控制与决策
Control and Decision

求解非线性方程组的智能优化算法综述

高卫峰，罗宇婷，原杨飞

求解非线性方程组的智能优化算法综述

高卫峰†，罗宇婷，原杨飞

(西安电子科技大学 数学与统计学院，西安 710126)

摘　要: 非线性方程组的求解是优化领域的一个重要研究课题. 近年来,利用智能优化算法求解非线性方程组已成为一个重要方向. 首先介绍非线性方程组的定义;其次,根据智能优化算法求解非线性方程组问题的基本框架,从转化方法和智能优化算法两方面入手,对求解非线性方程组的算法的研究进展进行归纳总结;再次,对非线性方程组的测试函数及评价指标进行描述,对比了 5 个具有代表性算法的性能,分析了目前利用智能优化算法求解非线性方程组亟待解决的问题;最后,指出值得进一步研究的方向.

关键词: 非线性方程组；转化方法；智能优化算法；多目标优化

中图分类号: TP18　　**文献标志码:** A

DOI: 10.13195/j.kzyjc.2020.0379　　开放科学 (资源服务) 标识码 (**OSID**):

引用格式: 高为峰,罗宇婷,原杨飞. 求解非线性方程组的智能优化算法综述[J]. 控制与决策, 2021, 36(4): 769-778.

Overview of intelligent optimization algorithms for solving nonlinear equation systems

GAO Wei-feng†, *LUO Yu-ting*, *YUAN Yang-fei*

(School of Mathematics and Statistics，Xidian University，Xi’an 710126，China)

Abstract: Solving nonlinear equation systems is an important research topic in optimization field. In recent years, using intelligent optimization algorithms to solve nonlinear equation systems has become an important research area. In this paper, the definition of nonlinear equation systems is firstly introduced. And then, based on the basic principles of intelligent optimization algorithms for solving nonlinear equations, state-of-the-art algorithms of solving nonlinear equation systems are surveyed from the aspects of transformation methods and intelligent optimization algorithms. In addition, the benchmark test functions and performance criteria of nonlinear equation systems are described, and the performances of five representative algorithms are compared, meanwhile, the problems that need to be solved are analyzed. Finally, the open research issues in this field are pointed out.

Keywords: nonlinear equation systems；transformation methods；intelligent optimization algorithms；multiobjective optimization

0　引　言

在电力系统、机械制造、神经网络、模式识别、生产调度、网络通信、投资组合、图像处理等诸多领域[1-2] 都会涉及到非线性方程组 (nonlinear equation systems, NESs), 因而 NESs 的求解成为一个十分重要的研究课题.

目前, 求解 NESs 的算法有很多, 可以简单地划分为传统优化算法和智能优化算法这两类. 其中传统优化算法求解 NESs 通常是基于梯度信息的迭代方法[3-4], 例如: 共轭梯度法、牛顿法、拟牛顿法、最速下降法等. 这些方法依赖于初始点的选取, 如果初始点选取不当, 则可能找不到根. 由于需要梯度信息, 只适用于可微函数, 而且容易陷入局部最优解. 更重要的是, 这些方法在一次运行中仅能找到一个根. 而智能优化算法是一种基于群体的优化算法, 从多个点出发同时进行搜索, 而不是单个点搜索, 具有隐形并行性. 而且对初始点要求不高, 对不具有可微性 NESs 依然适用, 求解范围广, 具有高效性、鲁棒性等特点, 极大地方便了对 NESs 的求解, 解决了传统优化算法的局限性. 因此, 利用智能优化算法求解 NESs 受到越来越多的关注, 近年来已成为国内外学者的一个研究热点.

收稿日期: 2020-04-03；**修回日期:** 2020-05-25.
基金项目: 国家自然科学基金项目 (61772391).
责任编委: 林崇.
†通讯作者. E-mail: gaoweifeng2004@126.com.

近10年来,在求解NESs的领域出现了许多智能优化算法. 吴龙等[5]在2014年总结了利用遗传算法求解NESs的各种方法,但对求解NESs的其他算法及最近几年出现的算法讨论较少. 鉴于最近几年NESs的求解算法发展迅猛,本文认为有必要对该领域的最新研究成果进行全面的介绍和探讨,并对目前求解NESs中存在的挑战和问题进行描述.

本文首先介绍NESs的定义及利用智能优化算法求解NESs的基本框架;然后从NESs的转化方法和智能优化算法两方面出发,对利用智能优化算法求解NESs的研究及其进展进行综述;此外,对NESs的评价指标及实验结果进行比较分析;对存在的主要挑战进行了探讨;最后,指出进一步研究的方向.

1 NESs问题

1.1 NESs的定义

不失一般性,NESs可以定义为如下形式:

$$\begin{cases} e_1(X)=0, \\ e_2(X)=0, \\ \quad\vdots \\ e_m(X)=0. \end{cases} \tag{1}$$

其中: $X=(x_1,x_2,\ldots,x_n)\in S$是$n$维的决策向量; $S\subseteq R^n$表示决策空间,它满足如下约束条件:

$$S=\prod_{i=1}^{n}[L_i,U_i], \tag{2}$$

L_i和U_i分别代表x_i的下界和上界; $e_j(X)=0$ $(j\in\{1,2,\ldots,m\})$表示第j个方程,m是方程的个数.

如果$e_j(X^*)=0$对于任意的$j\in\{1,2,\ldots,m\}$均成立,则X^*是问题(1)的一个根. 通常,一个NES有多个根,例如下面这个NES:

$$\begin{cases} x_1-\cos(4\pi x_2)=0, \\ x_1^2+x_2^2=1, \\ -1\leqslant x_1,x_2\leqslant 1. \end{cases} \tag{3}$$

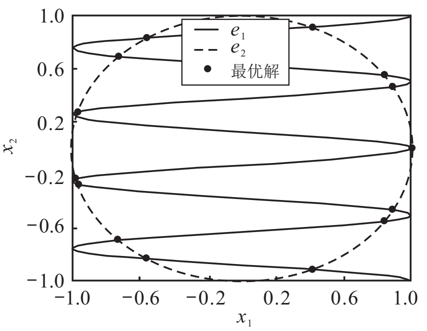

图1 一个非线性方程组有多个根的实例

从图1很容易看出,这个NES有15个根. 利用智能优化算法求解NESs的主要任务是在一次运行中找到所有根,以便于用户根据实际问题进行选择.

1.2 利用智能优化算法求解NESs的基本框架

Song等[6]总结出了利用智能优化算法求解NESs的基本框架,如图2所示.

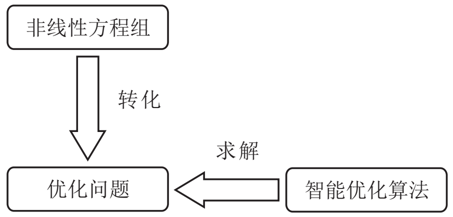

图2 利用智能优化算法求解非线性方程组的框图

容易看出,利用智能优化算法求解NESs大体可以分为两步:

1) 将一个NESs转化成一种优化问题(例如:单目标优化、约束优化、多目标优化等);

2) 利用智能优化算法求解转化后的优化问题.

值得注意的是,转化后的优化问题与原NESs拥有相同的最优解(即,问题(1)的一个根).

2 NESs的转化方法

近年来,研究者提出了多种将NESs转化为优化问题的方法. 根据目标函数个数的不同和有无约束条件,本文将它们划分为以下3类: 1)单目标转化方法; 2)约束转化方法; 3)多目标转化方法.

2.1 单目标转化方法

单目标转化方法因其简单易执行而得到广泛应用,主要分为平方和法和绝对值法两种形式. 平方和法的一般形式为

$$\min\sum_{i=1}^{m}(e_i(X))^2, \tag{4}$$

绝对值法的一般形式为

$$\min\sum_{i=1}^{m}|e_i(X)|. \tag{5}$$

2.2 约束转化方法

由于智能优化算法本质上是一种无约束的搜索算法,用它来求解约束优化问题时,需要与一定的约束处理技术相结合. 因此,把NESs转化成约束优化问题来求解的研究不是很多,主要有以下两种形式的转化:

$$\begin{cases} \min\ \sum_{i\in S_1}|e_i(X)|, \\ \text{s.t. } e_j(X)\geqslant 0,\ j\in S_2; \end{cases} \tag{6}$$

和

$$\begin{cases} \min\ \sum_{i\in S_1}|e_i(X)|, \\ \text{s.t. } e_j(X)=0,\ j\in S_2. \end{cases} \tag{7}$$

其中: 集合 $S_1 \subseteq \{1,2,\ldots,m\}$, 集合 $S_2 \subseteq \{1,2,\ldots,m\}/S_1$.

2.3 多目标转化方法

多目标转化方法是将NESs转化成多目标优化问题来求解, 在这里先引入多目标优化问题的定义. 不失一般性, 一个多目标优化问题(multiobjective optimization problems, MOPs)可以描述为如下形式(本文考虑最小化情形):

$$\min\ f(X)=(f_1(X),f_2(X),\ldots,f_M(X)). \tag{8}$$

其中: $X=(x_1,x_2,\ldots,x_n)\in S$ 是 n 维的决策向量, $S\subseteq R^n$ 表示决策空间, $f(X)\subseteq Y$ 是目标向量, $Y\subseteq R^M$ 表示目标空间.

值得注意的是, 多目标优化问题中, 多个目标函数之间是相互排斥关系.

在将NESs转化成多目标优化问题时, 通常有转化成多个目标(大于等于3)和双目标两种方法. 2008年, Grosan等[7]提出了一种CA转化方法, 将每个方程视为一个目标函数, 这种方法简单直观, 容易操作, 具体转化形式如下:

$$\begin{cases}\min\ |e_1(X)|,\\ \min\ |e_2(X)|,\\ \quad\vdots\\ \min\ |e_m(X)|.\end{cases} \tag{9}$$

然而, CA的主要缺陷是随着方程个数的增加, 算法的性能会受到严重影响. 当方程个数变得很大时, 这种方法容易陷入“维数灾难”. 为了解决这个问题, Song等[6]提出了一种MONES (multiobjective optimization for nonlinear equation systems)转化方法, 将NESs转化成两个目标函数, 每个目标函数由位置函数和系统函数两部分构成, 具体转化形式如下:

$$\begin{cases}\min\ f_1(X)=x_1+\sum\limits_{i=1}^{m}|e_i(X)|,\\ \min\ f_2(X)=1-x_1+m\cdot\max(|e_1(X)|,\\ \qquad\qquad\ldots,|e_m(X)|),\end{cases} \tag{10}$$

其中 x_1 是一个NES的第1维决策向量.

MONES只需要同时优化两个目标函数, 大大降低了计算的复杂度. 但由于在位置函数中只考虑了一个维度的信息, 在优化过程中可能会丢失一部分最优解. 受到MONES的启发, 2017年, Gong等[8]提出A-Web (weighted biobjective optimization problem)算法, 这种算法在位置函数中给每个决策向量分配了权重, 综合了各个决策向量的信息, 大幅度提高了算法的性能. A-Web这种转化方法的具体定义形式如下:

$$\begin{cases}\min\ f_1(X)=\dfrac{\sum\limits_{j=1}^{n}\omega_j\cdot x_j}{\sum\limits_{j=1}^{n}\omega_j}+\sum\limits_{i=1}^{m}|e_i(X)|,\\ \min\ f_2(X)=1-\dfrac{\sum\limits_{j=1}^{n}\omega_j\cdot x_j}{\sum\limits_{j=1}^{n}\omega_j}+\sum\limits_{i=1}^{m}|e_i(X)|.\end{cases} \tag{11}$$

其中: $x_j(j\in 1,2,\ldots,n)$ 是一个NES的第 j 维决策向量; $\omega=(\omega_1,\omega_2,\ldots,\omega_n)$ 是一组权重向量; ω_j 对应于 x_j 的权重, 它的值是从0到1的一个随机数.

在以上3种转化方法中: 单目标转化方法和约束转化方法由于目标单一, 在优化过程中可能找不到多个最优解, 设计优化算法时要注意增加多样性; 而多目标转化方法在一次运行中能够找到多个最优解.

3 求解转化问题的智能优化算法

智能优化算法作为一种基于群体的搜索算法, 因其优良的性能越来越多地应用于NESs的求解中, 研究者提出了大量的求解NESs的智能优化算法. 下面根据近年来的研究趋势, 本文将它们划分为以下4类: 1) 基于排斥技术的算法; 2) 基于聚类技术的算法; 3) 基于多目标技术的算法; 4) 其他算法. 1)和2)这两类技术主要针对转化后的单目标优化问题, 目的是增加多样性以便找到NESs的多个根; 而3)主要是用来求解转化后的多目标优化问题. 下面根据这4类展开详细介绍.

3.1 基于排斥技术的算法

排斥技术是求多个最优解时常用到的方法, 其主要思想是: 基于目标函数设计一个排斥函数, 在已发现的最优解周围形成排斥区域, 迫使算法在其他区域进行搜索. 排斥技术旨在激发优化算法探索新的搜索区域和寻找新的最优解. 为了便于理解排斥函数在算法运行过程中的实际作用, 首先给出一个具体例子[9].

如图3所示, 图3(a)是原函数 $f(x)=\sin^2(x/\pi)$, $x\in[-10,10]$ 的图像, 图3(b)是排斥函数 $R(x)$ 的图像. 从图3(a)可以看出, $x=0$ 是原函数 $f(x)=0$ 的一个最优解. 假设在算法运行过程中 $x=0$ 这个最优解被找到, 接下来利用排斥函数 $R(x)$, 使 $x=0$ 的附近点函数值增大, 形成一个排斥区域. 之后继续搜索 $f(x)=0$ 的最优解时, 就会在新的区域寻找.

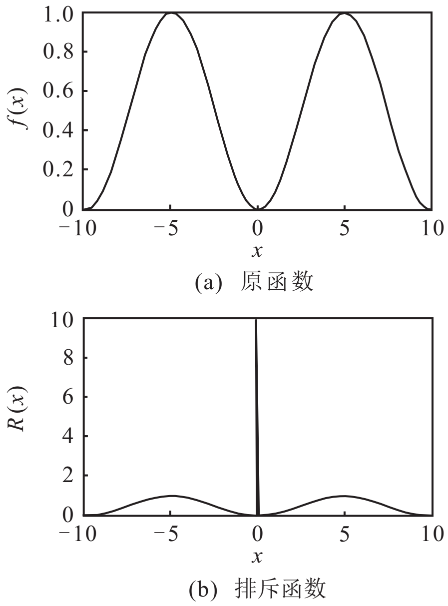

图 3 一个排斥技术的实例

接下来,根据常用到的乘法排斥技术和加法排斥技术两种方法对近些年的研究工作展开介绍.

1) 乘法排斥技术(multiplicative repulsion technique).

Pourjafari 等[10] 提出了如下形式的乘法排斥技术:

$$\min\ R(X)=(f(X)+\varepsilon)\sum_{j=1}^{K}|\coth(\alpha\|X-X_j^*\|)|. \tag{12}$$

其中: K 是已找到最优解的个数; ε 是一个接近于0的正常数; α 是排斥半径,用来调节排斥区域的大小; X_j^* 是已找到的第 j 个最优解.

Ramadas 等[11] 提出了基于误差函数"erf"的另一种乘法排斥技术,具体形式如下:

$$\min\ R(X)=(f(X)+\varepsilon)\sum_{j=1}^{K}\zeta_{\rho'}(\gamma,\|X-X_j^*\|). \tag{13}$$

其中: γ 是度量排斥程度的量, ρ' 是排斥半径.

2) 加法排斥技术(additive repulsion technique).

Hirsch 等[12] 提出了加法排斥技术,用来求解 NESs,具体形式如下:

$$\min\ R(X)=f(X)+\beta\sum_{j=1}^{K}\mathrm{e}^{-\|X-X_j^*\|}\chi_\rho(\|X-X_j^*\|). \tag{14}$$

其中: χ_ρ 表示特征函数; ρ 是一个小的常数,用来调整排斥半径; β 是度量排斥程度的量.

文献[9]将加法排斥技术与差分算法(differential evolution, DE)[13] 相结合,提出了一种新的优化算法 RADE (repulsion-based adaptive differential evolution) 来求解 NESs. 该方法首先将 NESs 转化为单目标优化问题;然后在找到一个最优解时,利用加法排斥技术调整目标函数;在下次迭代过程中增大已找到最优解附近的函数值,使之周围形成排斥区域,迫使算法继续搜索新的区域以寻找新的最优解;为了增加种群的多样性,在原始差分算法的基础上加入邻域变异和拥挤选择这两种多样性机制;最后,考虑到经典差分算法两个控制参数 F 和 CR 影响搜索性能,该方法结合 SHADE[14] 和 JADE[15] 两种差分算法,提出了自适应的参数调整策略以提高 RADE 的搜索性能. 结合这些技术从而提高了算法的性能.

排斥半径在排斥技术中占有举足轻重的地位. 若排斥半径的值设得太大,则排斥区域就会很大,搜索时可能会丢失掉那些距已找到最优解较近的最优解;相反,若排斥半径的值设得太小,则排斥区域就会很小,很多计算资源会浪费在探索已找到最优解附近的区域,导致远离已找到最优解的区域被搜索的概率降低,最终漏掉一些最优解. 上述文献中排斥半径的值是由研究者根据先验知识和实验分析预设的一个固定值,对测试问题的依赖性较大,不利于推广到一般情况中.

针对以上问题, Liao 等[16] 提出了一种基于动态排斥的进化算法. 在迭代过程中,排斥半径的值是随着迭代次数的变化而动态调整的. 算法运行初期,为了对远离已找到最优解的区域进行高效搜索,应设置较大的排斥半径;随着迭代次数的增加,排斥半径逐渐减小,以便精确搜索已找到最优解附近存在的新的最优解. 这种动态的排斥半径避免了试错过程,提高了优化效率. 其次,提出了一种基于动态排斥的进化算法框架(dynamic repulsion-based evolutionary algorithms, DREA),能与不同的进化算法和不同的排斥技术结合. 最后,利用种群重新初始化策略来维持种群的多样性. 该方法极大地提高了找到最优解的概率.

3.2 基于聚类技术的算法

聚类技术因为执行简单而得到了广泛的应用[17],其主要思想是:将一组个体划分进不同的类. 一个好的聚类结果是同一个类中的个体具有很高的相似度,不同类中的个体具有很大的差异. 对于求解 NESs,一些研究者使用聚类技术将候选解划分进不同的类,再从每个类中搜索 NESs 的根.

Tsoulos 等[18] 提出了一种基于聚类的全局搜索

算法来求解NESs的多个根. 首先,将NESs转化成平方和形式的单目标优化问题;然后,利用全局搜索算法Multistart和Minfinder来求解转化问题. Minfinder是一种聚类算法,主要目的是定位多维连续可微函数在决策空间中的所有局部最优解.

Sacco等[19]在设计优化算法求解NESs时,结合了模糊聚类方法(fuzzy clustering means, FCM). 首先,将NESs转化成单目标优化问题;然后,利用设计的优化算法来求解. 该方法在第1阶段,利用Luus-Jaakola算法在整个决策空间中进行搜索,得到大量候选解;第2阶段,为了能定位到多个最优解,利用FCM将候选解进行分类;最后阶段,利用N-M局部搜索算法在每个类中进行搜索,加快种群的收敛.

He等[20]提出了FNODE(fuzzy neighborhood-based differential evolution with orientation)算法,该算法将NESs转化为平方和形式的单目标优化问题. 通过改进模糊邻域技术,根据模糊规则和个体分布选择合适的个体以形成邻域,提高种群的探索能力. 在利用DE搜索阶段,提出了基于方向信息的变异操作,将邻域个体迁移的方向信息融合到变异中以产生有希望的子代.

Liao等[21]首先将NESs转化成单目标优化问题,然后提出了DDE/R (decomposition-based differential evolution with reinitialization)算法来求解转化问题. 为了提高种群多样性,找到NESs的所有根,将分解技术、子种群控制策略和子种群再初始化机制相结合进行求解. 分解技术划分种群进入不同的子种群,子种群控制策略用来提高在个体太少或太多的子种群中优化算法的搜索效率,而子种群再初始化机制可以提高多样性,更好地搜索到新的根. 这3种机制的合理设计增加了找到根的概率.

Liao等[22]又提出了一种基于划分种群思想的NESs求解算法,该算法被称为MENI-EA (memetic niching-based evolutionary algorithms). 首先,将NESs转化为平方和形式的单目标优化问题,在求解过程中,利用小生境技术将整个种群划分为多个子种群,随着算法的运行,个体逐渐向可能包含最优解的不同子种群收敛,如果有满足相应条件的个体,则可以将其视为初始解;然后利用数值方法对它们进行细化,得到NESs精确的根. 该算法将传统数值方法与智能优化算法结合得当,大大提高了求解性能.

尽管聚类技术在NESs的求解中易于实现,但依然存在一些缺陷,最主要的不足是聚类个数的选取较为困难. 若选取的参数过大,由于有些类中可能不存在最优解仍然消耗计算资源在该区域搜索,则导致有些存在最优解的类搜索不到根;若选取的参数过小,由于在局部搜索过程中每个类只会找出一个根,则忽略了多个根被分到同一个类中的情况. 这两种情况都会导致NESs根的丢失.

3.3 基于多目标技术的算法

多目标技术是处理多目标优化问题的常用算法之一[23],其目的在于找到一组Pareto最优解. 近些年,出现了采用多目标技术求解NESs问题的研究成果. 通常,主要有两种多目标技术运用于处理转化后的优化问题:

1) 使用Pareto排序选择个体;

2) 将种群划分为若干个子种群,每个子种群对应一个目标函数,这种机制称为基于群体的方法.

因为随后将要介绍的算法普遍基于NSGA-II (nondominated sorting genetic algorithm II)[24]算法,所以接下来先给出NSGA-II的具体流程示意图,见图4.

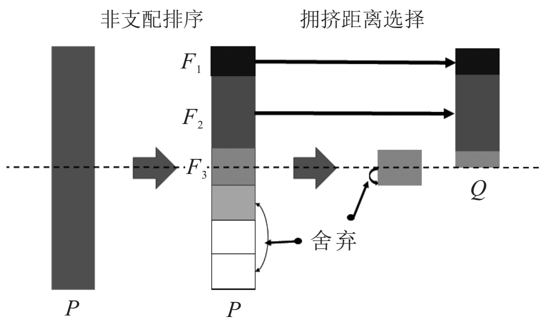

图4 NSGA-II流程图

图4中: P代表父代与子代合并后的种群, Q代表进入下一代的种群. 在一次迭代过程中,种群P经过非支配排序和拥挤距离选择得到种群Q.

下面介绍几种具有代表性的算法.

Song等[6]提出了MONES算法以求解NESs. 在该算法中,将NESs视为一个双目标优化问题来处理,通过NSGA-II来求解转化问题. 随机初始化种群之后,经过选择、交叉、变异产生子代种群;然后合并父代和子代,利用非支配排序和拥挤距离选择选出进入下一代的个体.

Gong等[8]在MONES的基础上进一步改进,提出了基于权值的双目标优化算法(A-Web). 在该算法中,目标函数中的权重值从0～1随机产生. 在优化过程中,结合SHADE和NSGA-II两种搜索算法,经过变异、交叉产生子代. 这一过程中参数是自适应调整的,从而提高了搜索的精确度. 之后个体之间的选择由

非支配排序决定.

Gao 等[25] 设计了两阶段算法TPEA(two-phase evolutionary algorithm)来求解NESs. 在该算法中,将NESs转化为单目标优化问题. 第1阶段,利用小生境技术NCDE构造了基于高斯核函数的多样性指标以共同维持种群的多样性,达到收敛性与多样性的平衡. 在具体迭代中,NCDE与NSGA-II交替进行,产生高质量的候选解. 第2阶段,设计了一种探测方法来定位有前途的区域(即最优解可能存在的区域),将DE作为局部搜索算法最终找到NESs的根.

许伟伟等[26] 首先将NESs转化为多模态多目标优化问题;然后,通过非支配排序和决策空间拥挤距离选择机制,挑选种群中的一半优质个体进行变异,在变异过程中设计了一种新的变异策略和边界处理方法以增加解的多样性;最后,通过交叉和选择机制使优质个体得到进化,直到搜索到全部最优解.

Naidu 等[27] 提出了 HCMOIWO (hybrid cooperative multiobjective optimization IWO) 算法,将NESs视为双目标优化问题. 在该算法中,先将种群分成相等规模的两个子种群,每个子种群对应一个目标函数,基于IWO和STS (space transformation search)对每个子种群进行搜索;然后联合所有的子种群,再用非支配排序选择下一代个体,将每一次迭代产生的个体中非劣个体存放在预先给定的存档中;最终输出存档中的个体.

3.4 其他算法

除了基于上述3类技术的智能优化算法之外,研究者还提出了许多不同的框架. 与上述3类方法相比,由于这些框架所形成的算法或者不具备较好的优化性能,或者通用性欠佳,本文称其为其他算法. 下面选取部分算法进行说明.

1) 遗传算法(genetic algorithms, GA)[28] 作为主框架的算法: 文献[29]结合Gauss-Legendre数值积分和GA算法来求解NESs. 文献[30]利用免疫遗传算法(immune genetic algorithm, IGA)寻找NESs的根,并采用基于适应度值的个体距离比较法,这样既保留了优秀个体,又减少了相似个体的选择,从而保证了整个种群的多样性. 文献[31]提出了一种混合小生境遗传算法与拟牛顿算法求解NESs的方法. 首先由小生境遗传算法产生一个种群,找到适应值最好的个体;然后,以一定的概率选择剩余种群中的一个个体作为初始点,采用拟牛顿法进行局部搜索. 如果该局部最优个体适应度值好于之前,则进行替换;否则,不进行替换操作.

2) 粒子群优化算法(particle swarm optimization, PSO)[32] 作为主框架的算法: 与GA算法相比,PSO算法是一种有记忆的算法,好的个体信息会保存利用,并且在大多数情况下收敛速度快于GA. 将PSO算法用于求解NESs问题的研究十分广泛. Mo 等[33] 提出了 CDPSO(conjugate direction particle swarm optimization)算法,将共轭方向法用于PSO以改善PSO在高维中易陷入局部最优问题,便于求解NESs. Brits 等[34] 进一步改进PSO,提出了nbest (neighborhood best)的操作,增加了算法的多样性,使得该算法易于定位NESs的多个根.

3) 模拟退火算法(simulated annealing, SA)[35] 作为主框架的算法: 文献[36]在寻找多个最优解时,为了避免SA 算法过早收敛,陷入局部最优解,利用模糊自适应SA算法来克服这一缺陷. 文献[37]提出了一种极化技术(polarization technique),类似于排斥技术,在确定一个新最优解之后连续地修改目标函数,在先前找到的根的邻域中产生排斥区域,解决了SA算法陷入局部最优问题.

除此之外,也有研究者运用蝙蝠算法、社会情感优化算法、帝国主义竞争算法等[38-40] 智能优化算法来求解NESs.

4 求解NESs问题的智能优化算法性能比较

4.1 测试函数和评价指标

为了评估算法性能,近些年来,一般采用30个标准(benchmark)测试函数($F01 \sim F30$)来进行试验研究(测试函数见文献[9]). 这些测试函数的基本特征由表1给出.

表1中: n 为决策向量的个数, range为决策向量取值范围, LE为方程组中含有线性方程的个数, NE为方程组中含有非线性方程的个数, NOR为方程组真实根的个数, MaxFes为最大评价次数.

为了比较不同算法之间的性能,给出了如下两个常用的评价指标[25].

1) 找到根率(roots found rate, RR): 利用该指标来评价多次运行找到NESs根的概率,具体计算公式如下:

$$\mathrm{RR} = \frac{\sum_{j=1}^{N_r} N_s^j}{\mathrm{NOR} \cdot N_r}. \tag{15}$$

其中: N_r 是算法运行次数, N_s^j 是第 j 次运行找到根的个数, NOR是方程组真实根的个数.

表1 30个标准测试函数特征

prob	n	range	LE	NE	NOR	MaxFes
$F01$	20	$[-1,1]^n$	0	2	2	50 000
$F02$	2	$[-1,1]^n$	1	1	11	50 000
$F03$	2	$[-1,1]^n$	0	2	15	50 000
$F04$	2	$[-10,10]^n$	0	2	13	50 000
$F05$	10	$[-2,2]^n$	0	10	1	50 000
$F06$	2	$[-1,1]^n$	1	1	8	50 000
$F07$	2	$[0,1],[-10,0]$	0	2	2	50 000
$F08$	2	$[0,1]^n$	0	2	7	50 000
$F09$	5	$[-10,10]^n$	4	1	3	100 000
$F10$	3	$[-5,5],[-1,3],[-5,5]$	0	3	2	50 000
$F11$	2	$[-1,1],[-10,10]$	0	2	4	50 000
$F12$	2	$[-1,2]^n$	0	2	10	50 000
$F13$	3	$[-0.6,6],[-0.6,0.6],[-5,5]$	0	3	12	50 000
$F14$	2	$[-5,5]^n$	0	2	9	50 000
$F15$	2	$[0.25,1],[1.5,2\pi]$	0	2	2	50 000
$F16$	2	$[0,2\pi]^n$	0	2	13	50 000
$F17$	8	$[-1,1]^n$	1	7	16	100 000
$F18$	2	$[-2,2]^n$	0	2	6	50 000
$F19$	20	$[-2,2]^n$	19	1	2	200 000
$F20$	3	$[-1,1]^n$	0	3	7	50 000
$F21$	2	$[-2,2]^n$	0	2	4	50 000
$F22$	3	$[-2,2]^n$	0	2	6	50 000
$F23$	3	$[-20,20]^n$	0	3	16	500 000
$F24$	3	$[0,1]^n$	0	3	8	100 000
$F25$	3	$[-3,3]^n$	0	3	2	100 000
$F26$	2	$[-1,-0.1],[-2,2]$	0	2	2	50 000
$F27$	2	$[-5,1.5],[0,5]$	0	2	3	50 000
$F28$	2	$[0,2],[10,30]$	0	2	2	50 000
$F29$	3	$[0,2],[-10,10],[-1,1]$	0	3	5	50 000
$F30$	2	$[-2,2],[0,1.1]$	0	2	4	50 000

2) 成功率 (success rate, SR): 该指标度量的是所有运行中的成功运行率, 如果在一次运行中找到NESs的所有根, 则认为它是一次成功的运行. 具体计算公式如下:

$$\mathrm{SR}=\frac{N_s}{N_r}, \tag{16}$$

其中N_s是成功运行的次数.

4.2 算法性能比较

表2中给出了5种具有代表性算法在30个NESs上的实验结果, 表2中的结果均从文献中收集得到.

表2中的评价指标包括找到根率 (RR) 和成功率 (SR). 此外, 5种进化算法分别是MONES[6]、A-WeB[8]、RADE[9]、DR-JADE[16]和TPEA[25].

在30个测试函数中, 测试函数$F01$和$F19$目标函数维数较高, 主要考察算法的开发能力, 设计算法时要注意种群的收敛性; 测试函数$F02$、$F03$、$F04$、$F12$、$F13$、$F16$、$F17$和$F23$的最优解个数较多, 主要考察算法的探索能力, 设计算法时要注意种群的多样性; 其他函数主要考察算法的综合能力.

通过表2, 可以从整体上对利用智能优化算法求解NESs的性能有一个较为清晰的认识. 这有助于更加全面了解该领域智能优化算法的性能, 有针对性地提高已有算法的性能以及提出更加有效的新的算法, 并进一步发现和考虑其中一些亟待解决的问题. 从表2中可以看出: MONES在收敛性和多样性方面均存在一些缺陷, 由SR的实验结果也能发现, MONES算法的鲁棒性较低, 在测试函数$F03$、$F04$等函数上算法的稳定性较差; 而算法A-WeB、RADE、DR-JADE和TPEA均表现出较好的优化效果, 这是由于这4种算法在平衡探索与开发能力上能够做到有机结合. 上述实验结果充分表明, 只有保证探索与开发的平衡, 即多样性与收敛性的平衡, 求解NESs才能取得较好的结果.

表2 5种算法对30个标准测试函数的实验结果比较(NA表示文献中没有提供实验结果)

prob	RR					SR				
	MONES	A-Web	RADE	DR-JADE	TPEA	MONES	A-Web	RADE	DR-JADE	TPEA
$F01$	0.945 0	0.620 0	1.000 0	0.000 0	0.920 0	0.93	0.36	1.00	0.00	0.82
$F02$	1.000 0	1.000 0	0.990 0	0.997 0	1.000 0	1.00	1.00	0.90	0.96	1.00
$F03$	0.965 3	0.957 3	0.996 0	0.957 8	1.000 0	0.58	0.58	0.95	0.46	1.00
$F04$	0.934 6	1.000 0	0.901 5	1.000 0	1.000 0	0.67	1.00	0.31	1.00	1.00
$F05$	1.000 0	1.000 0	1.000 0	1.000 0	1.000 0	1.00	1.00	1.00	1.00	1.00
$F06$	0.125 0	0.940 0	0.990 0	0.958 3	1.000 0	0.00	0.60	0.93	0.66	1.00
$F07$	0.990 0	1.000 0	1.000 0	1.000 0	1.000 0	0.98	1.00	1.00	1.00	1.00
$F08$	0.438 6	0.837 1	0.997 1	1.000 0	1.000 0	0.00	0.12	0.98	1.00	1.00
$F09$	0.886 7	0.893 3	0.970 0	1.000 0	1.000 0	0.66	0.68	0.91	1.00	1.00
$F10$	0.965 0	1.000 0	1.000 0	1.000 0	1.000 0	0.93	1.00	1.00	1.00	1.00
$F11$	0.000 0	1.000 0	1.000 0	1.000 0	1.000 0	0.00	1.00	1.00	1.00	1.00
$F12$	0.436 0	0.888 0	0.631 0	0.876 7	0.942 0	0.00	0.28	0.00	0.03	0.40
$F13$	0.543 3	0.093 3	0.890 8	0.916 7	0.916 7	0.00	0.00	0.19	0.26	0.36
$F14$	0.990 0	0.973 3	0.986 7	1.000 0	1.000 0	0.92	0.76	0.89	1.00	1.00
$F15$	0.980 0	1.000 0	1.000 0	0.500 0	1.000 0	0.96	1.00	1.00	0.00	1.00
$F16$	0.436 9	1.000 0	0.995 4	1.000 0	1.000 0	0.00	1.00	0.94	1.00	1.00
$F17$	0.166 3	0.668 8	0.944 4	1.000 0	0.975 0	0.00	0.00	0.43	0.03	0.82
$F18$	0.508 3	0.943 3	1.000 0	1.000 0	1.000 0	0.00	0.66	1.00	1.00	1.00
$F19$	0.305 0	0.620 0	0.795 0	0.000 0	0.720 0	0.07	0.24	0.69	0.00	0.64
$F20$	0.544 3	0.951 4	1.000 0	1.000 0	1.000 0	0.00	0.70	1.00	1.00	1.00
$F21$	0.535 0	0.995 0	1.000 0	1.000 0	1.000 0	0.01	0.98	1.00	1.00	1.00
$F22$	0.530 0	1.000 0	1.000 0	1.000 0	1.000 0	0.00	1.00	1.00	1.00	1.00
$F23$	0.146 3	NA	0.561 9	0.797 9	0.975 0	0.00	NA	0.00	0.00	0.80
$F24$	0.640 0	0.855 0	0.998 8	0.912 5	1.000 0	0.00	0.14	0.99	0.43	1.00
$F25$	0.315 0	NA	0.835 0	1.000 0	1.000 0	0.07	NA	0.67	1.00	1.00
$F26$	1.000 0	1.000 0	1.000 0	1.000 0	1.000 0	1.00	1.00	1.00	1.00	1.00
$F27$	1.000 0	1.000 0	0.996 7	1.000 0	1.000 0	1.00	1.00	0.99	1.00	1.00
$F28$	0.875 0	0.940 0	1.000 0	1.000 0	1.000 0	0.78	0.88	1.00	1.00	1.00
$F29$	0.822 0	0.932 0	0.994 0	0.953 3	1.000 0	0.13	0.66	0.97	0.76	1.00
$F30$	1.000 0	1.000 0	1.000 0	1.000 0	1.000 0	1.00	1.00	1.00	1.00	1.00
mean	0.667 5	0.896 7	0.949 1	0.895 7	0.981 6	0.42	0.70	0.82	0.72	0.93

5 亟待解决的问题

虽然利用智能优化算法求解NESs已经取得了一定的成果,但是仍然存在很多问题需要解决,主要体现在以下几点.

1) 理论性. 数学描述的完善程度是衡量一种理论成熟的标志. 用传统优化算法求解优化问题拥有完善的理论依据,而智能优化算法的理论研究一直滞后于它的应用. 如何证明算法收敛性还需要进一步深入研究. 此外,算法的收敛速度也是一个重要的研究课题.

2) 转化方法. 目前,研究者大多都是根据算法设计的需要或者操作难易程度,人为地将NESs转化成单目标、约束或者多目标优化问题,没有考虑到方程组的特点. 不同的转化对求解效率的影响是不同的,因此,设计一种能通过方程组的特点判断选择哪种转化方法更加有效,这将是未来研究的方向之一.

3) 混合性. 由于求解NESs问题的复杂性和多样性,近年来,研究者提出了混合多种技术的算法(如前面提到的排斥技术、聚类技术、多目标技术等). 这类算法的优点是多样性好,搜索效率高,而且特别适合有多个根的NESs. 但是,混合的技术不同,可能会导致算法结果参差不齐. 如何设计混合算法以进一步优化算法性能,提高NESs找到根的概率,是一个值得广泛研究的问题.

4) 无穷根的问题. 现有的求解NESs的算法大多都只适用于有限个根的问题,A-WeB和DR-JADE这两种算法对无穷根问题进行了简单的描述与分析. 但是,大部分研究对于如何求解无穷根的问题以

及设计怎样的评价指标来评价算法的有效性则涉及较少,这仍然是一个巨大的挑战.

5) 测试函数. 实际生活是丰富多彩的,并不是每一个实际的NESs问题都包含已有的测试函数的一些特性. 因此,设计一些能够反映实际NESs问题基本特征的标准测试函数,是该领域需要解决的一个重要问题.

6 结 论

本文根据智能优化算法求解NESs问题的基本框架,从转化方法和智能优化算法两方面出发,对求解NESs的算法研究进展进行了综述. 此外,对NESs的测试函数及评价指标进行了描述,并选取5种具有代表性的算法进行了对比,分析了求解NESs亟待解决的问题. 由于问题所涉及的方面过多,本文不可能面面俱到,希望与研究者们沟通交流,共同促进该领域的发展.

参考文献(References)

[1] Smale S. Mathematical problems for the next century[J]. The Mathematical Intelligencer, 1998, 20(2): 7-15.

[2] Guo D S, Nie Z Y, Yan L C. The application of noise-tolerant ZD design formula to robots’ kinematic control via time-varying nonlinear equations solving[J]. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2018, 48(12): 2188-2197.

[3] Broyden C G. A class of methods for solving nonlinear simultaneous equations[J]. Mathematics of Computation, 1965, 19(92): 577-593.

[4] Ramos H, Monteiro M T T. A new approach based on the Newton′s method to solve systems of nonlinear equations[J]. Journal of Computational and Applied Mathematics, 2017, 318: 3-13.

[5] 吴龙, 任红民, 毕惟红. 遗传算法求解非线性方程组研究综述[J]. 电子科技, 2014, 27(4): 173-178.
(Wu L, Ren H M, Bi W H. Review of the genetic algorithm for nonlinear equations[J]. Electronic Science and Technology, 2014, 27(4): 173-178.)

[6] Song W, Wang Y, Li H X, et al. Locating multiple optimal solutions of nonlinear equation systems based on multiobjective optimization[J]. IEEE Transactions on Evolutionary Computation, 2015, 19(3): 414-431.

[7] Grosan C, Abraham A. A new approach for solving nonlinear equations systems[J]. IEEE Transactions on Systems, Man and Cybernetics-Part A, 2008, 38(3): 698-724.

[8] Gong W Y, Wang Y, Cai Z H, et al. A weighted biobjective transformation technique for locating multiple optimal solutions of nonlinear equation systems[J]. IEEE Transactions on Evolutionary Computation, 2017, 21(5): 697-713.

[9] Gong W Y, Wang Y, Cai Z H, et al. Finding multiple roots of nonlinear equation systems via a repulsion-based adaptive differential evolution[J]. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2020, 50(4): 1499-1513.

[10] Pourjafari E, Mojallali H. Solving nonlinear equations systems with a new approach based on invasive weed optimization algorithm and clustering[J]. Swarm and Evolutionary Computation, 2012, 4: 33-43.

[11] Ramadas G C, Fernandes E M, Rocha A A. Multiple roots of systems of equations by repulsion merit functions[J]. Computational Science and Its Applications, 2014(2): 126-139.

[12] Hirsch M J, Pardalos P M, Resende M G C. Solving systems of nonlinear equations with continuous GRASP[J]. Nonlinear Analysis: Real World Applications, 2009, 10(4): 2000-2006.

[13] Storn R, Price K. Differential evolution — A simple and efficient heuristic for global optimization over continuous spaces[J]. Journal of Global Optimization, 1997, 11(4): 341-359.

[14] Tanabe R, Fukunaga A. Success-history based parameter adaptaion for differential evolution[C]. Proceedings of the 2013 IEEE Congress on Evolutionary Computation. Cancun, 2013: 71-78.

[15] Zhang J Q, Sanderson A C. JADE: Adaptive differential evolution with optional external archive[J]. IEEE Transactions on Evolutionary Computation, 2009, 13(5): 945-958.

[16] Liao Z W, Gong W Y, Yan X S, et al. Solving nonlinear equations system with dynamic repulsion-based evolutionary algorithms[J]. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2020, 50(4): 1590-1601.

[17] Wu X H, Wu B, Sun J, et al. Mixed fuzzy inter-cluster separation clustering algorithm[J]. Applied Mathematical Modelling, 2011, 35(10): 4790-4795.

[18] Tsoulos I G, Stavrakoudis A. On locating all roots of systems of nonlinear equations inside bounded domain using global optimization methods[J]. Nonlinear Analysis: Real World Application, 2010, 11(4): 2465-2471.

[19] Sacco W F, Henderson N. Finding all solutions of nonlinear systems using a hybrid metaheuristic with fuzzy clustering means[J]. Applied Soft Computing, 2011, 11(8): 5424-5432.

[20] He W, Gong W Y, Wang L, et al. Fuzzy neighborhood based differential evolution with orientation for nonlinear equation system[J]. Knowledge-Based Systems, 2019, 182: 104796.

[21] Liao Z W, Gong W Y, Wang L, et al. A decomposition-based differential evolution with reinitialization for nonlinear equations systems[J]. Knowledge-Based Systems, 2020, 191: 105312.

[22] Liao Z W, Gong W Y, Wang L. Memetic niching-based evolutionary algorithms for solving nonlinear equation system[J]. Expert Systems with Applications, 2020, 149: 113261.

[23] Zhang Q F, Li H. MOEA/D: A multiobjective evolutionary algorithm based on decomposition[J]. IEEE Transactions on Evolutionary Computation, 2007, 11(6): 712-731.

[24] Deb K, Pratap A, Agarwal S, et al. A fast and elitist multiobjective genetic algorithm: NSGA-II[J]. IEEE Transactions on Evolutionary Computation, 2002, 6(2): 182-197.

[25] Gao W F, Li G H, Zhang Q F, et al. Solving nonlinear equation systems by a two-phase evolutionary algorithm[J]. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2019: 1-12.

[26] 许伟伟, 梁静, 岳彩通, 等. 多模态多目标差分进化算法求解非线性方程组[J]. 计算机应用研究, 2019, 36(5): 1305-1310.
(Xu W W, Liang J, Yue C T, et al. Multimodal multi-objective differential evolution algorithm for solving nonlinear equations[J]. Application Research of Computers, 2019, 36(5): 1305-1310.)

[27] Naidu Y R, Ojha A K. Solving multiobjective optimization problems using hybrid cooperative invasive weed optimization with multiple populations[J]. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2018, 48(6): 821-832.

[28] Gaafar L K, Masoud S A. Genetic algorithms and simulated annealing for scheduling in agile manufacturing[J]. International Journal of Production Research, 2005, 43(14): 3069-3085.

[29] Mhetre P S. Genetic algorithm for linear and nonlinear equation[J]. International Journal of Advanced Engineering Technology, 2012(2): 114-118.

[30] Wang J. Immune genetic algorithm for solving nonlinear equation[C]. Proceedings of the 2011 International Conference on Mechatronic Science, Electric Engineering and Computer. Jilin, 2011: 2094-2097.

[31] 周丽, 姜长生. 非线性方程组求解的一种新方法[J]. 小型微型计算机系统, 2008, 29(9): 1709-1713.
(Zhou L, Jiang C S. New method for solving nonlinear equation systems[J]. Journal of Chinese Conputer Systems, 2008, 29(9): 1709-1713.)

[32] Kennedy J, Eberhart R C. Particle swarm optimization[C]. Proceedings of the 1995 IEEE International Conference on Neural Networks. Perth: IEEE Service Center, 1995: 1942-1948.

[33] Mo Y B, Liu H T, Wang Q. Conjugate direction particle swarm optimization solving systems of nonlinear equations[J]. Computers & Mathematics with Applications, 2009, 57(10): 1877-1882.

[34] Brits R, Engelbrecht A P. Solving systems of unconstrained equations using particle swarm optimization[C]. Proceedings of the 2002 IEEE International Conference Systems, Man and Cybernetics. Yasmine Hammamet, 2002: 6-9.

[35] Corana A, Marchesi M, Martini C, et al. Minimizing multimodal functions of continuous variables with the “simulated annealing” algorithm[J]. ACM Transactions on Mathematical Software, 1987, 13(3): 262-280.

[36] Oliveira H A E, Petraglia A. Solving nonlinear systems of functional equations with fuzzy adaptive simulated annealing[J]. Applied Soft Computing, 2013, 13(11): 4349-4357.

[37] Henderson N, Sacco W F, Platt G M. Finding more than one root of nonlinear equations via a polarization technique: An application to double retrograde vaporization[J]. Chemical Engineering Research & Design, 2010, 88(5): 551-561.

[38] Xie J, Zhou Y Q, Chen H. A novel bat algorithm based on differential operator and Lévy flights trajectory[J]. Computational Intelligence and Neuroscience, 2013, 2013: 453812.

[39] Wu J, Cui Z, Liu J. Using hybrid social emotional optimization algorithm with metropolis rule to solve nonlinear equations[C]. Proceedings of the 10th IEEE International Conference on Cognitive Informatics and Cognitive Computing. Banff, 2011: 405-411.

[40] Abdollahi M, Isazadeh A, Abdollahi D. Imperialist competitive algorithm for solving systems of nonlinear equations[J]. Computers & Mathematics with Applications, 2013, 65(12): 1894-1908.

作者简介

高卫峰 (1985−), 男, 教授, 博士生导师, 从事最优化方法、进化计算理论及应用等研究, E-mail: gaoweifeng2004@126.com;

罗宇婷 (1995−), 女, 硕士生, 从事进化计算及应用的研究, E-mail: 809293635@qq.com;

原杨飞 (1995−), 男, 硕士生, 从事进化计算及应用的研究, E-mail: 791583819@qq.com.

(责任编辑：李君玲)