

文章编号: 1001-0920(2005)01-0036-05

基于粒子群优化算法的约束布局优化

周 驰, 高 亮, 高海兵

(华中科技大学 工业工程系, 湖北 武汉 430074)

摘 要: 布局优化是 NP 难问题, 也是复杂的非线性约束优化问题。针对这个问题, 将新的基于粒子群优化的方法应用于布局参数的优化, 提出了适合粒子群优化的约束处理, 并通过与直接搜索算法的混合, 加强了算法在局部区域的搜索能力。通过实例将该算法与乘法以及基于遗传算法的布局优化方法进行了比较。仿真结果表明, 该算法可以提高布局优化问题解的质量, 同时降低计算费用。

关键词: 粒子群优化; 布局优化; 约束处理; 局部搜索

中图分类号: TP391; TP18

文献标识码: A

Particle swarm optimization based algorithm for constrained layout optimization

ZHOU Chi, GAO Liang, GAO Hai-bing

(Department of Industrial Engineering, Huazhong University of Science and Technology, Wuhan 430074, China)

Correspondent: ZHOU Chi, E-mail: cutecheng@163.com)

Abstract: Layout optimization is an NP-hard problem. It also belongs to complex nonlinear constrained optimization problem. In view of this problem, a new methodology based on particle swarm optimization (PSO) is developed to optimize layout parameters. A constraint handling strategy suit for PSO is proposed. Furthermore, improvement is made by using direct search to intensify local search ability of PSO algorithm. Simulation results show that the proposed algorithm improves the quality of the solution while lowering the computational cost.

Key words: particle swarm optimization; layout optimization; constraints handling; local search

1 引 言

布局优化是经典的 NP 难问题。机械设计与制造、工业工程、电子工程等领域的许多实际问题(如机构设计、生产车间设备布局、集成电路设计等)可转化为布局优化问题求解。因此, 针对布局优化问题的研究具有较好的应用前景。考虑实际应用, 布局优化问题往往包含多种实际约束, 一般建模为高维、多约束的复杂非线性优化问题。传统的优化算法(如线性规划、二次规划、动态规划等)仅适用于特定问题的求解, 且容易陷入局部最优。进化算法是常用的全局优化算法, 适用于各种复杂问题(包括约束优化问题)的求解。但是, 进化算法复杂的进化操作(如选

择、复制、交叉、变异)使其计算费用随问题的规模以及复杂程度呈指数级增长。而且, 由于其缺乏有效的局部区域搜索机制, 算法搜索后期容易出现收敛停滞现象^[1]。

本文提出了基于粒子群优化(PSO)的约束布局优化算法, 给出了针对 PSO 算法的约束处理机制, 并通过与局部搜索算法(直接搜索)的混合来提高算法在局部区域精细搜索的能力。通过实例将该算法与乘法以及遗传算法进行了分析比较。仿真结果表明, 该算法在加速收敛的同时可提高解的精度。

2 约束布局优化问题的数学建模

以待布局物体的形心坐标作为待优化的参数,

收稿日期: 2004-02-15; 修回日期: 2004-05-24

基金项目: 国家自然科学基金项目(50305008)。

作者简介: 周驰(1981—), 男, 湖北枣阳人, 硕士生, 从事群体智能、智能制造等研究; 高亮(1974—), 男, 山东临清人, 副教授, 博士, 从事运筹学与优化、需求链管理等研究。

建立带约束的二维布局优化问题的数学模型,即确定布局优化问题的目标函数和约束条件^[2]。假设在以角速度 ω 旋转的容器中为若干圆形物体布局,且各物体的圆心与质心重合

2.1 目标函数

旋转容器中的布局物体应尽量向容器中心聚集,即最小化所有布局物体的最大包络圆半径。其数学表达式为

$$\min F_1(X) = \min \{ \max \{ \sqrt{x_i^2 + y_i^2} + r_i \} \}. \quad (1)$$

此外,应尽量减小给定角速度 ω 下布局物体的动不平衡量,即

$$\min F_2(X) = \sqrt{\left(\sum_{i=1}^n m_i \omega^2 x_i \right)^2 + \left(\sum_{i=1}^n m_i \omega^2 y_i \right)^2}. \quad (2)$$

因此,布局优化问题的目标函数可表示为

$$\min F(X) = \alpha F_1(X) + \beta F_2(X). \quad (3)$$

其中: x_i, y_i, r_i 和 m_i 为第 i 个布局物形心的横纵坐标、半径及质量; n 为布局物体数目; α 和 β 为目标函数中单目标权重系数

2.2 约束条件

容器中的布局物体之间以及布局物体与容器之间不能发生干涉。首先,为避免布局物体与容器之间的干涉,应保证布局物体不超越容器边界,约束的数学表达式为

$$\sqrt{x_i^2 + y_i^2} \leq R - r_i; \quad (4)$$

其次,必须避免布局物体之间发生干涉,即

$$\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \geq r_i + r_j; \quad (5)$$

最后,给出布局物体的形心坐标约束为

$$-R \leq x_i \leq R, \quad -R \leq y_i \leq R. \quad (6)$$

其中: R 代表旋转容器半径, $i \in [1, n]$

3 粒子群优化算法

通过对生物群体的观察和研究发现,生物群体内个体间的合作与竞争等复杂性行为产生的群体智能,往往能为某些特定的问题提供高效的解决方法。Kennedy 等受鸟群觅食行为的启发,于 1995 年提出粒子群优化算法 (PSO)^[3]。与进化算法相比,PSO 保留了基于种群的全局搜索策略,但其所采用的速度-位移搜索模型操作简单,避免了复杂的进化操作,在解决复杂非线性函数优化^[4]和电压稳定控制^[5]等实际问题中取得了成功的应用

PSO 算法采用以下公式进行单位迭代次数的更新:

$$v_i = w \times v_i + c_1 \times \text{rand}() \times (p_i -$$

$$x_i) + c_2 \times \text{rand}() \times (l_i - x_i), \quad (7)$$

$$x_i = x_i + v_i; \quad (8)$$

其中: w 为惯性权重; $\text{rand}()$ 为均匀分布在 $(0, 1)$ 之间的随机数; c_1 和 c_2 为学习因子。

式(7)中的速度 $v_i = (v_{i1}, v_{i2}, \dots, v_{id})$ 决定粒子在解空间内单位迭代次数的位移,它反映了粒子所代表的解 $x_i = (x_{i1}, x_{i2}, \dots, x_{id})$ 单位迭代次数的变化,如式(8)所示。每一次迭代,粒子通过两个极值更新其速度和位置。第 1 个是粒子从算法迭代初始到当前迭代搜索所生成的最优解,即个体极值 $p_i = (p_{i1}, p_{i2}, \dots, p_{id})$; 第 2 个则是粒子所在邻域对应的最优解,即邻域极值 $l_i = (l_{i1}, l_{i2}, \dots, l_{id})$ 。粒子在解空间内不断跟踪个体极值和邻域极值进行搜索,直到满足迭代停止条件,即达到规定的迭代次数或满足规定的误差标准

PSO 算法基本步骤如下:

Step 1: 随机初始化粒子种群,即初始化种群中所有粒子的速度和位置(可行解);

Step 2: 根据适应度函数对粒子种群进行评价;

Step 3: 更新粒子的个体极值;

Step 4: 更新粒子的邻域极值;

Step 5: 根据式(7)和(8)进行速度和位置(解)的迭代;

Step 6: 重复 Step 2~ Step 5,直到满足算法停止迭代的条件

4 基于粒子群优化算法的约束处理

布局优化问题包含大量复杂的非线性约束,约束处理对于布局优化解的质量有决定性的影响。本文提出了适合 PSO 优化机理的约束处理方法,其主要特性包括:

1) 在迭代过程中,用 Particle Has Been Feasible (PF) 记录当前粒子是否曾经满足所有约束条件,即粒子通过记忆保留的历史约束状态;

2) 用 PF 并结合粒子当前约束状态及带惩罚项的目标函数更新粒子种群的个体和邻域极值;

3) 根据 PF 选择粒子的速度更新策略

需要说明的是,历史约束状态 PF 仅代表历史信息,与粒子的当前约束状态并没有必然联系。PF 在初始阶段设置为 false,一旦 PF 更新为 true,则该粒子的个体极值始终为可行解。粒子当前可能违反约束,但在其个体极值和邻域极值均为可行解的情况下,粒子可尽快返回可行解空间。同时,不可行解的存在可以保证算法在可行解空间边缘的搜索能力

基于粒子群优化的约束处理步骤如下:

Step 1: 随机初始化粒子种群,并令每个粒子的 PF = false。初始种群为随机产生,无需保证存在可

行解

Step2: 判断当前粒子是否违反约束 若违反, 则计算含惩罚项的目标函数; 否则计算初始目标函数, 并将该粒子的 PF 设置为 true

Step3: 根据粒子记录的历史约束状态、当前约束状态以及目标函数, 更新粒子种群的个体极值和邻域极值 具体过程如下:

If 当前粒子 PF = true Then

If 当前粒子优于个体极值 p_i and 当前粒子满足约束 Then

更新 p_i

If p_i 优于邻域极值 l_i Then

更新 l_i

END If

END If

Else If 当前粒子 PF = false Then

If 当前粒子满足约束 Then

更新 p_i

If p_i 优于 l_i Then

更新 l_i

END If

Else If 当前粒子优于 p_i Then

更新 p_i

END If

END If

对每一个粒子均作以上处理

Step4: 如果当前粒子 PF = true, 则采用原始速度和位置更新公式; 否则令速度迭代公式中 $c_1 = 0$, 其余不变, 将该粒子尽快引入可行解空间 需要说明的是, PF = true 时粒子当前既可能满足也可能违反约束 当粒子违反约束时依然采用原始而非 $c_1 = 0$ 时的速度迭代公式是考虑到此时粒子的个体极值和邻域极值均为可行解, 采用原始迭代公式也可使不可行粒子尽快回到解空间内, 同时避免受邻域极值的过多干扰, 防止粒子陷入局部最优

需要指出, 以上约束处理无需保证初始种群存在可行解 粒子初始速度的随机性以及带惩罚项的目标函数可使粒子种群逐渐向可行解空间移动 一旦邻域种群中出现可行解, 便可保证随后每一次迭代粒子的邻域极值均为可行解 邻域中所有粒子均从邻域极值中获得迭代信息, 从而促使粒子尽快回到可行解空间 同时, 由于粒子的初始速度具有随机性, 使得粒子自身具有探索不可行解空间的能力 研究表明, 保留一定比例的不可行解的策略特别适用于最优解位于解空间边缘的情况 对于复杂非线性约束问题, 其可行解空间分布未知的情况下, 该策略

可以在一定程度上防止局部收敛

5 基于粒子群优化的混合算法

与遗传算法类似, PSO 为基于种群的随机搜索算法 算法可快速收敛到最优解所在区域, 但由于缺乏局部区域精细搜索能力, 算法在搜索后期出现收敛停滞现象 因此, 在基于约束处理的 PSO 算法达到最大迭代次数, 或在指定的迭代次数内解的质量无明显改进时, 引入局部搜索算法(直接搜索)对最优解所在的局部区域进行解的微调, 以加强算法的局部搜索能力

直接搜索是有效的局部搜索算法, 步骤简单, 无需优化问题的梯度信息和解空间为凸的要求, 并且可以处理各种等式及不等式约束^[6] 与基于约束处理的 PSO 算法混合的直接搜索的具体步骤如下:

Step1: 结合约束处理的 PSO 算法停止迭代时的最优解作为直接搜索算法的初始解 $P(0)$. 优化变量的初始变化范围矢量 $R(0)$ 定义为

$$R(0) = RMF * Range, \quad (9)$$

其中 Range 为优化参数变化范围矢量

Step2: 在 $P(0)$ 附近生成 n_s 个随机解, 其数学表达式为

$$P_i = P(0) + R(0) \cdot \text{rand}(1, k). \quad (10)$$

其中: P_i 为 $P(0)$ 产生的第 i 个邻域解, $\text{rand}(1, k)$ 代表由 k 个 $(-0.5, 0.5)$ 区间随机数组成的随机矢量, k 为优化变量的数目, \cdot 表示点积运算

Step3: 对 Step2 产生的解进行约束检验, 计算所有可行解的目标函数, 其中目标函数最优解定义为 P_{best} 更新 $P(0)$, 令 $P(0) = P_{\text{best}}$

Step4: 根据

$$R(0) = R(0) * (1 - \beta), \quad (11)$$

线性递减 $R(0)$. 其中 β 为优化参数的邻域搜索范围递减参数

Step5: 算法停止条件检验 若算法达到最大迭代次数或在指定的迭代次数 $P(0)$ 无变化, 则算法停止, 此时的 $P(0)$ 即为布局优化问题的解; 若未满足迭代停止条件, 则返回 Step2 继续迭代

6 实例分析及比较

6.1 参数设置

为便于与遗传算法的比较, 本文选用文献[7]中布局物体数目分别为 7 和 40 的实例 目标函数的相关权重: 最大包络圆半径权重 $\lambda_1 = 1$, 不平衡量权重 $\lambda_2 = 0.01$

针对以上实例, PSO 算法的种群数目分别为 $n = 80$ 和 $n = 100$; 初始权重 $w(0)$ 分别为 1.2 和 1.5, 均随迭代次数线性递减至 0.4; $c_1 = c_2 = 2$ 算法的最大迭代次数分别为 $\text{Itemax} = 10\ 000$ 和 Itemax

= 25 000

直接搜索中, $RMF = 0.5, \beta = 0.05$, 邻域解数目 $n_s = 40$

6.2 仿真结果比较与分析

PSO + LS 代表采用改进约束处理方法及局部搜索的 PSO 算法。对于以上实例, 算法分别在 PSO 迭代次数达到 6 000 和 17 000 时进行直接搜索。PSO 为仅采用改进约束处理的基本 PSO 算法; GA 代表文献[2]中的改进遗传算法; HCIGA 为文献[7]提出的人机交互遗传算法。表 1 和表 2 为不同算法的布局优化结果; 图 1~ 图 4 为对应实例的布局优化示意图。限于篇幅, 改进 GA 及人机交互 GA 的布局结构在此不再赘述。

表 1 实例 1 布局优化结果比较

算 法	包围圆半径/mm	不平衡量/(g·mm)	CPU time/s
PSO + LS	32.230	7.04E-05	9.545
PSO	32.308	8.95E-05	13.439
乘法法	32.559	7.87E-07	31.515
GA	32.837	0.102	288.010
HCIGA	32.662	0.029	166.332

表 2 实例 2 布局优化结果比较

算 法	包围圆半径/mm	不平衡量/(g·mm)	CPU time/s
PSO + LS	811.806	0.002	187.283
PSO	812.311	0.005	267.694
乘法法	869.750	2.32E-05	304.422
GA	874.830	11.395	274.896
HCIGA	870.331	0.006	225.428

传统约束优化方法往往需要保证算法初始解集中存在可行解, 但复杂非线性规划问题(如布局优化问题)包含大量非线性约束, 初始可行解需通过大量观察和试验产生。基于粒子群优化的约束处理可以随机初始化, 即使初始解集中不存在可行解, 同样能以较大概率收敛。

表 3 给出了将基于约束处理的 PSO 算法用于实例 1 问题的收敛过程。可行率表示当前粒子种群中可行解的比例, 曾经可行率为种群中 $PF = true$ 粒子的比例。算法迭代 4 696 次开始出现可行解。作为比较, 对与传统惩罚函数简单结合的 PSO 算法随机测试 20 次, 在相同迭代次数算法均未能收敛到可行解。

在基于约束处理的 PSO 算法基础上, 本文引入局部搜索加速算法在局部区域的收敛速度。仿真实例给出了该算法与传统数学方法(如乘法法及改进遗传算法、人机交互遗传算法)的比较。由表 1 和表 2 的结果可以看出, 局部搜索的引入提高了布局优

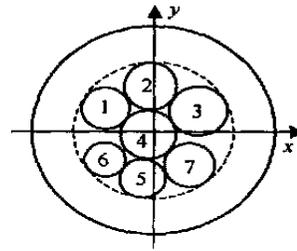


图 1 实例 1 PSO 算法布局优化结果

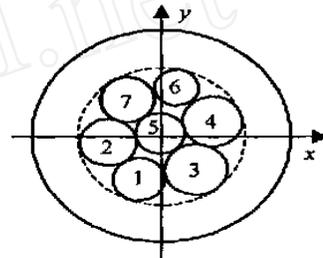


图 2 实例 1 乘法法布局优化结果

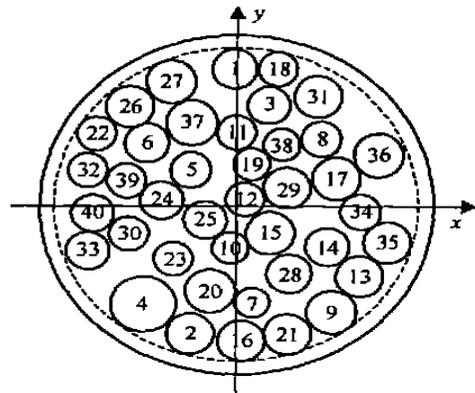


图 3 实例 2 PSO 算法布局优化结果

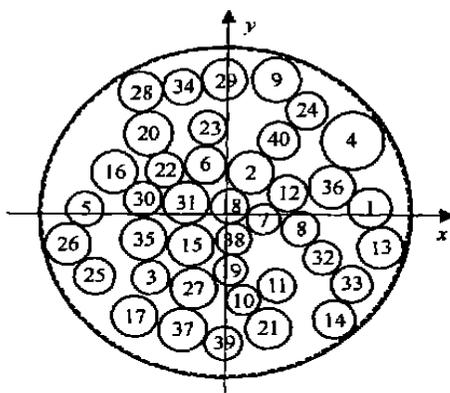


图 4 实例 2 乘法法布局优化结果

化问题解的精度, 同时加速了算法的收敛过程。即使不考虑局部搜索算法的改进, PSO 算法依然在收敛速度、目标函数及解的质量等指标上明显优于其他方法, 如实例 1 中的不平衡量、CPU 运算时间指标以及实例 2 中的包围圆半径指标。

表3 基于约束处理的粒子群优化算法收敛过程

迭代次数	包围圆半径/mm	不平衡量/(g·mm)	可行率	曾经可行率
5 000	32 351 0	0 724 19	0 04	0 15
5 500	32 337 8	0 002 410	0 11	0 19
6 000	32 310 8	0 003 786	0 12	0 22
6 500	32 309 2	0 000 159	0 14	0 22
7 000	32 309 0	0 000 882	0 2	0 22
7 500	32 308 9	0 001 318	0 16	0 23
8 000	32 308 8	0 000 117	0 18	0 23
8 500	32 308 8	0 000 117	0 2	0 24
9 000	32 308 8	0 001 099	0 18	0 24
9 500	32 308 7	0 000 185	0 21	0 26
10 000	32 308 7	0 000 176	0 21	0 26

7 结 语

本文研究了基于粒子群优化的约束布局优化算法,提出了适合 PSO 优化机理的约束处理机制。无需保证初始解集中存在可行解,基于该约束处理机制的 PSO 算法便可有效处理等式与不等式、线性及非线性约束。此外,通过与直接搜索混合,使得算法具有较强的局部搜索能力。通过实例,对本文算法与乘子法以及基于 GA 的布局优化方法进行了分析比较,仿真结果验证了该算法的有效性。结合本文提出的约束处理和局部搜索,PSO 算法可作为通用的非线性约束优化方法应用于其他可抽象为非线性规划模型的工程优化问题。

参考文献(References)

- [1] Franchini M. Use of a genetic algorithm combined with a local search method for the automatic calibration of

conceptual rainfall-runoff models[J]. *Hydrological Science J*, 1996, 41(1): 21-39

- [2] 唐飞, 滕弘飞. 一种改进的遗传算法及其在布局优化中的应用[J]. *软件学报*, 1999, 10(10): 1096-1102
(Tang F, Teng H F. A modified genetic algorithm and its application to layout optimization [J]. *J of Software*, 1999, 10(10): 1096-1102.)
- [3] Kennedy J, Eberhart R C. Particle swarm optimization [A]. *Proc of IEEE Int Conf on Neural Networks*[C]. Perth, 1995: 1942-1948
- [4] Shi Y H, Eberhart R C. Empirical study of particle swarm optimization [A]. *Proc of IEEE Congress on Evolutionary Computation*[C]. Washington, 1999: 6-9
- [5] Yoshida H, Kawata K, Yoshikazu Fukuyama. A particle swarm optimization for reactive power and voltage control considering voltage security assessment [J]. *IEEE Trans on Power System*, 2000, 15(4): 1232-1239
- [6] Chen C L, Chen N. Direct search method for solving economic dispatch problem considering transmission capacity constraints[J]. *IEEE Trans on Power System*, 2001, 16(4): 764-769
- [7] 钱志勤, 滕弘飞, 孙治国. 人机交互的遗传算法及其在约束布局优化中的应用[J]. *计算机学报*, 2001, 24(5): 553-559
(Qian Z Q, Teng H F, Sun Z G. Human-computer interactive genetic algorithm and its application to constrained layout optimization [J]. *J of Computers*, 2001, 24(5): 553-559.)

(上接第 35 页)

5 结 语

对于含两类动态机制的 Markov 跳变系统,在随机稳定性分析的基础上,给出了 L_2-L_∞ 性能指标下的鲁棒滤波器分析与综合结果。其鲁棒性不仅是针对各模式下系统参数的不确定性,而且考虑了模态间跳变转移概率的不确定性,使得滤波器对模态跳变机理具备一定的鲁棒性。数值示例说明了本文结果的有效性。

参考文献(References)

- [1] Xu S Y, Chen T W, Lam J. Robust H_∞ filtering for uncertain Markovian jump systems with mode-dependent time delays [J]. *IEEE Trans on Automatic Control*, 2003, 48(5): 900-907
- [2] Mahmoud M S, Shi P. Robust Kalman filtering for continuous time-lag systems with Markovian jump

parameters[J]. *IEEE Trans on Circuits and Systems I: Fundamental Theory and Applications*, 2003, 50(1): 98-105

- [3] Boukas E K, Liu Z K. Robust H_∞ filtering for polytopic uncertain time-delay systems with Markov jumps [J]. *Computers and Electrical Engineering*, 2002, 28(3): 171-193
- [4] Gao J, Huang B, Wang Z. LM I-based robust H_∞ control of uncertain linear jump systems with time-delay [J]. *Automatica*, 2001, 37(7): 1141-1146
- [5] Feng X, Loparo K A, Ji Y, Chizeck H J. Stochastic stability properties of jump linear systems [J]. *IEEE Trans on Automatic Control*, 1992, 37(1): 38-53
- [6] Rotea M A. The generalized H_2 control problem [J]. *Automatica*, 1993, 29(2): 373-385