

求解二层规划的混合微粒群算法

熊 鹰, 周树民, 祁 辉

(武汉理工大学 理学院, 湖北 武汉 430070)

摘 要: 对于二层规划问题有许多经典的求解方法, 如极点搜索法、分支定界法和罚函数法等。文中给出了基于微粒群算法的二层规划的一种新的求解方法。提出了分别先用单纯形法和内部映射牛顿法的子空间置信域法求解下层规划, 然后用微粒群算法求解上层规划的求解方法, 这两种混合微粒群算法分别用于求解线性二层规划和非线性二层规划。并结合实例的对比分析, 说明了这两种混合微粒群算法求解二层规划的可行性和有效性。

关键词: 二层规划; 微粒群算法; 约束规划

中图分类号: TP301.6

文献标识码: A

文章编号: 1673-629X(2007)04-0229-03

Hybrid Particle Swarm Optimization for Bilevel Programming

XIONG Ying, ZHOU Shu-min, QI Hui

(School of Sciences, Wuhan University of Technology, Wuhan 430070, China)

Abstract: There are many classical solution methods for bilevel programming such as culmination searching algorithm, branch and bound algorithm and penalty function algorithm. Put forward a new solution algorithm for bilevel programming based on particle swarm optimization. It resolves the lower - programming by simplex method and subspace trust region method based on the interior - reflective Newton method, and resolve the upper - programming by particle swarm optimization. The two hybrid particle swarm optimizations resolve the linear bilevel programming and non - linear bilevel programming independently. With many examples, it is feasible that two hybrid particle swarm optimizations resolve bilevel programming.

Key words: bilevel programming; particle swarm optimization; constrained programming

1 二层规划

二层规划问题是二层决策问题的数学模型, 它是一种具有二层递阶结构的系统优化模型。上层规划和下层规划都有各自的目标函数和约束条件。上层规划的目标函数和约束条件不仅与上层决策变量有关, 而且还依赖于下层规划的最优解或最优值, 而下层规划的最优解或最优值又受到上层决策变量的影响。如果上层规划和下层规划的目标函数和约束条件都是线性的, 则称此二层规划为线性二层规划, 其一般模型为:

$$(BLP) \begin{cases} (P_1) \min_x f_1(x, y) = a^T x + b^T y \\ \text{s. t. } Cx \leq s \\ \text{其中 } y \text{ 解} \\ (P_2) \min_y f_2(x, y) = c^T x + d^T y \\ \text{s. t. } Ax + By \leq r \end{cases}$$

这里 $a, c, x \in R^{n_1}; b, d, y \in R^{n_2}; s \in R^{m_1}; r \in R^{m_2}$ 分别表示相应维数的列向量。 $A \in R^{m_2 \times n_1}, B \in R^{m_2 \times n_2}, C \in R^{m_1 \times n_1}$, 总是假定约束条件 $Cx \leq s$ 和 $Ax + By \leq r$ 已经包含了 $x \geq 0$ 和 $y \geq 0$ 这种决策变量的约束, 并且假设给定 x 以后, 问题 (P_2) 仅有唯一最优解。称集合 $S = \{x | Cx \leq s\}$ 和 $S = \{(x, y) | Ax + By \leq r\}$ 分别为 BLP 的上层约束集和下层约束集。

如果上层规划或下层规划的目标函数和约束条件有一个为非线性的, 则此二层规划为非线性二层规划, 其一般模型为:

$$(BNP) \begin{cases} \min_x F(x, y) \\ \text{s. t. } \begin{cases} G(x) \leq 0 \\ \min_y f_j(x, y_j) \\ \text{s. t. } g_j(x, y_j) \leq 0 \quad j = 1, 2, \dots, N \end{cases} \end{cases}$$

这里 $x \in R^n; y_j \in R^{n_j}; f_j: R^n \times R^{n_j} \rightarrow R; g_j: R^n \times R^{n_j} \rightarrow R^{m_j}, j = 1, 2, \dots, N; F, G, f_j, g_j$ 至少有一个是非线性的。下层有 N 个平行的规划, 第 j 个规划有决策变量 n_j 个, 约束条件 m_j 个。

对于二层规划, 即使上、下层规划中的目标函数和

收稿日期: 2006-06-29

作者简介: 熊 鹰 (1981-), 男, 湖北应城人, 硕士研究生, 主要研究方向为系统控制及优化、智能算法; 周树民, 教授, 研究方向为系统控制及优化、运筹学、图论。

约束函数都是线性的,它一般也是一个非凸问题,而且上层的目标函数对于上层决策变量不是处处可微的。二层规划的本质非凸性以及非处处可微性给其数值求解带来极大的困难,特别是求非线性二层规划问题的全局最优解。因此,绝大多数关于求解二层规划问题的数值方法的研究,或者局限于具有某种特殊结构的问题,或者局限于求局部最优解。对于线性二层规划,某些文献给出了推广的单纯形法,但由于二层规划的非凸性致使它是一个 NP 困难问题,文献[1]提出了一种基于遗传算法求线性二层规划问题全局最优解的方法,遗传算法用于求解上层规划,对于遗传算法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。但文献[1]只考虑了上层规划没有约束的情况。对于非线性二层规划,文献[2]提出了一种网格搜索方法,它是建立在非线性二层规划问题的两种不同形式的最优性条件的基础上,需要求解一系列非凸的非线性规划问题。文中提出了两种混合微粒群算法,对于线性二层规划,上层约束规划用微粒群算法实现,下层用单纯形法实现;对于非线性二层规划,上层约束规划用微粒群算法实现,下层用内部映射牛顿法的子空间置信域法(有关这种算法,请参考文献[3, 4])。

2 二层规划的微粒群算法

二层规划模型的决策思想是对上层约束集中给定的 $x \in X$, 下层规划以 x 为参数, 求解函数 f_i 的极小值, 反馈到上层, 上层根据下层的反馈情况, 调整 x , 直到找到最优解。为了使问题有意义, 假设对于满足上层约束集的 x , 下层规划总存在最优值。二层规划的微粒群算法实现过程: 首先, 在上层约束集内随机产生 m 个初始微粒 $x_i (i = 1, 2, \dots, m)$ 作为上层规划的初始微粒群, 同时产生 m 个初始速度 $v_i (i = 1, 2, \dots, m)$ 。初始微粒 x_i 和初始速度 v_i 均有 D 个分量。可以用如下方法产生满足上层约束集的微粒: 找出一个满足上层约束集的内点 x_0 , 定义一个足够大的半径 r , 以保证所产生的微粒遍历整个上层约束集, 随机产生一个方向 d , d 的每个分量都是 $[-1, 1]$ 上的均匀分布随机数, 如果 $x = x_0 + dr$ 在上层约束集内, 则 x 可作为一个微粒, 否则置 r 为 0 到 r 之间的一个随机数, 直到 $x = x_0 + dr$ 在上层约束集内, 由于 x_0 为内点, 所以以上方法总可以找到上层约束集内的 x , 重复以上操作直到产生 m 个微粒为止。然后, 对微粒群中的每个微粒 $x_i (i = 1, 2, \dots, m)$, 下层规划以 $x_i (i = 1, 2, \dots, m)$ 为参数, 对于线性规划和非线性规划, 分别用单纯形法和内部映射牛顿法的子空间置信域法求解, 通过下层规划

的反馈值, 连同上层规划所给的微粒 $x_i (i = 1, 2, \dots, m)$ 的位置, 计算每个微粒的适应值(即目标函数值), 通过比较适应值, 可以找出每个微粒的当前最好位置 $P_i = (P_{i1}, P_{i2}, \dots, P_{id})$ 和全局最好位置 $P_g = (P_{g1}, P_{g2}, \dots, P_{gd})$, 第一次迭代时, 微粒的当前最好位置即为微粒的初始位置。然后, 用公式(1)、(2)带收缩因子的微粒群算法更新微粒的速度和位置。

$$v_{id}(y, t+1) = \chi(v_{id}(y, t) + c_1 r_{1d}(y, t)(P_{id}(y, t) - x_{id}(y, t)) + c_2 r_{2d}(y, t)(P_{gd}(y, t) - x_{id}(y, t))) \quad (1)$$

$$x_{id}(y, t+1) = x_{id}(y, t) + v_{id}(y, t+1) \quad (2)$$

其中下标 $d (1 \leq d \leq D)$ 表示微粒的维索引, t 表示微粒群的进化代数, c_1 为自身加速常数, c_2 为全局加速常数, $\chi = 2 / |2 - l - \sqrt{l^2 - 4l}|$, 且 $l = c_1 + c_2, l > 4$, 一般地, $c_1 = c_2 = 2.05$ 。 $r_1 \sim U(0, 1), r_2 \sim U(0, 1)$ 为两个相互独立的随机变量, $v_{id}(y, t) \in [-v_{\max}, v_{\max}]$, $x_{id}(y, t) \in [-x_{\max}, x_{\max}]$, 如果 $v_{id}(y, t)$ 、 $x_{id}(y, t)$ 超出边界值, 就用边界值取代 $v_{id}(y, t)$ 、 $x_{id}(y, t)$, v_{\max} 、 x_{\max} 是依不同的目标函数和不同的搜索空间而不同的常数^[5]。最后, 对每个微粒 i , 将其适应值与其经历过的最好位置 P_i 做比较, 如果较好, 则将其作为当前的最好位置 P_i , 对每个微粒 i , 将其适应值与全局最好位置 P_g 作比较, 如果较好, 则将其作为当前全局最好位置 P_g 。如果 $x_i(y, t+1)$ 不在上层约束集内, 则 $\chi \leftarrow \chi \text{rand}$ (其中 rand 为 0 和 1 之间的均匀分布随机数) 也可以令 $\chi \leftarrow \chi \rho (0 < \rho < 1)$ 重新进化速度和位置, 直到产生可行的位置 $x_i(y, t+1)$ 。如果 $x_{id}(y, t)$ 已在上层约束集的边缘, 则 $v_{id}(x, t+1)$ 几乎为零时, $x_{id}(y, t+1)$ 落在上层约束集内, 而此时 $\chi \leftarrow \chi \text{rand}, \chi \leftarrow \chi \rho (0 < \rho < 1)$ 都能保证 $v_{id}(y, t+1)$ 收敛到零。如果不满足停机条件, 则返回下层规划的求解过程继续下一轮进化。一般停机条件为预设一个最大代数或全局最好位置的适应值连续多少次没有得到改进则停止。

3 实例及分析

实例 1: 线性二层规划模型(价格控制问题)^[6]:

$$\begin{cases} \max_x F = 2x - y \\ \max_y f = x^T y \\ \text{s. t. } -x + 2y \leq 12 \\ \quad x + y \leq 12 \\ \quad x - y \leq 6 \\ \quad x \geq 0, y \geq 0 \end{cases}$$

取上层规划的加速系数 $c_1 = c_2 = 0.205$, 微粒数 $n = 20$, 用带收缩因子的 PSO 迭代预设代数 50 代, 重复 10 次实验, 所得数据见表 1, 平均从第 6.5 代开始即得到最优点 $(x^*, y^*) = (9.0000, 3.0000)$, 此时 $F^* = 15$ 。整个过程平均耗时 142.3s。

表 1 实例 1 做 10 次实验所得结果(s)

收敛代数	5	8	6	7	6
耗时	138	150	148	144	136
收敛代数	6	7	7	7	6
耗时	140	148	138	141	140

实例 2:非线性二层规划模型:

$$\begin{cases} \min_x F = -x_1^2 - 2x_2^2 - y_1^2 - 2y_2^2 \\ \text{s. t.} & -2x_1 + 3x_2 \leq 9 \\ & 11x_1 + 7x_2 \leq 77 \\ & x_1, x_2 \geq 0 \end{cases}$$

$$\begin{cases} \min_y f = -2x_1^2 - x_2^2 - 2y_1^2 - y_2^2 \\ \text{s. t.} & -3x_1 - 2x_2 - 7y_1 + 9y_2 \leq 70 \\ & -9x_1 - 4x_2 + 12y_1 + 7y_2 \leq 100 \\ & y_1 \geq 0, y_2 \geq 0 \end{cases}$$

取上层规划的加速系数 $c_1 = c_2 = 2.05$, 微粒数 $n = 20$, 用带收缩因子的 PSO 迭代预设代数 50 代, 重复 10 次实验, 所得数据见表 2, 平均从第 24.8 代开始即得到最优点 $(x_1^*, x_2^*, y_1^*, y_2^*) = (3.5745, 5.3830, 12.2252, 1.0000)$, 此时 $F^* = -222.1847$ 。整个过程平均耗时 77.53s。

表 2 实例 2 做 10 次实验所得结果(s)

收敛代数	24	27	26	21	24
耗时	77.66	77.67	77.30	77.69	77.53
收敛代数	26	27	22	26	25
耗时	77.86	77.78	77.14	77.52	77.13

[分析 1]

实例 1 为线性二层规划, 决策变量只有两个, 且上层规划没有约束; 而实例 2 为非线性二层规划, 决策变量有四个, 且上层规划有约束, 比较而言, 实例 2 更为复杂, 那为何求解实例 2 所耗时间更加短暂呢? 在上层规划的初始化过程中, 寻找 m 个满足上层约束集的微粒的做法以及上层规划的微粒群迭代过程中, 如果新的微粒不在上层约束集内则重新进化微粒的位置和速度的做法将消耗大量的时间, 特别是某些微粒在可行集的边缘时, 需要许多次重新初始化或进化才能找到满足上层约束集的微粒, 所以在初始化时, 应尽量让微粒均匀分布于整个上层约束集, 这就使我们必须对上层约束集有相当的认识, 从而, 在寻找上层规划约束集的内点 x_0 时, 尽量让 x_0 位于上层约束集的中心。这样

就能够更快, 且更均匀地将个微粒散布于上层约束集内, 节省时间。正是因为实例 1 在这一步骤上不如实例 2 造成了实例 1 的简单模型求解时间反而长于实例 2 的复杂模型。

[分析 2]

定义: 设一个算法重复 n 次, 每次都有相同的截止代数 d_{\max} , d_{mean} ($d_{\text{mean}} < d_{\max}$) 为算法收敛的平均迭代次数, T_{mean} 为算法执行完毕的平均消耗时间, 则定义算法的代平均时间为 $\bar{t} = \frac{T_{\text{mean}}}{d_{\max}}$, 定义算法平均收敛时间为 $\bar{T} = \bar{t} \times d_{\text{mean}} = \frac{d_{\text{mean}} T_{\text{mean}}}{d_{\max}}$ 。设 \bar{t}_1, \bar{t}_2 , 分别表示实例 1 和实例 2 的代平均时间; \bar{T}_1, \bar{T}_2 分别表示实例 1 和实例 2 的平均收敛时间, 则 $\bar{t}_1 = 142.3 \div 50 = 2.8460$, $\bar{t}_2 = 77.53 \div 50 = 1.5506$, $\bar{t}_1 > \bar{t}_2$, $\bar{T}_1 = 142.3 \times 6.5 \div 50 = 18.4990$, $\bar{T}_2 = 77.53 \times 24.8 \div 50 = 38.4549$, $\bar{T}_1 < \bar{T}_2$ 。尽管实例 1 的代平均时间长于实例 2, 即整个过程, 实例 1 消耗时间长于实例 2, 但实例 1 的平均收敛时间短于实例 2。总体来讲, 微粒群算法对实例 1 的简单线性二层规划模型比实例 2 的复杂非线性二层规划模型有更快的收敛速度。

4 结束语

二层规划问题在社会经济、工程技术、军事指挥和企业管理中有着广泛的应用。文中用两种混合微粒群算法对其进行了研究, 解决了线性二层规划和非线性二层规划的求解问题, 并用之于实例得到了较好的结果, 表明微粒群算法解决二层规划是行之有效的。

参考文献:

[1] Mathieu R, Pittard L, Anandalingam G. Genetic algorithm based approach to bilevel linear programming[J]. R. A. I. R. O. Recherche Operationelle, 1994, 28: 1 – 21.

[2] Bard J F. Practical Bilevel Optimization: Algorithms and Applications[M]. Boston: Kluwer Academic Publishers, 1998.

[3] Coleman T F, Li Y. An Interior, Trust Region Approach for Nonlinear Minimization Subject to Bounds[J]. SIAM Journal on Optimization, 1996, 6: 418 – 445.

[4] Coleman T F, Li Y. On the Convergence of Reflective Newton Methods for Large – Scale Nonlinear Minimization Subject to Bounds[J]. Mathematical programming, 1994, 67(2): 189 – 224.

[5] 曾建潮, 介 婧, 崔志华. 微粒群算法[M]. 北京: 科学出版社, 2004.

[6] 滕春贤, 李智慧. 二层规划的理论与应用[M]. 北京: 科学出版社, 2002.