

文章编号: 1006—2106(2009)06—0098—05

基于自适应遗传算法的 3D 钢桁架优化设计研究^{*}

田承昊^{1**} 董 城¹ 刘 明¹ 王铁成²

(1. 铁道第三勘察设计院集团有限公司, 天津 300025; 2 天津大学, 天津 300072)

摘要: 研究目的: 针对标准遗传算法出现的未成熟收敛、振荡、随机性太大等缺点, 通过研究对其改进, 提出自适应遗传算法。利用自适应遗传算法建立钢桁架结构优化设计模型, 采用 matlab 语言编写自适应遗传算法优化分析程序, 用于铁路车站工程一雨棚的优化设计。

研究结论: 针对标准遗传算法存在的缺点, 通过研究提出了自适应遗传算法。该算法在计算过程中依据个体的适应值, 来确定染色体的交叉位置、交叉率、变异率, 使杂交沿着有利于算法收敛的方向进行。确保了子代个体优于父本, 使进化前期变异明显, 后期趋于稳定, 确保搜索到全局最优解, 从而保证了种群开发和搜索的平衡及全局收敛性。并建立了桁架结构优化设计模型, 以 matlab 软件为平台, 对铁路车站一雨棚桁架结构进行了优化设计, 取得了满意的效果, 证明自适应遗传算法是一种理想的建筑结构优化设计方法。

关键词: 自适应遗传算法; 优化设计; 自适应算子; 惩罚函数

中图分类号: TU311.4 文献标识码: A

Research on the Optimal Design of 3D Trusses with Adaptive Genetic Algorithm

TIAN Cheng-hao, DONG Cheng, LIU Ming, WANG Tie-cheng

(1. The Third Railway Survey and Design Institute Group Corporation, Tianjin 300142, China; 2. Tianjin University, Tianjin 300072, China)

Abstract: Research purposes: Aiming to the problem of the standard Genetic Algorithms (SGA), such as premature convergence, oscillation and over-randomization in iterative process, a series of adaptive genetic algorithms is proposed to improve SGA. The optimal design model for 3D trusses is established with the adaptive genetic algorithms and the optimal analysis program for the adaptive genetic algorithms is made with matlab language for optimal design of the canopy of railway station.

Research conclusions: The adaptive genetic algorithms decides the cross location, cross rate and variance rate of chromosome based on the individual adaptive value in calculation process to make the cross moving to the algorithms convergence direction, which ensures the final generation is better than father generation, make the start phase variable obviously and later phase stable slowly and ensure obtaining the optimal solution as a whole for population development, seeking balance and entire convergence. The optimal design of the trusses structure of the railway station canopy is made with the optimal design model for 3D trusses and matlab software. The result shows the adaptive genetic algorithms is an ideal optimal design method for building structure.

Key words: adaptive genetic algorithms; optimal design; adaptive operator; punishment function

标准遗传算法 (SGA) 来源于进化论和群体遗传学, 是计算智能理论的一个重要分支^[1]。它具有以决策变量编码作为运算对象, 以目标函数值作为搜索信

息, 具有隐含并行性, 采用概率搜索技术的特点^[2]。这些特点使 SGA 成为解决以不确定性、非线性、时间不可逆为内涵的工程结构优化问题的有力工具之

* 收稿日期: 2008-11-06

** 作者简介: 田承昊, 1980 年出生, 男, 助理工程师, 硕士研究生。

—^[3]。与传统优化方法相比, SGA 尽管可以更有效地找到全局最优解, 但在迭代过程中经常表现出未成熟收敛、振荡、随机性太大等缺点。为此, 通过研究提出一套新的自适应操作算子, 用于对标准遗传算法的改进。并通过算例, 对包含自适应算子的自适应遗传算法 (AGA) 与 SGA 对比, 验证自适应遗传算法的收敛特性、迭代次数、优化设计效果。

1 SGA 基本原理

SGA 是通过模拟自然进化过程搜索最优解的方法, 其寻优过程是一个迭代过程。它模拟自然界的演化, 即生物通过遗传和变异适应外部环境, 一代一代的优胜劣汰, 繁衍进化, 把搜索空间映射为遗传空间, 将参变量映射为染色体, 向量的每个元素成为基因, 所有染色体组成种群, 并按预定的目标函数对每个染色体进行评价, 据其结果给出一个适应值。在这种机制下, 每一代个体的基因特征可以通过染色体遗传到下一代。代表一个群体的设计方案相互之间可以通过复制和杂交, 并以一定的概率发生变异, 杂交倾向于由群体中较为优秀的个体承担, 相互匹配的个体最好特性结合, 使产生的后代比父代具有更优良的特性。

1.1 基本概念

SGA 的操作对象是染色体, 每条染色体代表求解问题的一个可行解, 用向量 $X = X_1, X_2, X_3, \dots, X_n$ 表示, 把每个 $X_i (i = 1, 2, 3, \dots, n)$ 看作一个遗传基因, 染色体也称个体 X 对每一个个体 X 它对应可行解的某一决策变量。

群体或种群是由一组染色体描述 SGA 搜索的遗传空间, 在搜索过程中, 用适应值 fitness 来评价每条染色体的优劣, 如果优化目标是寻找最大解, 那么 fitness 越大, 相应染色体所代表的可行解越优。

1.2 基本操作

1.2.1 选择

根据每个个体的适应值, 按照一定的规则或方法, 从第 t 代群体 $P(t)$ 中选择出一些优良的个体, 遗传到下一代群体 $P(t+1)$ 中。

1.2.2 交叉

将群体 $P(t)$ 内各个个体随机搭配成对, 对每一对个体, 以某个概论 (称为交叉概率 P_c) 交换它们之间的部分染色体基因值。

1.2.3 变异

对于群体 $P(t)$ 中的每一个个体, 以某种概率 (称为变异概率 P_m) 改变某一个或某些染色体上的基因值。

2 自适应算子的设计

在遗传算法中, 遗传操作算子是决定遗传算法性态的核心, 因此, 自适应策略主要表现在遗传操作的适应性。操作算子的好坏直接影响算法的收敛性和最终的优化结果。为此, 本文设计了一套自适应操作算子, 这种自适应操作算子一方面保证了算法的全局搜索性, 另一方面也改善了算法的收敛性。

2.1 交叉算子和变异算子的设计

SGA 参数中交叉率 P_c 及变异率 P_m 的选择是影响 SGA 行为和性能的关键所在, 直接影响算法的收敛性和群体多样性。SGA 的交叉和变异操作分别是以 P_c 和 P_m 为基础, P_c 和 P_m 各自控制着交叉、变异操作被使用的频值。较大的 P_c 和 P_m 可增强 SGA 开辟新的搜索区域的能力, 但适应值高的基因遭到破坏的可能性增大, 可能产生较大的代沟, 从而使搜索走向随机化。低的 P_c 和 P_m 能够保持一个连续的解空间, 但进化的速度很慢; 若 P_c 和 P_m 太低, 就会有较多的个体直接复制到下一代, 使搜索陷入迟钝状态。简单的 SGA 中, 针对不同的优化问题, 需要反复实验来确定 P_c 和 P_m , 这是一种很烦琐的事情, 而且很难找到适应于问题的最佳值。这使许多学者认识到这些参数需要随着遗传进程而自适应变化。这种具有自适应性能的遗传算法, 在保持群体多样性的同时, 保证 AGA 以合理速度收敛, 具有更高的鲁棒性、全局最优性和效率。Davis 在文献^[2-3]中提出一种 P_c, P_m 随基因操作的在线性能自适应变化的有效方法, 性能提高则 P_c 增加, 反之, 则 P_m 增加。一些其它学者也做了一些研究^[4-6], 对于求最小值问题, 典型的 AGA 交叉算子和变异算子^[7]为:

$$P_c = \begin{cases} P_{c_1} (f_{in} - f) / (f_{in} - f_{vg}), & \leq f_{vg} \\ P_{c_1} & , > f_{vg} \end{cases} \quad (1)$$

$$P_m = \begin{cases} P_{m_1} (f_{in} - f) / (f_{in} - f_{vg}), & \leq f_{vg} \\ P_{m_1} & , > f_{vg} \end{cases} \quad (2)$$

式中 f_{in} ——群体的最小适应值;
 f_{vg} ——每代群体中的平均适应值;
 f ——要交叉的 2 个群体中较小的适应值;
 P_{c_1}, P_{m_1} ——取 (0, 1) 区间的值。

对于框架优化设计的非线性组合优化问题, 式 (1)、式 (2) 表示的遗传算子在进化初期搜索速度慢, 在进化后期易使进化收敛于局部最优解, 所以必须改进自适应遗传算子。

由于 P_c, P_m 对 SGA 的运行收敛有较大的影响, 故应防止 P_c, P_m 的值选择不当造成的进化过程过早收敛或收敛速度极慢的现象出现。自适应算子值的选择,

对远离最值的个体采用较大的 P_c 、 P_m ，以提高搜索速度；对接近最优值的个体采用较小的 P_c 、 P_m ，保证不收敛于局部最优解。为此， P_c 、 P_m 的选取就不能简单的随适应值线性变化，本文研究中对自适应算子进行了改进，使其具有更好的自适应能力，本文构造的自适应遗传算子为：

$$P_c = \begin{cases} \frac{P_{c1} + P_{c2}}{2} + \frac{P_{c1} - P_{c2}}{2} \cos\left(\frac{f - f_{avg}}{f_{min} - f_{avg}} \pi\right), & f \leq f_{avg} \\ P_{c1} & , f > f_{avg} \end{cases} \quad (3)$$

$$P_m = \begin{cases} \frac{P_{m1} + P_{m2}}{2} + \frac{P_{m1} - P_{m2}}{2} \cos\left(\frac{f - f_{avg}}{f_{min} - f_{avg}} \pi\right), & f \leq f_{avg} \\ P_{c1} & , f > f_{avg} \end{cases} \quad (4)$$

改进后的自适应遗传算子的比较如图 1所示。由图 1可知，改进后的 P_c 和 P_m 的值，在适应值大的区域比较大，在中间区域变化迅速，在适应值小的区域比较小，但不直接趋于零。此值一方面保证了不破坏高适应值的个体，另一方面对低适应值的个体采用较高的交叉率可以提高空间搜索能力。

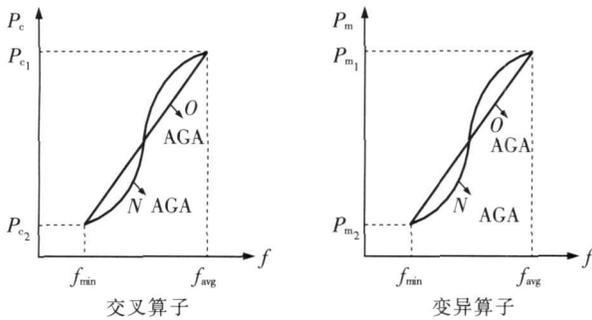


图 1 改进前后的自适应操作算子

2.2 交叉点的自适应控制

标准的SGA采用点式杂交，在2个父代串中随机选取杂交点，然后交换其对应子串。当群体的染色体变得类似时，交叉算子的影响力就会减弱，使搜索变得无效。为了改变这种交叉点选择的盲目性和随机性，本文依据父本染色体的适应值来控制交叉点的位置，适应值大的个体遗传到下一代的子串大，这样的交叉操作能够保持种群传递的连续性。随机的从群体中抽取父本，假定适应值为 f_1 、 f_2 则交叉点的串长为：

$$l_1 = f_1 / (f_1 + f_2) \quad (5)$$

$$l_2 = l - l_1 \quad (6)$$

式中 l —— 染色体的长值。

父本依据式(5)、式(6)确定的交叉点位置进行运算。在进行交叉操作之后，对新生成的染色体进行评

价。如果新染色体的适应值高于原来的染色体，将父代染色体取代，否则保留原来的染色体。在计算中淘汰了子代较父代差的个体，不但使得子代群体的适应值提高，也加速了计算收敛性。

笔者针对一个二维多极值函数的最大值搜索问题(式(7))，对其构造的自适应算子进行了检验，结果表明，在进化代数和最优值方面都显示出改进后的自适应算子的优越性。

$$f = 3(1-x)^2 \exp[(1-x)^2 - (1+y)^2] - 10\left(\frac{x}{5} - \hat{x} - \hat{y}\right) \exp(-x^2 - y^2) - \frac{1}{3} \exp[-(x+1)^2 - y^2] \quad (7)$$

3 适应值函数的设计

3.1 目标函数

适应值用于区分群体中个体的好坏，适应值越大的个体越好，反之，适应值个体越小的个体越差。由于优化问题，要找出满足约束的条件，同时使目标函数达到最小的设计向量解，因此，需要针对结构设计优化问题的个体确定评价方法，即建立遗传算法的适应值函数，一般是通过目标函数的适应值函数映射得到，目标函数为：

$$W = \rho \sum_{i=1}^{N_c} l_i \Lambda \quad (8)$$

式中 N_c —— 单元数；
 l_i —— 单元长度；
 ρ —— 容重；
 Λ —— 单元截面积。

3.2 约束条件

3.2.1 强度约束条件

强度约束条件的表示式为：

$$\frac{|\sigma_i|}{f} - 1 \leq 0 \quad \sigma_i \geq 0 \quad (9)$$

$$\frac{|\sigma_i|}{\sigma_E} - 1 \leq 0 \quad \sigma_i < 0 \quad (10)$$

式中 σ_i —— 杆件的计算应力；
 f —— 钢材的强度设计值；
 σ_E —— 压杆的欧拉临界应力。

3.2.2 杆件宽度比的约束条件

杆件宽度比约束条件的表示式为：

$$\frac{D_i}{t} \leq 100 \quad (11)$$

式中 D_i —— 杆件的管径；
 t —— 壁厚。

3.2.3 截面最大和最小尺寸的约束条件

各杆件截面可供选择的钢管规格见文献[6]附录

L 中表 L-1 截面最大和最小尺寸的约束条件为:

$$\left. \begin{aligned} D \geq 32 \geq 2.5 \text{ mm} \\ D \leq 630 \leq 16 \text{ mm} \end{aligned} \right\} \quad (12)$$

3.2.4 挠度约束条件

挠度约束条件的表示式为:

$$\delta / [\delta] - 1 \leq 0 \quad (13)$$

式中 δ —悬挑端 节点竖向挠度值。

3.2.5 杆件长细比的约束条件

杆件长细比约束条件的表示式为:

$$\frac{4l}{\sqrt{D_i + (D_i - 2t)^2}} \leq 150 \quad \sigma < 0 \quad (14)$$

$$\frac{4l}{\sqrt{D_i + (D_i - 2t)^2}} \leq 250 \quad \sigma \geq 0 \quad (15)$$

其中, 给定的各杆件截面变量域应满足杆件宽度比的约束条件和截面最大及最小尺寸的约束条件。

3.3 目标函数向适应值函数的映射

因 AGA 是用来解决无约束优化问题的, 必须把约束问题转化为无约束问题, 采用罚函数法加以转化。罚函数的基本思想^[7]是: 设法对个体违背约束条件的情况给予惩罚, 并将本次惩罚体现在适应值函数设计中, 这样就把一个约束问题转化为一个附带惩罚因子的非约束问题。罚函数的选取必须谨慎, 若罚函数太大, 设计进程也许会收敛过快, 很可能在局部最优解处收敛; 如果罚函数太小, 则收敛进程会很慢, 极大地增加了运算量, 这 2 种情况都不理想。本文采用式 (16) 作为罚函数。

$$F = W(1 + C) \quad (16)$$

$$\text{其中: } C = \sum_{i=1}^{N_e} \zeta_i + \sum_{i=1}^{N_n} \zeta_i^* + \sum_{i=1}^{N_e} \zeta_i^* + \sum_{i=1}^{N_e} \zeta_i^* + \sum_{i=1}^{N_e} \zeta_i^* + \sum_{i=1}^{N_e} \zeta_i^* \quad (17)$$

式中 F —适应值函数;

C —含约束变量函数;

ζ_i —应力约束变量函数;

ζ_i^* —节点位移约束变量函数;

ζ_i^* —转换公式;

N_e —单元数;

N_n —单元节点;

ζ_i —约束变量函数值, 表示式为:

$$\zeta_i = \begin{cases} 0 & \alpha \leq 0 \\ \alpha^2 & \alpha > 0 \end{cases} \quad (18)$$

对于式 (18) 当为 ζ_i 时:

$$\alpha_i = \frac{|\sigma_i|}{f} - 1 \quad (\sigma \geq 0) \quad (19)$$

$$\alpha_i = \frac{|\sigma_i|}{f} - 1 \quad (\sigma < 0)$$

对于式 (18) 当为 ζ_i^* 时:

$$\alpha_i = \delta / [\delta] - 1 \quad (20)$$

对于式 (18) 当为 ζ_i^* 时:

$$\alpha_i = \frac{4l}{\sqrt{D_i + (D_i - 2t)^2}} - 150 \quad \sigma < 0 \quad (21)$$

$$\alpha_i = \frac{4l}{\sqrt{D_i + (D_i - 2t)^2}} - 250 \quad \sigma \geq 0 \quad (22)$$

用上面所述的罚函数, 惩罚量将总是结构重量的一部分, 违规量越大, 惩罚量也越大。

3.4 程序实现

本文采用二进制编码串表示构件的决策变量进行编程, 程序包括适应值的评估、线性结构分析、规范的执行、罚函数的计算 4 个模块。本研究中包含自适应操作算子遗传算法实现的基本流程如图 2 所示。

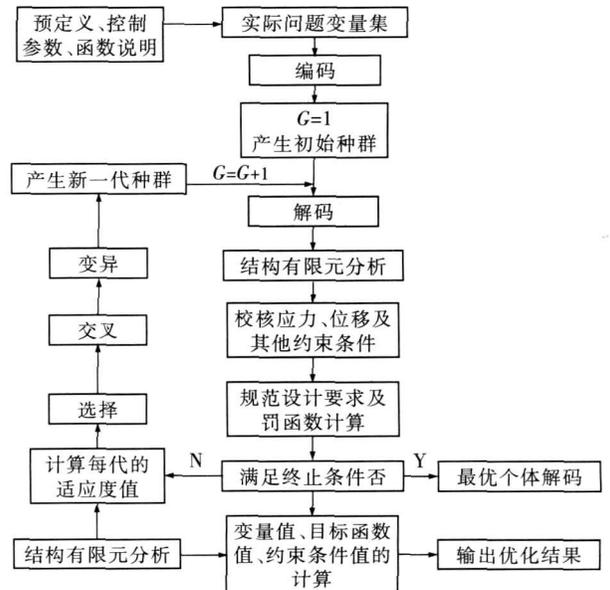


图 2 自适应遗传算法流程图

4 钢桁架优化设计实例

在太原车站既有高架站房改造工程中, 需在高架站房一侧新加装外挑 12 m 的雨棚, 设计中采用正三角钢桁架作为主受力构件, 钢桁架结构如图 3、图 4 所示。节点 1、2、3 铰接, 节点 10 受竖向刚度为 $k=5.0 \times 10^4 \text{ N/mm}$ 的弹簧约束, 材料弹性模 $E=2.05 \times 10^5 \text{ MPa}$, $f=215 \text{ MPa}$, 悬挑端节点 16 挠度允许值 $[\delta]=48 \text{ mm}$, 材料密度 $\rho=7.850 \text{ kg/m}^3$, $\sigma_E=4EA/l$ 。

受力情况为: 在起控制作用的工况组合作用下, 上弦各节点 (不包括节点 1) 分别受值为 50 kN 的竖直向下的荷载作用; 要求上弦、下弦分别采用一种截面, 其余杆件采用同一种截面。用 matlab 语言编写了优化

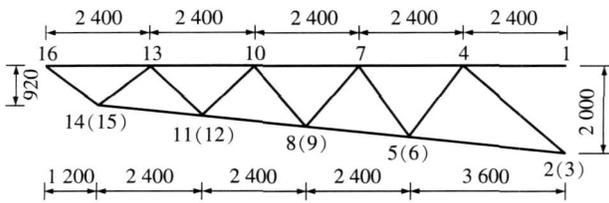


图 3 桁架侧视图 (单位: mm)

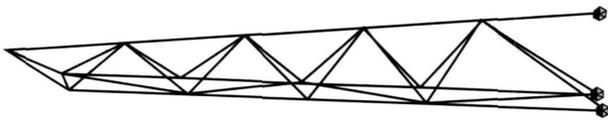


图 4 桁架透视图

分析程序 SFOPT, M对钢桁架进行分析。初始参数设置为: 种群大小 100, 最大世代数 20, 染色体长度 75, 忽略《钢结构规范》中所要求的二阶效应的影响进行线性分析设计。

运行所设计的程序, 迭代至 18次得到最优解: 上弦截面 219×9 , 下弦截面 121×7 , 其它 70×4 设计保持跳跃, 收敛进程如图 5所示。并采用 midas软件对设计所得的最优设计结果进行了静力分析, 结果表明: 节点 16的挠度为 43.7 mm; 大部分杆件应力比为 0.8 在规范范围内。由此验证了所编程序运算结果的可信性。

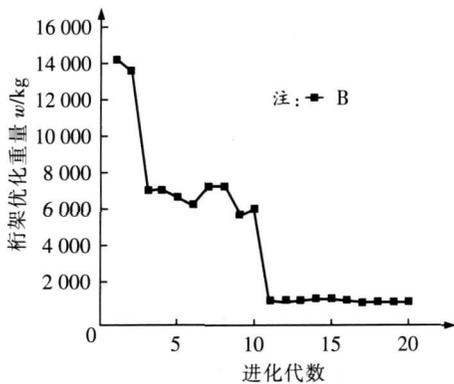


图 5 框架结构设计 2 优化进程图

5 结论

遗传算法是基于生物进化原理的一种全局优化算法, 其基本原理简单, 操作方便, 但在迭代过程中经常表现出未成熟收敛、振荡、随机性太大等缺点。而本文提出的自适应遗传算法, 在计算过程中依据个体的适应值, 来确定染色体的交叉位置、交叉率、变异率, 使杂交沿着有利于算法收敛的方向进行, 确保了子代个体优于父本。使进化前期变异明显, 后期趋于稳定, 确保搜索到全局最优解, 从而保证了种群开发和搜索的平衡及全局收敛性。采用自适应遗传算法建立了连续性结构优化模型, 对铁路车站一雨棚桁架结构进行了优化设计, 取得了比较满意的优化设计方案。计算结果表明: 该算法在收敛速度和获得全局最优解两个方面, 都优于标准遗传算法。通过不断的理论研究, 在结构的优化设计方面有着广泛的应用前景。

参考文献:

- [1] 席裕庚. 遗传算法综述 [J]. 控制理论与应用, 1996 13 (6): 697—706
- [2] 刘勇. 非数值并行算法。遗传算法 [M]. 北京: 科学出版社, 1995
- [3] Jenkins W. M. Structural Optimization with the Genetic Algorithm [J]. The Structural engineer, 1991 69 (24): 418—422
- [4] Goldberg D E. Genetic Algorithms in Search Optimization and Machine Learning [M]. Addison-wesley 1989
- [5] 陈国良, 王熙法. 遗传算法及其应用 [M]. 北京: 人民邮电出版社, 1996
- [6] 宋曼华. 钢结构设计与计算 [M]. 北京: 机械工业出版社, 2006
- [7] 王光远. 结构优化设计 [M]. 北京: 高等教育出版社, 1987.
- [8] Davis L. Adapting Operator Probabilities in Genetics Algorithms Proc 3rd Conf Genetic Algorithms 1989 104—109

(编辑 张 滨)