

# 基于PSO的约束优化

# 约束优化问题

## ■ 问题描述

$$\text{Min } F(\mathbf{x}, \boldsymbol{\xi}) = (f_1(\mathbf{x}, \boldsymbol{\xi}), \dots, f_m(\mathbf{x}, \boldsymbol{\xi}))$$

$$g_i(\mathbf{x}, \boldsymbol{\xi}) \leq 0 \quad i = 1, 2, \dots, n$$

$$\text{S.t. } h_j(\mathbf{x}, \boldsymbol{\xi}) = 0 \quad j = 1, 2, \dots, p$$

$$l_k \leq x_k \leq u_k \quad k = 1, 2, \dots, d$$

多目标，噪声，带约束



$$\text{Min } f(\mathbf{x})$$

$$g_i(\mathbf{x}) \leq 0 \quad i = 1, 2, \dots, n$$

$$\text{S.t. } h_j(\mathbf{x}) = 0 \quad j = 1, 2, \dots, p$$

$$l_k \leq x_k \leq u_k \quad k = 1, 2, \dots, d$$

约束优化问题



$$\text{Min } f(\mathbf{x})$$

无约束优化问题

## ■ 问题难点

- ❑ 优化空间的复杂性
- ❑ 不可行域的存在性
- ❑ 约束满足和目标优化的不平衡

# 基本概念

- 可行解
- 不可行解
- 可行域F
- 搜索空间S
- 约束违反量

$$\text{Min } f(\mathbf{x})$$

$$g_i(\mathbf{x}) \leq 0 \quad i = 1, 2, \dots, n$$

$$\text{S.t. } h_j(\mathbf{x}) = 0 \quad j = 1, 2, \dots, p$$

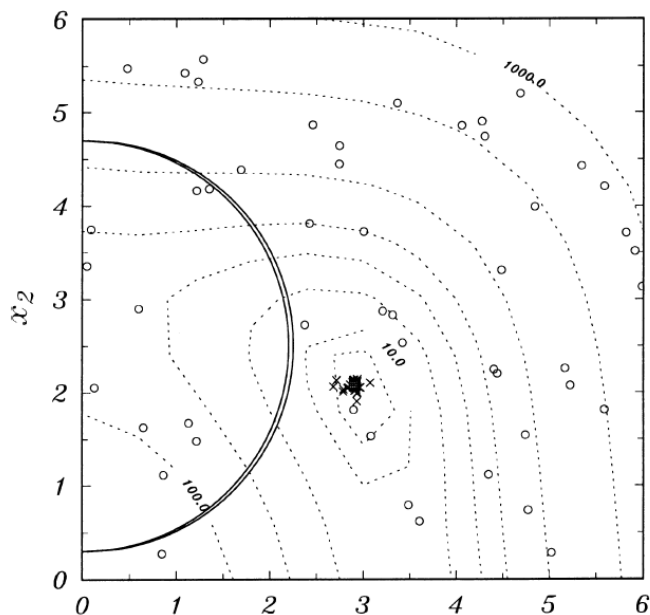
$$l_k \leq x_k \leq u_k \quad k = 1, 2, \dots, d$$

$$G_i(\mathbf{x}) = \max[0, g_i(\mathbf{x})] \text{ 或 } G_i(\mathbf{x}) = \{\max[0, g_i(\mathbf{x})]\}^2$$

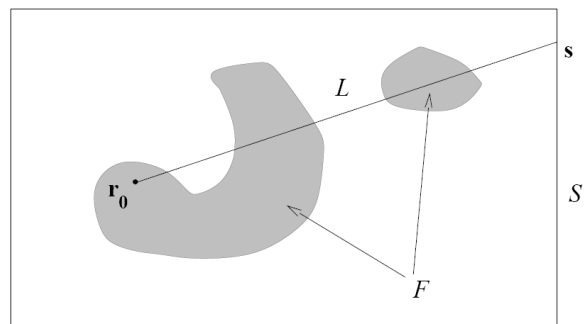
$$H_j(\mathbf{x}) = |h_j(\mathbf{x})| \text{ 或 } H_j(\mathbf{x}) = [h_j(\mathbf{x})]^2$$

# 问题难点

## 强约束问题



## 不连通可行域



## 各种复杂类型

问题	维数	$f(x)$	$\rho$	LI	NI	LE	NE
g01	13	二次	0.0002%	9	0	0	0
g02	20	非线性	99.9969%	1	1	0	0
g03	10	非线性	0.0019%	0	0	0	1
g04	5	二次	26.9664%	0	6	0	0
g05	4	非线性	0.0000%	2	0	0	3
g06	2	非线性	0.0065%	0	2	0	0
g07	10	二次	0.0002%	3	5	0	0
g08	2	非线性	0.8454%	0	2	0	0
g09	7	非线性	0.5369%	0	4	0	0
g10	8	线性	0.0008%	3	3	0	0

# 数学规划方法的不足

## ■ 对函数解析性质要求严苛

- Kuhn-Tucker一阶条件要求  $f, g, h$  可微，活动约束的导数线性独立， $f$  为凸函数， $g$  为凹函数， $h$  为线性函数
- Kuhn-Tucker二阶条件要求  $f, g, h$  二阶可微，活动约束的导数线性独立

## ■ 只保证局部最优

- 导数为局部性质，算法性能受初值影响，易陷入局部极小

## ■ 不适于求解强约束问题

- 复合法，内点法均需要一个初始可行解
- 如何利用不可行解的信息？

# 微粒群算法

## ■ Particle Swarm Optimization, PSO

## ■ 与数学规划方法相比

- 无需导数信息，通用性广
- 全局搜索能力

## ■ 与进化算法相比

- 特有记忆(*pbest*)和信息共享(*gbest*)部分
- 算子简单，更适于求解连续优化问题
- 收敛速度更快，计算效率高

## ■ 核心公式

$$v_i^{t+1} = wv_i^t + c_1R_1(p_i^t - x_i^t) + c_2R_2(p_g^t - x_i^t)$$

$$x_i^{t+1} = x_i^t + v_i^{t+1}$$

# 智能约束处理技术



# 基于协进化模型的PSO算法

## ■ 约束处理技术 – 罚函数法

$$\Phi(\mathbf{x}) = f(\mathbf{x}) + \sum_{i=1}^n r_i \times G_i(\mathbf{x}) + \sum_{j=1}^p s_j \times H_j(\mathbf{x})$$

## ■ 主要难点

- ❑ 罚因子  $r_i, s_j$  的大小对算法性能影响巨大
- ❑ 合适的罚因子需要**手动调试**得到，大大降低算法效率

## ■ 国际研究现状

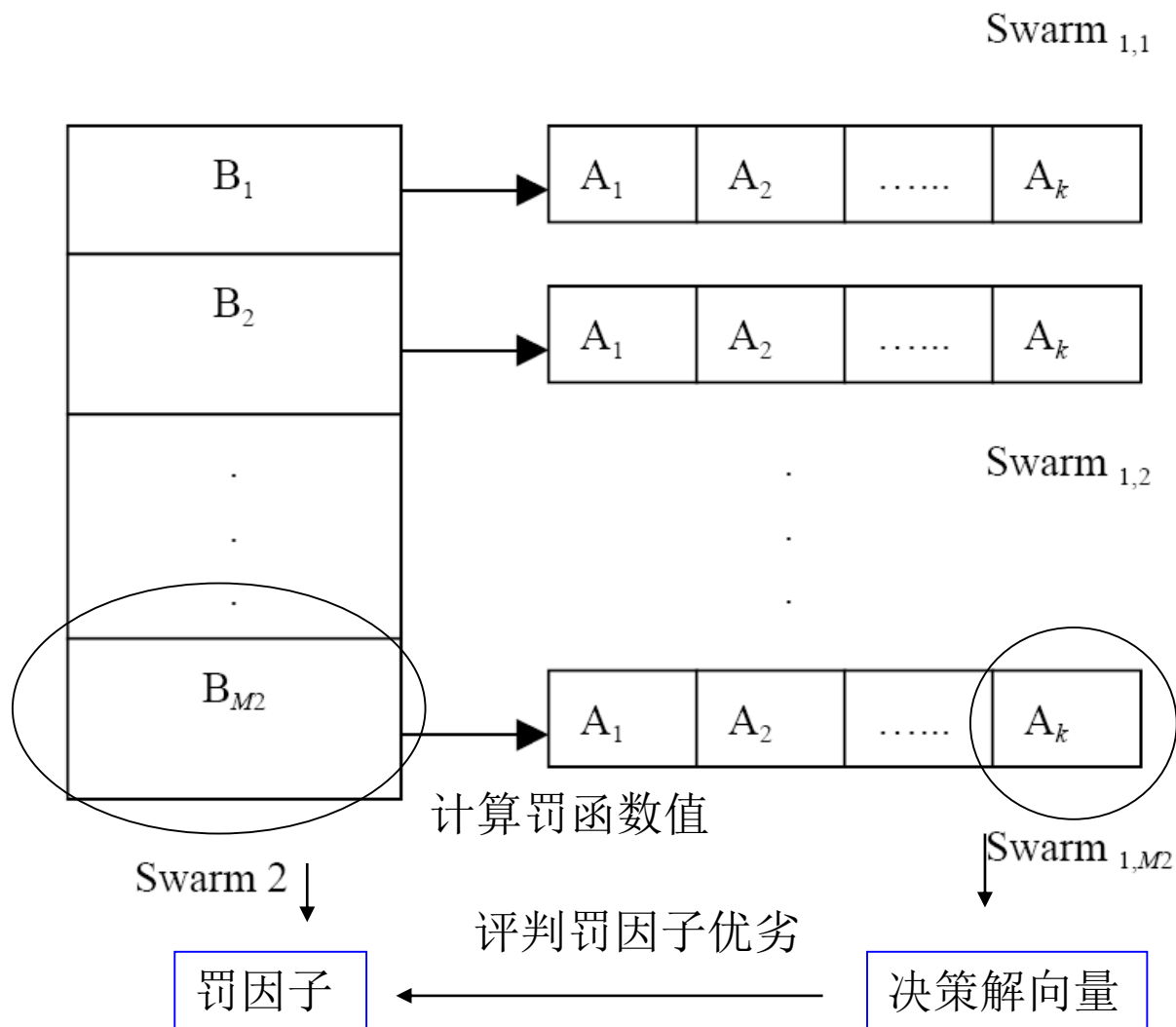
- ❑ 分级罚函数，动态罚函数，退火罚函数等
- ❑ 存在问题：人为制定罚因子变化规则，通用性不强；未充分利用不可行解

## ■ 思路

- ❑ 自适应调整罚因子的框架
- ❑ 设计新的罚函数形式



# 协进化模型 - 研究思路



两类种群，一类寻求最优解，一类寻求合适罚因子，**协同进化**

罚因子也作为优化自变量，**自适应地动态调整**

# 基于协进化模型的PSO

## ■ 种群2（罚因子）的评价函数

- 若 $B_j$ 对应种群1中至少有一个可行解，则称该 $B_j$ 为有效 (valid) 罚因子

$$P(B_j) = \frac{\sum f_{feasible}}{num\_feasible} - num\_feasible$$

- 可行解数量越多，函数值 $f(x)$  越小，对该罚因子 $B_j$ 的评价越好

- 若 $B_j$ 对应种群1中无可行解，则称该 $B_j$ 为无效 (invalid) 罚因子

$$P(B_j) = \max(P_{valid}) + \frac{\sum sum\_viol}{\sum num\_viol} + \sum num\_viol$$

- 不可行解数量越少，约束违反量越小，对该罚因子 $B_j$ 的评价越好

# 基于协进化模型的PSO

## ■ 种群1的评价函数

$$F_i(\mathbf{x}) = f_i(\mathbf{x}) + sum\_viol \times w_1 + num\_viol \times w_2$$

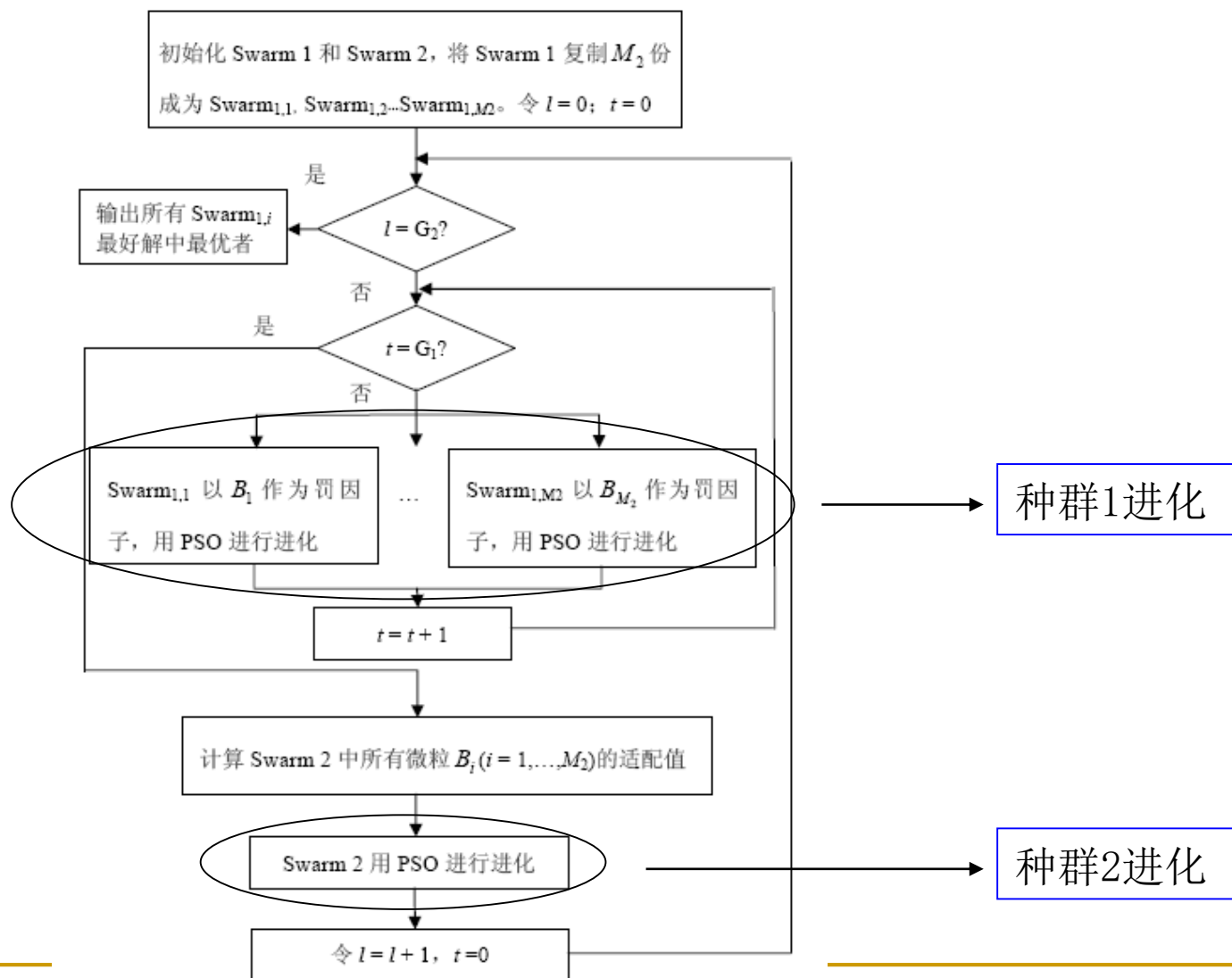
- 不仅考虑**约束违反量**，而且考虑**约束违反个数**信息，罚函数形式更为合理

## ■ 两类种群速度位移进化公式

$$v_i^{t+1} = wv_i^t + c_1R_1(p_i^t - x_i^t) + c_2R_2(p_g^t - x_i^t)$$

$$x_i^{t+1} = x_i^t + v_i^{t+1}$$

# 基于协进化模型的PSO 算法框架



# 仿真结果与比较

## 焊接条设计问题

$$\text{Minimize } f(\mathbf{x}) = 1.10471x_1^2x_2 + 0.04811x_3x_4(14.0 + x_2)$$

$$\text{Subject to } g_1(\mathbf{x}) = \tau(\mathbf{x}) - 13000 \leq 0$$

$$g_2(\mathbf{x}) = \sigma(\mathbf{x}) - 30000 \leq 0$$

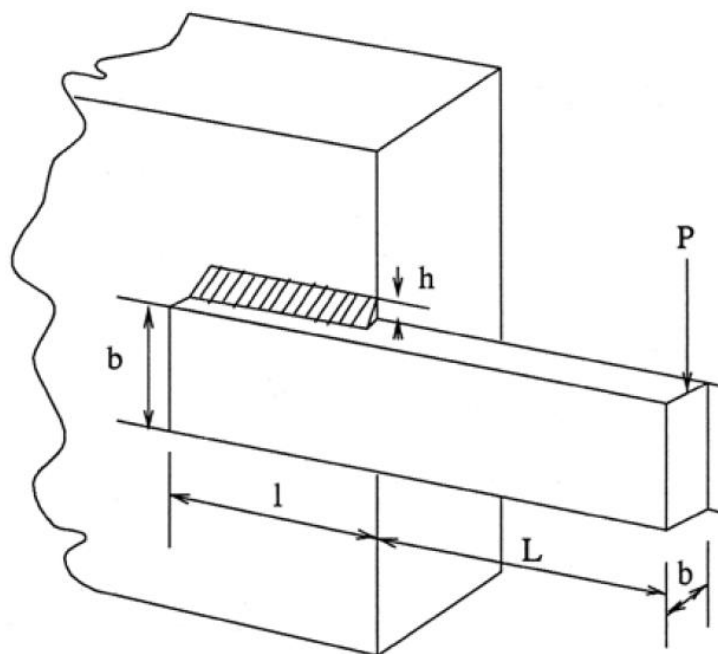
$$g_3(\mathbf{x}) = x_1 - x_4 \leq 0$$

$$g_4(\mathbf{x}) = 0.10471x_1^2 + 0.04811x_3x_4(14.0 + x_2) - 5.0 \leq 0$$

$$g_5(\mathbf{x}) = 0.125 - x_1 \leq 0$$

$$g_6(\mathbf{x}) = \delta(\mathbf{x}) - 0.25 \leq 0$$

$$g_7(\mathbf{x}) = 6000 - P_c(\mathbf{x}) \leq 0$$



# 仿真结果与比较

决策变量	CPSO	文献[51]	文献[52]	文献[53]	文献[39]
$x_1(h)$	0.2024	0.2455	0.2489	0.2088	0.2060
$x_2(l)$	3.5442	6.1960	6.1730	3.4205	3.4713
$x_3(t)$	9.0482	8.2730	8.1789	8.9975	9.0202
$x_4(b)$	0.2057	0.2455	0.2533	0.21	0.2065
$g_1(\mathbf{x})$	-12.839796	-5743.826517	-5758.603777	-0.337812	-0.074092
$g_2(\mathbf{x})$	-1.247467	-4.715097	-255.576901	-353.902604	-0.266227
$g_3(\mathbf{x})$	-0.001498	0	-0.004400	-0.0012	-0.000495
$g_4(\mathbf{x})$	-3.429347	-3.020289	-2.982866	-3.411865	-3.430043
$g_5(\mathbf{x})$	-0.079381	-0.120500	-0.123900	-0.0838	-0.080986
$g_6(\mathbf{x})$	-0.235536	-0.234208	-0.234160	-0.235649	-0.235514
$g_7(\mathbf{x})$	-11.681355	-3604.275002	-4465.270928	-363.232384	-58.66644
$f(\mathbf{x})$	<b>1.728024</b>	2.385937	2.433116	1.748309	1.728226

方法	最优解	平均解	最差解	标准偏差
CPSO	<b>1.728024</b>	<b>1.748831</b>	<b>1.782143</b>	0.012926
文献[51]	2.385937	N/A	N/A	N/A
文献[52]	2.433116	N/A	N/A	N/A
文献[53]	1.748309	1.771973	1.785835	<b>0.011220</b>
文献[39]	1.728226	1.792654	1.993408	N/A

# 算法搜索效率

## ■ 最优解首次代数

Pro.	最小首次代数	平均首次代数	最大首次代数
焊接条设计	30	34.5	39
伸缩绳设计	23	32.8	40
压力管设计	24	32.5	36

## ■ 计算消耗 – 解评价总次数（FFE）

方法	CPSO	文献[53]	文献[39]
FFE	200,000	900,000	80,000

# 对CPSO算法的改进

## ■ 措施I 约束违反量的归一化处理

$$sum\_viol = \sum_{i=1}^N G_i(\mathbf{x}) \quad \longrightarrow \quad sum\_viol = \sum_{i=1}^N \frac{G_i(\mathbf{x}_j)}{\max_j [G_i(\mathbf{x}_j)]}$$

□ 作用：消除不同约束之间尺度的差异

## ■ 措施II 使约束违反和目标函数处于同一尺度

$$F_i(\mathbf{x}) = f_i(\mathbf{x}) + sum\_viol \times w_1 + num\_viol \times w_2$$



$$F_i(\mathbf{x}) = f_i(\mathbf{x}) + \max_i (f_i(\mathbf{x}))$$

$$\times (sum\_viol \times w_1 + num\_viol \times w_2)$$

□ 作用：使得罚因子可以在很小的范围内搜索，并能提供足够的惩罚



# 对CPSO算法的改进

- 措施III 用最优解替代种群中的个体
  - 每进化一代，用全局最优个体代替第一类种群中10%的个体
  - 作用
    - 减少算法在不可行域内的不必要搜索，加快收敛速度
    - 充分利用可行解的信息
    - 加强种群间信息的交流

# 改进结果

表2.11 焊接条设计问题求解统计结果比较

方法	最优解	平均解	最差解	标准偏差
CPSO	1.728024	1.748831	1.782143	0.012926
I+II+III	1.724870	1.725086	1.726160	2.9276e-004
I	1.725217	1.728408	1.734958	0.0027143
II	1.725097	1.730651	1.735682	0.0033761
III	1.725042	1.729845	1.745277	0.0062416
I+II	1.725735	1.733143	1.754402	0.0078539
I+III	1.724863	<b>1.725015</b>	<b>1.725254</b>	<b>1.1468e-004</b>
II+III	<b>1.724860</b>	1.725075	1.725664	2.1692e-004

措施III起主要作用

表2.12 压力管设计问题求解统计结果比较

方法	最优解	平均解	最差解	标准偏差
CPSO	6061.0777	6147.1332	6363.8041	86.4545
I+II+III	<b>6059.7143</b>	6092.2145	6370.7798	67.3090
I	659.3802	1926.5345	3639.9858	767.6291
II	6059.7341	6082.4961	6151.5268	21.5595
III	1046.4521	1127.3367	1215.0715	42.7009
I+II	6059.7157	<b>6075.1899</b>	6090.5412	15.2423
I+III	43.9092	82.7870	140.8168	27.1442
II+III	<b>6059.7143</b>	6080.1936	<b>6090.5291</b>	<b>14.5697</b>

措施II起主要作用

# 改进结果

表2.13 伸缩绳设计问题求解统计结果比较

方法	最优解	平均解	最差解	标准偏差
CPSO	0.012856	0.013630	0.014417	5.1011e-004
I+II+III	0.012718	<b>0.012931</b>	0.013322	1.6489e-004
I	0.013161	0.013662	0.014281	3.3238e-004
II	0.012953	0.013671	0.014850	4.9111e-004
III	0.012715	0.012945	0.013402	1.7816e-004
I+II	0.013166	0.013832	0.014744	4.7759e-004
I+III	0.012736	0.012992	<b>0.013308</b>	<b>1.5667e-004</b>
II+III	<b>0.012681</b>	0.012941	0.013686	2.5289e-004

措施III起主要作用

## ■ 结论

- ❑ 针对**约束最优和无约束最优不同且相差一定量级的问题**，措施II能有效避免算法收敛到不可行域，且能提高搜索效率
- ❑ 措施III能加强种群间信息的交流，针对**可行域小的问题**，措施III能加快种群到达可行域的速度
- ❑ 措施I能在一定程度上减小算法的方差

# 基于可行性规则的混合PSO算法

## ■ 研究思路

- 彻底抛弃罚函数
- 将约束处理和寻求最优分开处理

## ■ 可行性规则

- [Deb2000]首次提出，用于GA锦标赛选择操作时两个体的比较
- 具体规则
  - 可行解总优于不可行解
  - 若两者同为可行解，目标函数值小的更优
  - 若两者同为不可行解，违反约束量小的更优

# 基于可行性规则的混合PSO算法

## ■ 创新点

- [Deb2000]中约束处理技术仍为罚函数法，可行性规则只影响个体的选择
- 本文直接利用可行性规则处理约束，**不再含有罚因子**

## ■ 与PSO的结合

- 个体历史最优位置 $pbest$ 的更新
- 种群全局最优位置 $gbest$ 的更新

- (1)  $p_i^t$  为不可行解，而  $x_i^{t+1}$  为可行解；
  - (2)  $p_i^t$  和  $x_i^{t+1}$  均为可行解，且有  $f(x_i^{t+1}) < f(p_i^t)$ ；
  - (3)  $p_i^t$  和  $x_i^{t+1}$  均为不可行解，且有  $viol(x_i^{t+1}) < viol(p_i^t)$



$$p_i^{t+1} = x_i^{t+1}$$

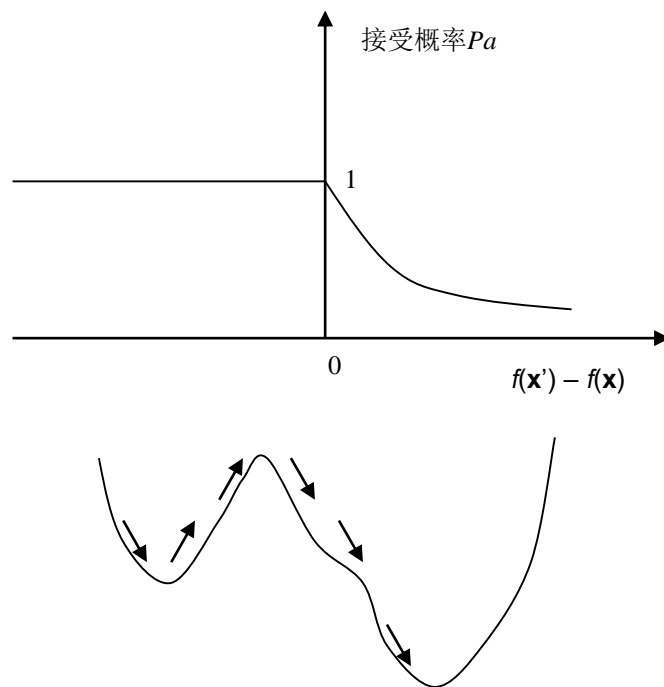
# 基于可行性规则的混合PSO算法

## ■ 单纯采用可行性规则的弊端

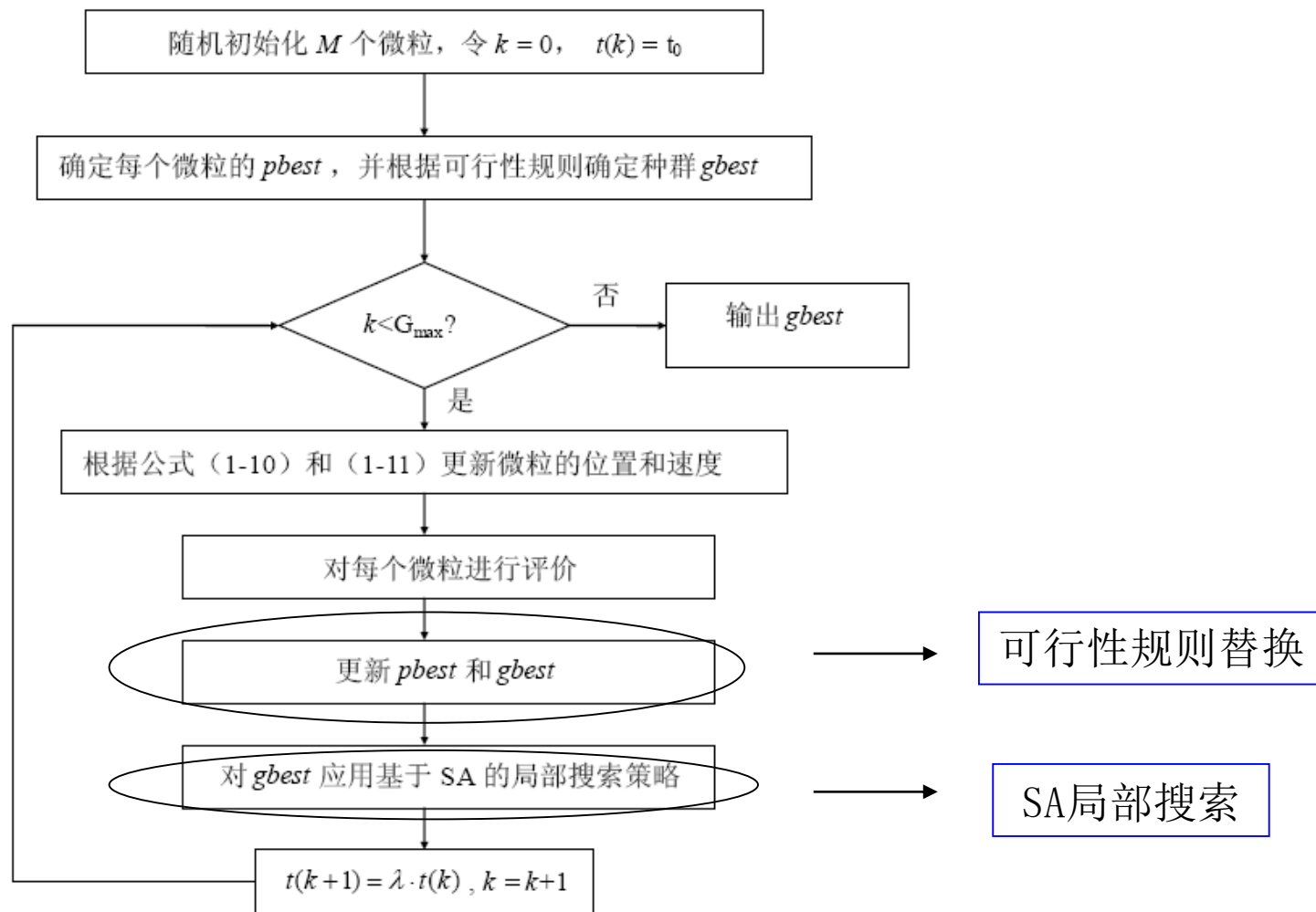
- 可行解总优于不可行解，不可行解面临选择压力过大，算法容易陷入局部极小，早熟收敛

## ■ 应对措施

- 模拟退火算法SA – 概率突跳特性
- 对 $g_{best}$ 用SA进行局部搜索
  - 在 $g_{best}$ 附近产生新解
  - 计算新解适配值和接受概率
  - 以接受概率替换 $g_{best}$
  - 若达到规定搜索步数则终止



# 基于可行性规则的混合PSO算法



# 仿真结果与比较

表3.4 不同算法求解伸缩绳设计问题所得最优解比较

方法	$x_1(d)$	$x_2(D)$	$x_3(P)$	$f(\mathbf{x})$
文献[55]	0.050000	0.315900	14.250000	0.0128334
文献[54]	0.053396	0.399180	9.185400	0.0127303
文献[53]	0.051480	0.351661	11.632201	0.0127048
文献[39]	0.051989	0.363965	10.890522	0.0126810
文献[47]	0.050000	0.317395	14.031795	0.0127210
文献[60]	0.051728	0.357644	11.244543	0.0126747
HPSO	0.051706	0.357126	11.265083	<b>0.0126652</b>

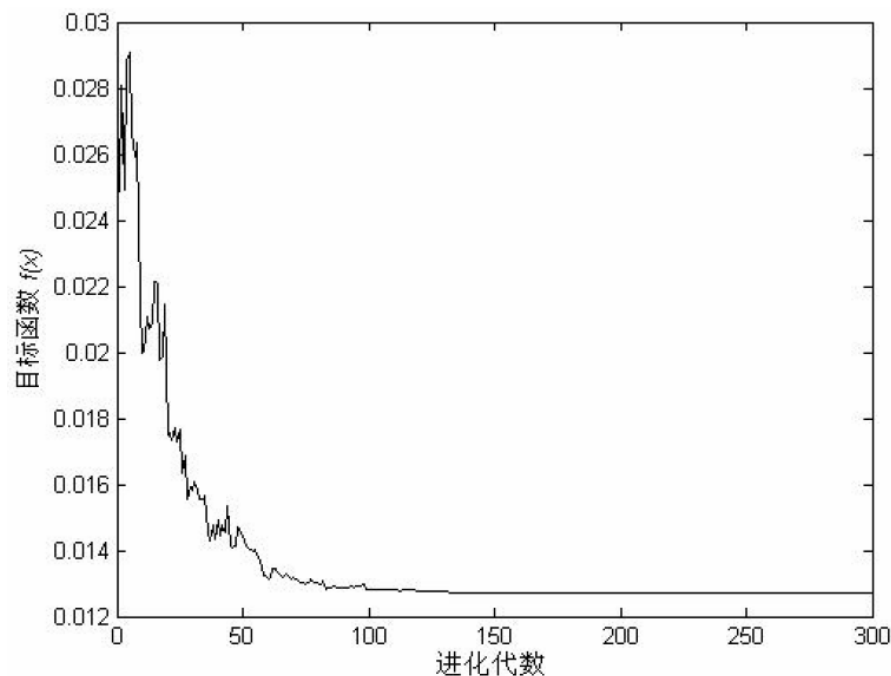
表3.5 不同算法求解伸缩绳设计问题统计结果比较

方法	最优解	平均解	最差解	标准偏差.
文献[55]	0.0128334	N/A	N/A	N/A
文献[54]	0.0127303	N/A	N/A	N/A
文献[53]	0.0127048	0.0127690	0.012822	3.9390e-005
文献[39]	0.0126810	0.0127420	0.012973	5.9000e-005
文献[47]	0.0127210	0.0135681	0.015116	8.4152e-004
文献[60]	0.0126747	0.0127300	0.012924	5.1985e-004
HPSO	<b>0.0126652</b>	<b>0.0127072</b>	<b>0.0127191</b>	<b>1.5824e-005</b>



# 算法搜索效率

## ■ 典型进化曲线



## ■ 计算消耗 - 解评价总次数 (FFE)

方法	HPSO	CPSO[60]	文献[53]	文献[39]
FFE	75,000	200,000	900,000	80,000

# 电力系统经济调度问题

## ■ 电力系统中最基本的优化问题

## ■ 数学描述

□ 目标：最小化燃料成本 Minimize  $f = \sum_{i=1}^n F_i(P_i)$

□ 约束

■ 能量守恒

$$\sum_{i=1}^n P_i - P_L - P_D = 0$$

■ 发电机自身约束

$$P_i^{\min} \leq P_i \leq P_i^{\max}$$

## ■ 特点

□ 目标函数非线性  $\tilde{F}_i(P_i) = a_i P_i^2 + b_i P_i + c_i + |e_i \sin(f_i \cdot (P_i^{\min} - P_i))|$

□ 线性约束

□ 维数较高（有3维，13维，最高为40维）

# 等式约束处理

- 消去等式约束，降维

$$P_n = P_L + P_D - \sum_{i=1}^{n-1} P_i$$

- 增加两个不等式约束

$$P_n^{\min} \leq P_L + P_D - \sum_{i=1}^{n-1} P_i \leq P_n^{\max}$$

# 仿真结果与比较

表4.5 3维ED问题统计结果比较

方法	最优解	平均解	最差解	标准偏差
IFEP <sup>[64]</sup>	8234.07	8234.16	8234.54	N/A
PSO-SQP <sup>[65]</sup>	8234.07	8234.07	N/A	N/A
MPSO <sup>[66]</sup>	8234.07	N/A	N/A	N/A
ECPSO	<b>8234.07</b>	<b>8234.07</b>	<b>8234.07</b>	<b>0</b>
HPSO	<b>8234.07</b>	8240.55	8241.66	2.2240

■ ECPSO与HPSO均能获得与国际上最新算法相当的结果

■ 随着问题维数的增加，算法鲁棒性略微变差

表4.7 13维ED问题统计结果比较

方法	最优解	平均解	最差解	标准偏差
IFEP <sup>[64]</sup>	17994.07	18127.06	18267.42	N/A
PSO-SQP <sup>[65]</sup>	17969.93	18029.99	N/A	N/A
DEC-SQP <sup>[63]</sup>	17938.95	17943.13	17944.81	1.9735
ECPSO	18025.27	18072.80	18095.97	20.0247
HPSO	18078.41	18178.94	18315.47	75.2334

表4.8 40维ED问题统计结果比较

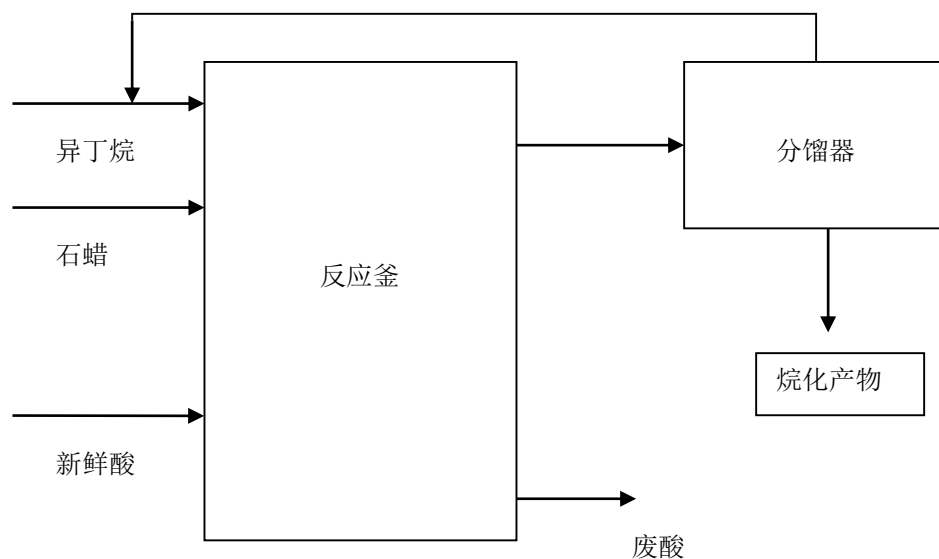
方法	最优解	平均解	最差解	标准偏差
IFEP <sup>[64]</sup>	122624.35	123382.00	125740.63	N/A
PSO-SQP <sup>[65]</sup>	122094.67	122245.25	N/A	N/A
DEC-SQP <sup>[63]</sup>	121741.98	122295.13	122839.29	386.1809
ECPSO	127550.25	128690.02	129944.33	665.6898
HPSO	<b>120952.72</b>	123190.79	128502.42	1868.2823

# 烷化过程优化问题

## ■ 化工优化问题特点

- 由于反应设备、热平衡、化学平衡等因素的制约，化工过程优化问题一般带有**多个非线性约束**，**可行域较小**

## ■ 烷化过程优化



# 烷化过程优化问题

## ■ 数学描述

$$\text{Minimize } c_1 x_1 + c_2 x_1 x_6 + c_3 x_3 + c_4 x_2 + c_5 - c_6 x_3 x_5$$

$$\text{s.t. } c_7 x_6^2 + c_8 x_1^{-1} x_3 - c_9 x_6 \leq 1$$

$$c_{10} x_1 x_3^{-1} + c_{11} x_1 x_3^{-1} x_6 - c_{12} x_1 x_3^{-1} x_6^2 \leq 1$$

$$c_{13} x_6^2 + c_{14} x_5 - c_{15} x_4 - c_{16} x_6 \leq 1$$

$$c_{17} x_5^{-1} + c_{18} x_5^{-1} x_6 + c_{19} x_4 x_5^{-1} - c_{20} x_5^{-1} x_6^2 \leq 1$$

■ 自变量7维，14个非线性约束

■ 可行域非常小  $\rho = 0.0026\%$

# 仿真结果

表4.9 烷化过程优化问题统计结果比较

方法	最优解	平均解	最差解	标准偏差
GGP <sup>[67]</sup>	1227	N/A	N/A	N/A
$\alpha$ BB <sup>[68]</sup>	1227.23	N/A	N/A	N/A
MDE <sup>[61]</sup>	1233.64	N/A	N/A	N/A
ECPSO	1300.41	1348.63	1466.99	44.2829
HPSO	1273.65	1352.86	1511.37	57.3283

## ■ 结论

- 每次运行均能找到可行解
- 其它方法均存在一定问题