文章编号: 1001-0920(2006)04-0396-04

离散非线性规划问题的改进遗传算法

何大阔, 王福利, 毛志忠
(东北大学 a. 教育部暨辽宁省流程工业综合自动化重点实验室, b. 信息科学与工程学院, 沈阳 110004)

摘 要: 针对实际离散非线性规划问题,分析了离散与连续变量优化问题和求解方法的不同及特性. 根据离散变量与遗传算法的特点,将单纯形搜索与算术交叉思想相结合,提出离散单纯形交叉算子以提高遗传算法的局部寻优能力,将种群逐步向离散极值点进行引导,实现算法的快速离散寻优. 同时,设计了离散变异算子,使遗传算子真正在离散空间中进行搜索. 基于梯度下降思想提出离散修复算子,提高算法对非线性约束的处理能力. 实际离散非线性规划问题的应用研究验证了方法的有效性.

关键词: 离散非线性规划; 遗传算法; 离散单纯形交叉算子; 离散变异算子; 离散修复算子
中图分类号: TP13 **文献标识码**: A

Improved Genetic Algorithm in Discrete Variable Non-linear Programming Problems

HE Da-kuo, WANG Fu-li, MAO Zhi-zhong
(a. Key Laboratory of Process Industry Automation of Ministry of Education, b. College of Information and Engineering, Northeastern University, Shenyang 110004, China. Correspondent: HE Da-kuo, E-mail: hedakuo@mail.sy.ln.cn)

Abstract: For discrete variable non-linear programming problems in practice, characteristics of discrete variable optimization and continuous variable optimization and their solving methods are analyzed. Based on features of discrete variables and genetic algorithm. Simplex searching and arithmetic crossover are combined and a discrete Simplex crossover operator is presented to improve the local searching capability of the genetic algorithm. This method leads the population to local optimization and produces rapid discrete searching. At the same time, discrete mutation operator is presented to make the genetic operator search in discrete space. Discrete repair operator is presented on the basis of the theory of gradient to improve the performance of the algorithm in solving non-linear constraint. The application of practical discrete variable non-linear programming problems shows the validity of this algorithm.

Key words: Discrete variable non-linear programming; Genetic algorithm; Discrete simplex crossover operator; Discrete mutation operator; Discrete repair operator

1 引 言

遗传算法的并行全局搜索性能和对问题较强的适应性,使其不仅成功地应用于连续非线性规划问题[1~4],同时也适于处理离散变量非线性规划问题. 目前,遗传算法在连续变量优化问题中的应用相对较多,对离散变量问题的研究却较少. 但在一些工程实际中,离散变量优化往往比连续变量优化更有意义[5],如结构拓扑设计、机械优化设计等. 对于这类离散非线性规划问题直接应用一般的连续变量遗传算法,收敛性与准确性都很难得到保证. 因此,研究面向离散非线性规划问题的遗传算法具有重要的理论与现实意义.

收稿日期: 2005-03-01; **修回日期**: 2005-04-22.
基金项目: 国家自然科学基金项目(60374003);"973"子课题基金项目(2002CB312200).
作者简介: 何大阔(1975—),男,沈阳人,副教授,博士,从事过程建模及优化理论与应用的研究; 王福利(1957—),男,辽宁辽阳人,教授,博士生导师,从事控制系统故障诊断与容错、系统优化等研究.

本文通过对离散非线性规划问题的分析, 将传统方法与遗传算法相结合, 提出了一种针对离散非线性规划的改进遗传算法

2　离散非线性规划问题

离散非线性规划问题一般可描述为

$$\min f(x) = f(x_1, x_2, \cdots, x_n),$$
$$\text{s.t.} \begin{cases} g_i(x) \leq 0, i = 1, 2, \cdots, m_1; \\ h_i(x) = 0, i = m_1 + 1, \cdots, m_1 + m_2. \end{cases} \quad (1)$$

其中 $x \in X^D$, X^D 为离散空间

变量是离散的, 这是离散非线性规划问题最基本的特点与难点 变量的离散性不仅使问题目标函数和约束函数在其可行集内既不连续也不可微, 造成解析方法很难实现[6~7], 同时也给遗传算法这样的直接随机优化方法带来困难[8]. 求解连续非线性规划问题主要集中在搜索方法与约束处理两个方面, 而对于离散非线性规划问题, 离散变量的处理以及搜索与约束处理方法如何真正适于离散变量也是问题的关键

理论与实践证明,"先连续后圆整"的思路通常很难获得令人满意的结果 简单地运用连续变量优化方法是不能很好解决离散非线性规划问题的, 对于遗传算法而言也是如此 由于解空间的离散性, 致使离散变量优化问题解空间相对于连续解空间发生极大的变化 当问题最优峰较为陡峭时, 连续最优解与离散最优解差距极大, 局部最优解的分布也有很大差异 而一般的遗传算子主要针对连续变量, 通过计算搜索连续局部极值点, 如果直接应用于离散变量优化问题, 易将种群引向连续局部极值点导致离散优化失败 所以, 应用遗传算法解决离散非线性规划问题时, 遗传算子也应是离散算子, 即进行离散搜索的遗传算子, 真正实现在离散空间寻优

3　遗传算法的改进与实现

3.1　交叉算子

交叉运算不仅具有寻优功能, 也是产生新个体的主要方法, 交叉算子对遗传算法的收敛速度有很大影响 但传统的交叉算子随机性较强、方向性差, 无法保证交叉后的子代优于父代, 即所谓的非效率计算, 从而影响遗传算法的收敛速度 显然, 要改善遗传算法的性能, 首先应改良遗传算法的交叉算子.

单纯形搜索法是对于N 维问题采用对$N+1$个初始点为顶点组成的单纯形进行计算, 通过改良单纯形较劣顶点进行搜索并逐步逼近最优点的方法 可见, 单纯形搜索法是将单纯形中各顶点的信息加以交流来迭代寻优, 这与遗传算法的交叉算子交叉父代信息极为类似, 由于单纯形搜索法具有方向性, 所以寻优速度较快 基于此将单纯形搜索的思想引入遗传算法构成单纯形交叉算子, 使交叉计算具有一定的方向性, 从而提高遗传算法的收敛速度及准确性 在单纯形交叉算子的操作中, 单纯形顶点的随机选取使得每一代的迭代搜索较为随机, 只是对于每一次交叉计算而言方向是确定的, 这样就在一定程度上克服了过强的搜索方向易陷入局部极值的问题 另外, 单纯形法改进劣点的特性与遗传算法种群对种群并行搜索的优势相结合, 相当于多个单纯形进行寻优, 从而避免种群迅速向较优个体靠拢而导致过早收敛, 使算法进一步具有跳出局部极值的能力.

算术交叉算子是实数编码遗传算法常用的交叉算子, 可描述为

$$\begin{cases} x_1' = rx_1 + (1-r)x_2 = x_2 + r(x_1 - x_2), \\ x_2' = rx_2 + (1-r)x_1 = x_1 + r(x_2 - x_1). \end{cases} \quad (2)$$

其中: x_1 和 x_2 为父代个体, x_1' 和 x_2' 为交叉产生的后代个体, r 为[0, 1]上的均匀分布随机数 由式(2)可知, 算术交叉运算类似于由 x_1 和 x_2 分别在两个方向上作线性随机搜索 为进一步提高寻优速度, 本文将算术交叉运算中这种线性搜索的思想应用于单纯形交叉算子的计算中, 即把单纯形法中从最劣点对除最优点以外各顶点中心的反射改为沿最劣点向最优点方向的线性搜索 这种离散单纯形交叉算子既具有算术交叉算子线性搜索的特性, 又具备单纯形法包含多点信息、劣点改进的优势, 将种群向离散极值点进行引导, 使算法在全局上真正实现离散空间优化 具体做法如下:

对于N 维问题, 在种群中任取$N+1$个父代, 其中, 最劣个体为 x_1, 最优个体为 x_2, 由 x_1 在 $\delta = x_2 - x_1$ 上进行引导搜索

$$x_i' = x_{2i} + \mathrm{INT}[\delta_i \lambda / \epsilon_i]\epsilon_i. \quad (3)$$

其中: x_i'为交叉算子后代个体第 i 个分量, x_{2i} 为 x_2 的第 i 个分量, δ_i 为 δ 的第 i 个分量, λ为搜索步长, $\epsilon_i = x_{i,j+1} - x_{ij}$ 为 x_i 的最小离散增量

当搜索失败时, 采用加速收缩步长, 即

$$\lambda = \alpha(1/2)^n. \quad (4)$$

其中: α为步长因子, n 为搜索失败后的搜索次数 在实际应用中, 可取若干次搜索得到的更好解(即优于 x_1 的解) 作为交叉算子的后代个体, 若经若干次搜索未得到更好的解, 则将单纯形中的次劣点作为引导算子的后代个体

3.2　变异算子

对于离散非线性规划问题, 设计离散变异算子, 设 c 为[0, 1]上的均匀分布随机数, 则当 $c > 0.5$时,

$$x_i' = x_i + \mathrm{INT}[r(x_{im} - x_i)/\epsilon_i]\epsilon_i; \quad (5)$$

当 $c < 0.5$ 时,

$$x_i' = x_i - \mathrm{INT}[r(x_i - x_{i1})/\epsilon_i]\epsilon_i. \tag{6}$$

其中 INT 为取整运算. 这种方法相当于将随机变量映射到离散解集中, 从而实现离散变异.

3.3　修复算子

对于非线性约束优化问题仅仅依靠对不可行个体实施惩罚而使算法最终收敛于可行解是很难保证搜索效率的, 因为惩罚函数虽规范个体的可行性, 但是却难以规范遗传算子产生非可行解. 修复策略可使不可行解变成可行解, 但设计一种将不可行解完全修正为可行解的修正算法较为困难. 可见, 惩罚和修复两种策略各有长短, 惩罚函数提供了对个体可行性的度量与评价, 而修复策略则是对不可行个体的主动引导. 将两种方法相结合, 不仅实施惩罚, 还要对个体进行修复, 从而充分发挥两种策略的优势. 为了提高修复操作的效率, 本文不将非可行个体完全修复成可行解, 而只是对其进行改善, 这样就避免了完全修复所带来的问题, 减少算法处理不可行解的时间.

本文引入修复算子对不可行个体进行可行度的改善. 基于率先惩罚违反最大的约束而使个体逐渐趋于满足约束的思想, 设

$$g_{\max} = \max\{0, g_i(x), |h_i(x)|\}. \tag{7}$$

其中 $g_{\max}(x)$ 为 $g_{\max}$ 对应的约束. 若 $x \notin X$, 即 x 不在可行域内, 将 x 沿 $-\nabla g_{\max}(x)$ 方向搜索, 可以使 x 趋于满足 $g_{\max}(x)$[9]. 所以, 为改善不可行个体的可行度, 对不可行个体进行 $-\nabla g_{\max}(x)$ 方向的算子操作, 修复方向 $O(x)$ 为 $O(x) = -\nabla g_{\max}(x)$.

修复算子对个体进行操作, 即

$$x_k' = x_k + \lambda_k O(x_k). \tag{8}$$

其中 λ_k 为搜索步长. 搜索步长的确定对于修复算子是很重要的, 步长过大可能会影响其他约束的满足程度降低修复效果, 甚至影响搜索效率及速度; 而步长过小又无法发挥修复的作用. 另外, 对于不同个体搜索步长也不应是定值, 针对不同的个体应适当地实施不同的修复力度, 以充分发挥修复策略的性能. 当 $x \notin X$ 时, 令

$$S = \{i \,|\, g_i(x) > 0, i = 1, 2, \cdots, m_1; h_i(x) \neq 0, \\ i = m_1 + 1, \cdots, m; x \in X^D\}.$$

于是, 本文将搜索步长定义为

$$\begin{aligned} &\lambda_k = \gamma_k \beta_k, \\ &\beta_k = (g_{\max} - g_{\min})/g_{\max}, \\ &g_{\min} = \min\{|g_i(x)|, |h_i(x)|, i \in S\}, \\ &\gamma_k = M(n_1 + n_2). \end{aligned} \tag{9}$$

其中: M 为定值常数; β_k 为协调系数用来协调各约束的综合满足能力, 避免因向违反最大的约束方向修复而对其他约束满足程度产生过大影响; $g_{\min}$ 为约束中违反程度最小约束的值; n_1 为所有不可行个体按 fun 由优到差排序的序号; n_2 为所有不可行个体按 $g_{\max}$ 由小到大排序的序号(即可行度由好到差排序的序号). 若个体的 $n_1 + n_2$ 较小说明个体不仅 fun 优且可行度也很好, 于是, 修复步长较小以实现其保护; 若个体的 $n_1 + n_2$ 较大说明个体不仅 fun 差且可行度也很差, 所以, 修复步长较大以加大对其可行修复的力度.

以上修复步长的确定方法体现了对不可行个体的不同修复操作, 实现对较优不可行个体的扶植以及对较劣不可行个体的大力修复. 由于离散变量无法得到梯度信息, 所以采用次梯度方向作为梯度方向, 即

$$\begin{aligned} &-\nabla g = \left[\frac{\Delta g}{\Delta x_1}, \frac{\Delta g}{\Delta x_2}, \cdots, \frac{\Delta g}{\Delta x_N}\right]^{\mathrm{T}}, \\ &\frac{\Delta g}{\Delta x_i} = \frac{g(x + \epsilon_i) - g(x)}{\epsilon_i}, \\ &i = 1, 2, \cdots, N. \end{aligned} \tag{10}$$

4　实例应用

检验本文改进遗传算法的性能, 针对测试离散优化算法典型的飞机小型螺旋发动机齿轮减速器的重量优化问题[10] 进行应用研究, 初始种群采用随机离散变量, 为保持算法的收敛性, 采用最优保持策略. 该优化问题可描述为

$$\begin{aligned} &\min f(x) = \\ &0.785\,4(3.333x_3^2 + 14.933\,4x_3 - 43.093\,4) - \\ &1.508x_1(x_6^2 + x_7^2) + 7.477\,7(x_6^3 + x_7^3) + \\ &0.785\,4(x_4x_6^2 + x_5x_7^2), \\ &\text{s.t.}\ g_1(x) = 27x_1^{-1}x_2^{-2}x_3^{2} - 1 \leq 0, \\ &g_2(x) = 397.5x_1^{-1}x_2^{-2}x_3^{-2} - 1 \leq 0, \\ &g_3(x) = 1.93x_4^3x_2^{-1}x_3^{-1}x_6^{-4} - 1 \leq 0, \\ &g_4(x) = 1.93x_5^3x_2^{-1}x_3^{-1}x_7^{-4} - 1 \leq 0, \\ &g_5(x) = (110x_5^3)^{-1}[(745x_2^{-1}x_3^{-1}x_4)^2 + \\ &\qquad 16.9 \times 10^6]^{0.5} - 1 \leq 0, \\ &g_6(x) = (85x_7^3)^{-1}[(745x_2^{-1}x_3^{-1}x_5)^2 + \\ &\qquad 157.5 \times 10^6]^{0.5} - 1 \leq 0, \\ &g_7(x) = x_2x_3 - 40 \leq 0, \\ &g_8(x) = (1.5x_6 + 1.9)x_4^{-1} - 1 \leq 0, \\ &g_9(x) = x_1x_2^{-1} - 12 \leq 0, \\ &g_{10}(x) = 5 - x_1x_2^{-1} \leq 0, \\ &g_{11}(x) = (1.1x_7 + 1.7)x_5^{-1} - 1 \leq 0, \\ &2.6 \leq x_1 \leq 3.6, 0.7 \leq x_2 \leq 0.8, \\ &17 \leq x_3 \leq 28, 7.3 \leq x_4 \leq 8.3, \\ &7.3 \leq x_5 \leq 8.3, 2.9 \leq x_6 \leq 3.9, \end{aligned}$$

$$5 \leq x_7 \leq 5.5$$

对约束采用如下惩罚函数形式:

$$p(x) = \begin{cases} 0, x \text{ is feaible;} \\ \text{fun} \times g_{\max}, \text{ otherwise.} \end{cases}$$

其中 fun 为当前个体惩罚前的适应值. 应用本文算法进行求解, 取种群规模为 100, 交叉概率为 0.8, 变异概率为 0.38, 步长因子为 0.8, 交叉算子搜索次数为 3, 选择、交叉、变异、交叉算子的子代比例为 2:4:4, 终止代数为 200, 则可得到离散最优解 x^* 为 [3.5, 0.7, 17, 7.3, 7.3, 2.9, 5], 离散最优值 $f(x^*)$ 为 2 713.65. 而文献[10]的优化结果为 2 994, 在完全满足约束的前提下, 最优值有较大改善. 可见, 本文提出的遗传算法具有良好的离散寻优性能.

由于本文对交叉算子进行了改进, 又引入了离散修复算子, 应该说算法的计算复杂性增强了. 但由于离散修复算子只计算次梯度, 所以, 计算代价较传统实数编码遗传算法变化并不大. 另外, 传统实数编码遗传算法对于这种复杂离散优化问题, 很难或无法求出最优值, 从这个意义上讲, 计算代价上的增加也是必要的.

5　结　　论

针对离散非线性规划问题, 结合单纯形法与线性搜索的思想, 提出离散单纯形交叉算子, 将种群逐步向离散极值点进行引导, 实现加速离散寻优. 为更好地处理约束, 将惩罚与修复相结合, 不仅对个体实施惩罚, 还要进行修复, 充分发挥两种策略的优势. 实例应用研究验证了本文提出的解决离散非线性规划问题的遗传算法的合理性与有效性.

参考文献(References)

[1] Hansen J V. Genetic Search Methods in Air Traffic Control[J]. *Computers and Operations Research*, 2004, 31(3): 445-459.

[2] Saleh H A, Chelouah R. The Design of the Global Navigation Satellite System Surveying Networks Using Genetic Algorithms[J]. *Engineering Applications of Artificial Intelligence*, 2004, 17(1): 111-122.

[3] Juidette H, Youlal H. Fuzzy Dynamic Path Planning Using Genetic Algorithms[J]. *Electronics Letters*, 2000, 36(4): 374-376.

[4] Lyer, Srikanth K, Saxena, et al. Improved Genetic Algorithm for the Permutation Flowshop Scheduling Problem[J]. *Computer and Operations Research*, 2004, 31(4): 593-606.

[5] 朱朝艳, 刘斌, 郭鹏飞. 离散变量结构优化设计的复合形遗传算法[J]. *东北大学学报*, 2004, 25(7): 689-691.
(Zhu C Y, Liu B, Guo P F. Genetic Algorithm in Compound Form for Structural Optimization with Discrete Variables[J]. *J of Northeastern University*, 2004, 25(7): 689-691.)

[6] Chai S, Sun H C. A Relative Difference Quotient Algorithm for Discrete Optimization[J]. *Structural Optimization*, 1996, 12(1): 46-56.

[7] Bland J A. Discrete-variable Optimal Structural Design Using Tabu Search[J]. *Structural Optimization*, 1995, 10(2): 87-93.

[8] Krishnamoorthy R S. Discrete Optimization of Structures Using Genetic Algorithm[J]. *J of Structural Engineering*, 1999, 12(2): 175-184.

[9] Tang J, Wang D. A Hybrid Genetic Algorithm for a Type of Nonlinear Programming Problems[J]. *Computers and Mathematics with Applications*, 1998, 36(5): 11-21.

[10] 赵正佳, 黄洪钟, 王金诺. 混合离散变量优化的遗传算法研究[J]. *中国机械工程*, 1999, 10(12): 1375-1376, 1394.
(Zhao Z J, Huang H Z, Wang J N. The Search of Genetic Algorithms for Optimization with Mixed Discrete Variables[J]. *Machine Design and Research*, 1999, 10(12): 1375-1376, 1394.)