DOI:10.13878/j.cnki.jnuist.2020.05.007

官娟[1]　刘国华[1]　刘天祺[1]　秦健[1]　张淼森[1]

基于 MIMIC 算法和 RPCA 的混合蚁群优化算法

摘要

为提高连续域上蚁群算法的寻优性能,降低决策变量之间的相关性,设计一种基于 MIMIC 算法和 RPCA 的连续域上蚁群优化算法.本文首先介绍连续域上的蚁群算法;然后根据一些处理多变量相关性的方法,给出有效相关性的定义;接着提出一种基于 MIMIC 算法和 RPCA 的混合蚁群算法;最后,通过对标准测试函数进行优化求解实验,将所得结果与连续域上的蚁群优化算法相比较,可知该算法在寻优能力和收敛性方面都有明显的提高,是一种有效的优化算法.

关键词

蚁群优化算法;变量相关性;MIMIC 算法;鲁棒主成分分析(RPCA)

中图分类号 TP18

文献标志码 A

0　引言

元启发式算法是一类优化技术,在过去几年中得到了越来越快的发展,并被应用于许多问题.蚁群算法是一类比较典型和重要的元启发式算法,最早由 Colorni 等在 1992 年提出[1],算法主要针对自然界的蚂蚁觅食做了经典实验,通过将真实蚂蚁的行为模型化,得出了人工蚂蚁解决多目标决策问题的思路.基于此思想,Dorigo 等开创性地提出了蚁群算法[2].蚁群算法的本质是一个复杂的智能系统——蚂蚁系统(AS),它具有较强的鲁棒性、优良的分布式计算机制,以及易于与其他算法相结合等优点.

虽然蚁群算法具有许多优点,在低维问题中可取得较好的效果,但对于大规模问题全局寻优性能急剧下降,存在一些缺陷,如收敛速度慢、容易陷入局部最优解等,同时这也是大部分优化算法所面临的问题.针对此现状,近年来许多学者在蚁群算法的改进方面进行了研究,并提出了许多改进的蚁群算法,包括蚁群优化(ACO)的元启发式方法[3]、蚁群系统(ACS)[4]、基于 Q-学习的自适应蚁群算法[5]、最大最小蚂蚁系统(MMAS 等)[6].这些改进的算法主要从信息素释放方式、信息素更新规则、路径选择策略、参数的选择等方面入手,对蚁群算法进行了不同程度的改进,但仍存在一个很大的问题,即随机性对解搜索带来的较强干扰,使得解变量之间的相关性和抽样求解的效率太低、冗余度过高,严重制约了蚁群算法在高维空间中的优化性能.已有的蚁群优化算法中缺少对关于随机变量间相关性的研究,Socha 等[7]在研究连续域上的蚁群算法中有过相关问题的探讨,但并未对算法中变量相关性做进一步的研究.

为更有效地处理大规模优化问题,提高算法性能,本文在蚁群算法中,加入变量相关性分析并根据决策变量的相关性定义,提出一种新的关于解分量之间的相关性定义——有效相关性与冗余相关性;然后,根据得到的有效相关性矩阵,运用 MIMIC(Mutual Information Maximization for Input Clustering)分布估计算法对解存档建立决策变量间的概率图模型,并根据概率模型采样获得新的解存档,来提高蚁群优化算法的全局搜索能力;接着,利用 RPCA(Robust Principal Component Analysis)技术找到新的解存档的低秩结构矩阵,优化搜索空间、降低计算复杂度,进而在蚁群算法框架下,更新解存档;最后,通过实验仿真测试算法性能,将所得结果与连续域上的蚁群优化算法

收稿日期 2020-07-01

资助项目 国防科技创新特区项目(2019)

作者简介

官娟,女,硕士生,研究方向为智能算法、量子群与李理论.220171465@seu.edu.cn

刘国华(通信作者),女,博士,副教授,硕士生导师,研究方向为智能算法以及 Hopf 代数.liuguohua@seu.edu.cn

1　东南大学 数学学院,南京,210096

相比较,表明该算法可降低多个解分量之间的冗余相关性,在寻优能力和收敛性方面都有显著提高.

1 连续域上的蚁群优化算法

连续域上的蚁群优化算法 ACO_R(Ant Colony Optimization for Continuous Domains)[7] 均匀随机产生 k 个解作为初始解存档,如图 1 所示.每个解都是一个 D 维向量,其实值解决策分量 $x_i \in [x_{\min}, x_{\max}]$,其中 $i = 1,\cdots,D$.本文假定优化问题中只存在 box 有界约束.根据目标函数值从优至劣对解存档中 k 个解进行排序,每个解 $\boldsymbol{S}_k$ 都关联了一个权重 ω_k.该权重使用高斯函数计算,如下所示:

$$\omega_k = \frac{1}{qk\sqrt{2\pi}} e^{\frac{-(r(j)-1)}{2q^2k^2}}, \tag{1}$$

其中 $r(j)$ 是排序解存档中解 $\boldsymbol{S}_j$ 的序号,q 是算法的一个参数.

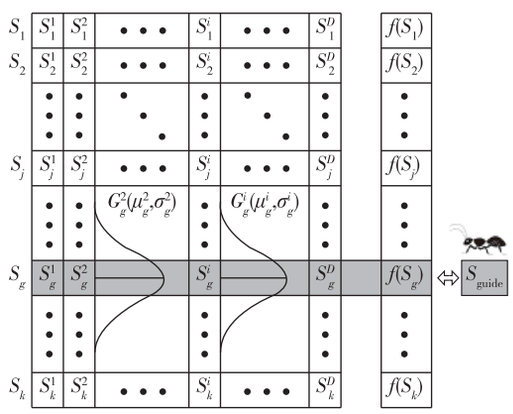

图 1 解存档的结构和采样的高斯函数

Fig. 1 Structure of the solution archive and the Gauss functions of sampling

从式(1)可知解存档中排序越靠前的解权重越大,权重间按照负指数递减.将式(1)的权重归一化即选择解 $\boldsymbol{S}_j$ 作为引导解的概率 $\omega_j / \sum_{\alpha=1}^{k} \omega_\alpha$,围绕该引导解生成新的候选解.一旦选择了引导解 $\boldsymbol{S}_{\text{guide}}$,则该算法就在引导解 $\boldsymbol{S}_{\text{guide}}$ 的第 i 个实值分量的邻域进行采样,采样的高斯密度函数均值为 $\mu^i_{\text{guide}} = S^i_{\text{guide}}$,方差为 $\sigma^i_{\text{guide}} = \xi \sum_{r=1}^{k} \frac{|S_r^i - S^i_{\text{guide}}|}{k-1}$,它是 $\boldsymbol{S}_{\text{guide}}$ 的第 i 个分量的值与解存档中其他解的第 i 个分量的值之间的平均距离,乘以一个参数 ξ.每一次迭代,选择一个引导解和生成一个候选解的过程被重复了总共 m 次(对应于“蚂蚁”的数目).在下一次迭代之前,算法更新解存档,只保留在解构建完成之后可用的 $k+m$ 个解中的最佳 k 个解.

2 基于 MIMIC 算法和 RPCA 的混合蚁群算法

2.1 变量相关性

现有的蚁群优化算法中,随机性对解的搜索带来了较强的干扰,使得解变量之间的相关性太低、抽样求解的效率太低、冗余度过高,严重制约了蚁群算法的优化性能.因此,显著提高蚁群算法的优化性能就必须克服抽样解选取过程中的相关性问题.

基于以上考虑,本文首先根据决策变量间的相关性定义,给出解分量之间的有效相关性和冗余相关性的定义.

2.1.1 解分量与解分量之间的相关性

根据文献[8-9],两个决策变量 x_i 和 x_j 被称为相关,如果 $\exists \boldsymbol{x} = (x_1, x_2, \cdots, x_n), x_i', x_j'$ 满足下列条件:

$$f(x_1,\cdots,x_i,\cdots,x_j,\cdots,x_n) < f(x_1,\cdots,x_i',\cdots,x_j,\cdots,x_n) \wedge f(x_1,\cdots,x_i,\cdots,x_j',\cdots,x_n) > f(x_1,\cdots,x_i',\cdots,x_j',\cdots,x_n), \tag{2}$$

其中 $\boldsymbol{x}$ 是候选决策向量,x_i', x_j' 分别被第 i 和第 j 个决策变量替换,“$\wedge$”表示逻辑符号“且”.

2.1.2 解分量与解分量之间的有效相关性

设 $\boldsymbol{X}$ 为当前解,sample_i 为对第 i 个分量进行采样的采样算子,$>_{\text{opt}}$ 代表更优化的比较算子,即只要存在一个可行解,满足式(2),则说明解分量 x_i 与解分量 x_j 关于目标函数 f 存在相关性,然而式(2)只刻画了变量间相关性的存在性,并未指出对优化过程具有有效促进作用的部分相关性,因此定义一个有效相关性如下:

$$\begin{aligned}
&\exists \boldsymbol{x} = (x_1, x_2, \cdots, x_n), \quad x_i' = \text{sample}_i(\boldsymbol{X}), \quad x_j' \\
&\text{s.t} \quad f(x_1,\cdots,x_i',\cdots,x_j',\cdots,x_n) >_{\text{opt}} f(x_1,\cdots,x_i,\cdots,x_j',\cdots,x_n) \\
&\vee \\
&\exists \boldsymbol{x} = (x_1, x_2, \cdots, x_n), \quad x_i', \quad x_j' = \text{sample}_i(\boldsymbol{X}) \\
&\text{s.t} \quad f(x_1,\cdots,x_i',\cdots,x_j',\cdots,x_n) >_{\text{opt}} f(x_1,\cdots,x_i',\cdots,x_j,\cdots,x_n)
\end{aligned} \tag{3}$$

其中“$\vee$”表示逻辑符号“或”.

2.1.3 解分量与解分量之间的冗余相关性

相应地,称:

$$\begin{aligned}
&\exists \boldsymbol{x} = (x_1, x_2, \cdots, x_n), \quad x_i' = \text{sample}_i(\boldsymbol{X}), \quad x_j' \\
&\text{s.t} \quad f(x_1,\cdots,x_i',\cdots,x_j',\cdots,x_n) <_{\text{opt}} f(x_1,\cdots,x_i,\cdots,x_j',\cdots,x_n)
\end{aligned}$$

$$\wedge$$
$$\exists \boldsymbol{x}=(x_1,x_2,\cdots,x_n),\quad x_i',\quad x_j'=\mathrm{sample}_i(\boldsymbol{X})$$
$$\text{s.t}\quad f(x_1,\cdots,x_i',\cdots,x_j',\cdots,x_n)<_{\mathrm{opt}} f(x_1,\cdots,x_i',\cdots,x_j,\cdots,x_n) \tag{4}$$

为解分量 x_i 与 x_j 关于 f 的冗余相关部分.

2.2 MIMIC$_c^G$ 分布估计算法

分布估计算法[10-11]是一种基于种群的搜索算法.它利用解存档建立概率分布模型,然后根据概率分布模型来指导个体进行搜索,其过程是一个不断更新概率模型,从而使概率模型越来越能反映优秀个体分布的过程.所以,分布估计算法具有比较强的全局搜索能力.

蚁群优化算法搜索时间长,易陷入局部最优解是其最为突出的缺点.为了提高蚁群算法的寻优能力,可以在蚁群优化算法中混合分布估计算法.

2.2.1 MIMIC 算法

MIMIC 算法,即输入聚类的最大互信息算法,最早由美国 MIT 人工智能实验室的 de Bonet 等提出的一种基于双变量相关模型的分布估计算法[12].在 MIMIC 算法中,变量之间的关系可以用链式有向图表示(图 2).

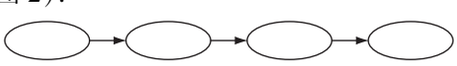

图 2 MIMIC 概率模型

Fig. 2 MIMIC probability model

设决策变量集合 $\boldsymbol{x}=(x_1,x_2,\cdots,x_n)$ 的联合概率分布为 $p(\boldsymbol{x})=p(x_1,x_2,\cdots,x_n)$,且

$$p(\boldsymbol{x})=p(x_1\mid x_2,\cdots,x_n)p(x_2\mid x_3,\cdots,x_n)\cdots p(x_{n-1}\mid x_n). \tag{5}$$

算法中解空间的概率模型可以描述为

$$p_l^{\pi}(\boldsymbol{x})=p_l(x_{i_1}\mid x_{i_2},\cdots,x_{i_n})p_l(x_{i_2}\mid x_{i_3},\cdots,x_{i_n})\cdots p_l(x_{i_{n-1}}\mid x_{i_n}), \tag{6}$$

其中 $\boldsymbol{\pi}=(i_1,i_2,\cdots,i_n)$ 表示变量 $x_1,x_2,\cdots,x_n$ 的一个排列,$p_l(x_{i_j}\mid x_{i_{j+1}})$ 表示第 i_{j+1} 个变量取值为 $x_{i_{j+1}}$ 时第 i_j 个变量取值为 x_{i_j} 的条件概率.

在构建概率模型的时候,我们希望得到一个最优排列 $\boldsymbol{\pi}$,使得 $p_l^{\pi}(\boldsymbol{x})$ 与每次迭代中优势群体的概率分布 $p(\boldsymbol{x})$ 尽可能接近,这里利用 K-L 距离(Kullback Liebler divergence)来衡量两个概率分布之间的距离,定义如下:

$$H_l^{\pi}(\boldsymbol{x})=h_l(\boldsymbol{X}_{i_n})+\sum_{j=1}^{n-1}p_l(\boldsymbol{X}_{i_j}\mid\boldsymbol{X}_{i_{j+1}}), \tag{7}$$

其中:

$$h(\boldsymbol{X})=-\sum_x p(\boldsymbol{X}=\boldsymbol{x})\log(\boldsymbol{X}=\boldsymbol{x}),$$
$$h(\boldsymbol{X}\mid\boldsymbol{Y})=-\sum_y h(\boldsymbol{X}\mid\boldsymbol{Y}=\boldsymbol{y})\log(\boldsymbol{Y}=\boldsymbol{y}),$$
$$h(\boldsymbol{X}\mid\boldsymbol{Y}=\boldsymbol{y})=-\sum_x(p(\boldsymbol{X}=\boldsymbol{x}\mid\boldsymbol{Y}=\boldsymbol{y})\log(\boldsymbol{X}=\boldsymbol{x}\mid\boldsymbol{Y}=\boldsymbol{y})).$$

2.2.2 MIMIC$_c^G$ 算法

MIMIC$_c^G$ 算法[13]是 MIMIC 在连续域中的扩展,其中,每对变量的潜在概率模型被假定为二元高斯分布.

对于优化变量 $\boldsymbol{x}=(x_1,x_2,\cdots,x_n)$,存在一个最优排列 $\boldsymbol{\pi}=(i_1,i_2,\cdots,i_n)$ 使得条件概率之积 $f_{\pi}(\boldsymbol{x})=f(x_{i_1}\mid x_{i_2})\cdots f(x_{i_{n-1}}\mid x_{i_n})\cdot f(x_{i_1})$ 与解存档的概率分布 $f(\boldsymbol{x})$ 之间的 K-L 距离达到最小.

MIMIC$_c^G$ 算法的估计概率模型操作是:用贪心算法缩小 $f_{\pi}(\boldsymbol{x})$ 与 $f(\boldsymbol{x})$ 之间的 K-L 距离 $J_{\pi}(\boldsymbol{x})$,得到最优排列 $\boldsymbol{\pi}=(i_1,i_2,\cdots,i_n)$.采样操作产生 x_{i_n},再以概率 $f(x_{i_{n-1}}\mid x_{i_n})$ 产生 $x_{i_{n-1}}$,依次以相应的条件概率产生 $x_{i_{n-2}},x_{i_{n-3}},\cdots,x_{i_1}$.

2.3 RPCA

如果搜索空间的维数过高,会严重影响蚁群算法的性能,需要一种技术来降低解分量之间的冗余相关性.在连续域上的蚁群优化算法和 MIMIC 算法中,候选解是根据高斯密度函数进行采样的,所以如果要对高维空间进行降维处理,并且降低解分量之间的冗余相关性,最常使用的技术是主成分分析(PCA)技术[14],但由于 PCA 技术的鲁棒性较差,在寻优过程中容易出现停滞的状况.所以本文选择使用 RPCA 技术,既可以降低解分量之间的冗余相关性,也可以保证蚁群系统的鲁棒性.

RPCA(Robust PCA)[15]是在 PCA 基础上加入一个可以测量在各个维度以内异常点的鲁棒函数,并且想办法去除异常点和修正恢复主要结构空间的一种方法.它很好地解决了传统 PCA 在处理大误差时不具有鲁棒性的问题.

2.3.1 RPCA 模型

设新的解矩阵 $\boldsymbol{S}'=\boldsymbol{B}+\boldsymbol{E}$,其中 $\boldsymbol{B}$ 是低秩矩阵,$\boldsymbol{E}$ 为稀疏(噪声)矩阵[16].RPCA 模型解决的问题是从带有稀疏大噪声的数据中精确地恢复出低秩矩阵.此模型可以表示为

$$\min\|\boldsymbol{B}\|_*+\lambda\|\boldsymbol{E}\|_1,\quad \text{s.t.}\quad \boldsymbol{S}'=\boldsymbol{B}+\boldsymbol{E}, \tag{8}$$

其中 $\|\cdot\|_1$ 为矩阵的 ℓ_1 范数,$\|\cdot\|_*$ 为矩阵的和范数,$\lambda=\dfrac{1}{\sqrt{\max(m,n)}}$,$\boldsymbol{B}$ 是需要恢复的低秩结构矩

阵,$\boldsymbol{E}$ 是未知的噪声矩阵,$\boldsymbol{S}'$ 是包含噪声的新的解矩阵.

2.3.2 RPCA 模型求解[17]

对模型(8)构建拉格朗日函数得:

$$L_\mu(\boldsymbol{B},\boldsymbol{E},\boldsymbol{Z}) = \|\boldsymbol{B}\|_* + \lambda\|\boldsymbol{E}\|_1 + \langle \boldsymbol{S}' - \boldsymbol{B} - \boldsymbol{E},\boldsymbol{Z}\rangle + \frac{\mu}{2}\|\boldsymbol{S}' - \boldsymbol{B} - \boldsymbol{E}\|_F^2, \quad (9)$$

迭代低秩矩阵 $\boldsymbol{B}$ 为

$$\begin{aligned}\boldsymbol{B}^* &= \arg\min_{\boldsymbol{B}} L_\mu(\boldsymbol{B},\boldsymbol{E},\boldsymbol{Z}) = \\ &\arg\min_{\boldsymbol{B}} \|\boldsymbol{B}\|_* + \langle \boldsymbol{S}' - \boldsymbol{B} - \boldsymbol{E},\boldsymbol{Z}\rangle + \\ &\frac{\mu}{2}\|\boldsymbol{S}' - \boldsymbol{B} - \boldsymbol{Z}\|_F^2 = \\ &D_{\mu^{-1}}\{\boldsymbol{S}' - \boldsymbol{E} + \mu^{-1}\boldsymbol{Z}\},\end{aligned} \quad (10)$$

迭代稀疏矩阵 $\boldsymbol{E}$ 为

$$\begin{aligned}\boldsymbol{E}^* &= \arg\min_{\boldsymbol{E}} L_\mu(\boldsymbol{B},\boldsymbol{E},\boldsymbol{Z}) = \\ &\arg\min_{\boldsymbol{E}} \|\boldsymbol{E}\|_1 + \langle \boldsymbol{S}' - \boldsymbol{B} - \boldsymbol{E},\boldsymbol{Z}\rangle + \\ &\frac{\mu}{2}\|\boldsymbol{S}' - \boldsymbol{B} - \boldsymbol{Z}\|_F^2 = \\ &S_{\lambda\mu^{-1}}\{\boldsymbol{S}' - \boldsymbol{B} + \mu^{-1}\boldsymbol{Z}\},\end{aligned} \quad (11)$$

其中矩阵 $\boldsymbol{Z}$ 是拉格朗日乘子,μ 为一个正数.

2.4 混合蚁群优化算法

蚁群算法在低维问题中取得了较好的效果,具有较强的局部搜索能力,但对于大规模问题全局优化性能急剧下降,而 MIMIC 算法的全局寻优能力强,局部搜索能力较弱,RPCA 可以恢复高维空间的低维结构空间,并保持系统的鲁棒性,所以在 ACO_R 算法的整体框架下加入 MIMIC 分布估计算法和 RPAC 以提高算法的寻优能力和效率.在 ACO_R 算法中,决策变量分量之间相互独立,未考虑解分量之间的相关性,导致抽样效率不高,所以在混合蚁群算法中,建立初始解存档之后,我们根据提出的有效相关性定义计算解存档的有效相关性矩阵指导后续算法采样.整个更新解存档的具体步骤如下:

步骤 1.对于 $\mathbf{R}^n$ 中优势解构成的解存档 $\boldsymbol{S} = (s_1 \quad s_2 \quad \cdots \quad s_N)^{\mathrm{T}}$,定义并计算解分量之间的一个有效相关性矩阵 $\boldsymbol{M}_{\mathrm{CR}}$.

对于解存档

$$\boldsymbol{S} = \begin{bmatrix} s_{11} & s_{12} & \cdots & s_{1n} \\ s_{21} & s_{21} & \cdots & s_{2n} \\ \vdots & \vdots & & \vdots \\ s_{N1} & s_{N2} & \cdots & s_{Nn} \end{bmatrix}, \quad (12)$$

根据定义式(3),计算解分量之间的有效相关性矩阵:

$$\boldsymbol{M}_{\mathrm{CR}} = \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & c_{21} & \cdots & c_{2n} \\ \vdots & \vdots & & \vdots \\ c_{n1} & c_{n2} & \cdots & c_{nn} \end{bmatrix}, \quad (13)$$

矩阵中 $c_{ii} = 0, c_{ij} = \dfrac{f_{ij}}{N^2}, i \neq j, f_{ij} = \sum_{k=1}^{N}\sum_{l=1}^{N} g(\mathrm{samsol}_{kl}, \mathrm{orsol}_{kl})$ 为有效相关性频数,其中

$$\mathrm{samsol}_{kl} = (s_{k1},\cdots,\mathrm{sample}_i(\boldsymbol{s}),\cdots,s_{lj},\cdots,s_{kn}),$$

$$g(\boldsymbol{x}',\boldsymbol{x}) = \begin{cases} 1, & f(\boldsymbol{x}') >_{\mathrm{opt}} f(\boldsymbol{x}), \\ 0, & \text{otherwise.} \end{cases}$$

步骤 2. 根据解分量之间的有效相关性矩阵(13),利用 MIMIC_c^G 分布估计算法估计解存档的高斯概率分布并根据解存档的分布模型进行采样,从而得到 m 个新搜索解 $\boldsymbol{s}_{N+1}, \boldsymbol{s}_{N+2}, \cdots, \boldsymbol{s}_{N+m}$.

1) 令解存档 $\boldsymbol{S}$ 服从 n 维高斯分布,对于解分量集合 $\boldsymbol{S} = (S_1, S_2, \cdots, S_n)$,根据 $i_n = \arg\min_j h(S_j)$ 找出排列 $\boldsymbol{\pi} = (i_1, i_2, \cdots, i_n)$ 中的 i_n;

2) 对任意 $k = n-1, n-2, \cdots, 1$,根据 $i_k = \arg\min_j h(S_j \mid S_{k+1})$(其中 $k \neq i_{k+1}, \cdots, i_n$)计算排列,$\boldsymbol{\pi} = (i_1, i_2, \cdots, i_n)$;

3) 根据有效相关性矩阵,计算概率分布 $f_\pi(\boldsymbol{s}) = f(s_{i_1} \mid s_{i_2}) f(s_{i_2} \mid s_{i_3}) \cdots f(s_{i_{n-1}} \mid s_{i_n}) f(s_{i_n})$.

步骤 3. 根据采样得到的新的解矩阵 $\boldsymbol{S}' = (\boldsymbol{s}_{N+1} \quad \boldsymbol{s}_{N+2} \quad \cdots \quad \boldsymbol{s}_{N+m})^{\mathrm{T}}$,利用 RPCA 找到其低秩结构矩阵 $\boldsymbol{B}$.

步骤 4.对步骤 3 得到低秩结构矩阵 $\boldsymbol{B}$ 的解分量进行评估,并更新解的目标函数值.

步骤 5.将 m 条新的解所对应的目标函数值与原来的解存档空间的 k 个解的目标函数值重新进行由优至劣的顺序排序,选取前 k 个解,作为新的解存档.

3 数值实验

3.1 测试函数

为测试算法的性能,并与连续域上的蚁群优化算法进行比较,本文对下面 6 个标准的测试函数进行测试.

Sumcan 函数:

$$f(\boldsymbol{x}) = -\left\{10^{-5} + \sum_{i=1}^{n} |\, y_i \,|\right\}^{-1}; y_1 = x_1; y_i = y_{i-1} + x_i, i = 2,\cdots,n; -0.16 \leqslant x_i \leqslant 0.16; \min f(x) = f(0,0,\cdots,0) = -10^5.$$

Sphere 函数:

$f(\boldsymbol{x})=\sum_{i=1}^{n}x_i^2,\ -10\leqslant x_i\leqslant 10;\min f(\boldsymbol{x})=f(0,0,\cdots,0)=0.$

Giewangk 函数:

$$f(\boldsymbol{x})=1+\sum_{i=1}^{n}\frac{x_i^2}{4\ 000}-\prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right)^2,\ -600\leqslant x_i\leqslant 600;\min f(\boldsymbol{x})=f(0,0,\cdots,0)=0.$$

Schwefel 函数:

$$f(\boldsymbol{x})=\sum_{i=1}^{n}|x_i|+\prod_{i=1}^{n}|x_i|,\ -10\leqslant x_i\leqslant 10;\min f(\boldsymbol{x})=f(0,0,\cdots,0)=0.$$

Rastrigin 函数:

$$f(\boldsymbol{x})=\sum_{i=1}^{n}(x_i^2-10\cos(2\pi x_i)+10),\ -5.12\leqslant x_i\leqslant 5.12;\min f(\boldsymbol{x})=f(0,0,\cdots,0)=0.$$

Easom 函数:

$$f(\boldsymbol{x})=-\cos(x_1)\cos(x_2)\mathrm{e}^{-((x_1-\pi)^2+(x_2-\pi)^2)},\ -100\leqslant x_1,x_2\leqslant 100;\min f(\boldsymbol{x})=f(\pi,\pi)=-1.$$

3.2 参数设置

在实验中,算法的参数设置如下:蚂蚁个数 $m=50$,保留最优解个数 $k=10$,最大迭代次数为 50,精度为 10^{-6},调试不同启发因子 q 与信息素挥发率 τ,分别进行 10 次实验.

3.3 实验结果与分析

为测试混合蚁群优化算法的性能,本文采用下列 4 种性能指标对实验结果进行分析,其中设定函数的维数 $n=10$.

1)在固定最大迭代次数内算法得到的最优值及平均最优值如表 1 所示.由表 1 知,混合 ACO_R 算法对 6 个测试函数进行 10 次实验后都可以很快地找到函数的最优值,并且与函数实际理论的最优值相同,说明混合 ACO_R 算法在固定的最大迭代次数内能够得到很好的优化结果,算法是可行的.同时,也可由表 1 看出,混合 ACO_R 算法的最优值和平均值都优于原来的 ACO_R 算法,这是由于混合 ACO_R 算法考虑变量之间的相关性,结合全局寻优能力强的 MIMIC 分布估计算法和 RPCA 技术,降低了变量之间的冗余相关性和陷入局部最优值的概率,收敛速度加快,在寻优能力上有了很大的提高.

表 1 50 次迭代内算法优化结果

Table 1 Optimization results of algorithms within 50 iterations

测试函数	$f(x^*)$ 理论最优值	混合 ACO_R		标准 ACO_R	
		最优值	平均值	最优值	平均值
Sumcan	-10^5	-10^5	-10^5	-69.769 3	-12.614 07
Sphere	0	0	0	0.000 010 788	7.804 636 077
Giewangk	0	0	0	0.059 188	16.468 821 8
Schwefel	0	0	0	0.026 068	18.352 690 5
Rastrigin	0	0	0	6.708 2	37.612 47
Easom	-1	-1	-1	-1	-0.999 971

2)表 2 是混合 ACO_R 算法与标准 ACO_R 算法达到理论最优值时的平均迭代次数的比较结果,其中 Sumcan 函数的最优值为 -10^5,Sphere、Giewangk、Schwefel 和 Rastrigin 函数最优值为 0,Easom 函数最优值为-1,"50+"代表算法在 50 次迭代后仍未达到理论最优值.由表 2 可以看出,对于 6 个测试函数,混合 ACO_R 算法的平均迭代次数远远优于标准 ACO_R 算法,平均迭代次数都很小,很快就达到了迭代次数少、收敛速度快的理想状态.

所有函数的标准 ACO_R 算法在最大迭代次数内没有达到理论最优值,部分函数后续迭代的最优值都在某个值上停留,所以认为算法很大可能已经陷入局部最优值.这说明与标准 ACO_R 算法相比,混合 ACO_R 算法收敛速度非常快,陷入局部最优解的概率非常小,基本能在最大迭代次数内收敛到最优值,大大减少收敛时的迭代次数,提高了解更新效率.

3)达标率指算法达到理论最优值的次数与实验总次数的比例.表 3 给出了标准 ACO_R 算法和混合 ACO_R 算法达到理论最优值时达标率的比较结果.由表 3 可以看出,对于 6 个测试函数,在混合 ACO_R 算法中都成功找到了函数理论最优值,达标率为 100%,即算法寻优过程中,陷入局部最优解的概率非常小;而标准的 ACO_R 算法除了 Easom 函数在 10 次实验中 9 次达到最优值,其他函数均未在固定迭代次数内找到理论最优值,达标率为 0,也就是说,标准的 ACO_R 算法在固定迭代次数内,函数收敛于局

表 2　到达理论最优值的平均迭代次数

Table 2　Average number of iterations reaching the theoretical optimal value

测试函数	混合 ACO_R	标准 ACO_R
Sumcan	2	50+
Sphere	4	50+
Giewangk	7	50+
Schwefel	5	50+
Rastrigin	8	50+
Easom	7	50+

表 3　到达理论最优值的次数与实验次数的比例

Table 3　Ratio of the number of arrivals at the theoretical optimal value to the number of experiments　%

测试函数	混合 ACO_R	标准 ACO_R
Sumcan	100	0
Sphere	100	0
Giewangk	100	0
Schwefel	100	0
Rastrigin	100	0
Easom	100	90

部最优值的概率较大,特别容易在寻优过程中发生停滞的现象.混合 ACO_R 算法的达标率远远优于标准 ACO_R 算法的达标率,这说明混合 ACO_R 算法有很高的有效性和可靠性.

图 3a—3f 分别为 6 个函数的标准 ACO_R 算法和混合 ACO_R 算法在相同迭代次数下的函数值的仿真实验结果.从图中可以很清晰地看出,与标准的 ACO_R 算法相比,混合的 ACO_R 算法可以在最初的几次迭代中快速成功找到函数的理论最优值,甚至如果选择合适的参数,对于部分函数,算法在第一次迭代就能成功找到函数的全局最优值,进而快速收敛;相比之下,标准的 ACO_R 算法在固定迭代次数内,全局寻优能力和收敛速度上都不是很理想.

4)在迭代次数不变的条件下,改变函数维数,比较两种算法的寻优性能(表 4 和表 5).下面是测试函数维数分别为 10 维和 30 维的情况下,标准 ACO_R 算法和混合 ACO_R 算法函数最优值与平均值的比较,实验结果可以体现不同维数对两种算法的影响.表 4 是不同维度下两种算法最优值的比较,表 5 是不同维度下两种算法平均值的比较,其中本次实验的固定迭代次数为 50 次.

由表 4 和表 5 可以看出,随着测试函数维数的增多,标准 ACO_R 算法寻找最优值的能力在减弱,但是混合 ACO_R 算法并没有表现出明显的寻优能力下降的趋势,不管是 30 维还是 10 维,混合 ACO_R 算法在实验中得出的最优值和平均值都比标准的 ACO_R 算法要好得多,且维数越高,优势越明显.这是由于当函数维数增大时,混合 ACO_R 算法考虑了变量之

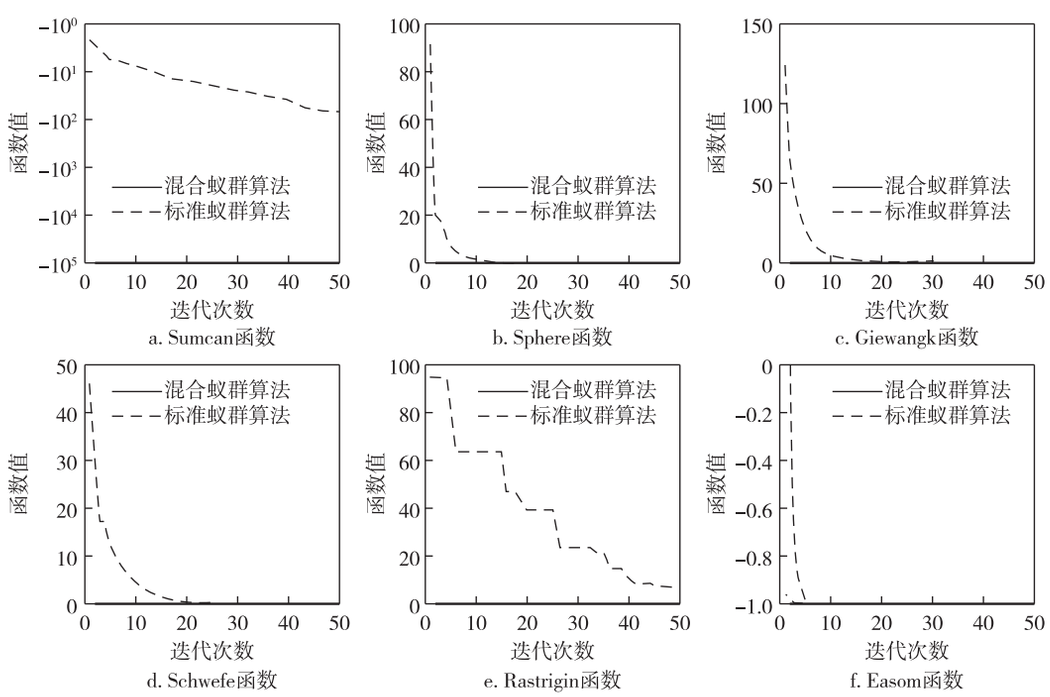

图 3　两种算法函数收敛曲线比较

Fig. 3　Comparison of convergence curves between the two algorithms

表 4 不同维度下两种算法最优值的比较

Table 4 Comparison of optimal values between the two algorithms under different dimensions

测试函数	$f(x^*)$ 理论最优值	$n=10$		$n=30$	
		ACO_R 最优值	混合 ACO_R 最优值	ACO_R 最优值	混合 ACO_R 最优值
Sumcan	-10^5	-69.769 3	-10^5	-1.014 1	-10^5
Sphere	0	0.000 010 788	0	52.544 6	0
Giewangk	0	0.059 188	0	5.777 2	0
Schwefel	0	0.026 068	0	32.239 2	0
Rastrigin	0	6.708 2	0	95.440 7	0

表 5 不同维度下两种算法平均值的比较

Table 5 Comparison of the average values between the two algorithms under different dimensions

测试函数	$f(x^*)$ 理论最优值	$n=10$		$n=30$	
		ACO_R 平均值	混合 ACO_R 平均值	ACO_R 平均值	混合 ACO_R 平均值
Sumcan	-10^5	-12.614 07	-10^5	-0.748 91	-10^5
Sphere	0	7.804 636 077	0	242.423 63	0
Giewangk	0	16.468 821 8	0	138.854 31	0
Schwefel	0	18.352 690 5	0	100.779 689 1	0
Rastrigin	0	37.612 47	0	231.691 64	0

间的相关性,并且利用 RPCA 技术尽可能精确地保留有效相关系数大的解分量,去除大的异常解分量和冗余的解分量,所以,虽然函数维数增大,但是最终得到的对函数值起主要作用的解分量并没有增加,反而因为维数的增加,优化了搜索空间,可以更容易找到全局最优值的搜索邻域,从而提高算法的寻优性能.

4 结束语

本文针对标准的 ACO_R 算法存在的不足,考虑算法中决策变量之间的相关性,根据变量有效相关性的定义,提出一种基于 MIMIC 算法和 RPCA 的混合蚁群算法.经仿真实验结果验证,与标准 ACO_R 算法相比,混合 ACO_R 算法可以十分精确地找到最优值,且到达理论最优值时的迭代次数和达标率都远远优于标准算法.混合蚁群优化算法有 MIMIC 算法全局搜索能力强的优点,又包含蚁群算法的局部求精能力强、收敛速度快的优点;对于高维空间上的寻优过程,由于混合 ACO_R 算法加入 RPCA 技术,在保证变量之间的有效相关性和系统鲁棒性的同时,保留主要解空间结构,降低计算复杂度与解变量之间的冗余相关性,优化搜索空间,使得该算法在解决高维优化问题时仍具有很强的寻优能力.

参考文献

References

[1] Colorni A, Dorigo M, Maniezzo V, et al. Distributed optimization by ant colonies[C]//Toward a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life. MIT Press/Bradford Books, 1992:134-142

[2] Dorigo M, Maniezzo V, Colorni A. Ant system: optimization by a colony of cooperating agents[J]. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 1996, 26(1):29-41

[3] Dorigo M, Birattari M, Stutzle T. Ant colony optimization [J]. IEEE Computational Intelligence Magazine, 2006, 1(4):28-39

[4] Gambardella L M, Dorigo M. Solving symmetric and asymmetric TSPs by ant colonies[C]//Proceedings of IEEE International Conference on Evolutionary Computation, 1996:622-627

[5] Dorigo M, Gambardella L M. A study of some properties of Ant-Q[C]//Parallel Problem Solving from Nature: PPSN IV. Berlin, Heidelberg: Springer Berlin Heidelberg, 1996: 656-665

[6] Stützle T, Hoos H H. MAX-MIN ant system[J]. Future Generation Computer Systems, 2000, 16(8):889-914

[7] Socha K, Dorigo M. Ant colony optimization for continuous domains[J]. European Journal of Operational Research, 2008, 185(3):1155-1173

[8] Tang K, Yáo X, Suganthan P N, et al. Benchmark functions for the CEC'2008 special session and competition on large scale global optimization [R]. Nature Inspired Computation and Applications Laboratory, USTC, China, 2008:1-18

[9] Chen W X,Weise T,Yang Z Y,et al.Large-scale global optimization using cooperative coevolution with variable interaction learning[C]// Parallel Problem Solving from Nature:PPSN XI,2010:300-309.DOI:10.1007/978-3-642-15871-1_31
[10] Larrañaga P,Lozano J A.Estimation of distribution algorithms[M].Boston,MA:Springer US,2002
[11] Mühlenbein H,Paaß G.From recombination of genes to the estimation of distributions Ⅰ:binary parameters[C]// Parallel Problem Solving from Nature:PPSN IV,1996:178-187.DOI:10.1007/3-540-61723-X_982
[12] de Bonet J S,Isbell Jr C L,Viola P A.MIMIC:finding optima by estimating probability densities[C]// Advances in Neural Information Processing Systems,1997:424-430
[13] Larrañaga P, Etxeberria R, Lozano J A, et al. Combinatorial optimization by learning and simulation of Bayesian networks[C]// Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence,2000:343-352
[14] Anderson T W.Asymptotic theory for principal component analysis[J].The Annals of Mathematical Statistics,1963,34(1):122-148
[15] Candès E J,Li X D,Ma Y,et al.Robust principal component analysis? [J].Journal of the ACM,2011,58(3):1-37
[16] Chandrasekaran V,Sanghavi S,Parrilo P A,et al.Rank-sparsity incoherence for matrix decomposition[J].SIAM Journal on Optimization,2011,21(2):572-596
[17] Goldfarb D, Qin Z. Robust low-rank tensor recovery: models and algorithms[J].SIAM Journal on Matrix Analysis and Applications,2014,35(1):225-253

A hybrid ant colony optimization algorithm based on MIMIC algorithm and RPCA

GUAN Juan[1] LIU Guohua[1] LIU Tianqi[1] QIN Jian[1] ZHANG Miaosen[1]
1 School of Mathematics,Southeast University,Nanjing 210096

Abstract To improve the optimization performance of the ant colony optimization algorithm for continuous domains and reduce the correlation among decision variables,a hybrid ant colony optimization algorithm based on MIMIC (Mutual Information Maximization for Input Clustering) algorithm and RPCA (Robust Principal Component Analysis) is designed. Firstly, the ant colony optimization algorithm for continuous domains is introduced. Then, a definition of effective correlation to deal with multivariable correlation is given and a hybrid ant colony optimization algorithm based on MIMIC algorithm and RPCA is proposed.Finally,through solving the standard test functions and comparing the results with that of the ant colony optimization algorithm for continuous domains,the proposed algorithm is proved to have great improvement in optimization ability and convergence,therefore,it is an effective optimization algorithm.

Key words ant colony optimization algorithm;variable correlation;MIMIC algorithm;robust principal component analysis (RPCA)