求解工程约束优化问题的 PSO-ABC 混合算法*

王珂珂，吕　强，赵汗青，白　帆
（装甲兵工程学院，北京 100072）

摘　要：针对包含约束条件的工程优化问题，提出了基于人工蜂群的粒子群优化 PSO-ABC 算法。将 PSO 中较优的粒子作为 ABC 算法的蜜源，并使用禁忌表存储其局部极值，克服粒子群优化算法易陷入局部最优的缺陷。采用可行性规则进行约束处理，将粒子种群分为可行子群和不可行子群，并在 ABC 算法产生蜜源的过程中保留部分较优的可行解和不可行解的信息，弥补了可行性规则处理最优点位于约束边界附近的问题时存在的不足。四个典型工程优化设计的实验结果表明，该算法能够寻得更优的约束最优化解，且稳健性更强。

关键词：粒子群优化；人工蜂群；工程约束优化；可行性规则；禁忌表
中图分类号：TP18　　**文献标志码**：A　　**文章编号**：1001-3695(2012)04-1230-04
doi:10.3969/j.issn.1001-3695.2012.04.008

Particle swarm optimization based on artificial bee colony for solving engineering constrained optimization problems

WANG Ke-ke, LV Qiang, ZHAO Han-qing, BAI Fan
(*Academy of Armored Force Engineering*, *Beijing* 100072, *China*)

Abstract: In order to solve engineering constrained optimization problems, this paper proposed a hybrid method combining particle swarm optimization(PSO) and artificial bee colony(ABC). The method selected the better particles in PSO as food sources for ABC algorithm, and used the tabu table to save the local optimization results in order to avoid PSO trapping into local optimum. And it used a feasibility-based rule to solve constrained problems, and divided the particle swarm into feasible subpopulation and infeasible subpopulation. So it produced the new food sources containing the information of good feasible and infeasible solution in the process of ABC, which could make up for the feasibility-based rule being invalid when the optimum was close to the boundary of constraint conditions. The algorithm was validated using four standard engineering design problems. The results indicate that PSO-ABC algorithm can find out better optimum and has stronger solidity.

Key words: particle swarm optimization(PSO); artificial bee colony(ABC); engineering constrained optimization; feasibility-based rule; tabu table

约束优化问题是工程应用领域经常遇到的一类数学规划问题，对此类问题的求解最终可归结为对一个带有约束条件的函数进行优化。与传统优化方法相比，粒子群优化算法对函数本身性质要求非常低，只要求被优化的函数值是可计算的，并不要求它具有连续性及可微性，所以它是一种可以广泛应用于工程领域复杂优化问题的有效算法。同时，它又是基于群体进化的算法，能以较大的概率收敛于全局最优解，所以它克服了一般的基于梯度优化方法局部收敛的一些缺点，在解决全局优化问题方面显示了极大的潜在优势。Eberhart 提出 PSO 算法后，又将其用于求解有约束条件的最优化问题[1]，证实了 PSO 算法在该领域的可行性。随后，文献[2～4]也对 PSO 算法求解约束优化问题进行了研究，结果表明，PSO 算法除了具有进化算法共有的优势之外，操作更简便、搜索速度更快，在约束优化问题领域具有很强的应用价值。

但是 PSO 算法也存在一些固有的缺陷，如搜索过程中易于陷入局部最优，在进化后期优化效率较低，甚至会进入停滞状态。为了解决这些问题，本文在前人的启发和研究成果的基础上，将 PSO 算法和 ABC 算法相结合，充分利用 PSO 算法的快速收敛性、较强的全局搜索能力和 ABC 算法具有跳出局部极值点区域的能力，给出一种新的基于 ABC 的 PSO(particle swarm optimization-artificial bee colony，PSO-ABC)算法，在 ABC 算法中引入禁忌表，并利用基于可行性规则的约束处理技术求解约束优化问题。

1　约束优化问题

1.1　问题的描述

约束优化问题(constrained optimization problems，COPs)是一类广泛存在于实际工程中但又较难求解的问题，因而对其研究具有十分重要的理论和实际意义。不失一般性，以最小化问题为例，约束优化问题可描述为

$$\min_{x\in R^n} F(x) = (f_1(x), f_2(x), \cdots, f_k(x))$$

收稿日期：2011-09-13；**修回日期**：2011-10-20　　**基金项目**：军队科研计划资助项目

作者简介：王珂珂(1982-)，男，河南夏邑人，博士研究生，主要研究方向为智能优化算法、自主车辆运动规划(pengkily@163.com)；吕强(1962-)，男，教授，博士，主要研究方向为智能控制技术；赵汗青(1961-)，男，副教授，博士，主要研究方向为自主车辆相关技术；白帆(1982-)，男，博士研究生，主要研究方向为智能火控系统、群智能优化算法.

$$\text{s.t.}\quad g_i(\boldsymbol{x}) \leqslant 0, i=1,\cdots,p$$
$$h_j(\boldsymbol{x}) = 0, j=1,\cdots,q$$

其中:$F(\boldsymbol{x})$为目标函数;$g_i(\boldsymbol{x})$、$h_j(\boldsymbol{x})$为约束条件;$\boldsymbol{x}=(x_1,x_2,\cdots,x_n)\in R^n$ 称为 n 维决策向量,$\boldsymbol{x}\in\Omega\subseteq S$。将满足所有约束条件的解空间 Ω 称为可行域,S 称为决策空间。S 为 R^n 中的 n 维长方体:$l(i)\leqslant x_i\leqslant u(i)$,$l(i)$、$u(i)$为常数,$i=1,\cdots,n$。$k=1$ 时为单目标优化问题,$k>1$ 时为多目标优化问题。$g_i(\boldsymbol{x})$是第 i 个不等式约束,$h_j(\boldsymbol{x})$是第 j 个等式约束。约束优化问题就是在保证满足上述等式和不等式约束条件的前提下,在变量的定义范围内寻找使得目标函数达到最优的解。

1.2　约束处理技术

智能算法已被广泛应用于求解约束优化问题。然而,其性能的好坏除了取决于算法的优化性能外,还受约束处理方法的影响,即如何将约束优化问题的目标函数和约束条件转换为适应值函数,因为适应值函数可以使搜索朝着合理的区域进行。因此,约束处理技术是求解约束优化问题的一个关键部分。

目前应用最广、研究最多的是罚函数法[5]。其基本思想是通过对违反约束条件的个体,通过在其目标函数 $f(\boldsymbol{x})$ 上增加一个惩罚项 $p(\boldsymbol{x})$来构造惩罚适应值函数 fitness($\boldsymbol{x}$),将约束优化问题转换为无约束优化问题进行处理。罚函数法的主要优点是简单易行,其主要缺点是难以选择合适的罚因子,用以确定对不可行解的合适的惩罚力度。虽然近年来出现了动态罚函数、自适应罚函数和退火罚函数等改进方法,但是这些方法本身又引入了大量的额外参数,且实现过程复杂。

近年来,将约束优化问题转换为多目标优化问题(multi-objective optimization problems,MOPs)来处理受到人们极大的重视。文献[6]给出了几种基于进化多目标优化概念的约束条件处理技术,这类算法的主要特点是将约束条件作为一个或多个目标来看待。基于上述思想的算法可分为两类:区分可行解与不可行解法和多目标优化法。区分可行解与不可行解法是将约束优化问题定义为具有两个目标的多目标优化问题。典型算法有 Deb[7] 提出的一种不需要任何罚参数的联赛选择算子,并采用三个可行性规则来比较成对个体的优劣;Mezura-Montes 等人[8] 也采用了类似的思想,提出一种简单的用于求解约束优化问题的多成员进化策略;林丹等人[9] 提出了一种将直接比较方法与在进化群体中自适应地保持不可行解比例的策略相结合来处理约束条件的新方法,用以在群体中保持一定比例的接近可行域边界的不可行解,这对于找到全局最优解很有帮助。

2　PSO-ABC 算法求解约束优化问题

2.1　标准 PSO 算法[10] 简介

考虑一个全局优化问题

$$\min\{f(\boldsymbol{x}):\boldsymbol{x}\in\Omega\subset R^n\}\quad f,\Omega\subset R^n$$

PSO 算法将优化问题的潜在解看做是搜索空间中一个没有质量和体积的粒子,它具有一定的飞行速度和位置,用有序三元组($\boldsymbol{x}^i$,v^i,P^i)来分别表示第 i 个粒子的位置、速度和自身经历的最好位置。令所有粒子经历的全局最优位置为 $G_{\text{best}}(t)=(G_1,G_2,\cdots,G_n)$,则粒子在 $t+1$ 代的速度和位置更新公式为

$$v_j^i(t+1)=v_j^i(t)+c_1r_1(P_j^i(t)-x_j^i(t))+c_2r_2(G_j(t)-x_j^i(t))$$
$$x_j^i(t+1)=x_j^i(t)+v_j^i(t+1)$$

其中:c_1 和 c_2 为加速度常数,用于调整粒子向自身经历的最优位置飞行的最大幅度和向全局最优位置飞行的最大幅度。此外,为了防止粒子飞到搜索空间之外,它的位置被边界 Ω 限制、速度被最大速度 $v_{\max}$限制。r_1 和 r_2 是[0,1]范围内服从均匀分布的随机数。速度更新公式中第一部分表示粒子先前的速度;第二部分为认知(cognition)部分,表示粒子自身的思考;第三部分为社会(social)部分,表示粒子之间的信息共享以及粒子对群体共有信息的认可程度。PSO 算法的重要原则是在寻求共同认知的过程中,个体会保留自己的最佳信念,同时也会考虑群体的信念,当群体的信念优于个体的信念时,需要动态更新个体的最佳信念。

2.2　PSO-ABC 算法

个体最优值和全局最优值的更新对 PSO 算法非常重要,因此,本文首先用 Deb 提出的可行性规则进行约束处理,并按照优劣程度对粒子进行排序。

规则 1　如果两个粒子都在可行域内,则适应值较高的粒子获胜,排序靠前。

规则 2　如果一个粒子在可行域内,而另一个在可行域外,则可行域内的粒子获胜,排序靠前。

规则 3　如果两个粒子都在可行域外,则违反约束程度小的粒子获胜,排序靠前。

显然,可行性规则将目标函数和约束违反程度信息分开考虑,无须额外的附加参数,易于实现,且没有问题依赖性。

本文将可行域外的粒子 $\boldsymbol{x}$ 对约束的违反程度定义如下:

$$\text{viol}(\boldsymbol{x})=\frac{\tau_1}{p}\sum_{i=1}^{p}\{\max(0,g_i(\boldsymbol{x}))\}+\frac{\tau_2}{q}\sum_{j=1}^{q}\{\max(0,|h_j(\boldsymbol{x})|-\varepsilon)\}$$

其中:τ_1 和 τ_2 分别表示该粒子违反不等式和等式约束条件的个数。

参考上述规则,将粒子群分为可行子群 $\boldsymbol{\Theta}_F$ 和不可行子群 $\boldsymbol{\Theta}_V$ 两组,其子群规模分别为 N_F 和 N_V。

为了改善 PSO 算法易陷入局部极值点、过早停滞等现象,本文在 PSO 迭代过程中引入 ABC 算法。在 ABC 算法中[11~13],人工蜂群包含三种个体:雇佣蜂、观察蜂和侦查蜂。在迭代过程中,蜂群对于蜂源的搜索遵循下述规则:雇佣蜂每次的搜索目标为上次获得的解的邻域解;在每次返回后,雇佣蜂将蜜源的信息反馈给观察蜂;观察蜂按照一定的策略在不同的蜜源中选择一个作为目标,并在其邻域内进行搜索;若雇佣蜂在设定的搜索次数 limit 内没有获得更好的蜜源,便放弃该蜜源,同时雇佣蜂变成侦查蜂,开始随机搜索可行的新蜜源。在迭代过程中,蜜源和雇佣蜂是一一对应关系,即雇佣蜂数目和蜜源数目相等;雇佣蜂和观察蜂各占蜂群总数的一半。在 ABC 算法中,蜜源的位置对应优化问题的一个解。

PSO-ABC 算法思想如下:从 $\boldsymbol{\Theta}_F$ 和 $\boldsymbol{\Theta}_V$ 中分别选择 N_{SF} 和 N_{SV} 个该子群中较优的粒子集合 $\boldsymbol{\Theta}_{SF}$ 和 $\boldsymbol{\Theta}_{SV}$ 构成蜜源,蜜源的数量为 N_S,$N_S=N_{SF}+N_{SV}$。每个雇佣蜂有一个确定的蜜源,并在迭代中对蜜源的邻域进行搜索。在每次返回之后,计算雇佣蜂对应蜜源的收益度 fit(i,t)。若蜜源为可行解,则蜜源的收益度对应问题的适应度函数,即 $\text{fit}_F(i,t)=\text{fitness}(i)$,$i=1,2,\cdots,N_{SF}$;若蜜源为不可行解,则蜜源的收益度对应解的约束违反程度,即 $\text{fit}_V(i,t)=\text{viol}(i)$,$i=1,2,\cdots,N_{SV}$。

雇佣蜂将蜜源的信息反馈给观察蜂,观察蜂根据轮盘赌方

法选择一个蜜源作为目标。对于可行蜜源,其选择概率为

$$P_{Fi}=\frac{F(i)}{\sum_{i=1}^{N_{SF}}F(i)}\quad i=1,2,\cdots,N_{SF}$$

其中:
$$F(i)=\begin{cases}\dfrac{1}{1+\mathrm{fit}_F(i,t)} & \text{if } \mathrm{fit}_F(i,t)\geqslant 0\\ 1+|\mathrm{fit}_F(i,t)| & \text{else}\end{cases}$$

对于不可行蜜源,其选择概率为

$$P_{Vi}=\frac{V(i)}{\sum_{i=1}^{N_{SV}}V(i)}\quad i=1,2,\cdots,N_{SV}$$

其中:
$$V(i)=\begin{cases}\dfrac{1}{1+\mathrm{fit}_V(i,t)} & \text{if } \mathrm{fit}_V(i,t)\geqslant 0\\ 1+|\mathrm{fit}_V(i,t)| & \text{else}\end{cases}$$

观察蜂选择蜜源之后,在其对应的邻域内选择新的蜜源进行访问,选择规则如下:

$$x_{i,t+1}=x_{i,t}+\psi_{i,t}$$

其中:$\psi_{i,t}$为随机步长。完成观察蜂对新蜜源的搜索后,将新的蜜源信息与 Θ_{SF}和 Θ_{SV}的粒子进行共享,返回 PSO 算法。若雇佣蜂在设定的搜索次数 limit 内没有获得更好的蜜源,便放弃该蜜源,并将该蜜源信息记录在禁忌表中。同时,雇佣蜂成为侦查蜂,并开始随机搜索新蜜源。

在每次迭代过程中,由 PSO 算法新产生的蜜源首先与原蜜源进行比较,若新产生的蜜源收益度比原蜜源的收益度高,则雇佣蜂更新现有蜜源;否则不更新。

2.3 PSO-ABC 求解约束优化流程

PSO-ABC 算法的实现过程如下:

a)执行一次标准 PSO 算法,并将种群分为可行子群 Θ_F 和不可行子群 Θ_V。

b)将 Θ_F 和 Θ_V 中的粒子进行排序,分别择优选取 N_{SF}个可行粒子和 N_{SV}个不可行粒子组成 N_S 个蜜源,并产生数量相同的雇佣蜂和观察蜂。

c)蜜源更新 I。将新选取的蜜源与前一次 ABC 迭代寻找的蜜源进行比较,如果收益度改善,则更新蜜源;否则保持原蜜源。

d)计算雇佣蜂所对应的蜜源收益度。

e)雇佣蜂与观察蜂分享蜜源的信息,观察蜂按照轮盘赌策略选择蜜源,并且在该蜜源邻域附近搜索新蜜源。

f)更新蜜源 Ⅱ。将搜索到的新蜜源先与禁忌表比较,如果新蜜源在禁忌表中,则重新从 Θ_F 和 Θ_V 选择新的蜜源。确定新蜜源不在其中之后对新蜜源计算适应度值,并根据贪婪机制更新蜜源。若迭代超过限定次数后,蜜源得不到更新,则放弃该蜜源,与之相应的雇佣蜂转变为侦查蜂搜索新的蜜源。同时,将局部最优值点存储在禁忌表内。

g)更新 Θ_{SF}和 Θ_{SV}。将新的蜜源信息与 Θ_{SF}和 Θ_{SV}中的粒子进行信息共享,并查看是否满足设定的终止条件。若满足,则终止迭代,输出最优解;否则,返回 PSO 算法。

上述步骤 b)中选择蜜源的方式综合考虑了当前最优的可行解信息和对约束违反程度最低的不可行解信息,其中 N_{SF}和 N_{SV}的取值决定它们所占比重。这种做法既可以保持当前全局最优值 G_{best},维持粒子群的“群体信念”特性不丧失,又能够将必要的优秀的不可行解信息“遗传”下来,对求解一些最优解在约束条件边界上或附近的问题非常有利。

整个 PSO-ABC 算法的流程如图 1 所示。

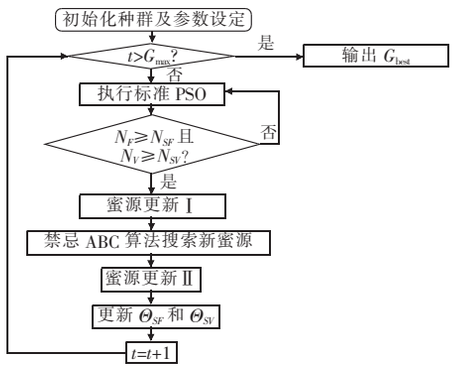

图 1 PSO-ABC 算法流程

3 性能验证

3.1 工程约束优化问题实例

工程优化设计问题通常被用来检验约束优化算法的有效性,例如典型的拉力/压力弹簧优化设计、焊接梁优化设计、齿轮传动机构优化设计和压力容器优化设计问题等。本文将采用上述四种典型的优化设计问题检验文中 PSO-ABC 算法的有效性,并与参考文献[3]所得的最优解相比较。四个典型工程优化设计问题的表述来自文献[3]。

1)拉力/压力弹簧优化设计 拉力/压力弹簧优化设计的目标是在满足最小挠度、剪应力和振动频率等约束下最小化其质量。设计变量分别为弹簧线圈直径 d、簧圈平均直径 D 和有效绕圈数 N,用向量 $\boldsymbol{x}=(x_1,x_2,x_3)$表示,如图 2 所示。

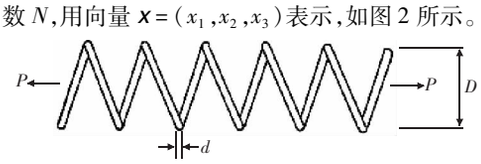

图 2 拉力/压力弹簧设计问题

目标函数和约束条件如下所示:

$$\min f(\boldsymbol{x})=(x_3+2)x_1^2x_2$$

$$\text{s.t.}\quad g_1(\boldsymbol{x})=1-\frac{x_2^3x_3}{71785x_1^4}\leqslant 0$$

$$g_2(\boldsymbol{x})=\frac{4x_2^2-x_1x_2}{12566x_1^3x_2-x_1^4}+\frac{1}{5108x_1^2}-1\leqslant 0$$

$$g_3(\boldsymbol{x})=1-\frac{140.45x_1}{x_2^2x_3}\leqslant 0,\quad g_4(\boldsymbol{x})=\frac{x_1+x_2}{1.5}-1\leqslant 0$$

设计变量的取值范围为

$$0.05\leqslant x_1\leqslant 2.0,\quad 0.25\leqslant x_2\leqslant 1.3,\quad 2.0\leqslant x_3\leqslant 15.0$$

2)焊接梁优化设计 焊接梁优化的目标是在一定的约束条件下使其总体成本最小。焊接梁的结构如图 3 所示,设计变量分别为 h、l、t 和 b,用向量 $\boldsymbol{x}=(x_1,x_2,x_3,x_4)$表示;约束变量为剪应力 τ,梁上的弯曲应力 σ,压曲临界载荷 Pc 和梁的尾端误差 δ。目标函数和约束条件如下所示:

$$\min f(\boldsymbol{x})=1.10471x_1^2x_2+0.04811x_3x_4(14.0+x_2)$$

$$\text{s.t.}\quad g_1(\boldsymbol{x})=\tau(\boldsymbol{x})-\tau_{\max}\leqslant 0,\quad g_2(\boldsymbol{x})=\sigma(\boldsymbol{x})-\sigma_{\max}\leqslant 0$$

$$g_3(\boldsymbol{x})=x_1-x_4\leqslant 0$$

$$g_4(\boldsymbol{x})=0.10471x_1^2+0.04811x_3x_4(14.0+x_2)-5.0\leqslant 0$$

$$g_5(\boldsymbol{x})=0.125-x_1\leqslant 0$$

$$g_6(\boldsymbol{x})=\delta(\boldsymbol{x})-\delta_{\max}\leqslant 0,\quad g_7(\boldsymbol{x})=P-Pc(\boldsymbol{x})\leqslant 0$$

其中,$\tau(\boldsymbol{x})=\sqrt{(\tau')^2+\dfrac{(2\tau'\tau'')x_2}{2R}+(\tau'')^2}$,$\tau'=\dfrac{P}{\sqrt{2}x_1x_2}$,$\tau''=$

$\frac{MR}{J}$, $M = P(L + \frac{x_2}{2})$, $R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2}$, $J = 2\{\sqrt{2}x_1x_2 \times [\frac{x_2^2}{12} + (\frac{x_1 + x_3}{2})^2]\}$, $\sigma(x) = \frac{6PL}{x_3^2x_4}$, $\delta(x) = \frac{4PL^3}{Ex_3^3x_4}$, $Pc(x) = \frac{4.013E}{L^2}\sqrt{\frac{x_3^2x_4^6}{36}}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right)$; $P = 6000$ (lb), $L = 14$ (in), $E = 3 \times 10^6$ (psi), $G = 12 \times 10^6$ (psi), $\tau_{max} = 13600$ (psi), $\sigma_{max} = 30000$ (psi), $\delta_{max} = 0.25$ (in)。

设计变量的取值范围为

$$0.1 \leqslant x_1, x_4 \leqslant 2.0; \quad 0.1 \leqslant x_2, x_3 \leqslant 10.0$$

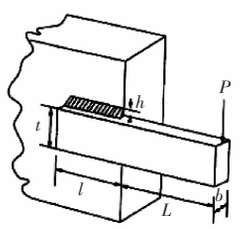

图 3　焊接梁设计问题

3)齿轮传动机构优化设计　其优化目标是如何设计齿轮盘的齿数使得传动机构的齿轮齿数比的代价最小,如图 4 所示。设计变量为齿轮齿数 n_A、n_B、n_D 和 n_F,均为整数,用向量 $x = (x_1, x_2, x_3, x_4)$ 表示。目标函数和约束条件如下所示:

$$\min f(x) = \left(\frac{1}{6.931} - \frac{x_3x_2}{x_1x_4}\right)^2$$

$$\text{s.t.} \quad 12 \leqslant x_i \leqslant 60, \quad i = 1, \cdots, 4, \quad x_i \in Z$$

4)压力容器优化设计　该问题是混合的约束优化问题,如图 5 所示,它的优化目标是造价最小。设计变量分别是两端帽的厚度 T_h、壳的厚度 T_s、内径 R、容器中间部分圆柱的长度 L,用向量 $x = (x_1, x_2, x_3, x_4)$ 表示。其中,变量 T_h 和 T_s 是 0.0625 in 的整数倍,因为它是标准冷扎钢板的有效厚度。其目标函数和约束条件如下:

$$\min f(x) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3$$

$$\text{s.t.} \quad g_1(x) = -x_1 + 0.0193x_3 \leqslant 0$$

$$g_2(x) = -x_2 + 0.00954x_3 \leqslant 0$$

$$g_3(x) = -\pi x_3^2x_4 - \frac{4}{3}\pi x_3^3 + 1296000 \leqslant 0$$

$$g_4(x) = x_4 - 240 \leqslant 0$$

变量的取值范围为

$$1 \times 0.0625 \leqslant x_1, x_2 \leqslant 99 \times 0.0625; \quad 10.0 \leqslant x_3, x_4 \leqslant 200.0$$

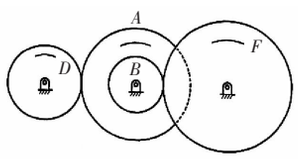

图 4　齿轮设计问题

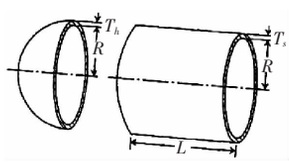

图 5　压力容器设计问题

3.2　参数设置与结果分析

本文用一定迭代周期之后寻找到的最优解的质量来考察 PSO-ABC 算法的性能。为减少随机性对算法性能的影响,算法对每个工程优化设计问题独立运行 100 次,并分析它们的统计特征,计算所得到的平均结果(平均值、标准差、最优值和最劣值),并与文献[3]中的算法性能进行比较。文献[3]中种群规模 $N = 20$,最大迭代次数 $T_{max} = 5\ 000$。本文中 PSO-ABC 算法的参数设置为 $N = 20$, $N_{SF} = 2$, $N_{SV} = 2$, $c_1 = c_2 = 1.454$, $[\omega_{min}, \omega_{max}] = [0.4, 0.9]$, $T_{max} = 1\ 000$。表 1 给出了 PSO-ABC 算法寻找到的各优化设计问题的最优解。通过对 PSO-ABC 算法的实验发现,上述参数设置中,当 $N_{SF} = N/10 \sim N/5$, $N_{SV} = N/10 \sim N/8$ 时,算法整体性能较好。

表 1　PSO-ABC 算法解得的最优解

设计 1	最优解	设计 2	最优解	设计 3	最优解	设计 4	最优解
x_1	0.050 00	x_1	0.205 73	x_1	34	x_1	0.812 5
x_2	0.374 43	x_2	3.470 45	x_2	20	x_2	0.437 5
x_3	3.200 11	x_3	9.036 62	x_3	13	x_3	42.098 45
$g_1(x)$	0	x_4	0.205 73	x_4	53	x_4	176.636 60
$g_2(x)$	0	$g_1(x)$	0			$g_1(x)$	0
$g_3(x)$	-14.653 00	$g_2(x)$	0			$g_2(x)$	-0.035 88
$g_4(x)$	-0.717 04	$g_3(x)$	0			$g_3(x)$	0
		$g_4(x)$	-3.432 98			$g_4(x)$	-63.363 40
		$g_5(x)$	-0.080 73				
		$g_6(x)$	-0.235 54				
		$g_7(x)$	-3.91083e-11				
$f(x)$	4.86774e-3	$f(x)$	1.724 85	$f(x)$	2.30781e-11	$f(x)$	6 059.714 34

PSO-ABC 算法所得的统计结果如表 2 所示。除齿轮传动机构优化设计外,PSO-ABC 算法对另外三个设计问题的寻优结果均比文献[3]提供的标准 UPSO 和 UPSOm 有较大提高。从统计结果来看,对于平均最优值,在解优化设计问题 1)~4)时, PSO-ABC 算法均优于 UPSO 算法;且 PSO-ABC 算法对设计 1)和 3)的评价优化结果比文献[3]获得的结果提高了一个数量级,这说明文中所给算法的整体性能有很大提高。同时, PSO-ABC 算法的均方差结果均优于 UPSO 算法,尤其对于设计 2)和 3)改善更为明显,说明文中所给算法的稳健性得到了很大提高。但是对于优化问题 3), PSO-ABC 求得的最优解较 UPSO 稍逊一些,这说明文中所给算法在求解整数优化问题时,优化精度有待进一步提高。从整体优化效果上看,在求解约束优化问题时 PSO-ABC 算法优于文献[3]提供的算法 UPSO。

表 2　实验统计结果比较

实例		标准 UPSO(*)				UPSOm(*)	PSO-ABC
		u=0	u=0.2	u=0.5	u=1		
1	mean	2.32563e-2	1.19291e-1	4.67351e-2	4.19581e-2	2.29478e-2	5.36297e-3
	StD	7.48230e-3	5.42710e-1	2.14505e-1	2.84724e-2	7.20571e-3	4.3230e-4
	min	1.28404e-2	1.31269e-2	1.28158e-2	1.30803e-2	1.31200e-2	4.86774e-3
	max	4.87550e-2	4.12260	1.57998	1.98921e-1	5.03651e-2	5.73656e-3
2	mean	2.588 69	2.297 18	1.968 20	4.279 85	2.837 21	1.773 76
	StD	5.01437e-1	4.10969e-1	1.55415e-1	1.36945	6.82980e-1	8.28525e-2
	min	1.830 08	1.824 40	1.765 58	1.918 53	1.921 99	1.724 85
	max	4.132 07	4.173 82	2.844 06	8.912 70	4.883 60	2.012 35
3	mean	3.92135e-8	7.55581e-8	2.83820e-7	1.64225e-6	3.80562e-8	4.97092e-9
	StD	7.71670e-8	1.83057e-7	6.87035e-7	8.28521e-6	1.09631e-7	6.95671e-9
	min	2.70085e-12	2.70085e-12	2.30781e-11	8.88761e-10	2.70085e-12	2.30781e-11
	max	6.41703e-7	8.94899e-7	5.69940e-6	8.19750e-5	8.94899e-7	4.47442e-8
4	mean	9.19585e+3	8.66971e+3	8.01637e+3	1.35035e+5	9.03255e+3	6.36100e+3
	StD	9.60268e+2	6.24907e+2	7.45869e+2	1.51116e+5	9.95573e+2	2.40055e+2
	min	7.56796e+3	6.77080e+3	6.15470e+3	7.52706e+3	6.54427e+3	6.05971e+3
	max	1.26720e+4	1.01895e+4	9.38777e+3	5.59300e+5	1.16382e+4	6.82041e+3

注:(*)表示文献[3]提供的原始数据

4　结束语

本文中将两种智能优化算法——PS0 算法和 ABC 优化算法相结合,给出了一种新的、针对有约束条件的工程优化问题的 PSO-ABC 算法。PSO-ABC 算法运用基于可行性规则的约束处理技术,充分利用 PSO 算法和 ABC 优化算法各自的特点,互补长短。通过求解四个典型的有约束的工程设计优化问题,证明了本文 PSO-ABC 算法的有效性,且该算法操作简单、精度更高,稳健性能也有较大改善。关于算法中相关参数设置的详细统计分析是下一步需要解决的问题。

(下转第 1266 页)

果中，路段 $l_{1,2}$（环城西路的环北路口—体育场路口路段）的畅通、正常、拥挤，拥堵的交通状态平均行程时间分别为 21 s、37 s、68 s、92 s，而路段 $l_{2,3}$（环城西路的体育场路口—凤起路口路段）的相同交通状态聚类中心分别为 64 s、99 s、192 s、243 s，即使在同一条主干道相邻路段的相同交通状态的聚类中心也存在较大的差异，因此若采用全局区域内同一的交通判断原型，可能会对某些空间单元的交通状态判别准确性有影响。交通管理部门发布的该区域内实际交通状态信息与其他方法进行的比对分析如表 3 所示。

表 2 区域道路交通状态分层结果

交通状态类型	正常	拥挤
点状	⑪	-
线状	③—④ ⑥—⑦ ⑩	④—⑨—⑪ ①—⑥ ②—⑤
面状	⑤—⑧ ⑨	③—④ ⑧—⑩ ⑪

表 3 路网单元交通状态结果误差分析

项目	统一标准	FCM	FCM
判别参数	定量分析参数	交通流参数	定量分析参数
误判个数	5	4	3
误判率/%	19.23	15.38	11.54

最后由表 2 可知，同一时刻，城市区域内各交叉口及路段也非呈现一致交通状态，各交通状态下的空间分布（包括点状、线状、面状）具有独立分布特性。

5 结束语

本文以平均延误及行程时间为交通状态参数对路网空间单元的交通状态进行定量分析，并利用其历史数据结合周相似性采用 FCM 分析得出其交通状态判断原型，在此基础上实现了城市路网空间单元实时交通状态判别。实例分析表明，该计算方法能在一定程度上降低交通状态误判率。在此基础上，通过建立区域路网模型，从空间尺度上分析判断各种交通状态的实时空间分布结果，为交通管理者和参与者提供了更为直观、细致的交通信息。基于 FCM 道路交通状态时空分层判别方法，实现了对城市道路交通状态从定量到定性的分析，并把历史数据与实时数据相结合，提高了判断的准确率。该模型涉及微观、中观、宏观多层次的路网交通状态空间分布，能直观地反映出城市交通流的时空变化特性，可为城市路网交通控制与交通诱导等实际应用提供理论依据与决策支持。

参考文献：

[1] MONTAZERI-GH M, FOTOUHI A. Traffic condition recognition using the K-means clustering method [J]. Scientia Iranica B, 2011, 18(4): 930-937.

[2] 杨祖元，徐姣，罗兵，等. 基于 SFLA-FCM 聚类的城市交通状态判别研究[J]. 计算机应用研究，2010，27(5)：1743-1745.

[3] 董红召，吴方国. 基于 FCM 的路段平均行程时间估计[J]. 科技通报，2011，27(3)：426-431.

[4] 张和生，张毅，胡东成，等. 区域交通状态分析的时空分层模型[J]. 清华大学学报：自然科学版，2007，47(1)：157-160.

[5] 陈红，章渺，赵禹乔. 区域高等级公路网交通状态识别与分析模型研究[J]. 公路，2011，2(2)：100-106.

[6] 张和生，张毅，胡东成. 一种区域交通状态定量分析方法[J]. 吉林大学学报：工学版，2009，39(2)：336-342.

[7] 美国交通研究委员会. 道路通行能力手册[M]. 3 版. 任福田，等译. 北京：中国建筑工业出版社，1991.

[8] 姜桂艳，李继伟，张春勤. 城市主干路行程时间估计的 BPR 修正模型[J]. 西南交通大学学报，2010，45(1)：124-129.

（上接第 1233 页）

参考文献：

[1] HU Xiao-hui, EBERHART R. Solving constrained nonlinear optimization problems with particle swarm optimization [C]//Proc of the 6th World Multiconference on Systematics, Cybernetics and Informatics. 2002: 203-206.

[2] SEDLACZEK K, EBERHARD P. Constrained particle swarm optimization of mechanical systems[C]//Proc of the 6th World Congresses of Structural and Multidisciplinary Optimization. 2005: 1-10.

[3] PARSOPOULOS K E, VRAHATIS M N. Unified particle swarm optimization for solving constrained engineering optimization problems [C]//Lecture Notes in Computer Science, vol 3612. Berlin: Springer, 2005: 582-591.

[4] HE Qie, WANG Ling. A hybrid particle swarm optimization with a feasibility-based rule for constrained optimization[J]. Applied Mathematics and Computation, 2007, 186(2): 1407-1422.

[5] 王勇，蔡自兴，周育人，等. 约束优化进化算法研究及其进展[J]. 软件学报，2009，20(1)：2691-2709.

[6] MEZURA-MONTES E, COELLO C A C. A survey of constraint-handing techniques based on evolutionary multi-objective optimization, EVOCINV-04-2006[R]. Mexico: Departamento de Computacion, Evolutionary Computation Group at CINVESTAV, 2006.

[7] DEB K. An efficient constraint handing method for genetic algorithms [J]. Methods for Applied Mechanical Engineering, 2000, 186(2-4): 311-338.

[8] MEZURA-MONTES E, COELLO C A C. A simple multimembered evolution strategy to solve constrained optimization problems [J]. IEEE Trans on Evolutionary Computation, 2005, 9(1): 1-17.

[9] 林丹，李敏强，寇纪淞. 基于遗传算法求解约束优化问题的一种算法[J]. 软件学报，2001，12(4)：628-632.

[10] KENNEDY J, EBERHERT R. Particle swarm optimization [C]//Proc of the 4th IEEE International Conference on Neural Networks. Piscataway: IEEE Press, 1995: 1942-1948.

[11] KARABOGA D. An idea based on honey bee swarm for numerical optimization[R]. Kayseri: Erciyes University, 2005.

[12] BASTURK B, KARABOGA D. An artificial bee colony (ABC) algorithm for numeric function optimization[C]//Proc of IEEE Swarm Intelligence Symposium. 2006: 3-4.

[13] KARABOGA D, BASTURK B, OZTURK C. Artificial bee colony (ABC) optimization algorithm for solving constrained optimization [C]//Advances in Soft Computing: Foundations of Fuzzy Logic and Soft Computing, vol 4529. 2007: 789-798.