基于改进粒子群的双层规划求解算法

赵志刚 王伟倩 黄树运

(广西大学计算机与电子信息学院 南宁 530004)

摘 要 提出一种采用粒子群优化算法求解双层规划模型的算法。首先对粒子群优化算法作了改进,然后用改进后的算法求解双层规划模型,通过两个粒子群优化算法之间的协同迭代,同步优化双层规划的上下层,最终求得双层规划模型的最优解。此算法将求解一般双层规划问题转化为通过两个粒子群优化算法的交互迭代来求解上下两层规划问题。通过对几种典型函数的测试,验证了此算法的有效性。

关键词 粒子群优化算法,双层规划,全局优化,惯性权重,变异算子

中图法分类号 TP18 **文献标识码** A

Bi-level Programming Problem Based on Improved Particle Swarm Algorithm

ZHAO Zhi-gang WANG Wei-qian HUANG Shu-yun

(College of Computer and Electronics Information, Guangxi University, Nanning 530004, China)

Abstract This paper proposed an algorithm which uses particle swarm optimization (PSO) method to solve the bi-level programming problem (BLPP). A PSO algorithm with adaptive mutation is put forward firstly to improve the performance of standard PSO. Then the modified PSO is used to solve the bi-level programming model. In the proposed algorithm, the interactive iteration between the two PSO optimizes synchronously the two levels of BLPP, and finally obtaining its optimal solution. The experimental results show that the new algorithm can be used to solve the general BLPP.

Keywords Particle swarm optimization algorithm, Bi-level programming problem, Global optimization, Inertia weight, Mutation operator

双层规划(Bi-level Programming Problem,简称 BLPP)是一类具有二层递阶关系的系统优化问题。在研究非平衡经济市场竞争时,双层规划被首先提出。1973 年,Bracken 和 McGill[1]在其文章中提出了双层规划的数学模型。1977 年,在 Candler 和 Norton 的科学报告[2]中正式出现了双层规划和多层规划这些名词。

双层规划的一般模型描述如下:

$$P_1\ \min_{x} F(x,y) \tag{1}$$

$$\text{s.t. } G(x,y)\leqslant 0 \tag{2}$$

其中,$y=y(x)$,是下式优化问题的最优解。

$$P_2\ \min_{y} f(x,y) \tag{3}$$

$$\text{s.t. } g(x,y)\leqslant 0 \tag{4}$$

双层规划模型由两个相互关联的子模型 P_1 和 P_2 组成,其中,P_1 为双层规划的上层,P_2 为双层规划的下层,F 为上层规划的目标函数,x 为上层规划的决策变量,G 是对变量 x 的约束条件;f 为下层规划的目标函数,y 为下层规划的决策变量,g 是对变量 y 的约束条件。上层决策者优先做出决策,通过控制 x 的值影响下层决策者,下层决策者在上层的决策信息下根据自己的利益做出反应,并将自己的最佳反应通过 y 反馈给上层,所以下层决策变量 y 是上层决策变量 x 的函数,即 $y=y(x)$,此函数一般称为反应函数。

在日常生活中,大多数问题可以看成是双层规划问题,如中央和地方的关系、总公司和分公司的关系、供销关系等。但是一般来说,求解双层规划问题是非常困难的。文献[3]首先指出线性双层规划本身是一个 NP-hard 问题。即使是简单的线性双层规划问题也很难求解,更不用说复杂的非线性双层规划问题。在双层规划上、下层中的目标函数和约束函数都是线性的情况下,它也可能是一个非凸问题,并且是非处处可微的。因此,双层规划本身的复杂性和非凸性是造成其求解困难的重要原因。

目前双层规划的求解方法大致可以分为精确算法、启发式算法和群智能算法。大多数精确算法都只是针对具有良好问题特性的双层规划问题,比如文献[4]采用极值点法求解只有一个下层约束条件、没有上层约束条件的 BLPP;文献[5]采用分支定界法构造有效下界求解 BLPP;文献[6]将求解带约束非线性优化问题转换为无约束优化问题并用信赖域方法进行求解;文献[7]将 BLPP 转化为带非凸约束条件的凸规划问题,然后采用分支定界法求解该问题;文献[8]针对目标函数系数和约束条件系数均为区间数的线性双层规划问题,提出了区间线性双层规划的最好最优解和最好最优值的定义,提出了 K 次最好法来求解最好最优解,并分析了下层目标函数的系数的变动对最好最优解的影响,数值例子验证了该方

本文受国家自然科学基金项目(61063031),广西教育厅科研项目(201106LX035)资助。

赵志刚(1973—),男,博士,副教授,主要研究方向为智能优化理论与算法,E-mail:zzgmail2002@163.com;**王伟倩**(1987—),女,硕士生,主要研究方向为智能系统与智能 CAD;**黄树运**(1985—),男,硕士生,主要研究方向为软件工程。

法的有效性和可行性，等。目前，最通用的精确算法都是基于库恩-塔克(Kuhn-Tucker，K-T)条件的。这些方法中，用 K-T 条件代替下层规划，将 BLPP 转化为一个带有附加条件的单层规划问题，然后再用其它的算法求解该问题。但是 K-T 条件的推导都必须基于目标函数或约束函数的可微性或者凸性等条件，因此这种类型的算法很难运用到一般的实际应用问题中，特别是带有不可微目标函数或者非凸搜索空间的复杂 BLPP。

许多文献在应用启发式算法(如遗传算法、进化算法等)求解 BLPP 方面作了研究，如求解线性 BLPP 的遗传算法[9]，求解非线性 BLPP 的进化算法[10]、遗传算法[11]，等等。文献[9]提出一种基于遗传算法求解线性 BLPP 的方法，即用遗传算法求解上层规划，对所产生的每个上层决策变量的值，利用单纯形法求解相应的下层规划。文献[10]将特定的非线性 BLPP 转化为等价的非线性单层规划问题，提出了求解该问题的进化算法，该进化算法可以求解上层目标函数不可微的非线性 BLPP。文献[11]针对上层函数及其约束函数不要求具有凸性和可微性、下层决策变量是凸二次规划的双层规划模型，提出了相应的遗传算法求解过程。启发式算法由于问题的复杂性及算法自身存在的制约因素，在求解复杂 BLPP 方面还存在不足，主要表现在：大多是针对特定的 BLPP，难以推广到一般情况；某些算法需要对原 BLPP 进行转化；算法本身结构较复杂，算法求解效率受到影响。

近几年，作为群智能优化算法的主要代表，粒子群优化(PSO)算法正日益受到广泛关注，已被应用于求解 BLPP[12-15]。文献[12]首先对标准 PSO 算法作了改进，进而采用改进 PSO 算法求解 BLPP，提出了基于改进 PSO 的双层规划模型的通用求解算法。文献[13]借助分层迭代的思想，提出了基于 PSO 的双层迭代算法，它能有效求解线性和非线性 BLPP。文献[14]通过两个 PSO 算法的变形算法来模拟 BLPP 的序列决策的过程，构建了一个基于 PSO 算法的层次算法框架来求解一般的 BLPP。文献[15]利用 PSO 算法求解线性 BLPP，获得了比遗传算法更好的计算结果。PSO 算法在求解 BLPP 时，相对于以往求解算法的优点是不需要目标函数的可微性、梯度信息和搜索空间的凸性等条件，也不需要对原 BLPP 进行转化。当前，采用 PSO 算法或基于 PSO 算法的优化方法求解 BLPP 已逐步发展成为一个值得深入研究的方向。

本文在分析和研究现有的一些优秀的算法思想的基础上，提出采用改进的 PSO 算法求解 BLPP。实验结果表明，本文提出的算法不仅能够有效地求解双层规划模型，获得高质量的全局最优解，而且该算法具有通用性和普遍性，不依懒于具体的双层规划模型，也不是基于特定的假设条件。

1 粒子群优化算法

1.1 基本粒子群优化算法

粒子群优化(Particle Swarm Optimization，PSO)算法是由美国社会心理学家 James Kennedy 和电气工程师 Russell Eberhart 于 1995 年提出的一种演化计算技术[16]。它源于对鸟群和鱼群群体觅食行为的模拟，是一种全局优化算法，最初用于解决连续优化问题，目前已可以用于处理大量非线性、不可微和多峰值等复杂问题优化，并已广泛用于函数优化、神经网络训练、模式识别和模糊系统控制等诸多领域。传统的 PSO 算法在处理复杂函数时，进化后期收敛速度明显变慢，同时，算法收敛精度不高，易陷入局部最优。针对这些问题，研究者们从不同的角度对基本的 PSO 算法进行了改进。文献[17]提出了基于群体适应度方差的自适应变异的粒子群优化算法，该算法根据群体适应度方差以及当前最优解的大小来确定当前最佳粒子的变异概率，增强了粒子群优化算法跳出局部最优解的能力。文献[18]提出了一种混合变异算子的自适应粒子群优化算法，通过提高局部精度搜索能力和破坏群体个体的单一性两种策略改进算法中的未成熟收敛现象，提高了算法的收敛速度和收敛精度。

PSO 算法初始化为一群随机粒子(随机解)，所有的粒子都有一个由被优化函数决定的适应值，以评价粒子当前位置的优劣，每个粒子还有一个速度来决定它们飞行的方向和距离。然后粒子们就追随当前的最优粒子在解空间中搜索，即通过迭代找到最优解。在每一次迭代中，粒子通过跟踪两个最优解来更新自己，一是粒子本身找到的最优解，即个体极值 *pBest*；二是整个种群目前找到的最优解，即全局最优解 *gBest*，粒子根据如下的公式来更新自己的速度和位置。

$$v(t+1)=\omega v(t)+c_1 r_1(pBest(t)-x(t))+c_2 r_2(gBest(t)-x(t)) \tag{5}$$

$$x(t+1)=x(t)+v(t+1) \tag{6}$$

式中，t 为当前的迭代次数；c_1、c_2 为学习因子，一般取值为 2；r_1、r_2 为 0 到 1 之间均匀分布的随机数；ω 为惯性系数，一般在 0.1 到 0.9 之间取值。文献[19]通过实验表明，若 ω 随算法迭代的进行而逐渐减小，算法的收敛性能显著提高。设 ωMax、ωMin 分别为最大和最小惯性系数；$iter$、$iterMax$ 分别为当前迭代次数和最大迭代次数，则有：

$$\omega=\omega\mathrm{Max}-(\omega\mathrm{Max}-\omega\mathrm{Min})\times iter/iter\mathrm{Max} \tag{7}$$

迭代过程中，粒子的位置和速度不断被更新，同时，*pBest* 与 *gBest* 也在不断更新，最后输出的 *gBest* 就是算法得到的最优解。迭代终止的条件一般为算法达到最大迭代次数或搜索到满足精度要求的最优解。

1.2 带自适应变异的粒子群优化算法

PSO 算法在运行过程中，当某个粒子的适应值大大超过当前粒子群平均个体适应值时，其他粒子将迅速向其靠拢，如果该粒子的位置是一个局部最优点而不是全局最优点，该粒子群将无法在解空间中进行重新搜索，算法陷入局部最优，出现了所谓的早熟收敛现象。实验证明，粒子群优化算法无论是全局收敛还是陷入局部收敛，都会出现粒子"聚集"的现象，即粒子聚集在搜索空间中的某一个或某几个特定的位置，这主要和问题本身的特性以及适应度函数的选择有关。为了判断粒子群优化算法是否陷入局部收敛，现在分别给出群体适应度方差和粒子聚集距离的定义。

定义 1[17] 设粒子群的粒子数为 n，f_i 为第 i 个粒子的适应度，f_{avg} 为粒子群目前平均的适应度，σ^2 为粒子群群体适应度方差，则 σ^2 可以定义为：

$$\sigma^2=\sum_{i=1}^{n}\left(\frac{f_i-f_{avg}}{f}\right)^2 \tag{8}$$

式中，f 为归一化定标因子，其作用是限制 σ^2 的大小，f 的取值如下：

$$f=\max\{1,\max\{|f_i-f_{avg}|\}\},i\in[1,n] \tag{9}$$

定义 1 表明：群体适应度方差 σ^2 反映的是粒子群中所有粒子的"收敛"程度。σ^2 越小，则粒子群越趋于收敛；反之，粒子群处于随机搜索状态。

文献[2]进一步证明，粒子群优化算法无论是全局收敛还是陷入局部收敛，粒子群中的粒子将聚集在搜索空间中的某一个或某几个特定的位置，且群体适应度方差 σ^2 等于零。但仅仅凭借适应度方差是否为零，并不能区分算法是早熟收敛还是全局收敛，还需要进一步判断算法此时得到的最优解是否为理论全局最优解或期望的最优解。

定义 2 粒子平均聚集距离[20]和聚集程度阈值[21]的定义分别为：

$$MeanDist=\frac{\sum_{i=1}^{n}\sqrt{\sum_{d=1}^{D}(p_{td}-x_{id})^2}}{n} \tag{10}$$

$$BorderDist=\sqrt{D(xMax)^2}/100 \tag{11}$$

式中，n 为粒子群的粒子数，D 为粒子的维数，p_{td} 为粒子群目前搜索到的最优位置，x_{id} 为每个粒子目前搜索到的最优位置，$xMax$ 为 x 的取值上限。$BorderDist$ 的表达式是根据经验定义的。

PSO 算法在运行过程中，当粒子的适应度方差趋于零且粒子的平均聚集距离大于聚集程度的阈值时，认为粒子达到全局收敛；当粒子的适应度方差趋于零且粒子的平均聚集距离小于聚集程度的阈值时，认为粒子陷入局部收敛，此时全局极值 $gBest$ 一定是局部最优解。结合式(5)，此时如果改变全局极值 $gBest$，就可以改变粒子的前进方向，摆脱局部极值点，从而让粒子进入搜索空间的其他区域去搜索，将有望找到全局最优解。

本文对 $gBest$ 依概率 P 进行变异，P 的计算公式如下所示：

$$P=\begin{cases}\beta, & \sigma^2<ld \text{ and } MeanDist<BorderDist\\ 0, & \text{others}\end{cases} \tag{12}$$

式中，ld 的取值与实际问题有关，一般远小于 σ^2。β 可取 [0.1,0.3]之间的任意值。

对 $gBest$ 进行变异操作，即给 $gBest$ 增加随机扰动。设 η 是服从标准正态分布的随机变量，即 $\eta\sim N(0,1)$，则有

$$gBest=gBest\times(1+\eta\times 0.5) \tag{13}$$

通过对 $gBest$ 进行随机变异来提高 PSO 算法跳出局部最优解的能力。

2 求解双层规划模型的算法描述

BLPP 已经被证明是 NP-hard 问题，尤其对于复杂的非线性 BLPP，其求解更为困难。针对复杂的 BLPP，研究基于 PSO 算法的通用的高效求解算法是非常重要的，也是可行的，但目前国内外这方面的研究还不充分。尽管目前没有一种算法能够求得双层规划模型的精确最优解，但也存在一些较好的算法。文献[13]提出采用 PSO 算法并借助分层迭代的思想来求解双层规划模型。即对上层规划采用 PSO 算法、下层规划采用传统优化方法进行求解，上层和下层之间反复迭代，求得最优解。文献[14]提出求解一般双层规划问题的层次粒子群算法，该算法将求解一般双层规划问题转化为通过两个变形粒子群算法的交互迭代来求解上下层规划问题。

在分析和对比现有方法的基础上，综合上节对 PSO 算法的改进，提出基于改进 PSO 的双层规划求解算法（BLPP based on Improved PSO，IPSO-BLPP）。即双层规划的上下两层均用改进后的 PSO 算法来实现，双层规划的上、下层问题都有各自的决策变量、目标函数和约束条件。上层决策者首先做出决策，先给定一个决策变量 x，下层以这个决策变量 x 为参量，根据自己的目标函数和约束条件，在可能的范围内求得一个最优解，并将自己的最优解反馈给上层，上层再在下层的最优解的基础上，在可能的范围内求得整体上的最优解，然后在上层和下层之间交互迭代，同步优化双层规划的上下层，最终求得双层规划模型的近似全局最优解。在 BLPP 中上层规划处于"主导"地位，上层规划的解的优劣决定着 BLPP 的解的品质，因此，下层决策者在优化自己的目标而选择决策时，不能违背上层的决策。

IPSO-BLPP 算法的基本流程如下：

Step1 初始化 PSO 算法中的参数；随机产生下层模型满足约束条件的初始解；随机初始化种群中粒子的位置 X_i 和速度 V_i，$i\in[1,n]$，n 为粒子的个数。

Step2 将第 i 个粒子的 $pBest$ 设置为该粒子的当前位置，$gBest$ 设置为初始种群中最佳粒子的位置。

Step3 对粒子群中的所有粒子执行如下操作：

①根据式(5)—式(7)更新粒子的位置和速度；

②将上层模型的解即粒子 i 的位置 X_i 代入下层模型，利用 PSO 算法求得下层模型的最优解 Y_i；

③将(X_i,Y_i)代入上层规划的目标函数，计算粒子 i 的适应度 $F(X_i,Y_i)$，$i\in[1,n]$；

④如果粒子 i 的适应度优于其 $pBest$ 的适应度，则该粒子的 $pBest$ 更新为当前位置 X_i；对应于 $pBest$ 的下层模型的最优解 $ypBest$ 相应地更新为 Y_i；

⑤如果粒子 i 的适应度优于当前 $gBest$ 的适应度，则 $gBest$ 更新为当前位置 X_i；对应于 $gBest$ 的下层模型的最优解 $ygBest$ 相应地更新为 Y_i。

Step4 判断算法是否达到最大迭代次数或搜索到满足精度要求的最优解，若是，转 Step6；否则转 Step5。

Step5 根据式(13)更新 $gBest$，利用 PSO 算法求出对应于 $gBest$ 的下层模型的最优解 $ygBest$，转 Step3。

Step6 输出双层规划模型的最优解 $gBest$ 和 $ygBest$，并相应求出上、下层规划目标函数值，算法运行结束。

3 数值实验

通过下面 10 个算例验证本文提出的算法的有效性，并与相关文献中的结果进行比较。IPSO-BLPP 算法中的参数设置如下：群体规模 $n=50$，维数 $D=30$，学习因子 $c_1=c_2=2$，ωMax 和 ωMin 分别为 0.9 和 0.4，收敛精度 $lamda=10^{-10}$，最大迭代次数 $iterMax=50$。算法在 MATLAB7.8.0 上实现。

例 1[13]

$$\min_x F(x,y)=-x_1^2-3x_2-4y_1+y_2^2$$

$$\text{s.t. } (x_1)^2+2x_2\leqslant 4,\ x_1\geqslant 0,\ x_2\geqslant 0$$

$$\min_y f(x,y)=2x_1^2+y_1^2-5y_2$$

$$\text{s.t. } x_1^2-2x_1+x_2^2-2y_1+y_2\geqslant -3$$

$$x_2+3y_1-4y_2\geqslant 4,\ y_1\geqslant 0,\ y_2\geqslant 0$$

例 2[13]

$$\min_x F(x,y)=2x_1+2x_2-3y_1-3y_2-60$$

$$\text{s. t. } x_1+x_2+y_1-2y_2\leqslant 40$$

$$0\leqslant x_1\leqslant 50, 0\leqslant x_2\leqslant 50$$

$$\min_y f(x,y)=(y_1-x_1+20)^2+(y_2-x_2+20)^2$$

$$\text{s. t. } 2y_1-x_1+10\leqslant 0, 2y_2-x_2+10\leqslant 0$$

$$-10\leqslant y_1\leqslant 20, -10\leqslant y_1\leqslant 20$$

例 3[22]

$$\min_x F(x,y)=(x_1-30)^2+(x_2-20)^2-20y_1+20y_2$$

$$\text{s. t. } -x_1-2x_2+30\leqslant 0, x_1+x_2-25\leqslant 0, x_2\leqslant 15$$

$$\min_y f(x,y)=(x_1-y_1)^2+(x_2-y_2)^2$$

$$\text{s. t. } 0\leqslant y_1\leqslant 10, 0\leqslant y_2\leqslant 10$$

例 4[13]

$$\min_x F(x,y)=k(x_1^2+x_2^2)-3y_1-4y_2+0.5(y_1^2+y_2^2)$$

$$\min_y f(x,y)=0.5\begin{bmatrix}y_1 & y_2\end{bmatrix}H\begin{bmatrix}y_1\\ y_2\end{bmatrix}-b(x)^{\mathrm{T}}\begin{bmatrix}y_1\\ y_2\end{bmatrix}$$

$$\text{s. t. } -0.333y_1+y_2-2\leqslant 0, y_1-0.333y_2-2\leqslant 0$$

$$y_1\geqslant 0, y_2\geqslant 0$$

a) $k=0.1, H=\begin{bmatrix}1 & -2\\ -2 & 5\end{bmatrix}, b(x)=\begin{bmatrix}x_1\\ x_2\end{bmatrix}$;

b) $k=1, H, b(x)$ 同 a)；

c) $k=0, H=\begin{bmatrix}1 & 3\\ 3 & 10\end{bmatrix}, b(x)=\begin{bmatrix}x_1\\ x_2\end{bmatrix}$;

d) $k=0.1, H, b(x)$ 同 c)；

例 5[14]

$$\min_x F(x,y)=x^2+(y-10)^2$$

$$\text{s. t. } x+2y-6\leqslant 0, -x\leqslant 0$$

$$\min_y f(x,y)=x^3-2y^3+x-2y-x^2$$

$$\text{s. t. } -x+2y-3\leqslant 0, -y\leqslant 0$$

例 6[14]

$$\min_x F(x,y)=(x-5)^2+(2y+1)^2$$

$$\text{s. t. } -x\leqslant 0$$

$$\min_y f(x,y)=(x-1)^2-1.5xy+x^3$$

$$\text{s. t. } -3x+y+3\leqslant 0, x-0.5y-4\leqslant 0$$

$$x+y-7\leqslant 0, -y\leqslant 0$$

例 7[14]：

$$\min_x F(x,y)=(x-5)^4+(2y+1)^4$$

$$\text{s. t. } x+y-4\leqslant 0, -x\leqslant 0$$

$$\min_y f(x,y)=e^{-x+y}+x^2+2xy+y^2+2x+6y$$

$$\text{s. t. } -x+y-2\leqslant 0, -y\leqslant 0$$

3.1 IPSO-BLPP 算法的计算结果

算法 IPSO-BLPP 对以上 10 个算例进行了优化计算，对每个算例独立运行 30 次，取最好解为近似全局最优解，实验结果如表 1 所列。对于每一个算例用如下指标进行统计：上层规划问题的目标函数的最优值(用“Best_U”表示)、最差值(用“Worst_U”表示)、平均值(用“Avg.”表示)，30 次实验的最优值的标准差(用“Std.”表示)，同时也给出了与上层规划问题目标函数最优值、最差值所对应的下层规划问题的目标函数值，分别表示为“Best_L”、“Worst_L”。

从表 1 中的结果可以看出，对于大部分算例来说，30 次试验的最优值的标准差都非常小，甚至有 4 个算例的标准差达到了 0，而仅有算例 3 的标准差较大，这表明，本文提出的算法鲁棒性比较高，对于本文中的 10 个算例，都能够找到比较好的解。

表 1 算法 HPSO-BLPP 实验结果

算例	上层目标函数值				下层目标函数值	
	Best_U	Worst_U	Avg.	Std.	Best_L	Worst_L
1	−14	−14	−14	0	−4	−4
2	−30	−30	−30	0	450	450
3	114.0106	117.1384	114.5700	0.9703	111.6889	110.7136
4a	−11.9985	−11.9913	−11.9981	0.0014	8.9859	8.9916
4b	−11.9985	−11.9932	−11.9979	0.0015	8.9808	8.9808
4c	−11.9985	−11.9985	−11.9985	0	−147.3320	−76.8135
4d	−11.9985	−11.9976	−11.9984	0.0002	76.3462	76.0246
5	62.3958	62.7152	62.4901	0.0995	−24.4604	−23.7856
6	2	2.0034	2.0010	0.0011	73	72.9082
7	2	2	2	0	24.0183	24.0183

3.2 与其他算法的比较结果

本节对算法 IPSO-BLPP 的结果与相关文献中的进行了比较，表 2 给出了比较结果。

从表 2 中可以看出，对每一个算例，算法 IPSO-BLPP 发现的近似全局最优解均优于其他算法。特别是对于除算例 4c、7 以外的其他算例，算法 IPSO-BLPP 都发现了比较好的解，远远优于文献中的结果。

算例 1 和 2，算法均能取得较好的解，上下层规划的目标函数值明显优于文献中的结果，并且算法运行 30 次，标准差均为 0，结果非常稳定。

对于算例 3，算法 IPSO-BLPP 发现了一个比较好的解(17.7267，7.2467，10，0.0365)，其对应的上层规划目标函数值是 114.0106，与文献中的函数值 225 相比，有较大的改进。

对于算例 4a 和 4b，算法 IPSO-BLPP 能够发现很多可行解，这些可行解具有相同的上层规划目标函数值－11.9985，但是下层规划目标函数值却不同，这是因为上层规划目标函数值取决于下层规划的决策变量 y，而与上层规划的决策变量 x 没有直接关系，它只会影响下层规划目标函数值。这些可行解具有相同的下层决策变量值 $y=(2.9985, 2.9985)$，而 x 具有不同的值。

对于算例 4d，算法发现了一个较好的解，上层规划目标函数值－11.9985 优于文献中的－3.6，对应的解为(0.0137，0.0121，2.9985，2.9985)。

对于算例 5，算法同样发现了一个较好的解(1.4851，2.

2418)，对应的上下两层目标函数值分别为 62.3958、-24.4604，均优于文献中的结果。

对于算例 6，算法 IPSO-BLPP 改变了文献[11]中算法所得出的上层目标函数的最优值 15.4400，本文得到的最优值为 2，对应的最优解为(4,0)。

综上所述，表 1 和表 2 中的数据结果表明，对于大多数测试函数，算法 IPSO-BLPP 都发现了比其他算法好的最优解，由此可见，本文中提出的 IPSO-BLPP 算法对于求解双层规划模型是非常有效的。

表 2 算法 IPSO-BLPP 与相关文献的数值结果比较

算例	上层目标函数最优值		下层目标函数最优值		最优解(x,y)	
	IPSO-BLPP	文献	IPSO-BLPP	文献	IPSO-BLPP	文献
1	−14	−12.6878	4	−1.01563	(0,2,2,0)	(0,2,1.875,0.90625)
2	−30	0	450	100	(0,0,−5,−5)	(0,30,−10,10)
3	114.0106	225	111.6889	100	(17.7267,7.2467,10,0.0365)	(20,5,10,5)
4a	−11.9985	−8.9172	8.9859	−6.157	(0.0017,0,2.9985,2.9985)	(1.03867,3.09868,2.5973,1.7938)
4b	−11.9985	−7.578458	8.9808	−0.57192	(0.0034,0,2.9985,2.9985)	(0.27878,0.47498,2.34383,1.03253)
4c	−11.9985	−11.9985	−147.3320	−178.07	(10.5727,64.0498,2.9985,2.9985)	(14.9086,69.9642,2.9985,2.9985)
4d	−11.9985	−3.6	76.3462	−2	(0.0137,0.0121,2.9985,2.9985)	(2,0,2,0)
5	62.3958	88.7757	−24.4604	−0.7698	(1.4851,2.2418)	*
6	2	15.4400	73	2.7280	(4,0)	*
7	2	2	24.0183	24.0190	(4,0)	*

*表示原文献中未提供结果

结束语 本文通过两个 PSO 算法之间的交互迭代，反映了双层规划问题的决策过程。提出的 IPSO-BLPP 算法不仅可以求解一般的双层规划模型，还可以获得高质量的近似全局最优解。本算法不需要借助任何的假设条件，也不需要对目标函数和约束条件进行转换处理，不依懒于具体的双层规划模型，是一种通用的求解双层规划模型的算法。

参考文献

[1] Bracken J, McGill J. Mathematical programs with optimization problems in the constraints [J]. Operation Research, 1973, 21: 37-44

[2] Candler W, Norton R. Multilevel programming [R]. Technical Report 20. World Bank Development Research Center, Washington D. C, 1977

[3] Jeroslow R. The polynomial hierarchy and a simple model for competitive analysis [J]. Mathematical Programming, 1985, 32: 146-164

[4] Candler W, Townsley R. A linear two-level programming problem[J]. Computers & Operations Research, 1982, 9(1): 59-76

[5] 安玉伟，严洪森. 一类两阶段生产系统生产计划与调度的集成优化[J]. 计算机集成制造系统，2012，18(4)：796-806

[6] 孙聪. 求解非线性双层规划的若干方法[D]. 长春：吉林大学，2010

[7] Muu L D, Quy N V. A global optimization method for solving convex quadratic bilevel programming problems[J]. Journal of Global Optimization, 2003, 26(2): 199-219

[8] 王建忠，杜纲. 区间线性双层规划的最好最优解[J]. 系统工程，2009，27(4)：100-103

[9] Mathieu R, Pittard L, Anandalingam G. Genetic algorithm based approach to bi-level linear programming [J]. Operations Research, 1994, 28(1): 1-21

[10] Wang Y, Jiao Y, Li H. An evolutionary algorithm for solving nonlinear bilevel programming based on a new constraint-handling scheme[J]. IEEE Transactions on Systems, Man, and Cybernetics, Part C, 2005, 35(2): 221-232

[11] 常永明，王宇平. 求解一类特殊的双层规划问题的遗传算法[J]. 计算机工程与应用，2009，43(3)：45-46

[12] Zhao Zhi-gang, Gu Xin-yi. Particle Swarm Optimization Based Algorithm for Bilevel Programming Problems[J]. Proceedings of the IEEE International Conference on Intelligent Systems Design and Applications. 2006: 951-956

[13] 赵志刚，顾新一，李陶深. 求双层规划模型的粒子群优化算法[J]. 系统工程理论与实践，2007，27(8)：92-98

[14] 李相勇，田澎. 双层规划问题的粒子群算法研究[J]. 管理科学学报，2008，11(5)：41-52

[15] Kuo R, Huang C. Application of particle swarm optimization algorithm for solving bi-level linear programming problem[J]. Computers and Mathematics with Applications, 2009, 58(4): 678-685

[16] Kennedy J, Eberhart R. Particle swarm optimization[C]//Proc. IEEE Int'l. Conf. on Neural Networks, IV. Piscataway, NJ: IEEE Service Center, 1995: 1942-1948

[17] 吕振肃，侯志荣. 自适应变异的粒子群优化算法[J]. 电子学报，2004，32(3)：416-420

[18] 安晓会，高岳林. 混合变异算子的自适应粒子群优化算法[J]. 计算机应用，2008，28：28-30

[19] Shi Y, Eberhart R. A Modified Particle Swarm Optimizer[C]//Proc. IEEE Int'l. Conf. on Evolutionary Computation. Piscataway, NJ: IEEE Press, 1998: 69-73

[20] Falk J, Soland R. An algorithm for separable nonconvex programming problems [J]. Management Science, 1969, 15(9): 550-569

[21] 李宁，刘飞，孙德宝. 基于带变异算子粒子群优化算法的约束布局优化研究[J]. 计算机学报，2004，27(7)：897-903

[22] 赵志刚，苏一丹. 基于粒子群优化算法求解双层规划模型[C]//第八届中国青年运筹信息管理学者大会论文集. 桂林，2006：525-533