 4. $\preceq$ is said to be impartial if

$$(y_1, y_2, \cdots, y_m) \simeq (y_{\tau(1)}, y_{\tau(2)}, \cdots, y_{\tau(m)})$$

for all $y \in R^m$, where τ stands for an arbitrary permutation of components of y.

Definition 5. $\preceq$ is said to satisfy the principle of transfers if $y_i > y_j \Rightarrow y - \epsilon e_i + \epsilon e_j \preceq y$, for all $y \in R^m$ and all $\epsilon \in [0, y_i - y_j]$.

Definition 6. A preference relation $\preceq$ defined on R^m is called a fair rational preference relation if it is reflexive, transitive, strictly monotonic, impartial and satisfies the principle of transfers.

The fair rational preference relations allow us to define the concept of fairly efficient solution.

Definition 7. Let $y', y'' \in Y$. We say that y' fairly dominates y'', and denote by $y' \prec_e y''$ iff $y' \prec y''$ for all fair rational preference relations $\preceq$.

An outcome vector y is fairly nondominated if and only if there does not exist another outcome vector y' such that y' fairly dominates y. Analogously, a feasible solution x is called a fairly efficient solution of the multiobjective problem (1) if and only if $y = f(x)$ is fairly nondominated.

Definition 8. Let $y \in R^m$.

1. The function $\Theta : R^m \to R^m$ is called an ordering map iff $\Theta(y) = (\theta_1(y), \theta_2(y), \cdots, \theta_m(y))$, where $\theta_1(y) \geq \theta_2(y) \geq \cdots \geq \theta_m(y)$ in which $\theta_i(y) = y_{\tau(i)}$ for $i = 1, 2, \cdots, m$, and τ is a permutation of the set $\{1, 2, \cdots, m\}$.

2. The function $\overline{\Theta} : R^m \to R^m$ is called a cumulative ordering map iff $\overline{\Theta}(y) = (\bar{\theta}_1(y), \bar{\theta}_2(y), \cdots, \bar{\theta}_m(y))$, where $\bar{\theta}_i(y) = \sum_{j=1}^{i} \theta_j(y)$ for $i = 1, 2, \cdots, m$.

To make it practical, fair efficiency is defined in terms of vector inequalities.

Proposition 1. *([3], Proposition 2.3) For any two vectors $y', y'' \in Y$*

$$y' \preceq_e y'' \Leftrightarrow \overline{\Theta}(y') \leqq \overline{\Theta}(y''),$$

where $\preceq_e$ is given by Definition 7.

3 Fair B-efficient solutions

In this section, we first introduce the concept of fairly B-efficient solution by fair rational preference relations. Then, similar to the fair dominance relation, we can express the fair B-dominance relation in terms of vector inequality on the ordered outcome vectors.

In fair multiobjective optimization, the focus is on the distribution of outcome values while ignoring their ordering. This means that in the multiobjective optimization problem (1) we are interested in a set of values of the objectives without taking into account which objective is taking a specific value. In this respect, let $\{1, 2, \cdots, m\}$ be the index of the vector $\Theta(y) = (\theta_1(y), \theta_2(y), \cdots, \theta_m(y))$ and n be a positive integer such that $n \leq m$. Throughout this article, we suppose that $B = \{B_1, B_2, \cdots, B_n\}$ is a partition of the set $\{1, 2, \cdots, m\}$ such that

$$\max B_i < \min B_{i+1}, \qquad i = 1, 2, \cdots, n-1. \tag{4}$$

The following definition is a necessary notion for the solution concepts of interest in this paper.

Definition 9. Suppose that $n \leq m$ and $B = \{B_1, B_2, \cdots, B_n\}$ is a partition of $\{1, 2, \cdots, m\}$ such that condition (4) is satisfied. The cumulative map $\Theta_B : R^m \to R^n$ is defined by

$$\Theta_B(y) = \left(\sum_{j \in B_1} \theta_j(y), \sum_{j \in B_2} \theta_j(y), \cdots, \sum_{j \in B_n} \theta_j(y) \right),$$

where θ_j's are as defined in Definition 8.

Definition 10. Suppose that $y', y'' \in Y$ are two outcome vectors. We say that y' fairly B-dominates y'' and write $y' \prec_{Be} y''$ iff $\Theta_B(y') \prec \Theta_B(y'')$ for all fair rational preference relations $\preceq$.

Definition 11. We say that outcome vector $y \in Y$ is fairly B-nondominated

iff there does not exist $y' \in Y$ such that $y' \prec_{Be} y$.

Definition 12. We say that feasible solution $x \in X$ is a fairly B-efficient solution of the multiobjective problem (1), iff $y = f(x)$ is fairly B-nondominated.

Similar to the relation of fair B-dominance, we can define the relation of fair B-indifference (indifference for all fair rational preference relations) and the relation of fair weak B-dominance (weak preference for all fair rational preference relations). The relations of fair B-dominance $\prec_{Be}$, B-indifference $\simeq_{Be}$, and weak B-dominance $\preceq_{Be}$ satisfy conditions (2-3).

To make it practical, fair B-efficiency can be defined in terms of vector inequalities. In order to do that, we define certain mapping.

Definition 13. Suppose that $n \leq m$ and $B = \{B_1, B_2, \cdots, B_n\}$ is a partition of $\{1, 2, \cdots, m\}$ such that condition (4) is satisfied. The cumulative ordering map $\overline{\Theta}_B : R^m \to R^n$ is defined by

$$\overline{\Theta}_B(y) = \left(\sum_{j \in B_1} \theta_j(y), \sum_{j \in B_1 \cup B_2} \theta_j(y), \cdots, \sum_{j \in B_1 \cup B_2 \cup \cdots \cup B_n} \theta_j(y) \right),$$

where θ_j's are as defined in Definition 8.

Definition 14. Suppose that $y', y'' \in Y$ are two outcome vectors. The relation $\leq_{Bie}$ is defined as follows.

$$y' \leq_{Bie} y'' \Leftrightarrow \overline{\Theta}_B(y') \leqq \overline{\Theta}_B(y'').$$

Note that if $B_j = \{j\}$ for $j = 1, 2, \cdots, m$, then $\Theta_B(y) = \Theta(y)$ and $\overline{\Theta}_B(y) = \overline{\Theta}(y)$, so the relation $\preceq_{Be}$ becomes the relation $\preceq_e$.

The relations $<_{Bie}$ and $=_{Bie}$ are defined by

$$y' <_{Bie} y'' \Leftrightarrow (y' \leq_{Bie} y'' \textit{and not } y'' \leq_{Bie} y'),$$
$$y' =_{Bie} y'' \Leftrightarrow (y' \leq_{Bie} y'' \textit{and } y'' \leq_{Bie} y').$$

Below, we will discuss the relationship between two preference relations $\preceq_{Be}$ and $\leq_{Bie}$. In order to do that, we need the following proposition.

Proposition 2. *([3], Proposition 2.2) If $\overline{\Theta}(y') \leq \overline{\Theta}(y'')$, then $\Theta(y') \leq \Theta(y'')$ or there exists a finite sequence of vectors $y^0 = y'', y^1, \cdots, y^t$ such that $y^k = y^{k-1} - \epsilon_k e_{i'} + \epsilon_k e_{i''}$, $i', i'' \in \{1, 2, \cdots, m\}$, $0 < \epsilon_k < y^{k-1}_{i'} - y^{k-1}_{i''}$ for $k = 1, 2, \cdots, t$ and $\Theta(y') \leqq \Theta(y^t)$.*

Since each fair rational preference relation $\preceq$ satisfies the principle of transfers, by using Proposition 2, we have

$$\overline{\Theta}(y') \leqq \overline{\Theta}(y'') \Longrightarrow \Theta(y') \preceq \Theta(y'').$$

Theorem 1. *Let $|B_1| \geq |B_2| \geq \cdots \geq |B_n|$ and $y', y'' \in Y$ be two outcome vectors. We have*

$$y' \preceq_{Be} y'' \Longleftrightarrow y' \leq_{Bie} y'',$$
$$y' \prec_{Be} y'' \Longleftrightarrow y' <_{Bie} y''.$$

Proof. By assumption $|B_1| \geq |B_2| \geq \cdots \geq |B_n|$, we have

$$\sum_{j \in B_1} \theta_j(y) \geq \sum_{j \in B_2} \theta_j(y) \geq \cdots \geq \sum_{j \in B_n} \theta_j(y),$$

for all $y \in Y$. This means that the vector $\Theta_B(y)$ is decreasing, so $\overline{\Theta}(\Theta_B(y)) = \overline{\Theta}_B(y)$. It is clear that

$$y' \preceq_{Be} y'' \Longrightarrow y' \leq_{Bie} y'',$$

since the relation $\preceq_e$ is a fair rational preference relation. Conversely suppose that $y' \leq_{Bie} y''$, we deduce $\overline{\Theta}_B(y') \leqq \overline{\Theta}_B(y'')$. So

$$\overline{\Theta}(\Theta_B(y')) \leqq \overline{\Theta}(\Theta_B(y'')).$$

Due to Proposition 2, we have $\Theta_B(y') \preceq \Theta_B(y'')$ for any fair rational preference relation $\preceq$. Thus $y' \preceq_{Be} y''$. By the same reasoning, remaining cases are satisfied. □

Due to Theorem 1, we deduce that the relation $\preceq_{Be}$ is a fair rational preference relation, when $|B_1| \geq |B_2| \geq \cdots \geq |B_n|$. Preference relation $\preceq_{Be}$, in general, does not satisfy the principle of transfers axiom. The truth of this statement is examined by the following example.

Example 1. Let $B_1 = \{1\}$, $B_2 = \{2, 3\}$, $y = (4, 2.5, 2)$. If $\epsilon = 1$ and $y' = y - \epsilon e_1 + \epsilon e_3$, then $y' = (3, 2.5, 3)$. Since $\overline{\Theta}(\Theta_B(y)) = (4.5, 8.5)$ and $\overline{\Theta}(\Theta_B(y')) = (5.5, 8.5)$. We have $\overline{\Theta}(\Theta_B(y')) \nleqq \overline{\Theta}(\Theta_B(y))$, so $y' \not\preceq_{Be} y$.

It should be noted that in this example $\overline{\Theta}_B(y) = (4, 8.5)$ and $\overline{\Theta}_B(y') = (3, 8.5)$. So $y' \leq_{Bie} y$. Also if $y' = (4, 2, 1)$ and $y'' = (3.5, 3, 2)$, it is obvious that $y' \preceq_{Be} y''$ while the $y' \not\leq_{Bie} y''$. This reflects the fact that Theorem 1, in general, is not true for any partition B and condition $|B_1| \geq |B_2| \geq \cdots \geq |B_n|$ is necessary.
By applying Theorem 1 and Definition 14, we have the following statement.

Corollary 1. Suppose that $|B_1| \geq |B_2| \geq \cdots \geq |B_n|$ and $y', y'' \in Y$. We have

$$y' \preceq_{Be} y'' \Leftrightarrow \overline{\Theta}_B(y') \leqq \overline{\Theta}_B(y''),$$
$$y' \prec_{Be} y'' \Leftrightarrow \overline{\Theta}_B(y') \leq \overline{\Theta}_B(y'').$$

Remark 1. If $B_j = \{j\}$ for $j = 1, 2, \cdots, m$, we have Proposition 2.3 from [3].

Note that Corollary 1 permits one to express fair B-efficiency for problem (1) in terms of the standard efficiency for the multiobjective problem with objectives $\overline{\Theta}_B(f(x))$:

$$\min\{\overline{\Theta}_B(f(x)) : x \in X\}. \tag{5}$$

Theorem 2. *Let $|B_1| \geq |B_2| \geq \cdots \geq |B_n|$. A feasible solution $x \in X$ is a fairly B-efficient solution of the multiobjective problem (1) if and only if it is an efficient solution of the multiobjective problem (5).*

Proof. By applying Corollary 1, we obtain the desired result. □

Remark 2. If $B_j = \{j\}$ for $j = 1, 2, \cdots, m$, then we have Corollary 2.2 from [3].

The following theorem provides the relationship between fairly B-efficient solutions and fairly efficient solutions.

Theorem 3. *Let $|B_1| \geq |B_2| \geq \cdots \geq |B_n|$ and $y \in Y$ be a outcome vector. If y is fairly B-nondominated, then it is fairly nondominated.*

Proof. Suppose that y is not fairly nondominated. Then there exists a vector $y' \in Y$ such that $y' \prec_e y$. Due to Proposition 1, $\overline{\Theta}(y') \leq \overline{\Theta}(y)$, so $y' <_{Bie} y$. Since $|B_1| \geq |B_2| \geq \cdots \geq |B_n|$, by using Corollary 1, we deduce that $y' \prec_{Be} y$. □

Corollary 2. Let $|B_1| \geq |B_2| \geq \cdots \geq |B_n|$ and $x \in X$ be a feasible solution. If x is a fairly B-efficient solution of multiobjective problem (1), then it is an fairly efficient solution of (1).

This result suggests that the set of fairly B-efficient solutions is contained within the set of fairly efficient solutions, but the reverse inclusion doesn't hold in general. It should be noted the structure of fair dominance is discussed in Example 2.1 of [3] by Kostreva and Ogryczak. They showed that the set of fairly efficient solutions is contained within the set of efficient solutions. In the following examples, we investigate the effectiveness of fair B-dominance relation with respect to fair dominance.

Example 2. Let's consider the problem

$$\min\left\{(x_1, x_2) : 4x_1 + 5x_2 \geq 72, x_1 \geq 0, 0 \leq x_2 \leq 72/5\right\},$$

and $B = \{B_1\}$ with $B_1 = \{1, 2\}$. It is obvious that

$$E = \{(x_1, x_2) : 4x_1 + 5x_2 = 72, x_1 \geq 0, x_2 \geq 0\},$$
$$F = \{(x_1, x_2) : 4x_1 + 5x_2 = 72, 0 \leq x_1 \leq 8, 8 \leq x_2 \leq 72/5\},$$

and $G = \{(0, 72/5)\}$ are the set of efficient solutions, the set of fairly efficient solutions and the set of fairly B-efficient solutions, respectively. Thus $G \subsetneq F \subsetneq E$.

In the next example, a large number of random solutions are generated for scalable test functions. From this large set of solutions, the nondominated solutions with respect to rational dominance (Pareto dominance), fair dominance and fair B-dominance are calculated.

Example 3. The test problem considered is the VFM1 [14],

$$\begin{aligned} \min_{x \in R^2} y =& \{f_1(x), f_2(x), f_3(x)\} \\ f_1(x) =& x_1^2 + (x_2 - 1)^2 \\ f_2(x) =& x_1^2 + (x_2 + 1)^2 + 1 \\ f_3(x) =& (x_1 - 1)^2 + x_2^2 + 2 \\ & x_1, x_2 \in [-2, 2]. \end{aligned}$$

Figure 1 shows the Pareto, fair and fair B-dominance fronts (objective space).

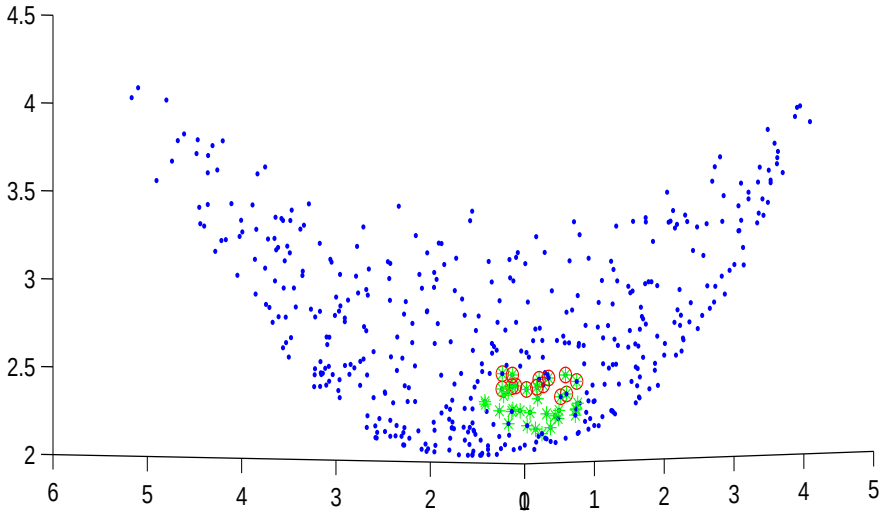


Figure 1: The Pareto, fair and fair B-dominance fronts (objective space) for the VFM1 problem (2 variables and 3 objectives)

From 5000 random solutions, 496 solutions (point) are rationally nondominated, 37 solutions (star) are fairly nondominated and 14 solutions (circle) are fairly B-nondominated, which are obtained by assuming $B_1 = \{1, 2\}, B_2 = \{3\}$.

In the rest of this section, we will investigate the structure of fairly B-efficient set. For this purpose, problem (1) is considered as a multiobjective linear programming problem. It is assumed that, $X \subset R^n$ denotes the feasible set defined by a system of linear equations and inequalities, and the objective functions are

$$f_i(x) = c_i x \qquad (i = 1, 2, \cdots, m), \tag{6}$$

where $c_i^T \in R^n$. Suppose that $n \leq m$ and $B = \{B_1, B_2, \cdots, B_n\}$ is a partition of $\{1, 2, \cdots, m\}$ such that condition (4) is satisfied. We put $b_i = \max B_i$ for $i = 1, 2, \cdots, n$, so

$$\begin{aligned}\overline{\Theta}_B(y) &= \left(\sum_{j=1}^{b_1} \theta_j(y), \sum_{j=1}^{b_2} \theta_j(y), \cdots, \sum_{j=1}^{b_n} \theta_j(y)\right) \\ &= \left(\bar{\theta}_{b_1}(y), \bar{\theta}_{b_2}(y), \cdots, \bar{\theta}_{b_n}(y)\right),\end{aligned}$$

where $\bar{\theta}_{b_i}(y) = \sum_{j=1}^{b_i} \theta_j(y)$ for $i = 1, 2, \cdots, n$, and $b_n = m$. In Theorem 2, we established the equivalence between the fairly B-efficient solutions of (1) and the efficient solutions of problem (5). Now, we further use this result to explore the structure of the fairly B-efficient set. Note that the individual objective functions of problem (5) are convex piecewise linear functions of $y = f(x)$. They can be written in the form

$$\bar{\theta}_{b_i}(y) = \max_{\tau \in \Pi} \left(\sum_{j=1}^{b_i} y_{\tau(j)}\right) \qquad (i = 1, 2, \cdots, n), \tag{7}$$

where Π denotes the set of all permutations τ of the set $\{1, 2, \cdots, m\}$. Thus, the corresponding problem (5) can be expressed in the form of multiobjective linear program

$$\min\left(z_{b_1}, z_{b_2}, \cdots, z_{b_n}\right) \tag{8}$$

subject to

$$x \in X, \tag{9}$$

$$y_i = c_i x \quad \textit{for } i = 1, 2, \cdots, m, \tag{10}$$

$$z_{b_i} \geq \sum_{j=1}^{b_i} y_{\tau(j)} \quad \textit{for } \tau \in \Pi\,; i = 1, 2, \cdots, n. \tag{11}$$

Put $z_B = (z_{b_1}, z_{b_2}, \cdots, z_{b_n})$. According to (7), the problem above is equivalent to problem (5) as stated in the following theorem.

Theorem 4. *A triple (x, y, z_B) is an efficient solution of (8)-(11), if and only if $y = Cx$, $z_B = \overline{\Theta}_B(y)$ and x is an efficient solution of the multiobjective problem (5).*

Remark 3. If $B_j = \{j\}$ for $j = 1, 2, \cdots, m$, then we have Proposition 4.1 from [3].

The question that arises is whether there exist fairly B-efficient solutions. The next result provides some sufficient conditions to answer this question. For this purpose, we need the following theorem.

Theorem 5. *([1], Theorem 6.5) If $X \neq \emptyset$ and there exists $y^0 \in Y$ such that $y^0 \leq Cx$ for all $x \in X$, then there exists a efficient solution of problem (1).*

Theorem 6. *Let $|B_1| \geq |B_2| \geq \cdots \geq |B_n|$. If $X \neq \emptyset$ and there exists $y^0 \in Y$ such that $y^0 \preceq_{Be} Cx$ for all $x \in X$, then there exists a fairly B-efficient solution of problem (1).*

Proof. Due to Corollary 1, $z_B^0 = \overline{\Theta}_B(y^0) \leqq \overline{\Theta}_B(Cx)$ for all $x \in X$. Thus, $z_B^0 \leqq z_B$ for any attainable achievement vector z_B of the multiobjective linear programming (8)-(11), which is by Theorem 4 equivalent to the problem (5). Hence, by using Theorem 5, there exists an efficient solution x^0 of (8)-(11). Due to Corollary 1, x^0 is a fairly B-efficient solution of problem (1). □

Remark 4. If $B_j = \{j\}$ for $j = 1, 2, \cdots, m$, then we have Proposition 4.3 from [3].

Kostreva and Ogryczak in [3] have shown that the set of fairly efficient solutions is nonempty provided that the set of efficient solutions is nonempty.

Proposition 3. *([3], Proposition 4.4) If there exists an efficient solution of problem (1), then there exists a fairly efficient solution of problem (1).*

By applying Corollary 2 and Proposition 3, we obtain the following statement.

Corollary 3. Let $|B_1| \geq |B_2| \geq \cdots \geq |B_n|$. If there exists an efficient solution of problem (1), then there exists a fairly B-efficient solution of problem (1).

In addition to the existence of the fairly B-efficient set and its relationship to the efficient set, we investigate the fact that the fairly B-efficient set is connected. To do so, we need the following theorem.

Theorem 7. *([6], Theorem 2.3) If the set of efficient solutions of a multiobjective linear programming problem is nonempty, then it is connected.*

Theorem 8. *If $|B_1| \geq |B_2| \geq \cdots \geq |B_n|$ and the set of efficient solutions of multiobjective linear programming problem (8)-(11) is nonempty. Then the set of fairly B-efficient solutions of problem (1) is connected.*

Proof. The set of fairly B-efficient solutions of problem (1) is the same as the set of efficient solutions of problem (5). By using Theorem 4, the efficient solutions of problem (5) are in one-to-one correspondence to the efficient solutions of (8)-(11) . Since (8)-(11) is a linear multiobjective optimization problem, its efficient set is a connected set according to Theorem 7. □

Remark 5. If $B_j = \{j\}$ for $j = 1, 2, \cdots, m$, then we have Proposition 4.5 from [3].

4 Generation techniques

In this section, we develop some scalarization-based methods to generate fairly B-efficient solutions. Scalarization is one of the most common approaches used to solve a multiobjective problem. Scalarizing functions are used to transform a given multiobjective problem into a single objective optimization problem, by aggregating the objectives of a multiobjective problem into a single objective. Typical solution concepts for multiobjective problems are defined by scalarizing functions $s : Y \to \mathbb{R}$ to be minimized. Thus the multiobjective problem (1) is replaced with the minimization problem

$$\min\{s(f(x)) : \ x \in X\}. \tag{12}$$

The preference relation corresponding to the problem (12) is defined as follows:

$$y' \preceq y'' \Leftrightarrow s(y') \leq s(y'').$$

For any strictly convex, increasing function $g : R \to R$, the scalarizing function defined by

$$s(y) = \sum_{i=1}^{m} g(y_i)$$

is a strictly monotonic and strictly Schur-convex function [7]. It has been shown in Proposition 3.1 from [3] that the preference relation corresponding to this scalarizing function is a fair rational preference relation. Also, every optimal solution of problem (12) is a fairly efficient solution of the original multiobjective problem.

In the following, we will generate the fairly B-efficient solutions by introducing certain scalarizing functions.

Theorem 9. *Let $g : R \to R$ be a strictly convex and increasing function. The optimal solution of the problem*

$$\min\left\{\sum_{i=1}^{n} g\left(\sum_{j\in Bi}\theta_j(f(x))\right): \quad x\in X\right\}, \tag{13}$$

is a fairly B-efficient solution of the multiobjective problem (1).

Proof. Suppose that x is not a fairly B-efficient solution of the multiobjective problem (1). Then a feasible vector x' must exist such that the vectors $f(x)$ and $f(x')$ satisfy $f(x') \prec_{Be} f(x)$. Namely for all fair rational preference relations $\prec$, we have $\Theta_B(f(x')) \prec \Theta_B(f(x))$. Since the preference relation corresponding to scalarizing function is a fair rational preference relation, we deduce that

$$\sum_{i=1}^{n} g\left(\sum_{j\in Bi}\theta_j(f(x'))\right) < \sum_{i=1}^{n} g\left(\sum_{j\in Bi}\theta_j(f(x))\right),$$

which contradicts the optimal solution of x for (13). □

Remark 6. If $B_j = \{j\}$ for $j = 1, 2, \cdots, m$, we have Corollary 3.1 from [3].

The weighted sum method is one of the most common ways of finding efficient solutions of multiobjective problem. Details of the method can be found in [2]. Further, Kostreva et al. [4] have proven every optimal solution of the weighted sum problem with strictly decreasing positive weights and ordering map $\Theta(f(x))$, is a fairly efficient solution of the original multiobjective optimization problem. If the weighted sum method is applied to problem (5), due to the definition of map $\overline{\Theta}_B$, we have

$$\min\left\{\sum_{i=1}^{n} w_i\left(\sum_{j\in B_1\cup B_2\cup\cdots\cup B_i}\theta_j(f(x))\right): x\in X\right\} \tag{14}$$

where $w \in R^n$ is any positive vector. The above problem is equivalent to

$$\min\left\{\sum_{i=1}^{n} \lambda_i\left(\sum_{j\in B_i}\theta_j(f(x))\right): x\in X\right\} \tag{15}$$

where $\lambda_i = \sum_{j=i}^{n} w_j$ for $i = 1, \cdots, n$.

Theorem 10. *Let $\lambda_1 > \lambda_2 > \cdots > \lambda_n > 0$. If $|B_1| \geq |B_2| \geq \cdots \geq |B_n|$, then the optimal solution of problem (15) is a fairly B-efficient solution of the multiobjective problem (1).*

Proof. Suppose that x is not a fairly B-efficient solution of the multiobjective problem (1). Then a feasible vector x' must exist such that the vectors $f(x)$ and $f(x')$ satisfy $f(x') \prec_{Be} f(x)$. By Corollary 1, we deduce that

$$\sum_{B_1\cup\cdots\cup B_k} \theta_j(f(x')) \leqslant \sum_{B_1\cup\cdots\cup B_k} \theta_j(f(x)) \quad k = 1, 2, \cdots, n,$$

where strict inequality holds at least once. If $w_i = \lambda_i - \lambda_{i+1}$, then $w_i > 0$ for $i = 1, 2, \cdots, n$. So,

$$\sum_{k=1}^{n} w_k \sum_{B_1\cup\cdots\cup B_k} \theta_j(f(x')) < \sum_{k=1}^{n} w_k \sum_{B_1\cup\cdots\cup B_k} \theta_j(f(x)).$$

This means that x cannot be the optimal solution of problem (14). Since problems (14) and (15) are equivalent, the desired result is obtained. □

Remark 7. If $B_j = \{j\}$ for $j = 1, 2, \cdots, m$, then we have Theorem 2 from [4].

A very important result in multobjective linear optimization is the association of efficient solutions with optimal solutions of the scalar weighting problem using positive weights [13]. The following result is the analogue for our fairly B-efficient solution set. It seems that such a result should thus play an important role in analysis of fair B-efficiency.

Theorem 11. *Suppose that $|B_1| \geq |B_2| \geq \cdots \geq |B_n|$ and the function f is defined as in (6). A feasible solution x^0 is a fairly B-efficient solution of problem (1), if and only if, there exists a weight vector $\lambda \in R^n$ with $\lambda_1 > \lambda_2 > \cdots > \lambda_n > 0$ such that x^0 is an optimal solution of problem (15).*

Proof. Sufficiency of the condition follows from Theorm 10. Thus we only need to show that for each fairly B-efficient solution x^0 there exists a weight vector $\lambda \in R^n$ with $\lambda_1 > \lambda_2 > \cdots > \lambda_n > 0$ such that x^0 is an optimal solution of the problem (15). Due to Theorms 2 and 4, if x^0 is a fairly B-efficient solution of (1), then $(x^0, Cx^0, \overline{\Theta}_B(Cx^0))$ is an efficient solution of multiobjective linear program (8)-(11). Thus, from the theory of multiobjective linear optimization [13], there exist positive weights $w_1, w_2, \cdots, w_n$ such that $(x^0, Cx^0, \overline{\Theta}_B(Cx^0))$ is an optimal solution of the problem

$$\min\left\{\sum_{i=1}^{n} w_i z_{b_i} : (8) - (11)\right\} \tag{16}$$

Due to positive weights w_i, the above problem is equivalent to the problem

$$\min\left\{\sum_{i=1}^{n} w_i \bar{\theta}_{b_i}(Cx) : (8) - (11)\right\} \tag{17}$$

where $\bar{\theta}_{b_i}(y) = \sum_{j=1}^{b_i} \theta_j(y)$ for $i = 1, 2, \cdots, n$. Problem (17) is equivalent to problem (15) where $\lambda_i = \sum_{j=i}^{n} w_j$ for $i = 1, \cdots, n$. This completes the proof of the theorem. □

Remark 8. If $B_j = \{j\}$ for $j = 1, 2, \cdots, m$, then we have Proposition 4.6 from [3].

Another way to generate fairly B-efficient solutions is lexicographic minimax approach. It is shown that any optimal solution of the lexicographic minimax problem is fairly efficient for the original multiobjective problem [4]. When applying the lexicographic optimization to the problem (5), we get the lexicographic problem

$$lexmin\{\overline{\Theta}_B(f(x)) : x \in X\}. \tag{18}$$

Due to Definition 13, problem (18) is equivalent to the problem

$$lexmin\{\Theta_B(f(x)) : x \in X\}. \tag{19}$$

We recall that, if $y' \neq y''$ and $s = \min\{i : y'_i \neq y''_i\}$, then $y' <_{lex} y''$ if and only if $y'_s < y''_s$. Also $y' \leq_{lex} y''$ if and only if $y'_s < y''_s$ or $y' = y''$.

Definition 15. A feasible solution $x \in X$ is lexicographically optimal or a lexicographic solution of the multiobjective problem (18), if there is no $x' \in X$ such that $\overline{\Theta}_B(f(x')) <_{lex} \overline{\Theta}_B(f(x))$.

Theorem 12. *Let $x \in X$ be a lexicographically optimal solution of the multiobjective problem (18). Then x is a fairly B-efficient solution of the multiobjective problem (1).*

Proof. Suppose that x is not a fairly B-efficient solution of the multiobjective problem (1). Then there exists a feasible vector x' such that $f(x') \prec_{Be} f(x)$. Due to Corollary 1, $\overline{\Theta}_B(f(x')) \leq \overline{\Theta}_B(f(x))$. So

$$\sum_{j \in B_1 \cup B_2 \cup \cdots \cup B_k} \theta_j(f(x')) < \sum_{j \in B_1 \cup B_2 \cup \cdots \cup B_k} \theta_j(f(x)),$$

for some $k \in \{1, ..., r\}$. Defining

$$s = \min\left\{k : \sum_{j \in B_1 \cup B_2 \cup \cdots \cup B_k} \theta_j(f(x')) < \sum_{j \in B_1 \cup B_2 \cup \cdots \cup B_k} \theta_j(f(x))\right\}.$$

we get

$$\sum_{j \in B_1 \cup B_2 \cup \cdots \cup B_k} \theta_j(f(x')) = \sum_{j \in B_1 \cup B_2 \cup \cdots \cup B_k} \theta_j(f(x)),$$

for $k = 1, 2, \cdots, s-1$ and

$$\sum_{j \in B_1 \cup B_2 \cup \cdots \cup B_s} \theta_j(f(x')) < \sum_{j \in B_1 \cup B_2 \cup \cdots \cup B_s} \theta_j(f(x)).$$

Therefore $\overline{\Theta}_B(f(x')) <_{lex} \overline{\Theta}_B(f(x))$ contradicting the lexicographic optimality of x. □

Remark 9. If $B_j = \{j\}$ for $j = 1, 2, \cdots, m$, then we have Corollary 3.3 from [3].

Since an efficient solution with equal outcomes is a lexicographic minimax solution. By Theorm 12, such a solution is fairly B-efficient.

Corollary 4. If there exists any efficient solution x^0 of problem (1) with equal outcomes $f_1(x^0) = f_2(x^0) = \cdots = f_m(x^0)$, then it is a fairly B-efficient solution.

5 Conclusion

In this paper, we introduced a theoretical development of a new concept of solution of a multiobjective optimization problem. The concept of fair B-efficiency is obtained from fair rational preference relations on a certain cumulative ordered vector. We introduced a new multiobjective optimization problem and by seeking efficient solutions of this new problem, we found fairly B-efficient solutions of the original problem. Furthermore, we examined some properties of the set of fairly B-efficient solutions. These include sufficient conditions for existence, connectivity of the fairly B-efficient set, scalarization methods, and characterizations related to weighting problems.

References

1. Ehrgott, M. *Multicriteria optimization*, Springer Verlag, Berlin, 2005.

2. Geoffrion, A. M. *Proper efficiency and the theory of vector maximization*, Journal of Mathematical Analysis and Applications 22 (1968), 618-630.

3. Kostreva, M.M., and Ogryczak, W. *Linear optimization with multiple equitable criteria*, RAIRO Operations Research 33 (3) (1999), 275-297.

4. Kostreva, M.M., Ogryczak, W., and Wierzbicki, A. *Equitable aggregations in multiple criteria analysis*, European Journal of Operational Research 158 (2) (2004), 362-377.

5. Lorenz, M.O. *Methods of measuring the concentration of wealth*, American Statistical Association, New Series 70 (1905), 209-219.

6. Luc, D.T. *Theory of Vector Optimization*, Springer Verlag, Berlin, 1989.

7. Marshall, A.W., and Olkin, I. *Inequalities: Theory of Majorization and Its Applications*, Academic Press, New York, 1979.

8. Ogryczak, W. *Multiple criteria linear programming model for portfolio selection*, Annals of Operations Research 97 (2000), 143-162.

9. Ogryczak, W. *Inequality measures and equitable approaches to location problems*, European Journal of Operational Research 122 (2) (2000), 374-391.

10. Ogryczak, W., Luss, H., Pioro, M., Nace, D., and Tomaszewski, A. *Fair optimization and networks: a survey*, Journal of Applied Mathematics 2014 (2014), Article ID 612018, 25 pages.

11. Ogryczak, W., Wierzbicki, A., and Milewski, M. *A multi-criteria approach to fair and efficient bandwidth allocation*, Omega 36 (2008), 451-463.

12. Ogryczak, W., and Zawadzki, M. *Conditional median: a parametric solution concept for location problems*, Annals of Operations Research 110 (2002), 167-181.

13. Steuer, R.E. *Multiple Criteria Optimization Theory*, Computation and Applications, John Wiley and Sons, New York, 1986.

14. Viennet, R., Fonteix, C., and Marc, I. *Multicriteria optimization using a genetic algorithm for determining a Pareto set*, International Journal of Systems Science 27 (2) (1996), 255-260.

Iranian Journal of Numerical Analysis and Optimization
Vol. 7, No. 1, (2017), pp 65-82
DOI: 10.22067/ijnao.v7i1.52961

Efficient methods for goal square Weber location problem

J. Fathali* and A. Jamalian

Abstract

In this paper, we consider a special case of Weber location problem which we call goal location problem. The Weber location problem asks to find location of a point in the plane such that the sum of weighted distances between this point and n existing points is minimized. In the goal location problem each existing point P_i has a relevant radius r_i and it's ideal for us to locate a new facility on the distance r_i from P_i for $i = 1, ..., n$. Since in the most instances there does not exist the location of a new facility such that its distance to each point P_i be exactly equal to r_i. So we try to minimize the sum of the weighted square errors. We consider the case that the distances in the plane are measured by the Euclidean norm. We propose a Weiszfeld like algorithm for solving the problem and also we use two modifications of particle swarm optimization method for solving this problem. Finally the results of these algorithms are compared with results of BSSS algorithm.

Keywords: Location theory; Weiszfeld method; Particle swarm optimization.

1 Introduction

Location theory is one of the useful and interesting fields of operations research and has many applications in real life. Single facility location problems are fundamental problems in location theory in which n existing demand points are located at known distinct points $P_1, \cdots, P_n$, and a new facility should be located at a point X such that the sum of weighted distances from

*Corresponding author
Received 13 January 2016; revised 18 June 2016; accepted 14 September 2016
J. Fathali
Department of Mathematics, Shahrood University of Technology, University Blvd., Shahrood, Iran. email: fathali@shahroodut.ac.ir

A. Jamalian
Department of Computer Science, Faculty of Mathematical Sciences, University of Guilan, Rasht, Iran. e-mail: a.jamalian.math@gmail.com

it to demand points is minimized. Let $d(X, P_i)$ represent the distance traveled per trip between points X and P_i, the weight w_i represents the product of cost per unit distance traveled and number of trips made per year between the new facility and existing demand point i and some times shows value of demand in this point. The total cost is given by:

$$\min F(X) = \sum_{i=1}^{n} w_i d(X, P_i). \tag{1}$$

The Euclidean problem is obviously referred to as the Steiner-Weber problem or the general Fermat problem, and has an extraordinary longevity (see [8]). In fact, a version of the problem for the case $n = 3$ and $w_i = 1$ for $i = 1, 2, 3$, was posed purely as a problem in geometry, by Fermat early in the seventeenth century, and was solved by Toricelli prior to 1640 (see [16]). The problem was studied by Steiner, a Swiss mathematician, in the nineteenth century, and by Weber, a German economist, early in the twentieth century. Single facility location with Euclidean distance is also called Weber problem and an iterative procedure was proposed for solving this problem by Weiszfeld [24] in 1937 and rediscovered by Kuhn [17] in 1962. This algorithm is widely used because of its simplicity and effectiveness. The algorithm can be generalized to other location problems where the cost is a function of the Euclidean distance rather than just being proportional to the distance (see e.g. [10]). Some other cases of Weber location problem can be find in [4,5,23]).

Fathali et al. [12], considered a special case of Weber problem in which a specified radius for every demand point is considered and the distance between the new facility and the demand point P_i is equal to the corresponding radius, r_i. However since in the real instances rarely exist the location of a new facility such that its distance to each point p_i is exactly r_i. So they tried to minimize the sum of the weighted square errors. We call this problem as Goal Square Weber Location Problem (GSWLP). Jamalian and Fathali [14] presented a linear model for the location problem with minimum absolute error on the block norms.

In what follows we explain the GSWLP in Section 2. Model properties and results are stated in Section 3. In this section main results and a proposed algorithm for solving problem are given. Because the objective function may have many local optima, we also use particle swarm optimization method for solving this problem. In Section 4, the particle swarm optimization method is explained. In Section 5, the computational results of comparing four algorithms WLA, PSO, PSOC and BSSS for this problem are given.

2 Goal Square Weber location problem

In single facility location problem, we want to find location of a new facility such that the sum of weighted distances from the facility to all demand points is minimized. In Fathali et al. [12] a radius, r_i, and value of demand, w_i correspond to every point P_i is considered. We want to find the location of a new facility such that the distances between it and the demand points $P_i,\ i = 1, \ldots, n$ is equal to r_i. Since this situation occurs in real applications rarely we try to minimize sum of weighted squared error.

In fact, in this problem for every point a goal distance is considered and we want to minimize the error of satisfaction location of facility in these goals. It is desirable for us that new facility be located on circle of correspond radius and center of demand point. For an example of this problem, if $r_i = r$, $i = 1, 2, \cdots, n$ and demand points be located on a circle of radius r, the optimal point is center of circle. In the case $r_i = r$, $i = 1, 2, \cdots, n$ the problem converted to the square Weber problem, so we call this problem as Goal Square Weber Problem.

The problem studied in this paper is different from the covering location problem (CLP). Two important covering models are the maximal and minimal covering location problems which are suitable for siting desirable and undesirable facilities, respectively (see e.g., [3,6]). In both of these problems a facility covers a demand point if the demand point lies within a pre-specified coverage radius. The objective is to locate a number of facilities so as to minimize or maximize the total coverage. There are some differences between CLP and GSWLP. Firstly, CLP is a multifacility location problem, while in the GSWLP we want to find the location of a single facility. Secondly, in CLP a radius is associated with facilities, in contrast to the GSWLP, where a radius r_i is associated with demand point P_i. Thirdly, in the maximal CLP, it is desirable if the distance between the demand points and facilities be less than or equal to a pre-determined radius and in the minimal CLP this distance should be more than or equal to a given radius, while in GSWLP it is desirable if the new facility exactly be in the given distances of the demand points.

Determination of the location of a company in the vicinities of some cities such that the setting up costs and the transportation costs and minimized can be an application of this problem as mentioned in [12] . Suppose that the cost of establishing a facility in the regions that are farther than a given distance r_i from city r_i is very low. On the other hand a move away from a city causes the transportation costs to increase. Therefore a tradeoff between the setting up costs and the transportation costs seems to be reasonable. GSWLP also has some other applications for locating facilities that are either desirable or undesirable. In these cases we want the facility not be close than a specified distance to facility centers, because of its undesirability, on the other hand, if the facility be so far from the facility centers, cost of providing security, human forces, transportation installation, and other costs will increase.

3 Model Properties

Let $P_i = (a_i, b_i)$ for $i = 1, \cdots, n$ be given points in the plane. Formulation of GSWLP is as follows:

$$\min F(X) = \sum_{i=1}^{n} w_i \left(d(X, P_i) - r_i\right)^2 \tag{2}$$

where $X = (x, y)$, $d(X, P_i) = \sqrt{(x - a_i)^2 + (y - b_i)^2}$, and r_i and w_i are corresponding radius and weight of demand point P_i, respectively. Fathali et al. [12] show that the objective function of model (2), is non convex and optimal solution is in extended rectangular hull of demand points. Extended rectangular hull of demand points is the smallest rectangle which contains all the demand points with their circles. We present a new proof by Farkas lemma in the Theorem 1. The advantage for our new proof is that in this proof we show there is an improvement direction from any point which is out of rectangular hull.

Farkas lemma is one of the theorems of alternatives a good discussion of which is given by Mangasarian [19]. It can also be found in Dantzig and Thapa [9] and Bazaraa et al. [2], among others.

Lemma 1. *[**Farkas Lemma (1902)**] For a given matrix A and a vector C, either $AX \geq 0$ and $CX < 0$ has a solution, or there exists a vector like W such that $WA = C$ and $W \geq 0$.*

Theorem 1. *The optimal solution of Equation* (2) *is in the extended rectangular hull of demand points.*

Proof. First we define four points as follow:

$$\begin{aligned} RH_1 &= \min\{a_i - r_i | i = 1, 2, \cdots, n\} \\ RH_2 &= \max\{b_i + r_i | i = 1, 2, \cdots, n\} \\ RH_3 &= \max\{a_i + r_i | i = 1, 2, \cdots, n\} \\ RH_4 &= \min\{b_i - r_i | i = 1, 2, \cdots, n\}. \end{aligned}$$

We show that the optimal point is in the rectangular hull of these four points. By contradiction, we suppose that $X_b = (x_b, y_b)^T$ is the optimal point of model (2) and is out of rectangle. Therefor it is out of cone generated either by $(RH_1 - RH_3, 0)$ and $(0, RH_4 - RH_2)$ or $(RH_3 - RH_1, 0)$ and $(0, RH_2 - RH_4)$. **Case 1**: suppose that the optimal point is out of cone generated by $(RH_1 - RH_3, 0)$ and $(0, RH_4 - RH_2)$. So the vector $X_b - (RH_3, RH_2)^T$ is not in the cone, i.e. following system has not solution:

$$\begin{cases} v_1(RH_1 - RH_3, 0) + v_2(0, RH_4 - RH_2) = (x_b - RH_3, y_b - RH_2) \\ v_1, v_2 \geq 0. \end{cases}$$

By Farkas' lemma, the following system have solution:

$$\begin{cases} \begin{bmatrix} RH_1 - RH_3 & 0 \\ 0 & RH_4 - RH_2 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \geq 0 \\ \\ [x_b - RH_3, y_b - RH_2] \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} < 0. \end{cases} \tag{3}$$

So by the first inequality of (3) we have:

$$\begin{cases} (RH_1 - RH_3)u_1 \geq 0 \\ (RH_4 - RH_2)u_2 \geq 0 \end{cases}$$

since $RH_1 \leq RH_3$ and $RH_4 \leq RH_2$ therefore $u_1 \leq 0$ and $u_2 \leq 0$. And by the second inequality of (3) we obtain

$$(x_b - RH_3)u_1 + (y_b - RH_2)u_2 < 0.$$

Moreover $a_i \leq RH_3$ and $b_i \leq RH_2$ for $i = 1, 2, \cdots, n$ so
$(x_b - a_i)u_1 + (y_b - b_i)u_2 \leq (x_b - RH_3)u_1 + (y_b - RH_2)u_2 < 0.$

Now, if $x_b - RH_3 < 0$ and $y_b - RH_2 > 0$ then let $U = \begin{pmatrix} 0 \\ u_2 \end{pmatrix}$ and if $x_b - RH_3 > 0$ and $y_b - RH_2 < 0$ then let $U = \begin{pmatrix} u_1 \\ 0 \end{pmatrix}$ and if $x_b - RH_3 > 0$ and $y_b - RH_2 > 0$ then let $U = \begin{pmatrix} u_1 \\ u_2 \end{pmatrix}$. So in each case

$$\nabla d(X_b, P_i).U = \frac{1}{d(X_b, P_i)}(x_b - a_i, y_b - b_i).U < 0.$$

Therefore

$$\begin{aligned} \nabla F(X_b).U &= 2\sum_{i=1}^{n} w_i \left(1 - \frac{r_i}{d(X_b, P_i)}\right)(x_b - a_i)u_1 \\ &\quad + 2\sum_{i=1}^{n} w_i \left(1 - \frac{r_i}{d(X_b, P_i)}\right)(y_b - b_i)u_2 \\ &= 2\sum_{i=1}^{n} w_i \left(1 - \frac{r_i}{d(X_b, P_i)}\right)[(x_b - a_i)u_1 + (y_b - b_i)u_2] < 0. \end{aligned}$$

This means that U is a descent direction and X_b can not be optimal point. Note that in the last equality statement $1 - \frac{r_i}{d(X_b,P_i)}$ is positive for all i, because X_b is out of extended rectangular hull and $d(X_b, P_i) \geq r_i$ so $\frac{r_i}{d(X_b,P_i)} \leq 1$.

Case 2: suppose that optimal point is out of cone generated by $(RH_3 - RH_1, 0)$ and $(0, RH_2 - RH_4)$. Proof of this case is the same as Case 1. So the optimal point is in rectangular hull generated by RH_1, RH_2, RH_3 and RH_4. □

4 Weiszfeld-like algorithm

In this section, we propose an iterative algorithm for solving GSWLP. Our algorithm is based on the Weiszfeld method. The original Weiszfeld algorithm is an iterative method for solving the following Weber problem.

$$MinF(X) = \sum_{i=1}^{n} w_i \sqrt{(x - a_i)^2 + (y - b_i)^2}. \tag{4}$$

From the necessary condition of optimality $\frac{\partial F(X^*)}{\partial X} = 0$ for this problem, the Weiszfeld method starts with an initial solution and try to find optimal solution by the following iterative relations;

$$x_{k+1} = \frac{\sum_{i=1}^{n} \frac{w_i a_i}{\sqrt{(x_k - a_i)^2 + (y_k - b_i)^2}}}{\sum_{i=1}^{n} \frac{w_i}{\sqrt{(x_k - a_i)^2 + (y_k - b_i)^2}}}$$

$$y_{k+1} = \frac{\sum_{i=1}^{n} \frac{w_i b_i}{\sqrt{(x_k - a_i)^2 + (y_k - b_i)^2}}}{\sum_{i=1}^{n} \frac{w_i}{\sqrt{(x_k - a_i)^2 + (y_k - b_i)^2}}}.$$

Now consider the GSWLP. The same as Weiszfeld method from the necessary optimality condition to obtain a candidate for optimal solution we set

$$\frac{\partial F}{\partial X} = 2 \sum_{i=1}^{n} w_i \left(1 - \frac{r_i}{d(X, P_i)}\right) (X - P_i) = 0 \tag{5}$$

therefore

$$X = \frac{\sum_{i=1}^{n} w_i \left(1 - \frac{r_i}{d(X,P_i)}\right) P_i}{\sum_{i=1}^{n} w_i \left(1 - \frac{r_i}{d(X,P_i)}\right)}. \tag{6}$$

Like in the Weiszfeld algorithm, there is a possibility that during the iterations, $d(X, P_i)$ is zero for one of the points, and then the ratio is infinite and the process cannot be continued. For practical purposes, the whole issue

of singular points may be avoided by replacing each distance in (6) by its hyperbolic approximation (e.g., see [18]) given by:

$$X = \frac{\sum_{i=1}^{n} w_i \left(1 - \frac{r_i}{d(X,P_i)+\varepsilon}\right) P_i}{\sum_{i=1}^{n} w_i \left(1 - \frac{r_i}{d(X,P_i)+\varepsilon}\right)} \tag{7}$$

where $\varepsilon > 0$ is a smoothing constant. Therefor we can use the following iterative method.

Algorithm [WLA].
Initialization:

1. Choose an initial solution, X_0 and set $k = 0$.

Iteration step:
While not stopping criterion do

1.
$$x_{k+1} = \frac{\sum_{i=1}^{n} w_i \left(1 - \frac{r_i}{\sqrt{(x_k-a_i)^2+(y_k-b_i)^2}+\varepsilon}\right) a_i}{\sum_{i=1}^{n} w_i \left(1 - \frac{r_i}{\sqrt{(x_k-a_i)^2+(y_k-b_i)^2}+\varepsilon}\right)}$$
$$y_{k+1} = \frac{\sum_{i=1}^{n} w_i \left(1 - \frac{r_i}{\sqrt{(x_k-a_i)^2+(y_k-b_i)^2}+\varepsilon}\right) b_i}{\sum_{i=1}^{n} w_i \left(1 - \frac{r_i}{\sqrt{(x_k-a_i)^2+(y_k-b_i)^2}+\varepsilon}\right)}$$

2. k:=k+1

endwhile

The stopping criterion rule is usually one of the following criteria: maximum number of iterations or reaching a tolerance.

Now we discuss the convergency of WLA. From Equation (6) if we consider the right hand side of this equation as X_k and the left hand side of it as X_{k+1}, so we have

$$X_{k+1} = \frac{\sum_{i=1}^{n} w_i \left(1 - \frac{r_i}{d(X_k,P_i)}\right) P_i}{\sum_{i=1}^{n} w_i \left(1 - \frac{r_i}{d(X_k,P_i)}\right)}, \tag{8}$$

then by choosing an initial solution, X_0, for $k = 1, 2, \cdots$ we have

$$X_{k+1} = X_k - \frac{1}{\sum_{i=1}^{n} w_i \left(1 - \frac{r_i}{d(X_k,P_i)}\right)} \left[\sum_{i=1}^{n} w_i \left(1 - \frac{r_i}{d(X_k,P_i)}\right)(X_k - P_i)\right]. \tag{9}$$

Let us assume that

$$d_k = -\frac{1}{2}\sum_{i=1}^{n} w_i \left(1 - \frac{r_i}{d(X_k,P_i)}\right)(X_k - P_i) \tag{10}$$

it is clear that d_k is a descent direction, i.e. $d_k^T.\nabla F(X_k) < 0$, so the Equation (9) is a gradient method and the sequence $\{F(X_k)\}$ is non increasing. Let

$$h(X) = X - \frac{1}{2\sum_{i=1}^{n} w_i \left(1 - \frac{r_i}{d(X,P_i)}\right)} \frac{\partial F(X)}{\partial X}. \tag{11}$$

The function $F(X)$ is two times differentiable and $h(X)$ in $X \neq P_i$, $i = 1, \cdots, n$, is continuous.

Lemma 2. *The sequence generated by Equation* (8) *is in affine hull of the* $P_i, i = 1, 2, \cdots, n$.

Proof. In each iteration, we have $X_{k+1} = \frac{\sum_{i=1}^{n} \alpha_i P_i}{\sum_{i=1}^{n} \alpha_i}$, in which $\alpha_i = \sum_{i=1}^{n} w_i \left(1 - \frac{r_i}{d(X_k,P_i)}\right)$. Let $\gamma_i = \frac{\alpha_i}{\sum_{i=1}^{n} \alpha_i}$, we have $\gamma_i \in [-1, 1]$ and $\sum_{i=1}^{n} \gamma_i = 1$. From Equation (8) the sequence $\{X_k\}$ is a affine combination of demand points. So the sequence $\{X_k\}$ is in affine hull of $P_i, i = 1, 2, \cdots n$. □

Lemma 3. *If* $X_k = X_{k+1}$ *then* $\frac{\partial F(X_k)}{\partial X} = 0$.

Proof. From the Equation (9), we have $X_{k+1} = h(X_k) = X_k - \frac{1}{2}\lambda_k \frac{\partial F(X_k)}{\partial X}$ where $\lambda_k = \frac{1}{\sum_{i=1}^{n} w_i \left(1 - \frac{r_i}{d(X_k,P_i)}\right)}$. If $X_{k+1} = X_k$ then $-\frac{1}{2}\lambda_k \frac{\partial F(X_k)}{\partial X} = 0$, i.e. $\frac{\partial F(X_k)}{\partial X} = 0$. □

Lemma 4. *If* $X_k \neq X_{k+1}$ *then* $F(X_{k+1}) < F(X_k)$, *and If* $F(X_k) = F(X_{k+1})$ *then* $X_k = X_{k+1}$.

Proof. From the Equation (9), if $X_k \neq X_{k+1}$, we have $\frac{\partial F(X_k)}{\partial X} \neq 0$. Let $D = -\frac{\partial F(X_k)}{\partial X}$ so $D^T.\nabla F(X_k) < 0$ i.e. $F(X_{k+1}) < F(X_k)$.
To show the second part of lemma if $F(X_k) = F(X_{k+1})$ then since for every k, the direction d_k is non increasing we have $X_k = X_{k+1}$. □

Note that the direction d_k in iteration k, is a descent direction, so the sequence $\{F(X_k)\}$ is non increasing. Since this sequence is bounded from below by zero, the limit $\lim_{k\to\infty} F(X_k)$ exists.

Theorem 2. *The sequence* $\{X_k\}$ *converges to a local minimum or a saddle point of function* $F(X)$.

Proof. By Lemma 2, the sequence $\{X_k\}$ is in affine hull of demand points and since by explanation of Equation (9), d_k^T are descent direction so the sequence $\{X_k\}$ is converged to a point like $\hat{X}$. Since $\lim_{k\to\infty} F(X_k)$ exists, $\lim_{k\to\infty} F(X_{k+1}) - F(X_k) = 0$ and $F(h(\hat{X})) = F(\hat{X})$. By Lemma 4, we have $h(\hat{X}) = \hat{X}$. If $X_{k+1} \neq X_k$ then the inequality $F(X_{k+1}) \leq F(X_k)$ holds strictly. From continuity of function $h(X)$, if X_k tends to $\hat{X}$ then X_{k+1} will tend to $h(\hat{X}) = \hat{X}$ and $\{X_k\}$ converges to $\hat{X}$. According to Lemma 3, $\frac{\partial F(\hat{X})}{\partial X} = 0$ and since $\{F(X_k)\}$ is non increasing and bounded from below by zero, the point $\hat{X}$ con not be local maxima, in other word, it is a local minimum or saddle point of function $F(X)$. □

We presented an analytic method for solving GSWLP that do not guarantee converging to a minimum and may fall in saddle point or local minima which is no minimum. Since the objective function of GSWLP is non convex, the local minimum may not be global (see Example 1). So we should use a global optimization method. In the next section we apply a particle swarm optimization algorithm to find the best solution of problem.

Example 1. Consider the points, their weights and their radiuses that are given in Table 1. This problem is solved in [12] by big square small square method and the value of optimal solution is 182. But if we solve this problem by WLA we will see this method convergence to a point which its value of objective function is 209.

Table 1: Data for 18 points problem.

(a, b, w, r)	(a, b, w, r)	(a, b, w, r)
$(1, 2, 3, 2)$	$(4, 4, 1, 2)$	$(7, 1, 2, 1)$
$(1, 3, 2, 2)$	$(4, 9, 2, 1)$	$(7, 2, 3, 2)$
$(2, 5, 1, 3)$	$(5, 3, 2, 2)$	$(8, 5, 1, 2)$
$(3, 6, 3, 2)$	$(5, 5, 1, 2)$	$(8, 8, 3, 1)$
$(4, 8, 2, 2)$	$(6, 6, 3, 3)$	$(9, 7, 3, 2)$
$(4, 1, 3, 1)$	$(6, 3, 3, 1)$	$(9, 6, 2, 3)$

Note that this is an special example that WLA traps in a local minima. This case rarely happens. We examined many other test problems but we could not find any example with this property. Also as we shall see in the computational experiments section, in non of the examined test problems in this section this case dose not happen.

5 Particle Swarm Optimization

Nature has always been an inspiration for researchers in designing solutions for many problems. Particle Swarm Optimization (PSO) is a meta-heuristic algorithm that is inspired of the behaviors of social models like bird flocking or fish schooling. It is introduced by Eberhart and Kennedy [11], as an optimization method for continuous nonlinear functions. Later, it has been applied to wide range of problems due to its conceptual and implementation simplicity (see e.g. [1, 15]).

Particle swarm optimization is a stochastic optimization technique based on individual improvement, social cooperation and competition in a population. PSO is famous for its fast convergence to a solution close to the optimal, by balancing the local search and global search i.e., exploitation and exploration, respectively. PSO is a population based method in which a swarm includes several individuals called particles. Each particle has a position and velocity vector. Particles are initially positioned randomly. The position of a particle is a candidate solution and it is updated at each iteration by using the particle's current velocity. The velocity of a particle is updated by using the particle's inertia weight as well as the social interaction (its tendency to follow the best direction experienced by the group) and personal experience (the best direction experienced by itself) of the particle. In the course of several iterations, particles make use of this experience and are supposed to move towards the optimum position.

The stopping criterion rule is usually one of the following criteria: maximum number of iterations, the maximum CPU time, the number of successive best objective function values without any improvements or reaching a predetermined tolerance. During the update step, for each particle, the velocity and position vector of the particle at iteration $t+1$ are calculated by using the Equations (12) and (13).

$$v_{i,j}^{t+1} = wv_{i,j}^{t} + c_1 r_1 \left(pbest_{i,j}^{t} - x_{i,j}^{t}\right) + c_2 r_2 \left(gbest_{j}^{t} - x_{i,j}^{t}\right) \tag{12}$$

$$x_{i,j}^{t+1} = x_{i,j}^{t} + v_{i,j}^{t+1}. \tag{13}$$

In these equations, $v_{i,j}^{t}$ and $x_{i,j}^{t}$ are the velocity and position of the j^{th} dimension of the i^{th} particle at iteration t, respectively. The parameters c_1 and c_2 are coefficients of learning factors, which are the weights of contributions of personal experience and social interaction. The stochastic behavior of PSO is achieved by r_1 and r_2 which are random numbers, generally in $(0, 1)$. The parameter w is the inertia weight which is the balancing factor between exploration and exploitation. Small inertia weight facilitates more exploitation while large inertia weight enables more exploration.

After the update step, the fitness function value is calculated for each particle based on its position. The local best position *pbest* of each particle and

the global best position *gbest* are updated using these fitness values as follow:

$$pbest_i^{t+1} = \begin{cases} pbest_i^t & F(x_i^{t+1}) \geq F(pbest_i^t) \\ x_i^{t+1} & F(x_i^{t+1}) > F(pbest_i^t) \end{cases} \tag{14}$$

$$gbest^{t+1} = argmin_{i\in\{1,2,\cdots,n\}} F(pbest_i^{t+1}). \tag{15}$$

These ideas lead to the following algorithm.

Algorithm [PSO].

Initialization:

1. Initialize iteration counter.
2. Initialize N random position of particles and store them in S.
3. Initialize N random velocities and store them in V.
4. Initialize N *pbest* and store them in P.
5. Set $gbest^t$ equal the best *pbest* in P.

Iteration step:
While not stopping criterion do

1. For each i^{th} particle:
   (a) Update V: Calculate velocity v_i^{t+1} using (12)
   (b) Update S: Calculate position x_i^{t+1} using (13)
   (c) Update P: Calculate position $pbest_i^{t+1}$ using (14)
2. Update gbest: $gbest^{t+1} = \arg\min_{i\in\{1,2,\cdots,N\}} \left\{f(pbest_i^{t+1})\right\}$
3. t:=t+1

endwhile

Where V is the set of velocities and P is the set of pbests. To apply this algorithm for Goal Square Weber Location Problem we set S the region of optimal solution, i.e. extended rectangular hull of points and each particle as a solution. In this algorithm first we choose N random positions and velocities for each particle. From rules of particle swarm optimization we obtain the pbest and gbest of swarm. The algorithm with updating variables continues until termination conditions reached.

Many modification for PSO algorithm has been proposed to improve the performance of this algorithm. In standard PSO algorithm, the information

of individual best and global best are shared by next generation particles. Shi and Eberhart [21] use dynamic inertia weight that decreases according to iterative generation increasing. The weight w in (12) follows the relation $w_{t+1} = \alpha w_t$ where $0 < \alpha < 1$. A large inertia weight facilitates global exploration and a smaller inertia weight tends to facilitate local exploration to search the local neighborhoods in the current search area more effectively.

Clerc and Kennedy [7] present another good modification of PSO. They consider the form of a constriction coefficient K which controls all the three components in velocity update rule. This has an effect of reducing the velocity as the search progress. In this modification, the velocity update is given as:

$$v_{i,j}^{t+1} = K\left(v_{i,j}^{t} + c_1 r_1\left(pbest_{i,j}^{t} - x_{i,j}^{t}\right) + c_2 r_2\left(gbest_{j}^{t} - x_{i,j}^{t}\right)\right) \qquad (16)$$

where

$$K = \frac{2}{\mid 2 - \phi - \sqrt{\phi^2 - 4\phi} \mid} \qquad (17)$$

where $\phi = c_1 + c_2 > 4$. This version is referred to as PSO with constriction or PSOC. We apply the original PSO and PSOC for solving GSWLP.

In the next section we compare the WLA with PSO, PSO with constriction (PSOC) and big square small square (BSSS) method. Big Square Small Square (BSSS) method is a geometrical branch and bound algorithm originally suggested by Hansen et al. [13] for solving an obnoxious facility location problem. The idea is to divide the plane into regions (squares), over each of which a lower (upper) bound for the problem is found. If the objective over a given square is worse than an existing upper (lower) bound, then that square is fathomed. The procedure continues until a prescribed tolerance is achieved. This method also applied by some authors such as Plastria [20], Skriver and Andersen [22], Zaferanieh et al. [25] and Fathali et al. [12] for solving location problems.

6 Computational results

The proposed algorithms were tested on forty test problems with 100 to 3000 points that are generated randomly in MATLAB. In these problems demand point's coordinates, weights and radiuses, all are positive numbers and generated randomly in $[1, 60]$, $[1, 3]$ and $[1, 10]$, respectively.

We compare the Weiszfeld like algorithm (WLA), PSOC and PSO with BSSS method of Fathali et al. [12]. Algorithms are coded in MATLAB software and run on a Laptop with Core(TM)2 CPU with 2.00 GHz processor and 1 GB of RAM. In the PSO and PSOC algorithms we should generate randomly N initial position of particles and velocities. Therefore in different executions of these methods we may find different solutions, so we run 5

times this two methods for all problems and report the average results. In all methods the iteration step of the algorithm is repeated until the tolerance is achieved. The tolerance for all examples was taken to be 0.01.

In our experiments with different values of c_1 and c_2 ($c_1, c_2 \in [1, 3]$) we found the best values to be $c_1 = c_2 = 2.1$. The inertia weight is started with $w = 1.5$ and decrease in each iteration such that reach 0.2 at the end of algorithm. The population size is set $N = 50$ for problems with $n = 100, \cdots, 500$ and $N = 100$ for remaining problems. We also examined larger numbers of population size. Our tests showed that larger population size slow down the algorithm but did not lead to better solutions.

Table 2 shows the computational results. The column with the heading *%Error* indicates the relative error; i.e.,

$$|\frac{f - f_{best}}{f_{best}}| \cdot 1000$$

where f_{best} is the best solution obtained by the four methods and f is the value of objective function obtained by each method.

Note that the BSSS method continues until the side of the sub-square is less than a given tolerance. So the solutions which obtained by this method is near optimal with the given tolerance. By this method we can also compute a lower bound for the objective for each sub-square. The column with the heading LB in Table 2 indicates the lower bound for the objective for the best sub-square. This sub-square contains optimal solution. Also the column with the heading *gap* shows the relative gap between the lower bound and the best solution, i.e.,

$$|\frac{f_{best} - LB}{f_{best}}| \cdot 1000.$$

We observe the largest relative gap is 5.79×10^{-4}, which happens for the test problem number 23.

Table 3 contains the average total CPU times for test problems. The CPU times of WLA are less than other methods in all cases. However the other methods could find better solution in some test problems.

To compare speed of presented methods one may consider the time complexity of these methods. However since these methods are improvement methods then with more iterations we may obtain better solutions. Therefore in the following we give their time complexity per iteration.

The main step of WLA is computing x and y in each iteration. Since we need $23n$ operations for each of x and y, obviously, the time complexity per iteration is $O(n)$. Where n is number of existing points.

In step 1 of each iteration of PSO, we should calculate velocity v, position of x and *pbest* according to (12), (13) and (14), respectively. They need $O(nN)$ operations. Where N is the population size. Step 2 of Iteration step

Table 2: The objective functions and errors

		objective function					*%Error*				
NO.	n	LB	WLA	BSSS	PSO	PSOC	WLA	BSSS	PSO	PSOC	gap
1	100	59599	59617	59618	59617	59617	0.000	0.017	0.000	0.000	0.302
2	100	68958	68995	68995	68995	68995	0.000	0.000	0.000	0.000	0.536
3	100	82265	82287	82287	82288	82287	0.000	0.000	0.012	0.000	0.267
4	100	78156	78232	78180	78182	78180	0.665	0.000	0.026	0.000	0.435
5	100	76104	76145	76145	76145	76145	0.000	0.000	0.000	0.000	0.539
6	200	121305	121340	121341	121341	121340	0.000	0.008	0.008	0.000	0.288
7	200	166160	166254	166254	166255	166254	0.000	0.000	0.006	0.000	0.565
8	200	159632	159720	159720	159720	159721	0.000	0.000	0.000	0.006	0.551
9	200	162253	162312	162312	162312	162312	0.000	0.000	0.000	0.000	0.363
10	200	160276	160324	160324	160325	160324	0.000	0.000	0.006	0.000	0.299
11	300	233660	233765	233766	233765	233766	0.000	0.004	0.000	0.004	0.449
12	300	234183	234314	234314	234314	234315	0.000	0.000	0.000	0.004	0.559
13	300	245908	246045	246045	246046	246045	0.000	0.000	0.004	0.000	0.557
14	300	241698	241761	241761	241761	241761	0.000	0.000	0.000	0.000	0.267
15	300	235575	235674	235621	235642	235621	0.225	0.000	0.089	0.000	0.195
16	400	309670	309777	309777	309777	309776	0.003	0.003	0.003	0.000	0.342
17	400	337358	337543	337544	337543	337543	0.000	0.003	0.000	0.000	0.548
18	400	319847	320030	320030	320031	320032	0.000	0.000	0.003	0.003	0.493
19	400	326612	326734	326735	326734	326735	0.000	0.003	0.000	0.003	0.373
20	400	318651	318783	318783	318783	318783	0.000	0.000	0.000	0.000	0.389
21	500	394201	394364	394364	394364	394364	0.000	0.000	0.000	0.000	0.418
22	500	391482	391654	391598	391597	391598	0.143	0.003	0.000	0.003	0.294
23	500	407515	407751	407752	407751	407751	0.000	0.003	0.000	0.000	0.579
24	500	412614	412763	412763	412763	412763	0.000	0.000	0.000	0.000	0.361
25	500	402198	402348	402277	402277	402277	0.176	0.000	0.000	0.000	0.196
26	1000	786520	786731	786732	786732	786732	0.000	0.001	0.001	0.001	0.268
27	1000	796696	797148	797149	797148	797148	0.000	0.001	0.000	0.000	0.191
28	1000	812939	813407	813406	813407	813406	0.001	0.000	0.001	0.000	0.575
29	1000	805501	805634	805634	805634	805634	0.000	0.000	0.000	0.000	0.165
30	1000	802280	802431	802432	802431	802432	0.000	0.001	0.000	0.001	0.313
31	2000	1465017	1465731	1465705	1465718	1465705	0.018	0.000	0.009	0.000	0.469
32	2000	1582076	1582976	1582977	1582977	1582977	0.000	0.001	0.001	0.001	0.569
33	2000	1633068	1634000	1634000	1634000	1634000	0.000	0.000	0.000	0.000	0.570
34	2000	1563967	1564236	1564185	1564186	1564185	0.001	0.000	0.001	0.000	0.139
35	2000	1607103	1607640	1607640	1607641	1607640	0.000	0.000	0.001	0.000	0.334
36	3000	2297871	2298872	2298873	2298872	2298873	0.000	0.001	0.000	0.001	0.435
37	3000	2362318	2363663	2363663	2363663	2363663	0.000	0.000	0.000	0.000	0.569
38	3000	2403658	2405041	2405040	2405040	2405040	0.001	0.000	0.000	0.000	0.575
39	3000	2387987	2388745	2388746	2388745	2388753	0.000	0.001	0.000	0.003	0.317
40	3000	2324214	2325395	2325396	2325396	2325395	0.000	0.001	0.001	0.000	0.508
Total							1.233	0.051	0.172	0.030	

Table 3: The CPU Time of PSOC, PSO, WLA and BSSS algorithms

n	WLA	BSSS	PSO	PSOC
100	0.002	4.204	0.630	0.778
200	0.004	7.765	1.145	1.318
300	0.005	11.363	1.686	1.852
400	0.007	15.061	2.192	2.920
500	0.008	18.205	2.709	3.470
1000	0.014	83.607	5.348	5.929
2000	0.027	389.972	10.570	12.346
3000	0.040	576.231	15.750	17.230

Table 4: The objective functions and errors in a constant time

			$objective function$			$\%Error$		
test#	n	time	BSSS	PSO	PSOC	BSSS	PSO	PSOC
1	100	0.5	59630	59617	59617	0.218	0.000	0.000
2	100	0.5	68997	68996	68995	0.028	0.014	0.000
3	100	0.5	82290	82289	82287	0.036	0.024	0.000
4	100	0.5	78241	78205	78202	0.780	0.320	0.281
5	100	0.5	76147	76146	76145	0.026	0.013	0.000
6	200	1.0	121355	121345	121343	0.124	0.041	0.025
7	200	1.0	166799	166263	166260	3.278	0.054	0.036
8	200	1.0	160699	159771	159730	6.129	0.319	0.063
9	200	1.0	162330	162318	162314	0.111	0.037	0.012
10	200	1.0	160345	160325	160324	0.131	0.006	0.000
11	300	1.5	233809	233766	233767	0.188	0.004	0.008
12	300	1.5	234316	234316	234316	0.008	0.008	0.008
13	300	1.5	246072	246047	246045	0.110	0.008	0.000
14	300	1.5	241763	241770	241761	0.008	0.037	0.000
15	300	1.5	235704	235664	235651	0.352	0.182	0.127
16	400	2.0	309790	309779	309780	0.039	0.009	0.012
17	400	2.0	337568	337543	337543	0.075	0.000	0.000
18	400	2.0	320116	320090	320076	0.258	0.180	0.138
19	400	2.0	326814	326784	326784	0.248	0.153	0.153
20	400	2.0	318853	318783	318783	0.220	0.000	0.000
21	500	2.5	394384	394366	394365	0.061	0.005	0.003
22	500	2.5	391755	391598	391599	0.403	0.003	0.005
23	500	2.5	407801	407751	407751	0.151	0.000	0.000
24	500	2.5	412833	412772	412763	0.211	0.027	0.000
25	500	2.5	402451	402279	402280	0.433	0.006	0.009
26	1000	5.0	786851	786742	786733	0.153	0.011	0.002
27	1000	5.0	797153	797149	797149	0.004	0.001	0.001
28	1000	5.0	813407	813410	813407	0.001	0.004	0.001
29	1000	5.0	805710	805642	805639	0.094	0.008	0.005
30	1000	5.0	802671	802431	802433	0.299	0.000	0.002
31	2000	10.0	1465978	1465720	1465711	0.186	0.010	0.004
32	2000	10.0	1583288	1582977	1582977	0.197	0.001	0.001
33	2000	10.0	1634121	1634004	1634000	0.121	0.004	0.000
34	2000	10.0	1564480	1564186	1564186	0.189	0.001	0.001
35	2000	10.0	1607871	1607665	1607660	0.144	0.025	0.020
36	3000	15.0	2299001	2298876	2298874	0.056	0.004	0.002
37	3000	15.0	2363820	2363663	2363663	0.066	0.000	0.000
38	3000	15.0	2405131	2405041	2405041	0.037	0.001	0.000
39	3000	15.0	2388876	2388754	2388759	0.055	0.004	0.006
40	3000	15.0	2325480	2325396	2325395	0.037	0.001	0.000
			Total			15.265	1.525	1.04

calculate *gbest* which needs $O(N)$ operations. Therefor per iteration of PSO has $O(nN)$ time complexity.

PSOC is the same as PSO just we need calculate velocity v by (16). It also need $O(nN)$ operations. Therefor the time complexity of PSOC is $O(nN)$ for per iteration.

Due to [12] in each iteration of BSSS we should calculate the objective function for a given point in a square. This take $O(n^2)$ time, therefor each iteration of BSSS has $O(n^2)$ time complexity.

To compare the algorithms on equal terms, we allow the BSSS, PSO and PSOC algorithms run during the same time. Table 4 contains computational results for this case. Since WLA is very fast, its results for the given times in Table 4 is the same as Table 2.

7 Summary and conclusion

In this paper we studied a new version of the single facility location problem in which we want to find location of a new facility such that the sum of weighted squared errors is minimized. In general, this problem is non convex and we

showed that optimal solution of problem is in extended rectangular hull of demand points. We proposed two approaches for solving this problem. An iterative Weiszfeld-like procedure and two modifications of PSO algorithm. We compared the results with those obtained by BSSS algorithm. It was shown that for almost all problems the PSOC outperforms the other three approaches.

References

1. Ali, M.M. and Kaelo, P. *Improved particle swarm algorithm for global optimization,* Applied Mathematics and Computation, 2008, 196: 578-593.

2. Bazaraa, M.S., Sherali, H. and Jarvis, J. *Linear Programming and Network Flows*, Third ed., John Wiley & Sons; 2005.

3. Berman, O. and Huang, R. *The minimum weighted covering location problem with distance constraints,* Computers and Operations Research, 2008, 35: 356-372.

4. Brimberg, J. *The Fermat-Weber location problem revisited,* Mathematical Programming, 1995, 71: 71-76.

5. Chen, R. *Noniterative Solution of Some Fermat-Weber Location Problems,* Advances in Operations Research, Volume 2011 (2011), Article ID 379505, 10 pages.

6. Church, R. and ReVelle, C. *The maximal covering location problem,* Papers of Regional Science Association, 1974, 32: 101-18.

7. Clerc, M. and Kennedy, J. *The particle swarm-explosion, stability, and convergence in a multidimensional complex space,* IEEE Transactions on Evolutionary Computation, 2002, 1: 58-73.

8. Courant, R. and Robbins, H. *What is mathematics?* Oxford: Oxford University Press, 1941.

9. Dantzig, G.B. and Thapa, M.N. *Linear Programming 2: Theory and Extensions*. Springer; 2003.

10. Drezner, Z. *On convergence of the generalized Weiszfeld algorithm,* Ann Oper Res. 2008; 167: 327-336.

11. Eberhart, R.C. and Kennedy, J. *A new optimizer using particle swarm theory,* proc. 6th int. Symp. Micro Machine and Human science, Nagoya, Japan, 1995; 39-43.

12. Fathali, J., Zaferanieh, M. and Nezakati, A. *A BSSS algorithm for the location problem with minimum square error,* Advances in Operations Research, Volume 2009 (2011), Article ID 212040, 10 pages.

13. Hansen, P., Peeters, D. and Thisse, J.F. *On the location of an obnoxious facility,* Sistemi Urbani, 1981; 3: 299-317.

14. Jamalian, A. and Fathali, J. *Linear programming for the location problem with minimum absolute error,* World Applied Sciences Journal 2009, 7: 1423-1427.

15. Jin, Y.X., Cheng, H.Z., Yan, J. and Zhang, L. *New discrete method for particle swarm optimization and its application in transmission network expansion planning,* Electric Power Systems Research, 2007, 77: 227-233.

16. Kuhn, H.W. *On a pair of dual nonlinear programs. In: Nonlinear Programing,* J.Abadie (eds.), North-Holand Publishers Co. Amsterdam, 1967, 37-54.

17. Kuhn, H.W. and Kuenne, R.E. *An efficient algorithm for the Numerical Solution of the Generalized Weber Problem in spatial Economics,* Journal of Regional Science, 1967; 4: 21-33.

18. Love, R.F., Morris, J.G. and Wesolowsky, G.O. *Facility Location :models and methods.*, North-Holland,(1988).

19. Mangasarian, O.L. *Nonlinear Programming.* McGraw-Hill; 1969.

20. Plastria, F. *GBSSS: the generalized big square small square method for planar single-facility location,* European Journal of Operational Research, 1992; 62: 163-174.

21. Shi, Y. and Eberhart, R.C. *Empirical study of particle swarm optimization,* IEEE International Congress on Evolutionary Computation, 1999, 3: 101-106.

22. Skriver, A.J.V. and Andersen, K.A. *The bicriterion semi-obnoxious location (BSL) problem solved by an ϵ-approximation*, European Journal of Operational Research, 2003; 146: 517-528.

23. Trinh, M.H., Lee, B.H. and Ahn, H.S. *The Fermat-Weber location problem in single integrator dynamics using only local bearing angles,* Automatica, 2015; 59: 90-96.

24. Weiszfeld, E. *Sur Le Point Pour Lequel La Somme Des Distances De N Points Donnes Est Minimum,* Tohoku Mathematical Journal, 1937; 60: 355-386.

25. Zaferanieh, M., Kakhki, H.T., Brimberg, J. and Wesolowsky, G.O. *A BSSS algorithm for the single facility location problem in two regions with different norms,* European Journal of Operational Research, 2008; 190: 79-89.

Iranian Journal of Numerical Analysis and Optimization
Vol. 7, No. 1, (2017), pp 83-105
DOI: 10.22067/ijnao.v7i1.48652

A Reliable Approach for Terminating the GA Optimization Method

L. LotfiKatooli and A. Shahsavand*

Abstract

Genetic algorithm (GA) has been extensively used in recent decades to solve many optimization problems in various fields of science and engineering. In most cases, the number of iterations is the only criterion which is used to stop the GA. In practice, this criterion will lead to prolong execution times to ensure proper solution. A novel approach is presented in this article as the approximate number of decisive iterations ($ANDI$) which can be used to successfully terminate the GA optimization method with minimum execution time. Two simple correlations are presented which relate the new parameter ($ANDI$) with approximate degrees of freedom (Adf) of the merit function at hand. For complex merit functions, a linear smoother (such as Regularization network) can be used to estimate the required Adf. Four illustrative case studies are used to successfully validate the proposed approach by effectively finding the optimum point by using to the presented correlation. The linear correlation is more preferable because it is much simpler to use and the horizontal axis represents the approximate (not exact) degrees of freedom. It was also clearly shown that the Regularization Networks can successfully filter out the noise and mimic the true hyper-surface underlying a bunch of noisy data set.

Keywords: Genetic algorithm; Termination criterion; Approximate degrees of freedom; Approximate number of decisive iteration; Linear smoother concept; Regularization Networks.

---

*Corresponding author
Received 26 July 2015; revised 21 November 2015; accepted 28 September 2016
L. LotfiKatooli
Department of Chemical Engineering, Faculty of Engineering, Ferdowsi University of Mashhad, Mashhad, Iran. e-mail: Leyla.Lotfi@stu.um.ac.ir

A. Shahsavand
Department of Chemical Engineering, Faculty of Engineering, Ferdowsi University of Mashhad, Mashhad, Iran. e-mail: shahsavand@um.ac.ir

1 Introduction

Recently, there has been great interest in improving both operators and criteria of Global Optimization techniques [19, 34]. Genetic algorithm (GA) which is probably the most famous Global optimization technique, still suffers from not having a reliable automatic termination criterion. In other words, GA is usually unable to decide when or where it should terminate the optimization computations [16]. In practice, some pre-specified maximum number of generations is usually used as the termination criterion. In many practical applications, the stopping criteria can significantly influence both the final optimal solution and the overall duration of the entire optimization process [17].

Genetic algorithm is originally rooted in Hollands work [15] which presented a new approach in early sixties for problem solving that finally became widely popular. Afterwards, in late 60s, his students, Bagley [3], Rosenberg [29], and Cavicchio [6] paved the way for new generations. At the present, GA has found extensive applications in many fields of science, economy and engineering.

Many engineering design, modeling and planning problems are inherently quite complex and usually they can't be solved via conventional optimization methods. On the other hand, GA can overcome these issues and is able to successfully solve optimization problems with complex merit functions. GA has simple operators, low storage requirements, and its parallel and global outlook has been applied successfully in a wide variety of problem domains [10, 22], such as control system design [9], chemical kinetics [8] and robotic manipulator design [4].

All steps of GA are usually well defined except the number of stopping criterion iterations [5]. Several termination criteria, such as Generation Number, Evolution Time, Fitness Threshold, Fitness Convergence, Population Convergence and Gene Convergence are introduced to stop the evolutionary algorithms like GA [32]. However, the most commonly used stopping criterion for GA is Generation number which is usually determined via trial and error.

In 1992, Nix and Vose modeled a simple genetic algorithm as Markov chain [24]. They have not preserved the knowledge of previous best in their model. In the mid-90s, Rudolph [30] and Suzuki [33] preserved the knowledge of the previous best chromosomes in their model and proved faster convergence of GAs to the optimal string. In 1994, Meyer and Feng [21] developed the concept of a fuzzy stop criterion to provide a useful evaluation of GA's real-time performance. Their method is not based on a global or fixed value but it works on achieving a pre-specified user-defined level of performance for the given problem. Previously computed performances of earlier GAs were used as a frame of reference for estimation of the current GA performance.

In 1996 Aytug and Koehler [2] derived a theoretical bound on the number of iterations. They showed that a level of confidence is required to guarantee

that the GA can find an optimal solution. They reported that the bound on GA running time is a function of the mutation rate, size of the population strings and the population size but it is independent of the crossover rate and the fitness function. They used Markov chain formulation to find the bound. In a similar work, Aytug et al. [1] have also modeled run time behavior of GA using cardinality representation, as Markov chain.

In 1998 Murthy et al. [23] introduced concept of ε-optimal stopping time of Genetic algorithm. They showed that the probability of performing mutation play an important role in estimation of optimal stopping time of GA. They have provided an estimate for the number of iterations that a GA has to run to obtain an ε-optimal global solution. In 2002 Greenhalgh and Marshall [11] examined how long a genetic algorithm should run to ascertain that it can reach a reasonable optimal solution. They solved this problem by looking at the effect of mutation on the convergence of genetic algorithm.

In 2007 Hedar et al. [14] modified a genetic algorithm with new termination criteria and acceleration elements. They named the proposed method "Genetic Algorithm with Automatic Accelerated Termination (G3AT)". This method is based on some directing strategies: a) The Gene Matrix (GM) that assist the exploration process, b) The Mutagenesis a new type of mutation that accelerate the exploration and exploitation processes and c) The Memetic Algorithm in order to achieve faster convergence.

In 2011 Ong and Fukushima [25] developed another genetic algorithm that was called Genetic Algorithm with Automatic Termination and Search Space Rotation (GATR). This algorithm can terminate the search without conventional predefined criteria such as the maximum number of generations or Function evaluations. They attempted to improve performance of Hedar's work [14] more accurately and more versatile.

In 2012 Bhandari et al. [5] introduced a new stop criterion based on variance (ε)) for the best optimal solutions of the problem at hand. The proposed criterion uses only the fitness function values and takes into account the inherent properties of the objective function. They didnt suggest any automatic procedure for choosing the value of ε. Instead, the ε criterion should be chosen sufficiently small to ensure the required accuracy.

In the present article, a new stopping criterion based on the new concept of approximate number of decisive iterations ($ANDI$) and approximate degrees of freedom (Adf) of the merit function at hand is proposed. For a previously defined and explicitly known cost function (which are used as typical examples in this work), the maximum value of absolute error (MAE_y) which is readily computable from the uncertainties of the input variables, can be used as a simple criterion for selecting the proper value of ($ANDI$).

In real optimizations, where the ordinary least square is used as the merit function, the above approach is extremely time consuming. For these cases, a convenient procedure is introduced to find the appropriate value of ($ANDI$) using the approximate degrees of freedom (Adf) for the least square objective function under consideration. A certain class of artificial neural network will

be used to compute the *Adf* for both discrete and continuous cost functions. It is clearly shown that a straightforward correlation exists between *ANDI* and *Adf* which can be used to estimate the GA termination criterion for any global optimization problem.

In the next sections, initially our in-house algorithm for GA, *ANDI* and *Adf* will be presented briefly. Afterwards, the so called "Regularization network" of Poggio and Girosi [26] is introduced as a vehicle to compute the approximate degrees of freedom for any desired cost function. Several simple and relatively complex case studies (merit functions) are used to propose a new correlation between our so called "*ANDI*" criterion and the "*Adf*" of the objective function at hand.

2 A brief overview of our in-house GA algorithm

Genetic algorithm is based on the survival of better solutions evolved from previous generations until a near optimal solution is acquired. It uses the main three operations of selection, crossover and mutation to produce new generations from the old ones [7].

The initial population is formed of N chromosomes that covers search domain and is usually generated in a random manner. Selection operation is performed by considering individual and cumulative probabilities, at this step stronger and better chromosomes have a greater chance of being selected. The Crossover operator works by selecting two chromosomes at random that will exchange genetic features [20]. In nature, mutation rarely happens but in a conventional GA, it happens frequently producing a new generation in each iteration.

A modified version of GA is presented here which uses a new termination criterion parameter known as *ANDI*. Figure 1 depicts our in-house GA algorithm which combines the standard GA with *ANDI* concept.

As shown in Figure 1, all conventional steps of a standard GA such as initialization, parent selection, cross over and mutation are used in our in-house algorithm. To ensure preserving the best solution throughout the entire optimization calculations, the best fitness is saved in each iteration and this chromosome is expected to bypass any GA operation (crossover or mutation). As Figure 1 depicts, our in-house algorithm stops when the best solution does not change after two consecutive approximate number of decisive iterations (*ANDI*). In other words, GA runs *ANDI* times initially and saves the best optimal solution. Afterwards, it runs *ANDI* times again and compares new best fitness with previous one. If the difference between these two fitnesses is smaller than a very small pre-specified tolerance, then GA is converged. Otherwise the old fitness is replaced by the new fitness and the process will continue as long as the above condition is satisfied.

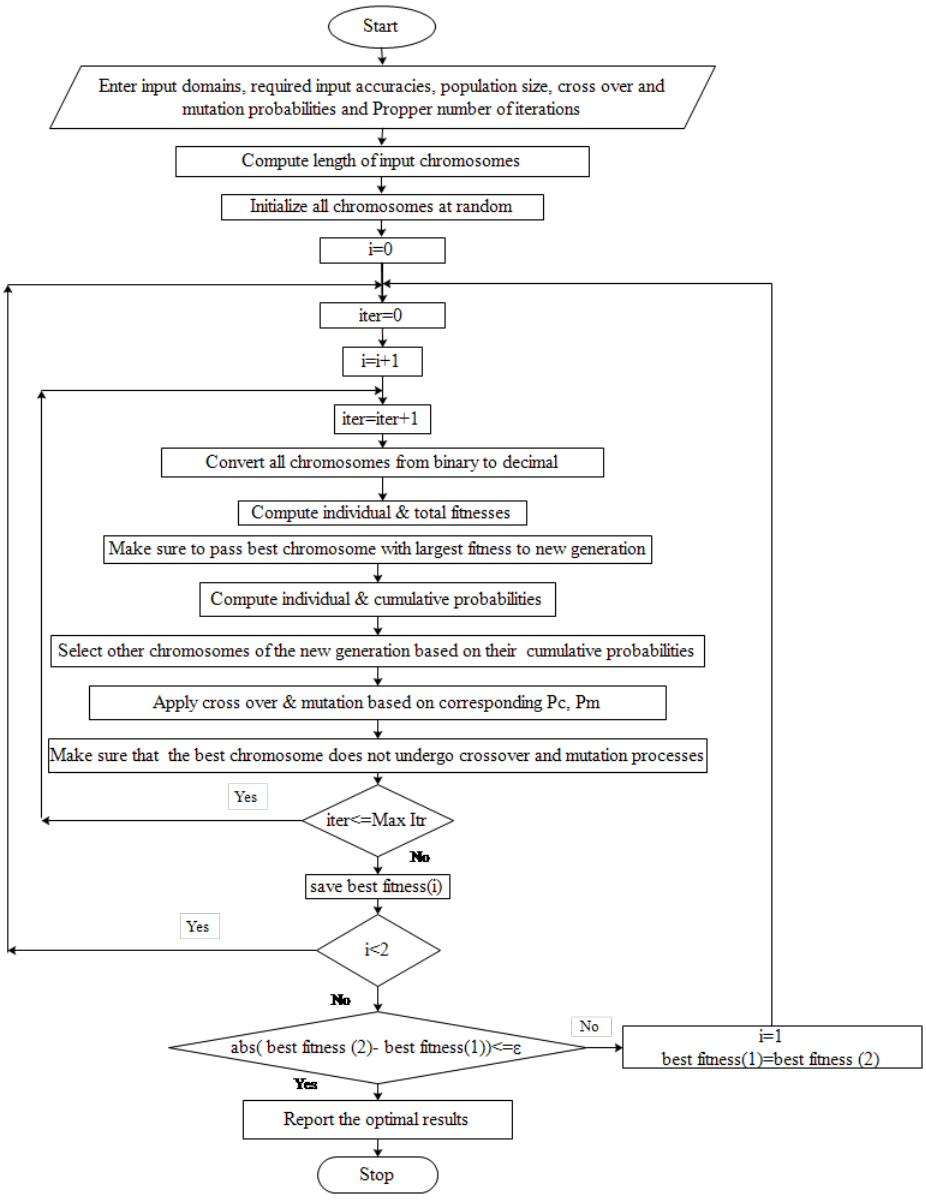


Figure 1: The in-house GA flowchart using *ANDI* concept as the termination criterion.

3 Description of *ANDI* and approximate *df*

As mentioned before, the approximate number of decisive iterations (*ANDI*) is defined as the minimum number of iterations that provides proper performance. This parameter has a crucial role in defining the termination criterion of our in-house algorithm. In practical optimization problems, it is essential to choose appropriate number of iterations (generations) which provides optimal answers with required accuracy in a reasonable time.

A simple but tedious way for selection of proper *ANDI* value is to plot the best optimal solution obtained after two consecutive successful number of iterations versus the corresponding number of iterations. Afterwards, a specific criterion is required to select the approximate number of decisive iterations (*ANDI*) based on difference between two optimal solutions obtained for two successive iterations. The maximum value of absolute error (MAE) which is readily computable from the uncertainties of the input variables, is a good candidate to be used as the above criterion for selecting the proper value of *ANDI*. The following section briefly explains a typical method which can be used to find MAE as an appropriate criterion for selection *ANDI*.

For a multivariate differentiable cost function the absolute error (AE) of the output variable (y) for a given set of inputs can be readily computed from the following equation provided that sufficiently small errors exist in input variables [18]:

$$AE_y = \sum_{i=1}^{p} |\partial f/\partial x_i| \Delta x_i \tag{1}$$

where $(\Delta x_i, i = 1, ..., p)$ are the uncertainties of the p input variables[1]. In GA, a vast domain of input variables is usually exist instead of fixed input values. Therefore, the value of objective function can drastically change with variations of inputs inside corresponding domain. In this work, the value of absolute error for a given cost function is initially computed when the input variables vary inside their domain. The maximum value of the absolute error encountered is then selected as the desired criteria (MAE_y) for selection of the required *ANDI*. Figure 2 illustrates a typical plot of AE for the following objective function:

$$\begin{aligned} f(x_1, x_2) = 100(1 - 3.3x_1 + 2.9x_1^2) exp(((x_1 - 0.5)/0.25)^2 (1 - 3.3x_2 + 2.9x_2^2) \\ exp(-((x_2 - 0.5)/0.25)^2) \end{aligned} \tag{2}$$

As can be seen, the four distinct hills with different heights shown in 3D plot section of Table 1 can be distinguished in Figure 2. Furthermore, the

[1] For discrete functions with known singularities, the finite difference method can be used to obtain the approximate values of partial derivatives $\frac{\partial f}{\partial x_i} = \frac{f^+ - f^-}{\partial x_i}$

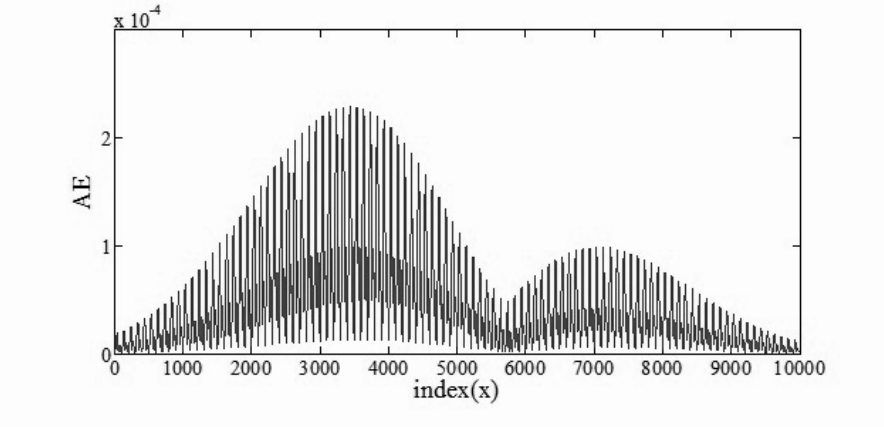


Figure 2: A typical plot of absolute error (AE) for various combinations of input variables inside input domain for cost function of equation (2).

maximum value of absolute error for the merit function described by equation (2) is around 0.022. This value will be used as the required criterion in the following sections to select the optimal value of *ANDI* for the given cost function. Figure 3 shows a typical plot of the best fitness values found after convergence versus the corresponding number of successive iterations for the given cost function inside the corresponding input domain. Assuming that the uncertainties for both input variables to be equal to 0.001, then the maximum value of absolute error (MAE_y) for the given cost function is around 0.006, when computed from equation (1). After determining the value of (MAE_y) and returning to Figure 3, sufficient number of consecutive successive iterations should be found which the variations in the corresponding best fitness values is less than (MAE_y). The square boxes shown in Figure 3 belong to different number of consecutive successive iterations (2-5) which the corresponding best fitnesses variations inside those number of consecutive iterations are smaller than the MAEy. As can be seen in Figure 3, after 3 consecutive successive iterations, the value of best fitness approaches its ultimate optimal value of around 2.95. It means that, at least 14 successive iterations are required to ensure that the optimal solution is successfully approximated. In this work, the above number of successive iterations (i.e. 14) is selected as the *ANDI* which can be used as the termination criterion for GA optimization method. To guarantee that sufficiently large values for *ANDI* is chosen, 5 consecutive successive iterations are used for all case study examples presented in latter sections. To the best of our knowledge, this procedure has not been addressed previously.

Evidently, selection of *ANDI* via the above simple but tedious procedure requires extensive computations which can be alleviated by resorting to the following technique.

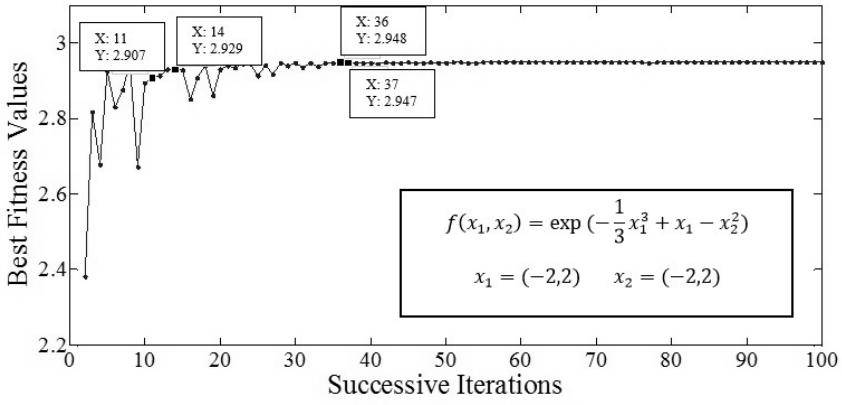


Figure 3: A typical plot of best fitness values (for the corresponding cost function) versus numbers of successive iterations.

A very useful concept usually known as approximate degrees of freedom (Adf) is borrowed from the literature [13] which depends on the complexity of the cost function at hand. As an example, the Adf is around 2 for a univariate linear function and it is around 4 for a bivariate linear objective function (in the absence of input interactions). For complex multivariate (cost) functions, the value of Adf can be computed by using various multivariate expansion series, which requires too many terms for successful reconstruction of the original (merit) function. In an another approach, the smoother matrix concept can be used when a linear smoother (e.g. Regularization network) is replaced for the original (cost) function [13, 27]. For a linear smoother model in the form of $f(x) = S(x)y$, the effective or approximate degrees of freedom (Adf) is defined as the trace of the smoother matrix $S(x)$. As we know the trace of a square matrix is its sum of diagonal elements which is equal to the sum of its eigenvalues. In this work, a certain class of linear smoother known as Regularization Network (RN) is used to successfully mimic the original cost function. Then the trace of the so called smoother matrix of the trained RN, learning from various samples (exemplars) provided by the true cost function will be used as the Adf of the original merit function [13].

4 A brief review of RN used as a linear smoother to mimic the true nonlinear cost function

Poggio and Girosi [26] illustrated that the solution of Euler-Lagrange partial differential equation (PDE) for the multivariate linear regularization problem

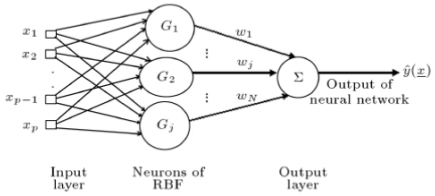


Figure 4: Schematic representation of Regularization network

leads to:

$$(G + I_N)\underline{w_\lambda} = \underline{y} \tag{3}$$

where G is $N \times N$ symmetric green matrix with elements $G_{(i,j)} = (x_i, x_j)$ and λ is the regularization level. The parameter λ should be sufficiently large to ensure that the matrix $(G + I_N)$ is well behaved and hence invertible. Equation (3) can be shown as the network illustrated in Figure 4. The so called Regularization network (RN) contains N neurons or centers $(x_j, j = 1, ..., N)$ positioned exactly over training data points $(x_i, i = 1, ..., N)$. The influence of the regularization parameter, λ, is embedded in the unknown synaptic weights, w_j's.

Using the appropriate differential operator in Euler-Lagrange PDE, the infinitely differentiable factorizable multidimensional Gaussian function in the following form will be final solution and denotes the shape of the basis functions:

$$G(x_i, x_j) = exp(\| \underline{x_i} - \underline{x_j} \|^2/\sigma_j^2) = \prod_{k=1}^{p} exp[-(x_{i,k} - x_{j,k})^2/(\sigma_j^2)] \tag{4}$$

The performance of the RN strongly depends on both the appropriate choice of the isotropic spread (σ) and the proper level of regularization (embedded in the synaptic weights) [31]. By resorting to the leave-one out cross validation (LOOCV or CV) criterion efficient and automatic computation of the optimum level of regularization (λ^*) can be achieved for any given value of σ.

$$CV(\lambda) = 1/N \sum_{k=1}^{N} [(\underline{e_k^T}(I_N - H(\lambda))\underline{y})/(\underline{e_k^T}(I_N - H(\lambda))\underline{e_k})]^2 \tag{5}$$

where e_k is the kth unit vector of size N, I_N is the $N \times N$ unit matrix and $H(\lambda)$ is the smoother matrix originally defined by Hastie and Tibshirani [13] for a linear smoother such as RN. As mentioned in the previous section, the trace of $H(\lambda)$ denoted the approximate degrees of freedom of the linear smoother. By definition, the smoother matrix for the RN can be computed

from the following relation:

$$H(\lambda) = S = G(G + \lambda I)^{-1} \tag{6}$$

5 Predictions of *Adf* via RN for different synthetic merit functions

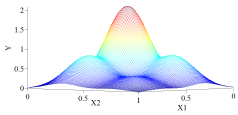
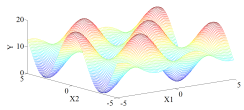
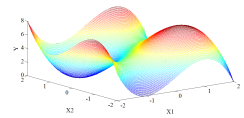
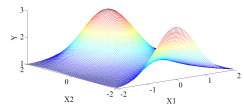
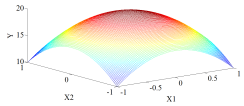
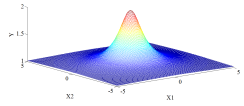
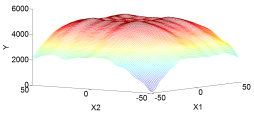
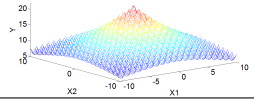
Several synthetic merit functions are used (as shown in 3D plot section of Table 1), to demonstrate the ability of the RN to successfully estimate the *Adf* of the original (merit) function and effectively reconstruct it.

A set of distinct exemplars are generated from the original (objective) function and then contaminated with a pre-specified level of noise. These data are initially normalized and then used to train and validate the corresponding optimized RNs with fixed spread of unity (to ensure sufficient overlap between adjacent centers). In practice, a set of distinct exemplars are usually replaces the merit function. In such cases, data generation step is not required anymore but other stages (as well as the initialization of the input variables) should be followed as they will be described in the following section.

For each merit function of Table 1, one hundred equispaced exemplars (on a 10×10 grid) are inside the corresponding domain and then contaminated with 10 percent noise level. These data sets are then used to train the equivalent optimized RNs for all functions. As the first step, all input variables are normalized and then the so called 100×100 Green matrix is constructed using equation (4) (with centers positioned on normalized training exemplars) and ($\sigma = 1$). Then equation 5 is used to find the optimum value of regularization parameter by minimizing the LOOCV criterion. The LOOCV criterion uses all training data for both training and validation purposes. Finally the trace of the smoother matrix ($H(\lambda)$) and linear weights of RN are computable from equations (6) and (3), respectively. These linear weights along with both maximum and minimum values of all training input data can then be used to compute the generalization performance of the optimally trained RN with optimum level of regularization.

A set of distinct exemplars are generated from the original (objective) function and then contaminated with a pre-specified level of noise. These data are initially normalized and then used to train and validate the corresponding optimized RNs with fixed spread of unity (to ensure sufficient overlap between adjacent centers). In practice, a set of distinct exemplars are usually replaces the merit function. In such cases, data generation step is not required anymore but other stages (as well as the initialization of the input variables) should be followed as they will be described in the following section.

Table 1: Typical bivariate non-linear functions

No.	Cost(merit)function	3D plot	Domain and Range
1	$100(1-3.3x_1+2.9x_1^2)exp(-((x_1-0.5)/0.25)^2$ $(1-3.3x_2+2.9x_2^2)exp(-((x_2-0.5)/0.25)^2))$		$X_1=(0,1)$ $X_2=(0,1)$ $Y=(0,2.061)$
2	$10cos(x_1)sin(x_2)+10$		$X_1=(-5,5)$ $X_2=(-5,5)$ $Y=(1.7,20)$
3	$x_1^3-x_2^3+3x_1-3x_2+4$		$X_1=(-2,2)$ $X_2=(-2,2)$ $Y=(0,8)$
4	$exp(-1/3x_1^3+x_1-x_2^2)$		$X_1=(-2,2)$ $X_2=(-2,2)$ $Y=(1,2.95)$
5	$-5(x_1^2+x_2^2)+20$		$X_1=(-1,1)$ $X_2=(-1,1)$ $Y=(10,20)$
6	$1+1/(1+x_1^2+x_2^2)$		$X_1=(-5,5)$ $X_2=(-5,5)$ $Y=(1,2)$
7	$f(x)=\sum_{i=1}^{D}(t_i+200D)+5000$ $t_i=\begin{cases}-200+x_i^2 & if \quad x_i<0\\ -80(2.5-x_i) & if \quad 0<x_i<2.5\\ -64(7.5-x_i) & if \quad 2.5<x_i<5\\ -64(x_i-2.5) & if \quad 5<x_i<7.5\\ -28(x_i-7.5) & if \quad 7.5<x_i<12.5\\ -28(17.5-x_i) & if \quad 12.5<x_i<17.5\\ -32(17.5-x_i) & if \quad 17.5<x_i<22.5\\ -32(27.5-x_i) & if \quad 22.5<x_i<27.5\\ -80(x_i-27.5) & if \quad 27.5<x_i<30\\ -200+(x_i-30)^2 & if \quad 30<x_i\end{cases}$		$X_1=(-50,50)$ $X_2=(-50,50)$ $Y=(0,5000)$
8	$20exp(-.2\sqrt{\frac{1}{D}\sum_{i=1}^{D}x_i^2})$ $+exp(\frac{1}{D}\sum_{i=1}^{D}cos(2\pi x_i))+40$		$X_1=(-10,10)$ $X_2=(-10,10)$ $Y=(5.56,22.72)$

For each merit function of Table 1, one hundred equispaced exemplars (on a 10×10 grid) are inside the corresponding domain and then contaminated with 10 percent noise level. These data sets are then used to train the equivalent optimized RNs for all functions. As the first step, all input variables are normalized and then the so called 100×100 Green matrix is constructed using equation (4) (with centers positioned on normalized training exemplars) and ($\sigma = 1$). Then equation 5 is used to find the optimum value of regularization parameter by minimizing the LOOCV criterion. The LOOCV criterion uses all training data for both training and validation purposes.

Finally the trace of the smoother matrix ($H(\lambda)$) and linear weights of RN are computable from equations (6) and (3), respectively. These linear weights along with both maximum and minimum values of all training input data can then be used to compute the generalization performance of the optimally trained RN with optimum level of regularization. Figure 5 shows the generalization performances of all 6 trained networks (RNs) on 100×100 grids inside the corresponding training domains.

Comparison of these generalizations performances with the original plots of Table 1 reveals that the optimal RNs provided impressive performances on reconstructing the true hyper-surface embedded in limited and noisy training data sets.

6 Predictions of *ANDI* values for various synthetic examples

The 3D case studies shown in Table 1 will be ultimately used to find a relatively simple correlation between *Adf* and *ANDI*. As mentioned earlier in section 3, the maximum value of absolute error (MAE_y) is used to select the proper value of *ANDI* for each and every (cost) function. The uncertainties in input values and the corresponding MAE_y values are depicted in Table 2.

As previously mentioned, the MAE_y values provide proper criteria for estimation of the *ANDI* values or the number of successive iterations that the GA requires to provide the optimum value of cost function, as described in the calculation procedure shown in Figure 1. In our present approach, the Genetic Algorithm is initially executed with different number of successive iterations (2 to 100) and the best fitness values are found for each and every iteration (after convergence) and plotted against the number of successive iterations as shown in Figure 6 for all (merit) functions of Table 1. Afterwards, the MAE_y values for the each (merit) function is used to select the proper value for *ANDI* for corresponding (merit) function.

Table 3 shows the corresponding best fitness values at different number of successive iterations for various numbers of consecutive successive iterations for all (merit) functions of Table 1. As mentioned before, to guarantee that

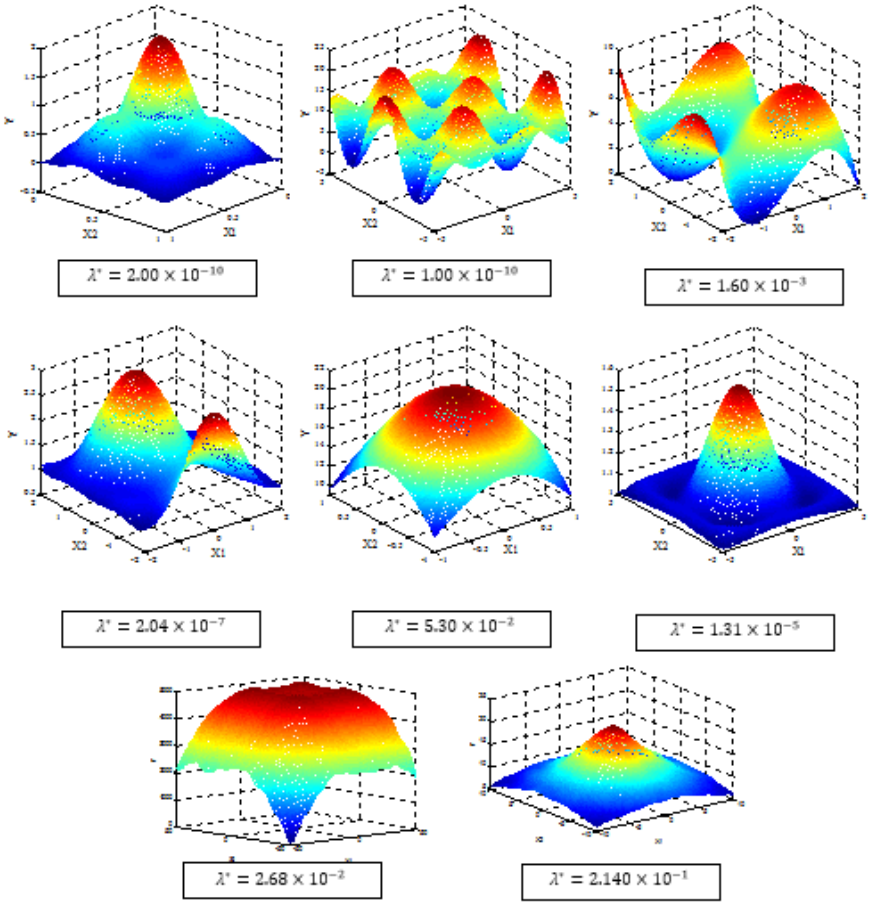


Figure 5: Generalization Performances of RNs ($\sigma = 1$) for all merit functions of Table 1 trained with 100 equi-spaced exemplars contaminated with 10 percent noise.

Table 2: MAE values for functions depicted in Table 1.

Function No.	MAE_{X_1}	MAE_{X_2}	MAE_y
1	.001	.001	.022
2	.001	.001	.010
3	.001	.001	.018
4	.001	.001	.006
5	.001	.001	.014
6	.001	.001	.010
7	.001	.001	.200
8	.001	.001	.070

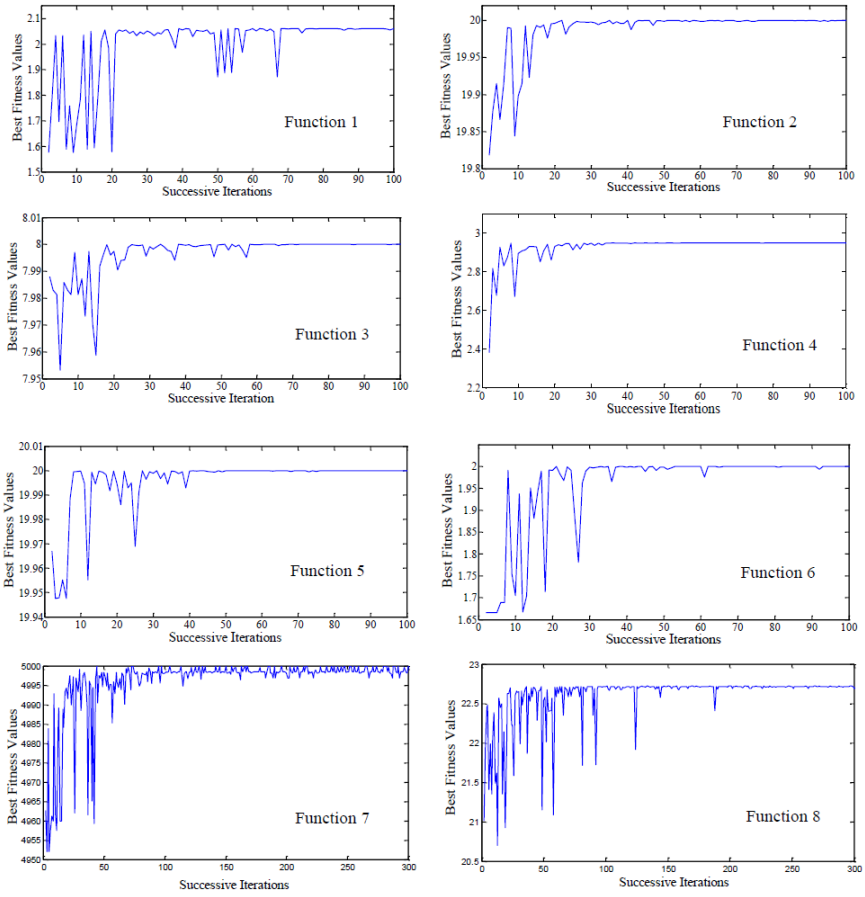


Figure 6: Plots of best fitness values versus number of successive iterations for all merit functions of Table 1.

Table 3: Best fitness values (BFV) at different number of successive iterations (NCSI) for various numbers of consecutive successive iterations (NCSI).

Function No. in Table 1	NCSI=2		NCSI=3		NCSI=4		NCSI=5	
	NSI	BFV	NSI	BFV	NSI	BFV	NSI=ANDI	BFV
1	21	2.055	22	2.050	24	2.042	71	2.060
2	27	19.997	53	19.998	78	19.999	79	19.999
3	2	7.982	3	7.981	18	7.996	19	7.997
4	11	2.907	14	2.929	36	2.948	37	2.947
5	2	19.947	3	19.947	5	19.947	10	19.994
6	2	1.666	3	1.666	4	1.666	33	1.999
7	14	4959.90	108	4988.29	137	4998.14	175	4998.42
8	38	21.98	71	22.71	135	22.65	148	22.72

sufficiently large values for *ANDI* is chosen, five consecutive successive iterations (CSI) are used for all (merit) functions as depicted in the left (NSI: number of successive iterations) column of the NCSI=5 of Table 3. For example, *ANDI* value for the first (merit) function of Table 1 is 71 where the corresponding best fitness value is around 2.060.

Evidently, the above procedure seems quite complex and can be avoided by resorting to the approximate degrees of freedom (*Adf*). In the absence of an explicit cost function, a linear smoother model such as RN can be used (as discussed in the previous section) to reconstruct the hyper-surface representing the original merit function and compute the *Adf* of the cost function at hand. The final goal is to present a relatively simple correlation between, *ANDI* and *Adf*. This approach has not been addressed previously.

In some cases the cost function is sufficiently simple and the exact df can be easily computed. For example, the exact degrees of freedom for the third and 5th (merit) functions of Table 1 are exactly eight and six, respectively. In the cases, no input interactions exist and two cubic or quadratic models ($f_1(x_1)$ and $f_2(x_2)$) can successfully reconstruct the true cost functions, respectively. In other cases, the Regularization network can be used to find the *Adf* for the complex cost function at hand. The following procedure is used to train six fully optimized RNs and compute the *Adf* values for all cost functions of Table 1.

1. Generate N training exemplars with sufficient noise level from the cost function at hand.
2. Normalize input variables values.
3. Compute the $N \times N$ elements of the Green matrix by positioning all the network centers on the normalized inputs and assuming a pre-specified value for the Gaussian isotropic spreads.
4. Compute the optimal value of regularization parameter (λ^*) by resorting to LOOCV criterion.
5. Calculate the *Adf* value as the trace of $N \times N$ smoother matrix $H(\lambda^*)$, which is already defined by equation (6).

Table 4: Approximate Degrees of freedom (*Adf*) for all merit functions of Table 1.

Func. No.	1	2	3	4	5	6	7	8
Adf	45	48	12	31	8	24	228	217
ANDI	71	79	19	37	10	33	175	148

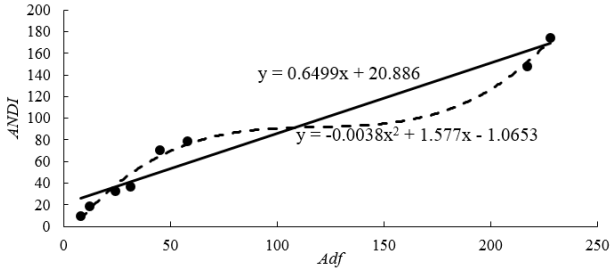


Figure 7: Two simple correlations introduced between *ANDI* and *Adf* values.

The second row of Table 4 shows the values of *Adf* for all merit functions of Table 1 computed via the above procedure. As can be seen, the approximate values for the degrees of freedom (*Adf*) of both third and 5th (merit) functions of Table 1 (which are: 12 and 8) are sufficiently close to their exact values (which are: 8 and 6). Other merit functions are too complex to directly compute their *Adf* values and the above procedure should be used to calculate the corresponding *Adf* values. The third row contains the *ANDI* values which are exactly transferred from the last column of Table 3.

7 Presenting simple correlations between *ANDI* and *Adf*

The data of Table 4 are used to plot the variations of *ANDI* values versus the corresponding *Adf* values, as depicted in Figure 7. Two linear and cubic correlations are extracted by fitting the data of Table 4. Although, the cubic equation almost passes from each and every point, however we prefer to use the linear fit because it is quite simpler and it should also be remember that the horizontal axis represents the approximate (not exact) degrees of freedom. Both of these correlations will be used in the following section to compute *ANDI* values for a relatively complex optimization problem.

Table 5: Comparison of best fitness and maximum values found via linear and cubic correlations for 2 different cost functions of Fig. 8.

Function	*Adf*	Correlation	*ANDI*	Best Fitness value	True Maximum
8.a	14	Linear	30	2.1479	2.1481
		Cubic	21	2.1471	
8.b	30	Linear	41	13.5842	13.5846
		Cubic	43	13.5836	

8 Validation of the proposed correlations

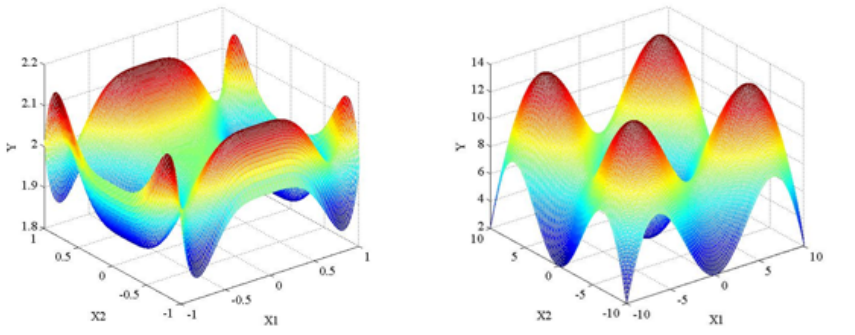
Various synthetic 2 and 5 dimensional in input domains case studies are used to validate the applicability of the proposed correlations for the termination criterion of the optimal genetic algorithm method. As the results clearly shows, the proposed heuristics successfully predicts the approximate number of decisive iterations (*ANDI*) for all 3D and 6D hyper-surfaces.

8.1 Two 3D hyper-surfaces synthetic case studies

Figure 8 illustrates two different 3D synthetic cost functions which are used to validate the effectiveness of the proposed correlations (shown on Figure 7). The first merit function (Figure 8.a) has no input interactions and the corresponding value for its degrees of freedom can be readily obtained as 14 by adding the maximum orders of each input plus number of input variables. Using both linear and cubic correlations, the *ANDI* values are computed and fixed at 18 and 22, respectively. Table 5 illustrates the same results for the second cost function (Figure 8.b). For such complex function, the exact degrees of freedom is not readily known because large interactions exist between two input variables. The fully optimized RN is used to estimate its approximate degrees of freedom around 30. Table 5 shows that both *ANDI* values computed from correlations of Figure 7 can successfully pinpoint the true maximum (optimum) point. As before, the linear correlation is much simpler and provides almost the same result as the cubic one.

8.2 Two 6D hyper-surface synthetic case studies

To demonstrate the robustness of the proposes algorithm and the excellent performance of the Regularization Network on filtering large noise levels and extracting the true underlying hyper-surface from a bunch of noisy data, two 6D synthetic examples are introduced in this section. Figure 9 depicts two



$f(x_1, x_2) = -x_1^6 + x_2^6 + x_1^4 - x_2^4 + 2$
$x_1 = (-1,1)\, x_2 = (-1,1)$
(a)

$f(x_1, x_2) = 2 + x_1 \sin(.1\pi x_1) + x_2 \sin(.1\pi x_2)$
$x_1 = (-10,10)\, x_2 = (-10,10)$
(b)

Figure 8: 3D plots and corresponding equations for two bivariate cost functions used for validation purposes.

typical contour maps of these case studies (borrowed from literature [28]). As can be seen, both functions are quite complex with relatively large oscillations and have sufficiently sizable degrees of freedom. To reduce the computational time while preserving the accuracy of the network predictions, 5 randomly spaced exemplars are generated in each input dimensions. To emphasize the excellent noise filtering capability of the Regularization network, the true generated hyper-surfaces is then contaminated with extremely large noise level of 50 percent .

The noisy data set is then used to train the optimal RN and Figure 10 illustrates the impressive performance of such fully optimized RN on capturing the true hyper-surface from heavily contaminated noisy data. The leave one out cross validation technique is used to select the optimum level of regularization [12]. Figure 10 clearly illustrates that the optimal RN which uses linear requlariza tion theory (also known as Tikhonov or Phillips and Twomey method [27]) with optimum level of regularization can successfully pull away the fitted hyper-surface from the noisy training data and reconstruct the true multidimensional surface by filtering out the large noises available in that data set. As table 6 shows, the RN predicts that the fitted hyper-surfaces have Adf s of 53 and 154 respectively, which are quite reasonable for such sophisticated functions. As before, both equations of Figure 7 can provide excellent maximums compared to the real maximum values of the cost functions at hand.

Figure 11 shows that both values of approximate numbers of decisive iterations computed from correlations of Figure 7 provide same results for the fitness values of both functions. Although, several oscillations may occur

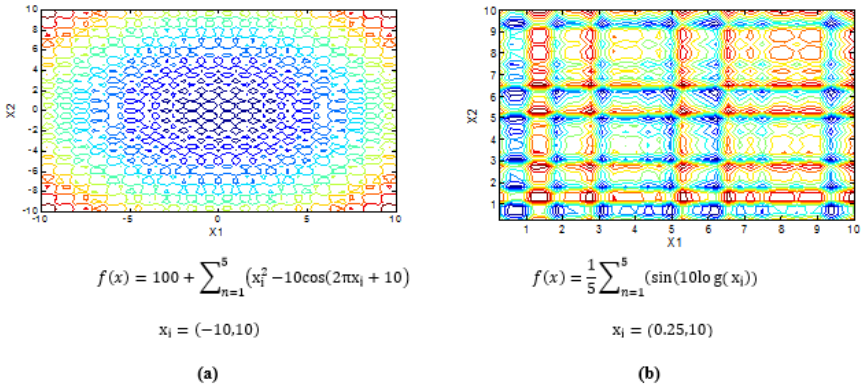


Figure 9: Typical two dimensional contour maps of two 5D synthetic case studies.

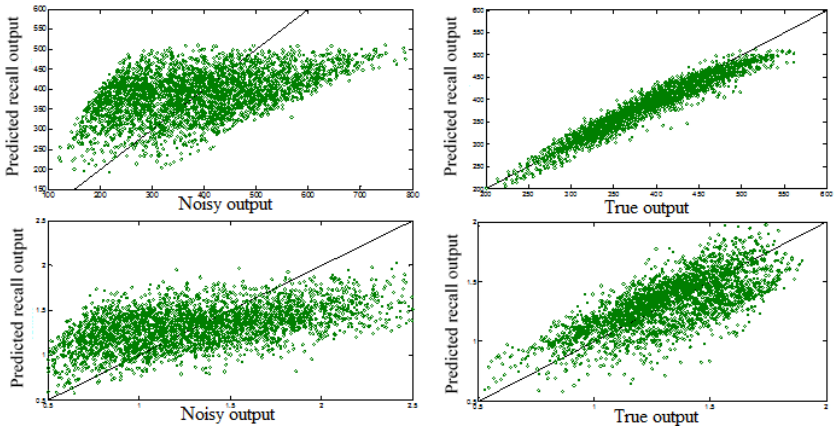


Figure 10: Effectiveness of RN on filtering the noise and reconstructing the true hyper surface in large multi-dimensional case studies.

Table 6: Comparison of best fitness and maximum values found via linear and cubic correlations for 2 different cost functions of Fig. 9.

Function	*Adf*	Correlation	*ANDI*	Best Fitness value	True Maximum
9.a	53	Linear	54	642.7	642.8
		Cubic	72	642.6	
9.b	154	Linear	121	1.993	1.993
		Cubic	152	1.993	

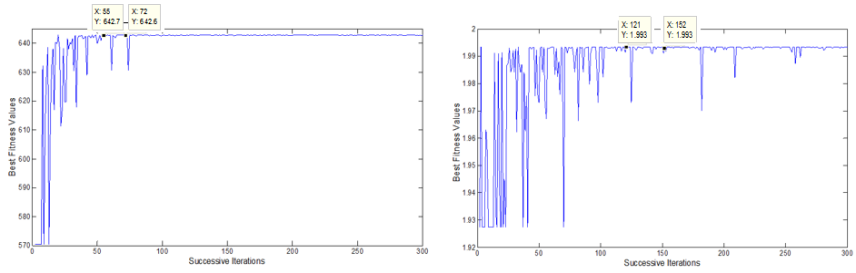


Figure 11: Plots of best fitness values versus number of successive iterations for all both 6D hyper-surfaces.

after the predicted *ANDI* values, however the maximum value of the cost function remains essentially constant. In other words, practically no better maximum exists for any iterations greater than *ANDI* values.

All above 3D and 6D hyper-surfaces case studies clearly demonstrate that the proposed method can successfully be used to predict the maximum value of any large dimensional cost function via genetic algorithm technique by resorting to approximate number of decisive iterations computable from correlations of Figure 7.

9 Conclusion

A new concept was introduced as the approximate number of decisive iterations (*ANDI*) which could be used as the termination criterion for genetic algorithm global optimization method. Two simple correlations were presented which related the novel *ANDI* value with the approximate degrees of freedom. A convenient method based of linear smoother concept was used to predict the approximate degrees of freedom for complex merit functions. The proposed approach was successfully tested for 4 different case studies. It was clearly illustrated that the Regularization Network (RN) can successfully filter out large noise levels and reconstruct the true hyper-surface from noisy data. The proposed approach has not addressed previously and can drastically reduce the computation time for complex optimization problems.

References

1. Aytug, H., Bhattacharrya, S. and Koehler, G.J. *A Markov chain analysis of genetic algorithms with power of 2 cardinality alphabets*, Eur. J. Oper.

Res. 96 (1996), 195-201.

2. Aytug, H. and Koehler, G.J. *Stopping Criteria for Finite Lenght Genetic Algorithms*, Informs Jo. Comput. 8 (1996), 183-191.

3. Bagley, J.D. *The behavior of adaptive systems which employ genetic and correlation algorithms*. Ph.D. Thesis, University of Michigan, 1967.

4. Bergamaschi, P.R., Saramago, S.F.P. and Coelho, L.S. *Comparative study of SQP and metaheuristics for robotic manipulator design*, Appl. Numer. Math. 58 (2008), 13961412.

5. Bhandari, D., Murthy, C.A. and Pal, S.K. *Variance as a Stopping Criterion for Genetic Algorithms with Elitist Model*, Fundamenta Informaticae. 120 (2012), 145-164.

6. Cavicchio, D.J. *Adaptive Search using Simulated Evolution*, Ph.D. Thesis, University of Michigan, 1970.

7. Conway, B.A. *A Survey of Methods Available for the Numerical Optimization of Continuous Dynamic Systems*, J. Optim. Theory Appl. 152 (2012), 271306.

8. Elliott, L., Ingham, D.B., Kyne, A.G., Mera, N.S., Pourkashanian, M. and Wilson, C.W. *Genetic algorithms for optimisation of chemical kinetics reaction mechanisms*, Prog. Energ. Combust. 30 (2004), 297328.

9. Fleming, P.J. and Purshouse, R.C. *Evolutionary algorithms in control systems engineering: a survey*, Control Eng. Pract. 10 (2002), 12231241.

10. Gen, M. and Cheng, R. *Genetic Algorithms amd engineering Design*. New York, John Wiley and Sons, 2000.

11. Greenhalgh, D. and Marshall, S. *Convergence Criteria for Genetic Algorithms*, SIAM J. Comput. 30(1) (2000), 269282.

12. Golub, G.H., Van Loan, C.G., *Matrix Computations*, third ed., Johns Hopkins UniversityPress, Baltimore, 1996.

13. Hastie, T.J. and Tibshirani, R.J., Generalized Additive Models, London, Chapman and Hall, 1990.

14. Hedar, A.R., Ong, B.T. and Fukushima, M. *Genetic Algorithms with Automatic Accelerated Termination*, Technical Report, 2007.

15. Holland, J.H. *Outline for a Logical Theory of Adaptive Systems*, J. ACM. 9 (1962), 297-314.

16. Jackson. W.C. and Norgard. J.D. A Hybrid *Genetic Algorithm with Boltzmann Convergence Properties*, J. Optim. Theory. Appl. 136 (2008), 431-443.

17. Kim, J. *Genetic algorithm stopping criteria for optimization of construction resource scheduling problems*, Construction Management and Economics. 31(1) (2013), 3-19.

18. Kopchenova, N.V. and Maron, I.A. *Computatinal Mathematics: worked examples ana problems with elements of theory*, Moscow, Mir Publishers, 1975.

19. Kumar. M. and Banka, H. Changing Mutation *Operator of Genetic Algorithms for Optimizing Multiple Sequence Alignment*, International Journal of Information and Computation Technology. 5(3) (2003), 465-470.

20. Lam, X.B, Kim, Y.S., Hoang, A.D. and Park. C.W. *Coupled Aerostructural Design Optimization Using the Kriging Model and Integrated Multiobjective Optimization Algorithm*, J. Optim. Theory Appl. 142 (2009), 533556.

21. Meyer, L. and Feng, X. *A Fuzzy Stop Criterion for Genetic Algorithms Using Performance Estimatio*n, In: Proceedings of the Third IEEE Conference on Fuzzy Systems, Orlano, FL, IEEE Service Center, pp. 1990-1995, (1994)

22. Michalewicz, Z. *Genetic Algorithm+Data Structure =Evolution Programs*, New York, Springer-Verlag, 1992.

23. Murthy, C.A., Bhandari, D. and Pal, S.K. *Optimal Stopping Time for Genetic Algorithm*, Fundamenta Informaticae. 35 (1998), 91-111.

24. Nix, A.E. and Vose, M.D. *Modeling genetic algorithms with Markov chains*, Ann. Math. Artif. Intel. 5 (1992) 79-88.

25. Ong, B.T. and Fukushima, M. *Genetic algorithm with automatic termination and search space rotation*, Memetic Comp. 3 (2011), 111-127.

26. Poggio, T. and Girosi, F. *Regularization algorithms for learning that are equivalent to multilayer networks*, Science. 247 (1990), 978-982.

27. Press, W.H., Teukolsky, S.A., Vetterling, W.T. and Flannery, B.P. *Numerical Recipes: The Art of Scientific* Computing, Cambridge, Cambridge University Press, 2007.

28. Qu, B.Y., Liang, J.J., Wang, Z.Y., Chen, Q., Suganthan, P.N.: *Novel benchmark functions for continuous multimodal optimization with comparative results*, Swarm and Evolutionary Computation 26 (2016), 23-34.

29. Rosenberg, R.S. *Simulation of genetic populations with biochemical properties*, Ph.D. Thesis, University of Michigan, 1967.

30. Rudolph, G. *Convergence Analysis of Canonical Genetic Algorithms*, IEEE T Neural Networ. 5(1) (1994), 96-101.

31. Shahsavand, A.: *An Optimal Radial Basis Function (RBF) Neural Network for Hyper-Surface Reconstruction.* Chemistry and Chemical Engineering 16 (2009), 41-53.

32. Sumathi, S., Hamsapriya, T. and Surekha, P. *Evolutionary Intelligence An Introduction to Theory and Applications with Matlab*, Berlin, Springer-Verlag, 2008.

33. Suzuki, J. A Markov chain analysis on a genetic algorithm, IEEE T. Syst. Man Cyb. 25(4) (1995) 655659.

34. Yuan, S., Skinner, B., Huang, S. and Liu, D. A *new crossover approach for solving the multiple travelling salesmen problem using genetic algorithms*, Eur. J. Oper. Res. 228 (2013), 7282.

Iranian Journal of Numerical Analysis and Optimization
Vol. 7, No. 1, (2017), pp 107-117
DOI: 10.22067/ijnao.v7i1.52283

Alternating direction method of multipliers for the extended trust region subproblem

M. Salahi* and A. Taati

Abstract

The extended trust region subproblem has been the focus of several research recently. Under various assumptions, strong duality and certain SOCP/SDP relaxations have been proposed for several classes of it. Due to its importance, in this paper, without any assumption on the problem, we apply the widely used alternating direction method of multipliers (ADMM) to solve it. The convergence of ADMM iterations to the first order stationary conditions is established. On several classes of test problems, the quality of the solution obtained by the ADMM for medium scale problems is compared with the SOCP/SDP relaxation. Moreover, the applicability of the method for solving large scale problems is shown by solving several large instances.

Keywords: Extended trust region subporblem; Alternating method; Nonconvex optimization; Semidefinite program; Second order cone program.

1 Introduction

Consider the following extended trust region subproblem with m linear inequalities (m-eTRS):

$$\begin{aligned} \min \quad & \frac{1}{2}x^T Ax + a^T x \\ & ||x||^2 \leq 1, \\ & b_i^T x \leq \beta_i, \quad i = 1, \cdots, m, \end{aligned} \tag{m-eTRS}$$

*Corresponding author
Received 19 December 2015; revised 31 July 2016; accepted 28 September 2016
M. Salahi
Department of Applied Mathematics, Faculty of Mathematical Sciences, University of Guilan, Rasht, Iran. e-mail: salahim@guilan.ac.ir

A. Taati
Department of Applied Mathematics, Faculty of Mathematical Sciences, University of Guilan, Rasht, Iran. e-mail: akram.taati@gmail.com

where $A \in \mathbb{R}^{n\times n}$ is a symmetric matrix but not necessarily positive definite, $a, b_i \in \mathbb{R}^n$ and $\beta_i \in \mathbb{R}$. Due to its importance and crucial role in solving general nonlinear optimization problems, several versions of it have been the focus of current research [6, 7, 11, 13, 17, 18]. In [13], the authors have shown that the dimension condition, $\dim(\text{ Ker}(A - \lambda_1 I_n)) \geq s + 1$, where λ_1 is the smallest eigenvalue of A and $s = \dim(\text{span}\{b_1, .., b_m\})$, together with the Slater condition ensure that a set of combined first and second-order Lagrange multiplier conditions are necessary and sufficient for the global optimality of (m-eTRS) and consequently for strong duality. In [11], the authors have proposed an induction technique that reduces (m-eTRS) to several small-sized trust region subproblems (TRS). When m is not too large, their method can be very efficient but requires finding local-nonglobal minimum (LNGM), which no efficient algorithm is known to find it. Moreover, no numerical evidences are reported by the authors to support their theoretical foundation. Also they have improved the dimension condition by Jeyakumar and Li under which (m-eTRS) admits an exact SDP relaxation. They proposed the following condition

$$\text{rank } ([A - \lambda_1 I_n \;\; b_1 \; b_2 \; \cdots \; b_m]) \leq n - 1. \tag{1}$$

This rank condition implies that the global optimal solution of the (m-eTRS) does not happen at the (LNGM) of (TRS) [8,15]. In a most recent study [7], the authors have proposed the following SOCP/SDP relaxation for (m-eTRS) based on the following assumption:

Assumption 1: For all $i < j$, there exists no feasible point x for (m-eTRS) such that $b_i^T x = \beta_i$ and $b_j^T x = \beta_j$.

$$\begin{aligned} min_{x,X} \;\; & \frac{1}{2} trace(AX) + a^T x \\ \text{s.t. } & trace(X) \leq 1, \; X \succeq xx^T, \\ & ||\beta_i x - X b_i|| \leq \beta_i - b_i^T x, \;\; 1 \leq i \leq m, \qquad (R_m) \\ & \beta_i \beta_j - \beta_j b_i^T x - \beta_i b_j^T x + b_i^T X b_j \geq 0, \;\; 1 \leq i < j \leq m. \end{aligned}$$

They also have proved that this relaxation is tight when we consider the following assumption instead of Assumption 1.

Assumption 2: For all $i < j$, there exists no x with $||x|| < 1$ such that $b_i^T x = \beta_i$ and $b_j^T x = \beta_j$.

Moreover, in several recent studies, when $m = 1$, efficient algorithms have been proposed to solve large instances of (m-eTRS) [17, 18]. Therefore, finding an efficient algorithm to solve (m-eTRS) problems when $m > 1$ is still an active research area as the proposed methods either are using certain assumptions to simplify the analysis or using the SOCP/SDP relaxation, which both are not applicable practically, specially for large scale instances.

Recently, the *Alternating Direction Method of Multipliers* (ADMM) have been widely used to solve optimization problems arising in machine learning, signal processing, matrix factorization, financial optimization and etc. [2,3,5, 12,14,19,22]. Although the method exhibits faster convergence in practice, its global convergence is still the subject of research. Under various assumptions on the sequence generated by the method like "if the limit point exists" or "if the Lagrange multipliers are bounded", its global convergence to a stationary point is established in the above-mentioned papers. In this paper, we apply ADMM to (m-eTRS) without any assumption on the geometry of the feasible region. The convergence of the method to the first order necessary optimality conditions is proved. Finally, on several medium scale instances its performance is compared with the SOCP/SDP relaxation of [7] and then several large instances also are solved by ADMM. To the best of our knowledge, there is no algorithm in the literature for large scale (m-eTRS) which we could provide comparison.

2 ADMM for (m-eTRS)

One can write (m-eTRS) in the following equivalent form:

$$\begin{aligned} \min \quad & \frac{1}{2}x^T Ax + a^T x \\ & ||x||^2 \leq 1, \\ & b_i^T z \leq \beta_i, \quad i = 1, \cdots, m \\ & x = z. \end{aligned} \tag{2}$$

Now consider the following *augmented Lagrangian* for (2):

$$L(x, z, \lambda) = \frac{1}{2}x^T Ax + a^T x + \lambda^T (x - z) + \frac{\rho}{2}||x - z||^2,$$

where λ_i's are Lagrange multipliers and $\rho > 0$ is the penalty parameter. The ADMM iterations for the given x^k and λ^k are as follows [5]:

- **Step 1:** $z^{k+1} = \text{argmin}_{b_i^T z \leq \beta_i,\ i=1,\cdots,m} L(x^k, z, \lambda^k)$.
- **Step 2:** $x^{k+1} = \text{argmin}_{||x||^2 \leq 1} L(x, z^{k+1}, \lambda^k)$.
- **Step 3:** $\lambda^{k+1} = \lambda^k + \gamma\rho(x^{k+1} - z^{k+1})$, where $\gamma \in (0, 1)$ is a constant.

In what follows, we discuss the above steps. In Step 1, we need to solve the following convex quadratic optimization problem with m linear inequality constraints which can be efficiently solved using existing convex optimization software packages like CVX [9]:

$$\begin{aligned} \min \quad & \tfrac{\rho}{2} z^T z - (\lambda^k + \rho x^k)^T z \\ & b_i^T z \leq \beta_i, \ i = 1, \cdots, m. \end{aligned} \tag{3}$$

In Step 2 we need to solve the following (TRS) problem:

$$\begin{aligned} \min \ & \tfrac{1}{2} x^T (A + \rho I) x + (a + \lambda^k - \rho z^{k+1})^T x \\ & ||x||^2 \leq 1. \end{aligned} \tag{4}$$

The (TRS) problems are widely used and studied in the literature [8] and they can be solved efficiently using the exiting eigenvalue approaches even for large instances [1,16]. Now, the ADMM algorithm for solving (m-eTRS) can be outlined as follows.

ADMM Algorithm for solving (m-eTRS)

Input parameters $tol > 0$, maxiter> 0. Choose appropriate penalty parameter $\rho > 0$ and $\gamma > 0$. Set $k = 0$ and choose appropriate x^k and λ^k
For $k = 1, \cdots,$ maxiter **do**
Solve quadratic optimization problem (3) and let its solution be z^{k+1}.
Solve the (TRS) problem (4) and let its solution be x^{k+1}.
If $||x^{k+1} - z^{k+1}|| \leq tol$, then exit with x^{k+1} as output.
end if
Set $\lambda^{k+1} = \lambda^k + \gamma\rho(x^{k+1} - z^{k+1})$ and $k = k + 1$.
end for.

As we see, the method is very easy to implement and subproblems of the Steps 1 and 2 are efficiently solvable even for large instances. In what follows, we discuss the convergence of the above algorithm to the stationary point of (m-eTRS). First we present the following lemma.

Lemma 1. *Suppose that* $\{\lambda^k\}$ *is bounded and* $\sum_{k=1}^{\infty} ||\lambda^{k+1} - \lambda^k||^2 < \infty$. *Then*

$$||x^{k+1} - x^k|| \to 0, \quad ||z^{k+1} - z^k|| \to 0 \ \text{ as } \ k \to \infty.$$

Proof. Since x^{k+1} solves problem (4) at k-th iteration and $x^k - x^{k+1}$ is a feasible direction with respect to the feasible region of (4), then

$$\nabla_x L(x^{k+1}, z^{k+1}, \lambda^k)^T (x^k - x^{k+1}) \geq 0. \tag{5}$$

Now

$$L(x^k, z^{k+1}, \lambda^k) - L(x^{k+1}, z^{k+1}, \lambda^k) = \frac{1}{2}(x^k - x^{k+1})^T(A + \rho I)(x^k - x^{k+1})$$
$$+\nabla_x L(x^{k+1}, z^{k+1}, \lambda^k)^T(x^k - x^{k+1}) \geq \frac{\lambda_1 + \rho}{2}||x^k - x^{k+1}||^2, \quad (6)$$

where the inequality follows from the definition of the smallest eigenvalue of A, λ_1, and (5). We also have

$$L(x^k, z^k, \lambda^k) - L(x^k, z^{k+1}, \lambda^k) \geq 0, \quad (7)$$

as z^{k+1} is the minimizer of $L(x^k, z, \lambda^k)$. On the other hand

$$\begin{aligned} L(x^{k+1}, z^{k+1}, \lambda^k) - L(x^{k+1}, z^{k+1}, \lambda^{k+1}) &= (\lambda^k - \lambda^{k+1})^T(x^{k+1} - z^{k+1}) \\ &= -\frac{1}{\gamma\rho}||\lambda^k - \lambda^{k+1}||^2. \end{aligned} \quad (8)$$

Now using (5), (7), and (8) we have

$$\begin{aligned} L(x^k, z^k, \lambda^k) \; -L(x^{k+1}, z^{k+1}, \lambda^{k+1}) &= L(x^k, z^k, \lambda^k) - L(x^k, z^{k+1}, \lambda^k) + \\ L(x^k, z^{k+1}, \lambda^k) - L(x^{k+1}, z^{k+1}, \lambda^k) &+ L(x^{k+1}, z^{k+1}, \lambda^k) - L(x^{k+1}, z^{k+1}, \lambda^{k+1}) \\ &\geq \frac{\lambda_1 + \rho}{2}||x^{k+1} - x^k||^2 - \frac{1}{\gamma\rho}||\lambda^{k+1} - \lambda^k||^2. \end{aligned} \quad (9)$$

Since $\{\lambda^k\}$ and $\{x^k\}$ are bounded, then from Step 3 of ADMM iterations, $\{z^k\}$ also is bounded. Therefore $\{L(x^k, z^k, \lambda^k)\}$ is bounded. Moreover, since by assumption $\sum_{k=1}^{\infty}||\lambda^{k+1} - \lambda^k||^2 < \infty$, thus from (9), $\sum_{k=1}^{\infty}||x^{k+1} - x^k||^2$ is a bounded series (in the sense that the sequence of partial sums is bounded)with nonnegative terms, thus it is convergent. Then $||x^k - x^{k+1}|| \to 0$, as $k \to \infty$. Moreover since by assumption $||\lambda^k - \lambda^{k+1}|| \to 0$, as $k \to \infty$, from the Step 3 we have $x^k - z^k \to 0$, as $k \to \infty$. Finally since

$$z^k - z^{k+1} = z^k - x^k + x^k - x^{k+1} + x^{k+1} - z^{k+1}$$

and we know $z^k - x^k \to 0$, $x^k - x^{k+1} \to 0$, $x^{k+1} - z^{k+1} \to 0$, as $k \to \infty$, then $z^k - z^{k+1} \to 0$. □

It should be noted the boundedness assumption of multipliers in the lemma is a standard assumption for convergence analysis of nonconvex optimization problems [14]. Similar assumptions are used to prove the convergence to the stationary point in [21, 22]. In what follows we prove the convergence to the first order stationary conditions

Theorem 1. *Let (x^*, z^*, λ^*) be any accumulation point of $\{(x^k, z^k, \lambda^k)\}$ generated by the ADMM Algorithm. Then by boundedness of $\{\lambda^k\}$ and $\sum_{k=1}^{\infty}||\lambda^{k+1} - \lambda^k||^2 < \infty$, x^* satisfies the first order stationary conditions.*

Proof. Since (x^*, z^*, λ^*) is an accumulation point of $\{(x^k, z^k, \lambda^k)\}$, then there exists a subsequence $\{(x^k, z^k, \lambda^k)\}_{k\in I}$ that converges to (x^*, z^*, λ^*). Now

consider subproblems that should be solved in Steps 1 and 2. As we mentioned, subproblem (3) in Step 1 is a convex quadratic optimization problem which its necessary and sufficient optimality conditions are as follows:

$$\begin{aligned}
&\rho z^{k+1} - (\lambda^k + \rho x^k) + \sum_{i=1}^{m} \nu_i^{k+1} b_i = 0, \\
&\nu_i^{k+1}(b_i^T z^{k+1} - \beta_i) = 0,\ b_i^T z^{k+1} \le \beta_i, \quad i = 1, \cdots, m,
\end{aligned} \tag{10}$$

where ν_i^{k+1}'s are the Lagrange multipliers. Moreover, subproblem in Step 2 is a (TRS) as given in (4) that we have the following necessary and sufficient optimality conditions for it:

$$\begin{aligned}
&(A + \rho I_n + 2\mu^{k+1} I_n) x^{k+1} = -(a + \lambda^k - \rho z^{k+1}), \\
&\mu^{k+1}(||x^{k+1}||^2 - 1) = 0,\ ||x^{k+1}||^2 \le 1, \\
&A + \rho I_n + 2\mu^{k+1} I_n \succeq 0_{n\times n}.
\end{aligned} \tag{11}$$

Now by taking the limit of both (10) and (11), we get

$$\begin{aligned}
&(A + \rho I_n + 2\mu^* I_n) x^* = -(a + \lambda^* - \rho z^*), \\
&\mu^*(||x^*||^2 - 1) = 0,\ ||x^*||^2 \le 1, \\
&A + \rho I_n + 2\mu^* I_n \succeq 0_{n\times n}, \\
&\rho z^* - (\lambda^* + \rho x^*) + \sum_{i=1}^{m} \nu_i^* b_i = 0, \\
&\nu_i^*(b_i^T z^* - \beta_i) = 0,\ b_i^T z^* \le \beta_i, \quad i = 1, \cdots, m.
\end{aligned} \tag{12}$$

From the first and forth equations of (12), we get

$$(A + 2\mu^* I_n) x^* = -a - \sum_{i=1}^{m} \nu_i^* b_i,$$

which with the second and the last equations are the first order stationary conditions. □

3 Numerical experiments

In this section, we present several randomly generated test problems to assess the performance of ADMM for solving (m-eTRS). For small dimension problems, we compare ADMM with the SOCP/SDP relaxation (R_m). All computations are performed in MATLAB R2015a on a 2.50 GHz laptop with 4 GB of RAM. To solve the SOCP/SDP reformulation, we have used CVX 1.2.1. For all test problems, we set $tol = 10^{-6}$ and maxiter= 100. Our exten-

Table 1: Comparison between ADMM and SOCP/SDP relaxation for the first class of test problems with density=0.1.

Dimension	Algorithm	Time (s)	
n=100	ADMM	1.4	
	SOCP/SDP	1.9	
n=300	ADMM	3.2	
	SOCP/SDP	27.4	
n=500	ADMM	4.8	
	SOCP/SDP	163.5	

sive testing showed that $\gamma = 0.9$ and $\rho = -2\lambda_1 + 1$ are appropriate choices. Moreover, we set $\lambda^0 = 2e$, where e denotes the all one vector and $x^0 = \frac{e}{\sqrt{n}}$. Finally, to solve the (TRS) problems within the algorithm we have used the algorithm in [1] and to solve the quadratic optimization problems we have used the 'quadprog' command of MATLAB.

- **First class of test problems:**
  For this class we consider $m = 2$ in (m-eTRS), $A = sprandsym(n, density)$, $a = randn(n, 1)$, $b_1 = e_1$, $b_2 = -e_1$, the unit vector in R^n and $\beta_1 = 0.1$ and $\beta_2 = 0.1$. As we see, the two linear inequality constraints are parallel, then the relaxation in [7] is exact. Results are summarized in Tables 1 to 3 for the average of 10 runs. In Tables 1 and 2 we compare ADMM with the SOCP/SDP relaxation for different densities. As we see, ADMM is much faster than SOCP/SDP relaxation. Moreover, for these two tables beside the time, we also have computed the relative objective function difference which is always below $O(10^{-8})$ for all test problems. This numerically verifies that ADMM converges to the same solution as SOCP/SDP relaxation does. In Table 3 we just report the results of ADMM as the relaxation is not applicable for these problems. In this table, KKT1 denotes the first order stationary condition, namely $||(A + 2\mu^* I_n)x^* + a + \sum_{i=1}^{m} \nu_i^* b_i||$.

- **Second class of test problems:**
  For this class we consider $m = 5$, the number of linear inequalities. As before, we consider $A = sprandsym(n, density)$ and $a = randn(n, 1)$ and consider $b = rand(n, m)$, $x = randn(n, 1)$ and $\beta = \frac{b^T x}{||x||}$. Obviously, the feasible region of (m-eTRS) is nonempty. Results of applying ADMM and SOCP/SDP relaxation to this class are summarized in Tables 4 to 6. Similar observations to the previous class hold here as well.

Table 2: Comparison between ADMM and SOCP/SDP relaxation for the first class of test problems with density=0.01.

Dimension	Algorithm	Time (s)	
n=100	ADMM	0.75	
	SOCP/SDP	1.1	
n=300	ADMM	2.6	
	SOCP/SDP	26.7	
n=500	ADMM	4.1	
	SOCP/SDP	141.5	

Table 3: Results of ADMM for the first class of test problems with density=0.001.

Dimension	KKT1	Time (s)	
n=3000	8.78×10^{-14}	21.5	
n=5000	5.51×10^{-14}	37.7	
n=8000	1.03×10^{-13}	70.8	

Table 4: Comparison between ADMM and SOCP/SDP relaxation for the second class of test problems with $m = 5$ and density=0.1.

Dimension	Algorithm	Time (s)	
n=100	ADMM	2.5	
	SOCP/SDP	12.4	
n=300	ADMM	3.9	
	SOCP/SDP	345.6	
n=500	ADMM	5.2	
	SOCP/SDP	1235.8	

Table 5: Comparison between ADMM and SOCP/SDP relaxation for the second class of test problems with $m = 5$ and density=0.01.

Dimension	Algorithm	Time (s)	
n=100	ADMM	2.1	
	SOCP/SDP	7.4	
n=300	ADMM	3.3	
	SOCP/SDP	287.3	
n=500	ADMM	4.5	
	SOCP/SDP	933.4	

Table 6: Results of ADMM for the second class of test problems with $m = 5$ and density=0.001.

Dimension	KKT1	Time (s)	
n=3000	9.83×10^{-14}	37.6	
n=5000	7.90×10^{-14}	56.9	
n=8000	3.56×10^{-14}	110.7	

4 Conclusions

In this paper, we have applied ADMM for solving the extended trust region subproblem. The convergence of the method to the first order stationary conditions is established and the quality of the solutions for small dimensions are compared with the known SOCP/SDP relaxation showing that ADMM converges to the global solution in significantly shorter time. Moreover, several large instances also are solved by ADMM to show its capability. The second order convergence analysis of the method is an interesting future research direction which one may follow.

5 Acknowledgments

The authors would like to thank both reviewers for their useful comments and questions which improved the paper and University of Guilan for supporting this research.

References

1. Adachi, S., Iwata, S., Nakatsukasa, Y. and Takeda, A. *Solving the trust region subproblem by a generalized eigenvalue problem,* Mathematical Engineering Technical Report(METR 2015-14), Department of Mathematical Informatics, Graduate School of Information Science and Technology, The University of Tokyo.

2. Bai, X. and Scheinberg, K. *Alternating direction methods for non convex optimization with applications to second-order least-squares and risk parity portfolio selection,* Optimization-Online, 2015.

3. Bai, X., Sun, J., Sun, S. and Zheng, X. *An alternating direction method for chance-constrained optimization problems with discrete distributions,* Optimization-Online, 2012.

4. Beck, A. and Eldar, Y.C. *Strong duality in nonconvex quadratic optimization with two quadratic constraints,* SIAM Journal on Optimization, 17(3), 844-860, 2006.

5. Boyd, S., Parikh, N., Chu, E., Peleato, B. and Eckstein, J. *Distributed optimization and statistical learning via the alternating direction method of multipliers,* Foundations and Trends in Machine Learning, 3(1), 1-122, 2011.

6. Burer,S. and Anstreicher, K.M. *Second-order-cone constraints for extended trust-region subproblems,* SIAM Journal on Optimization, 23(1), 432-451, 2013.

7. Burer,S. and Yang, B. *The trust region subproblem with non-intersecting linear constraints,* Mathematical Programming, 149, 253-264, 2015.

8. Conn, A.R., Gould, N.I. and Toint, P.L. *Trust Region Methods,* SIAM, Philadelphia, PA, 2000.

9. Grant, M. and Boyd, S. *CVX: Matlab software for disciplined convex programming,* version 2.0 beta. http://cvxr.com/cvx, September 2013.

10. Fortin, C. and Wolkowicz, H. *The trust region subproblem and semidefinite programming,* Optimization Methods and Software, 19(1), 41-67, 2004.

11. Hsia, Y. and Sheu, R.L. *Trust region subproblem with a fixed number of additional linear inequality constraints has polynomial complexity,* arXiv preprint arXiv, 1312.1398, 2013.

12. Hong, M. , Luo, Z.Q. and Razaviyan, M. *Convergence analysis of alternating direction method of multipliers for a family of nonconvex problems,* SIAM Journal on Optimization, 26(1), 337-364, 2016.

13. Jeyakumar, V. and Li, G.Y. *Trust-region problems with linear inequality constraints: exact SDP relaxation, global optimality and robust optimization,* Mathematical Programming, 147, 171-206, 2014.

14. Luo, H., Sun, X. and Wu, H. *Convergence properties of augmented Lagrangian methods for constrained global optimization,* Optimization Methods and Software, 23(5), 763-778, 2008.

15. Martinez, J.M. *Local minimizers of quadratic functions on Euclidean balls and spheres,* SIAM Journal on Optimization, 4(1), 159-176, 1994.

16. Rendl, F. and Wolkowicz, H. *A semidefinite framework for trust region subproblems with applications to large scale minimization,* Mathematical Programming, 77(1), 273-299, 1997.

17. Salahi, M. and Fallahi, S. *Trust region subproblem with an additional linear inequality constraint,* Optimization Letters, 10(4), 821-832, 2016.

18. Salahi, M. and Taati, A. *A fast eigenvalue approach for solving the trust region subproblem with an additional linear inequality,* Compuational and Applied Mathematics, DOI: 10.1007/s40314-016-0347-3, 2016.

19. Shen, Y., Wen, Z. and Zhang, Y. *Augmented Lagrangian alternating direction method for matrix separation based on low-rank factorization,* Optimization Methods and Software, 29(2), 239-263, 2014.

20. Sturm, J.F. and Zhang, S. *On cones of nonnegative quadratic functions,* Mathematics of Operations Research, 28(2), 246-267, 2003.

21. Xu, Y., Yin, W., Wen, Z. and Zhang, Y. *An alternating direction algorithm for matrix completion with nonnegative factors,* Journal of Frontiers of Mathematics in China, Special Issues on Computational Mathematics, 365-384, 2011.

22. Xu, L., Yu, B. and Zhang, Y. *An alternating direction and projection algorithm for structure-enforced matrix factorization,* Technical Reports, Department of Computational and Applied Mathematics, Rice University, 2013.

Persian Translation of Abstracts

محاسبه ی مقادیر ویژه ی مسائل اشتورم-لیوویل مرتبه ی چهار به روش گروه لی

حنیف میرزایی

تبریز، دانشگاه صنعتی سهند، دانشکده علوم پایه، گروه ریاضی

دریافت مقاله ۱۲ اسفند ۱۳۹۳، دریافت مقاله اصلاح شده ۱۹ اسفند ۱۳۹۴، پذیرش مقاله ۸ خرداد ۱۳۹۵

چکیده : در این مقاله، مسئله اشتورم-لیوویل مرتبه چهار را به صورت یک معادله دیفرانسیل ماتریسی در گروه لی فرمول بندی می کنیم. با حل این معادله ی ماتریسی به روش بسط مگنوس مقادیر ویژه ی مسئله محاسبه می گردند. بسط مگنوس یک سری نامتناهی از انتگرالهای چندگانه ی براکتهای لی می باشد. روش تقریب، قطع سری به تعداد متناهی جمله و سپس محاسبه ی انتگرالها به روش گاوس می باشد. در پایان چند مثال عددی آورده شده است.

کلمات کلیدی : روش گروه لی؛ مسئله اشتورم-لیوویل مرتبه ی چهار؛ بسط مگنوس.

تحليل پايداري سيستم هاي كسري منطبق شدني

هادي رضازاده[۱،۲]، *حسين اميني خواه*[۱] و *اميرحسين رفاهي شيخاني*[۳]

[۱] دانشگاه گيلان، دانشكده علوم رياضي، گروه رياضي كاربردي
[۲] دانشگاه تخصصي فناوري هاي نوين آمل، دانشكده فناوري هاي مهندسي
[۳] دانشگاه آزاد اسلامي واحد لاهيجان، گروه رياضي كاربردي

دريافت مقاله ۳۰ ارديبهشت ۱۳۹۴، دريافت مقاله اصلاح شده ۱۹ اسفند ۱۳۹۴، پذيرش مقاله ۴ خرداد ۱۳۹۵

چكيده : در اين مقاله، به بررسي تجزيه و تحليل پايداري سيستم هاي ديفرانسيل كسري شامل مشتق كسري منطبق شدني مي پردازيم. برخي از شرايط پايداري سيستم هاي ديفرانسيل كسري با استفاده از تابع نمايي كسري و تبديل لاپلاس كسري ارائه شده است. به علاوه براي نشان دادن اثربخشي و كارايي فرآيند ذكرشده، پايداري دستگاه كسري منطبق شدني لوتكا-ولترا را با روش چندگامي آشفتگي هوموتوپي بررسي مي كنيم.

كلمات كليدی : تحليل پايداری؛ پايداری مجانبی؛ مشتق كسری منطبق شدنی؛ دستگاه لوتكا-ولترا .

روش تکرار پارامتریک بهینه برای حل معادلات نوع Lane-Emden

رضا چهارپاشلو

دزفول، دانشگاه صنعتی جندی شاهپور، دانشکده علوم پایه

دریافت مقاله ۲۶ شهریور ۱۳۹۳، دریافت مقاله اصلاح شده ۲۷ مهر ۱۳۹۴، پذیرش مقاله ۲ مرداد ۱۳۹۵

چکیده : در این مقاله روش تحلیلی ای به نام روش تکرار پارامتریک (PIM) به منظور حل IVP های منفرد درجه دوم نوع Lane-Emden ارائه شده است و همگرایی داخلی اش مورد بحث قرار گرفته است. از آنجایی که دارا بودن راه حل تحلیلی تقریبی برای توصیف معادلات نوع Lane-Emden بویژه برای مواردی که راه حل های دقیقی برای شان موجود نیست، مفید است بنابراین، یک اصلاح (پیشرفت) موثر PIM نیز پیشنهاد شده است که قادر است راه حل تحلیلی تقریبی ای را بدست آورد.در نهایت، PIM اصلاح شده به عنوان الگوریتمی در دنباله ی فواصل برای یافتن راه حل تقریبی دقیق معادلات غیرخطی نوع Lane-Emden مورد توجه قرار می گیرد. همچنین نشان می دهیم که چگونه یک میزان بهینه ی تقریبی پارامتر افزایش همگرایی را در چارچوب این روش شناسایی می کنیم. مثال هایی ارائه شده است تا کارایی و درستی روش پیشنهادی را نشان دهد.

کلمات کلیدی : روش تکرار پارامتری piecewise-truncated ؛ روش تکرار پارامتریک truncated (برشی)؛ روش تکرار پارامتریک؛ معادلات غیرخطی نوع Lane-Emdeb .

مفهوم جواب B-موثر در مسائل بهینه‌سازی چندهدفه منصف

داود فروتن نیا [۱] *و علی محمودی نژاد* [۲]

[۱] دانشگاه ولیعصر رفسنجان، گروه ریاضی
[۲] دانشگاه صنعتی سیرجان، گروه ریاضی

دریافت مقاله ۱۱ بهمن ۱۳۹۳، دریافت مقاله اصلاح شده ۲۹ خرداد ۱۳۹۵، پذیرش مقاله ۲ مرداد ۱۳۹۵

چکیده : مسئله‌ای که گاهی مواقع در بهینه‌سازی چندهدفه رخ می‌دهد وجود مجموعه‌ای بزرگ از جواب‌های موثر منصف است. از اینرو تصمیم گیری مبنی بر انتخاب جواب ترجیح داده شده منحصربفرد مشکل است. مدل‌های درنظرگرفته شده با B-موثری منصف، کمی از بار تصمیم‌گیری را به وسیله کوچک کردن مجموعه جواب کم می‌کند، چون برای هر مسئله مجموعه جواب‌های B-موثر منصف زیرمجموعه‌ای از جواب‌های موثر منصف است. در این مقاله، ابتدا بعضی از مفاهیم تئوری و عملی جواب‌های B-موثر منصف بحث شده است. سپس، بعضی از تکنیک‌های عددی‌سازی برای تولید جواب‌های B-موثر منصف توسعه داده شده است

کلمات کلیدی : بهینه‌سازی منصف؛ غیر تسلطی؛ منصفانه؛ B-موثری؛ عددی‌سازی.

مساله مکانیابی وبر مربعی آرمانی

جعفر فتحعلی [۱] *و علی جمالیان* [۲]

[۱] دانشگاه صنعتی شاهرود، دانشکده علوم ریاضی
[۲] دانشگاه گیلان، دانشکده علوم ریاضی

دریافت مقاله ۲۳ دی ۱۳۹۴، دریافت مقاله اصلاح شده ۲۹ خرداد ۱۳۹۵، پذیرش مقاله ۲۴ شهریور ۱۳۹۵

چکیده : دراین مقاله به حالت خاصی از مساله مکانیابی وبر می پردازیم که آن را مساله مکانیابی آرمانی می نامیم. در مساله مکانیابی وبر هدف پیدا کردن نقطه ای در صفحه به گونه ای است که مجموع وزنی فاصله ها بین این نقطه و n نقطه موجود کمینه شود. در مساله مکانیابی آرمانی هر نقطه موجود مانند P_i یک شعاع متناظر r_i دارد و حالت ایده آل وقتی است که سرویس دهنده جدید در فاصله r_i از نقطه P_i به ازای $i = ۱, ۲, \cdots n$ قرار گیرد. اما واضح است که در اغلب مسائل چنین نقطه ای ممکن است وجود نداشته باشد. لذا ما سعی میکنیم مجموع وزنی خطای مربعی از نقطه ایده آل را کمینه کنیم. در این مقاله فاصله ها با نرم اقلیدسی درنظر گرفته می شوند . الگوریتمی شبیه روش وایزفلد و دو الگوریتم به روش بهینه سازی پرندگان برای مساله ارائه کرده و نتایج حاصل از آنها را با روش مربع بزرگ، مربع کوچک مقایسه می کنیم.

کلمات کلیدی : نظریه مکانیابی؛ الگوریتم وایزفلد؛ بهینه سازی پرندگان .

رویکردی قابل اطمینان برای اختتام روش بهینه سازی الگوریتم ژنتیک

لیلا لطفی کتولی و اکبر شاهسوند

دانشگاه فردوسی مشهد، دانشکده مهندسی، گروه مهندسی شیمی

دریافت مقاله ۴ مرداد ۱۳۹۴، دریافت مقاله اصلاح شده ۳۰ آبان ۱۳۹۴، پذیرش مقاله ۷ مهر ۱۳۹۵

چکیده : در دهه های اخیر، الگوریتم ژنتیک به شکل گسترده ای برای حل بسیاری از مسائل بهینه سازی در زمینه های مختلف علوم و مهندسی مورد استفاده قرار گرفته است. در اکثر موارد تعداد تکرارها تنها معیار برای متوقف کردن الگوریتم ژنتیک می باشد. ذر عمل، استفاده از این معیار موجب طولانی شدن زمان اجرای الگوریتم جهت رسیدن به نتیجه مطلوب می گردد. در این مقاله، رویکرد جدیدی با عنوان ''تعداد تقریبی تکرار های ضروری'' ارائه شده است که می تواند موجب توقف موفقیت آمیز روش بهینه سازی الگوریتم ژنتیک در کمترین زمان ممکن شود. همچنین، دو رابطه کاربردی ساده جهت ارتباط پارامتر جدید با درجه آزادی تقریبی برای توابع هدف مورد نظر ارایه گردیده است. برای توابع هدف پیچیده تر می توان از یک برازشگر خطی مانند شبکه عصبی موسوم به رگولاریزاسیون جهت محاسبه مقدار تقریبی درجه آزادی مربوطه استفاده نمود. به منظوراعتبارسنجی موفق رویکرد مطرح شده از چهار مثال موردی مناسب استفاده گردیده است که در تمامی موارد روش پیشنهادی با استفاده از روابط مذکور قادر به یافتن نقطه بهینه مربوطه شده است. بدلیل سادگی و از آنجایی در رابطه خطی ارائه شده، محور افقی نشانگر مقدار تقریبی (نه دقیق) درجه آزادی می باشد، استفاده از آن ارجح تر به نظر می رسد. به وضوح می توان دید که شبکه عصبی رگولاریزاسیون می تواند با موفقیت خطای موجود در داده های واقعی را حذف نموده و سطح چند بعدی حقیقی را از میان مجموعه ای از داده های آغشته به خطا بازسازی نماید.

کلمات کلیدی : ژنتیک؛ شرط اختتام؛ درجه آزادی تقریبی؛ تعداد تقریبی تکرار های ضروری؛ برازشگر خطی؛ شبکه عصبی رگولاریزاسیون.

روش جهت متناوب ضرایب برای زیر مساله ناحیه اطمینان توسیع یافته

مازیار صلاحی و اکرم طاعتی

دانشگاه گیلان، دانشکده علوم ریاضی، گروه ریاضی کاربردی

دریافت مقاله ۱۸ آذر ۱۳۹۴، دریافت مقاله اصلاح شده ۱۰ مرداد ۱۳۹۵، پذیرش مقاله ۷ مهر ۱۳۹۵

چکیده : اخیراً زیرمساله ناحیه اطمینان توسیع یافته در تحقیقات متعددی مورد توجه قرار گرفته است. تحت مفروضات گوناگون، دوگانی قوی و آزاد سازی مخروطی درجه دو-نیمه معین برای کلاس های مختلف آن ارایه شده است. با توجه به اهمیت زیر مساله ناحیه اطمینان توسیع یافته، در این مقاله بدون در نظر گرفتن هیچ فرضی روی مساله، روش جهت متناوب ضرایب را که بسیار مورد استفاده قرار گرفته است، برای حل آن بکار می‌بریم. همگرایی روش به شرایط ایستایی مرتبه اول ثابت می‌شود. همچنین روی کلاس های مختلفی از مسایل آزمون، کیفیت جواب محاسبه شده توسط این روش در ابعاد متوسط با آزاد سازی مخروطی درجه دو-نیمه معین مقایسه می شود. علاوه بر این، کاربرد روش در حل مسایل در ابعاد بزرگ با حل چندین مثال مقیاس بزرگ نشان داده می شود.

کلمات کلیدی : زیر مساله ناحیه اطمینان توسیع یافته؛ روش متناوب؛ بهینه سازی نامحدب؛ برنامه ریزی نیمه معین؛ برنامه ریزی مخروطی درجه دو.

Aims and scope

Iranian Journal of Numerical Analysis and Optimization (IJNAO) is published twice a year by the Department of Applied Mathematics, Faculty of Mathematical Sciences, Ferdowsi University of Mashhad. Papers dealing with different aspects of numerical analysis and optimization, theories and their applications in engineering and industry are considered for publication.

Journal Policy

After receiving an article, the editorial committee will assign referees. Refereeing process can be followed via the web site of the Journal.
The manuscripts are accepted for review with the understanding that the work has not been published and also it is not under consideration for publication by any other journal. All submissions should be accompanied by a written declaration signed by the author(s) that the paper has not been published before and has not been submitted for consideration elsewhere.

Instruction for Authors

The Journal publishes all papers in the fields of numerical analysis and optimization. Articles must be written in English.
All submitted papers will be refereed and the authors may be asked to revise their manuscripts according to the referee's reports. The Editorial Board of the Journal keeps the right to accept or reject the papers for publication.
The papers with more than one authors, should determine the corresponding author. The e-mail address of the corresponding author must appear at the end of the manuscript or as a footnote of the first page.
It is strongly recommended to set up the manuscript by Latex or Tex, using the template provided in the web site of the Journal. Manuscripts should be typed double-spaced with wide margins to provide enough room for editorial remarks.
References should be arranged in alphabetical order by the surname of the first author as examples below:
[1] Stoer, J. and Bulirsch, R. *Introduction to Numerical Analysis*, Springer-verlag, New York, 2002.
[2] Brunner, H. *A survey of recent advances in the numerical treatment of Volterra integral and integro-differential equations*, J. Comput. Appl. Math. 8 (1982), 213-229.

Submission of Manuscripts

Authors may submit their manuscripts by either of the following ways:

a) Online submission (pdf or dvi files) via the website of the Journal at:

http://ijnao.um.ac.ir

b) Via journal's email mjms@um.ac.ir

Copyright Agreement

Upon the acceptance of an article by the Journal, the corresponding author will be asked to sign a "Copyright Transfer Agreement" (see the web site) and send it to the Journal address. This will permit the publisher to publish and distribute the work.

Reviewers Acknowledgment

Peer reviewers are key to advancing scholarship and contributing to the quality of scholarly journals. We would like to sincerely thank the following reviewers who have taken part in the peer-review process for Iranian Journal of Numerical Analysis and Optimization during the last year.

Abbasbandy, S.
Abdeljawad, T.
Abdi, A.
Adibi, H.
Akhoundi Roshanavand, N.
Akrami, M.H.
Al-Horani, M.
Al-Amr, M. O.
Amini, K
Aminikhah, H.
Arab Ameri, M.
Asif Zahoor Raja, M.
Azarnavid, V.
Aziz, I.
Babu, G.V.R.
Baghani, O.
Barfeie, M.
Baymani, M.
Bazm, S.
Biazar, J.
Beik, F.
Bras, M.
Chen, W.
Darania, P.
Davoodi, N.
de Madrid. A.P.
Erfanian, M.
Ezzati. R.
Ezz-Eldien, S.
Fahuai Yi.
Farzi, J.
Fathali, J.
Fatemi, M.
Fazeli, S.
Gachpazan, M.
Ghanbari, R.
Ghorbani, A.
Ghoreishi, F.
Ghovatmand, M.
Golbabai, A.
Ghomanjani, F.
Goudarzi, M. A.
Hadizadeh Yazdi, M.
Haghighi, A
Hashemi Borzabadi, A.
Hashemi Mehne, S. H.
Hassanpur, H.
Heidari, A.
Hesameddini, S.
Hojjati, G. R.
Hajarian, M.
HinOoi, E.
Hussain Khan, S.
Jafari, H.
Jiang, W.
Kaeppeli, R.
Kaviani, M.A.
Kazem, S.
Khalil, R.
Khojasteh, D.
Khoshsiar, R.
Khoveyni, M.
Kirane, M.
Lakestani, M.
Liu, L.
Mashayekhi, S.
Mirzaee, F.
Mirzaei, D.
Mirzazadeh, M.
Moeen Moghaddas, F.
Mokhtari, R.
Molabahrami, A.
Namjoo, M.
Navidi, H.R.
Nazemi, A.R.
Niculescu, C.P.
Nigmatullin, R.
Ordokhani, Y.

Pakdaman, M.
Parand, K.
Peyghami, M.R.
Pishbin, S.
Rezaiee Pajand, M.
Saadatmandi, A.
Saberi Najafi, J.
Saeedpoor, M.
Safi, M.R.
Sajavicius, S.
Salahi, M.
Salamah, S.A.
Salehi, R.
Salehi Fathabadi, H.
Sevcovic, D.
Shahmorad, S.
Shamsara, E.
Shamsi, M.
Shi, Z.
Shivanian, E.
Solaymani Fard, O.
Soleimani, M.
Sun, W.
Tari, A.
Tavakoli, A.
Temimi, H.
Tohidi, E.
Toutounian, F.
Vahidian Kamyad, A.
Vitanov, N.K.
Yousefi, S. A.
Zaferanieh, M.
Zhang, H.
Zhao, G.
Ziari, S.

Iranian Journal of Numerical Analysis and Optimization

Former : MRJMS

CONTENTS Vol. 7, No.1, pp 1-135, 2017

web site : http://ijnao.um.ac.ir

Email : mjms@um.ac.ir

ISSN : 2423-6977

Serial Number: 11