第二章系统工程的基础理论与方法论

典型非线性函数

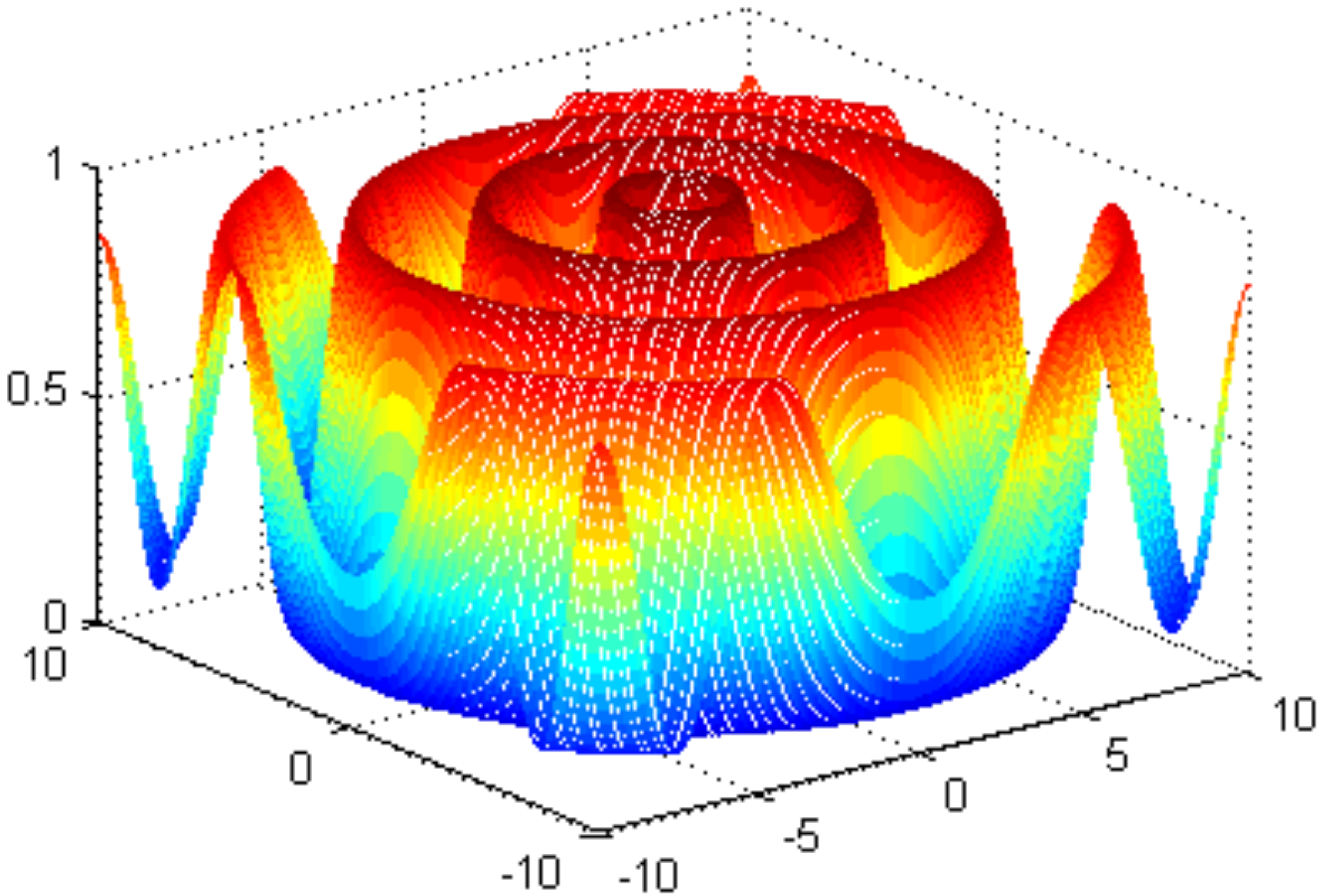
Schaffers 函数：自变量取值范围为：$-10 \le x, y \le 10$。

$$f(x,y) = 0.5 - \frac{\left(\sin\sqrt{x^2+y^2}\right)^2 - 0.5}{\left(1+0.001\cdot(x^2+y^2)\right)^2}$$

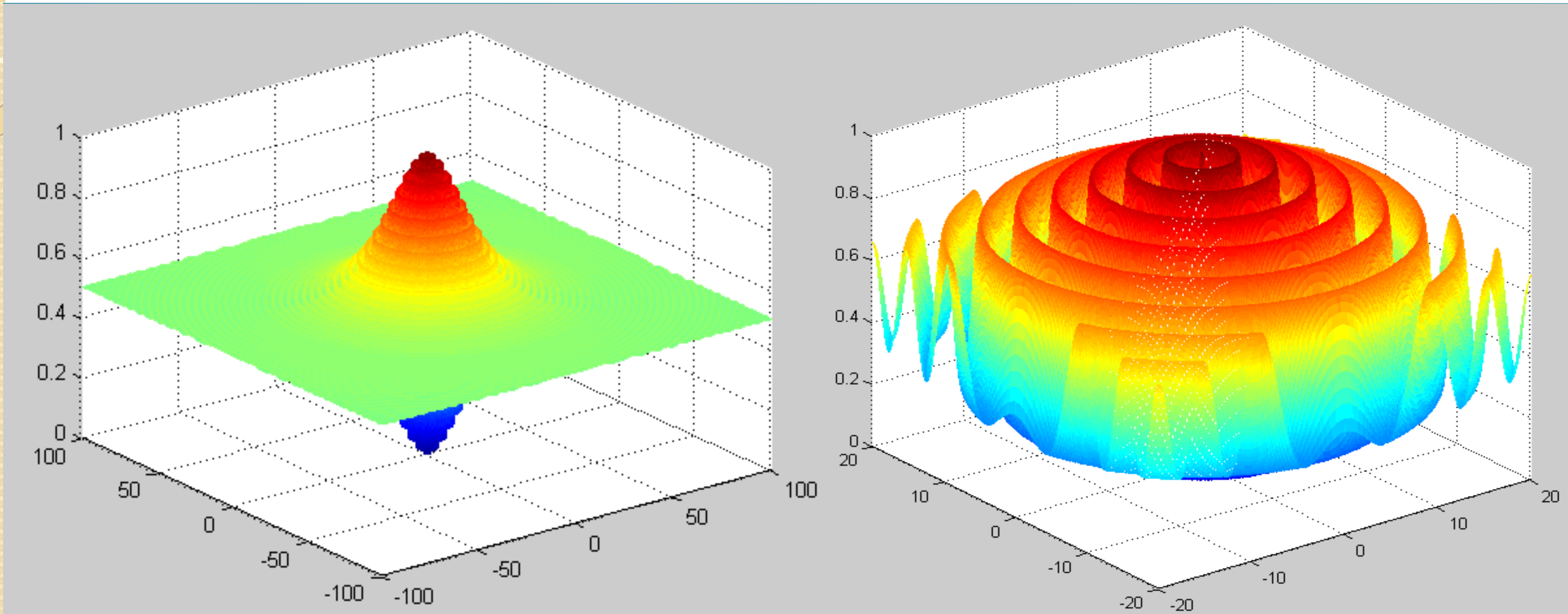
Ackley函数： $f(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right) + 20 + e$

Rastrigin函数：它在$\{x_i \mid x_i \in (-5.12, 5.12), i = 1, 2, \cdots n\}$内大约有10n个局部极小点；但仅当x*=0，其全局最小值为0：$f(x) = \sum_{i=1}^{n}\left(x_i^2 - 10\cos(2\pi x_i) + 10\right)$

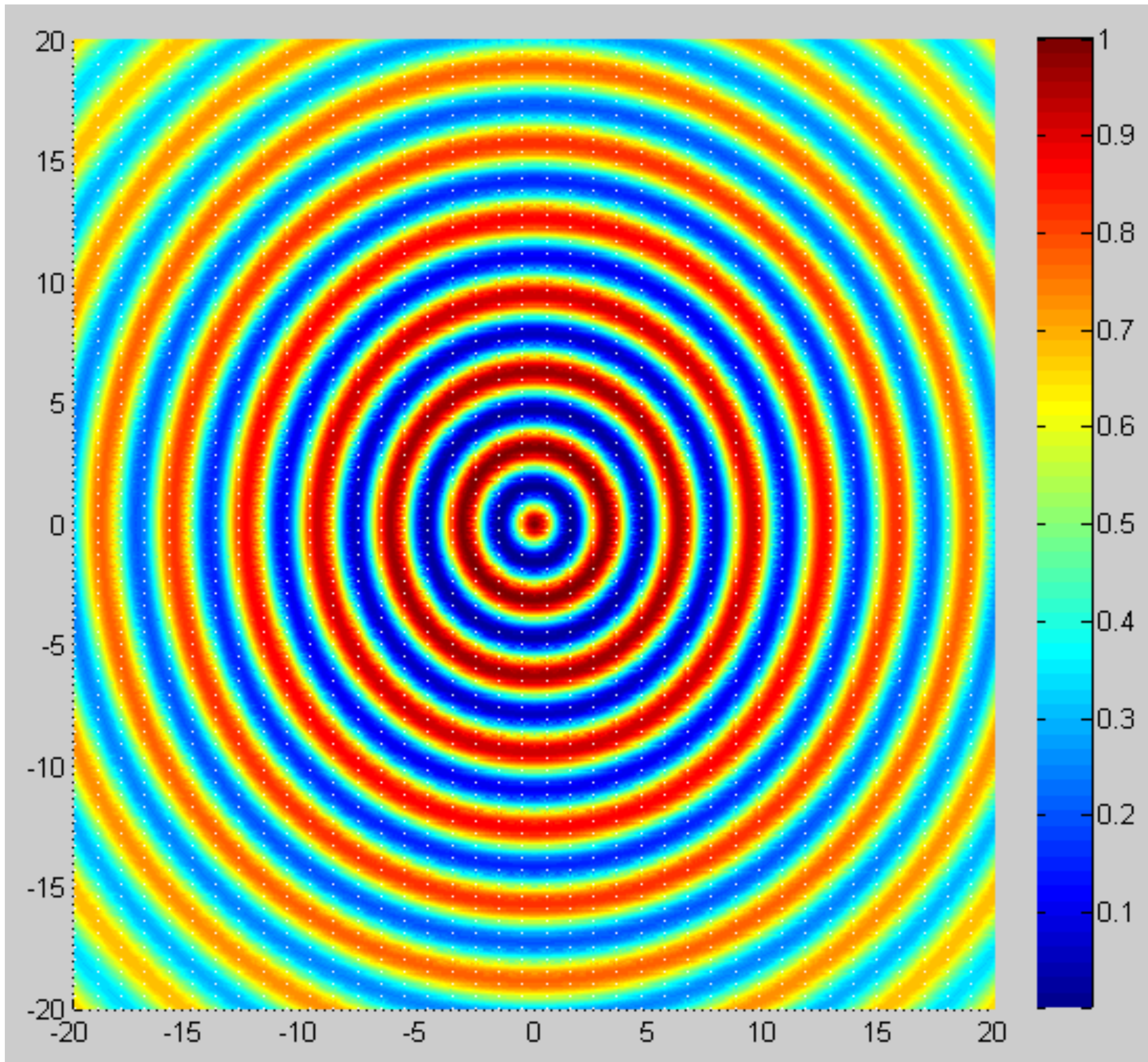
非线性函数（ Schaffers）



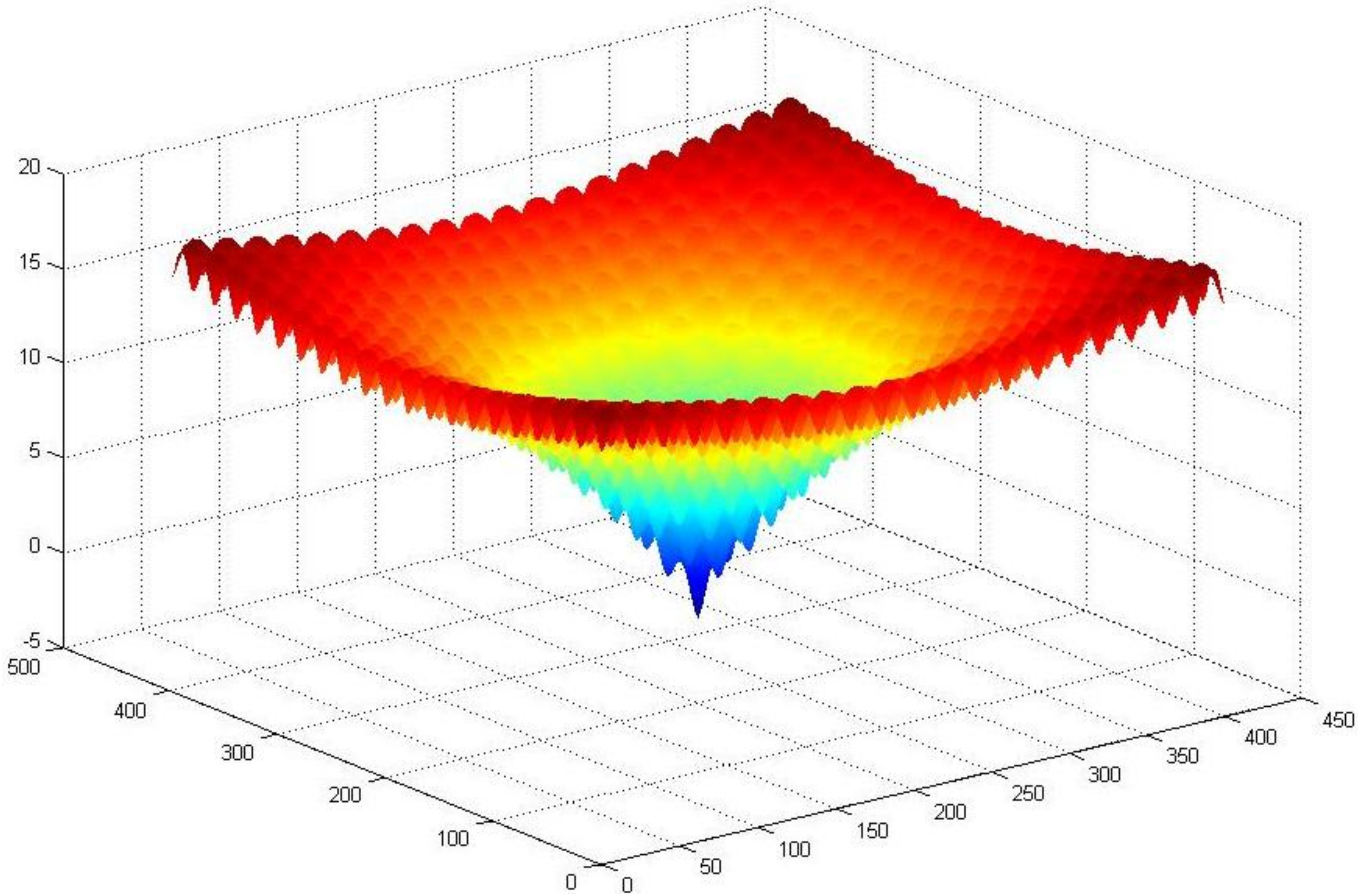
非线性函数（ Schaffers）



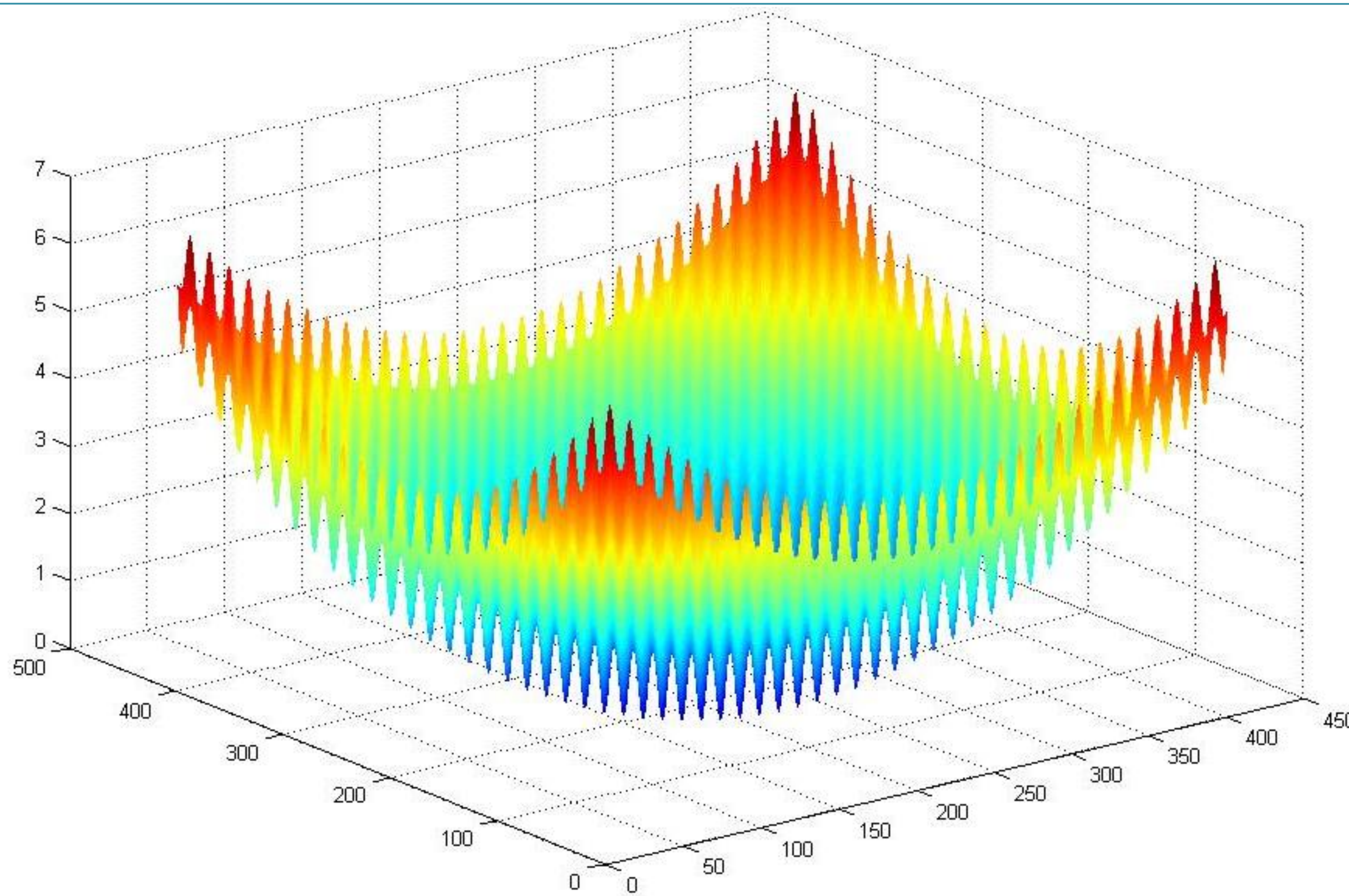
非线性函数（ Schaffers）



非线性函数（ Ackly）


20
15
10
5
0
-5
500
400
300
200
100
0
0
50
100
150
200
250
300
350
400
450

非线性函数（ Rastrigrin）



非线性优化问题的求解

进化优化算法-GA,DE,PSO,ABC等。

非单点迭代，采用群进化的方式进行求解。

不需要函数的导数信息。

函数可以是非连续的。

可以求解MINLP问题。

进化算法与群优化

- 进化算法：父代通过类似选择、交叉、变异等操作生成子代的过程。如遗传算法，差分进化算法，分布估计算法。
- 群优化算法：群体操作指导群中个体向最优解的方向移动。如粒子群优化算法，蚁群优化算法，蜂群优化算法。

进化算法的特点

① 从一个群体(也就是多个点)而不是一个点出发进行搜索；

② 根据适应值选择个体，不需要问题梯度信息；

③ 采用具有随机性的概率转换准则而不是确定性的转换准则；

④ 易于并行计算。

目前研究：**算法设计、算法理论和算法应用**。

遗传算法

遗传算法思想来源于生物进化过程，它是基于进化过程中的信息遗传机制和优胜劣汰的自然选择原则的搜索算法(以字符串表示状态空间)。遗传算法用概率搜索过程在该状态空间中搜索，产生新的样本。

遗传算法

- GA求解问题仅对每个染色体进行评价，并基于适应值来选择染色体，使适应性好的染色体有更多的繁殖机会。
- 在GA中，**位字符串**扮演染色体的作用，单个位扮演了**基因**的作用，随机产生一个体字符串的初始群体，每个个体给予一个数值评价(**适应度**)，选择高适应度的个体参加操作。
- 常用遗传算子:复制、杂交、变异和反转。

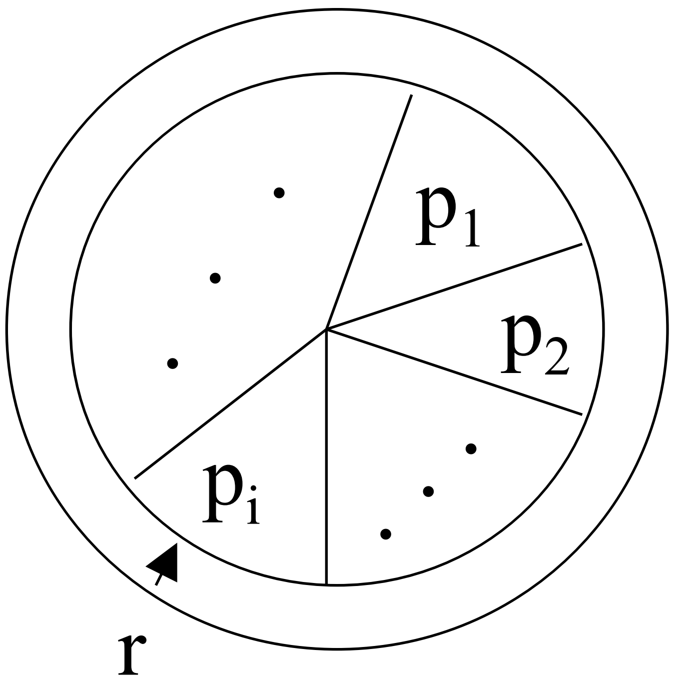
遗传算法–选择运算

- 进行“选择”操作。各染色体都有被选择的可能，这种可能性的大小用P_i来表示。对于某条染色体而言，它被选择的可能性大小取决于该染色体的适应度值及其他染色体的适应度值。最通用的一种方法为取： $P_i = f_i / \sum f_i$
- 式中f_i为个体的适应度值，这在众多文献中被称为“蒙特卡洛法”或“轮盘赌选择法”。

轮盘式选择

计算每个个体 i 被选中的概率

$$p_i = \frac{f(i)}{\sum_{j=1}^{n} f(j)}$$



遗传算法–交叉运算

- 交叉前 交叉后
- A $a_1\ a_2\ a_3\ a_4\ a_5\ |\ a_6\ a_7$ $a_1\ a_2\ a_3\ a_4\ a_5\ |\ b_6\ b_7$
- B $b_1\ b_2\ b_3\ b_4\ b_5|\ b_6\ b_7$ $b_1\ b_2\ b_3\ b_4\ b_5|\ a_6\ a_7$

设 $x_1=(x_{11}, x_{12}, \dots, x_{1n})$，$x_2=(x_{21}, x_{22}, \dots, x_{2n})$ 为两个个体，则 x_1，x_2 交叉生成的新个体 y_1，y_2 为：

$$\begin{cases} y_1 = \dfrac{1-t}{2}x_1 + \dfrac{1+t}{2}x_2 \\ y_2 = \dfrac{1-t}{2}x_2 + \dfrac{1+t}{2}x_1 \end{cases} \qquad t \in (0, T)$$

遗传算法–变异运算

设*x*为一个体，*y*为经变异产生后的个体，则

$$y = x + s \quad \text{或} \quad y = x \times s \qquad s \in (0, S)$$

其他的一些进化优化算法

- 差分进化算法
- 蚁群算法
- 蜂群算法
- 分布估计算法
- 和声搜索算法
- 模拟退火算法
- 文化算法。。。

粒子群优化算法

- 粒子群优化算法(PSO)就是由Eberhart博士和Kennedy博士于1995年提出的一种基于群智能方法的演化计算技术。
- 设想这样一个场景：一群鸟在随机搜寻食物，在这个区域里只有一块食物，所有的鸟都不知道食物在哪里，但是它们知道当前的位置离食物还有多远。

PSO的数学描述

在粒子群优化算法中，每个优化问题的解就是搜索空间中的“粒子”。所有的粒子都有一个由被优化函数决定的适应值，每个粒子还有一个速度决定他们飞翔的方向和距离。在指定的搜索空间中的每个粒子让如下的方程进行飞行：

$$v_{id}^{k+1}=\omega v_{id}^{k}+c_1 rand_1\left(pbest_{id}^{k}-x_{id}^{k}\right)+c_2 rand_2\left(gbest_{id}^{k}-x_{id}^{k}\right)$$

$$x_{id}^{k+1}=x_{id}^{k}+v_{id}^{k+1}$$

进化算法进行整数规划求解

A = 9 8 3 2 1 7 6 0 4 5

B = 2 3 5 0 9 6 1 7 8 4

首先交换A与B的两个匹配区域，得到：

A' = 9 8 3 0 9 6 6 0 4 5

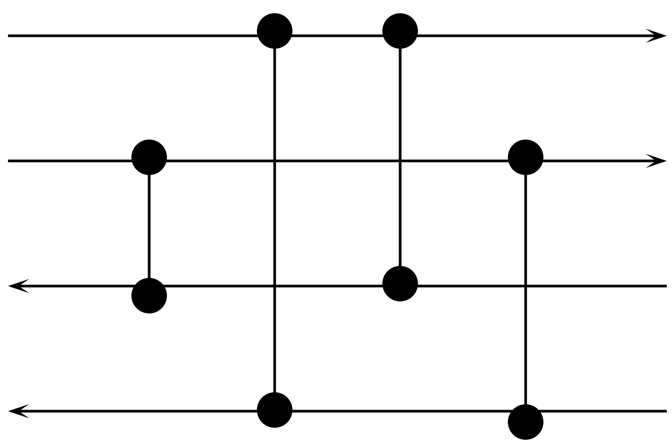
B' = 2 3 5 2 1 7 1 7 8 4

对于A'、B'两子串中匹配区域以外出现的遍历重复，依据匹配区域内的位置映射关系（即0<—>2，1<—>9，7<—>6），则得：

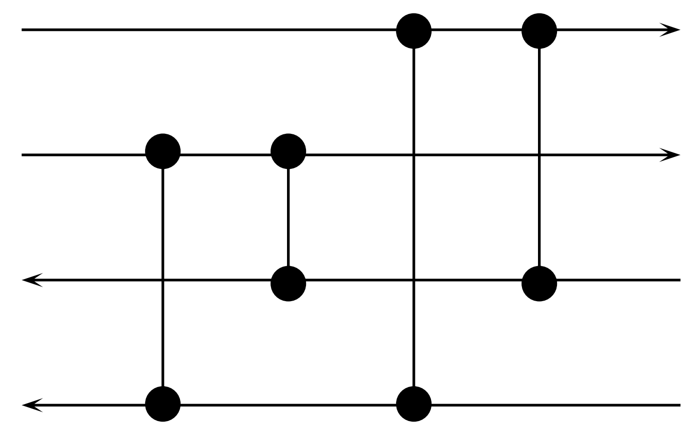
A'' = 1 8 3 0 9 6 7 2 4 5

B'' = 0 3 5 2 1 7 9 6 8 4

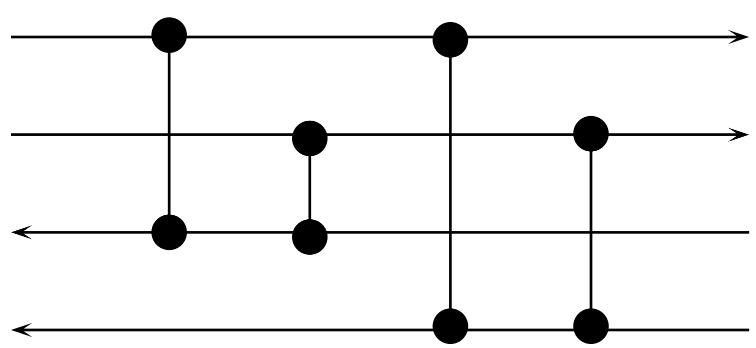
进化算法进行整数规划求解



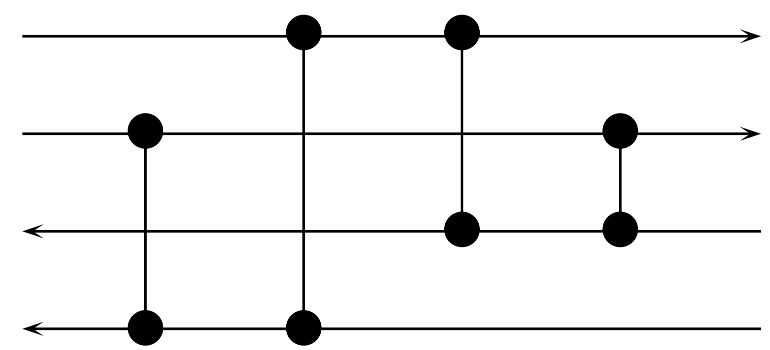
网络结构3214



网络结构4321



网络结构1324



网络结构3412

交叉后子代的结构示意

有约束的非线性规划

处理其约束条件进而实现数值迭代求解方法：

第一种方法:将迭代点序列严格控制在可行域内，从而执行的迭代实际上为无约束优化过程；

第二种方法:序列无约束优化方法(SUMT)。该方法通过将约束项处理成制约函数项加入到目标函数中形成新的广义目标函数，从而将有约束问题化为广义目标函数下的无约束问题。

有约束的非线性规划

第三种方法：在迭代点附近的序列线性化或序列二次函数逼近方法，通过运用迭代点附近的泰勒展开，将有约束的NLP近似为极易求解的LP或二次规化以实现迭代求解。

非线性规划

序列无约束优化方法中常用的制约函数有两类，一类为罚函数(外点法)，一类为障碍函数(内点法)。考虑约束非线性规划问题：

$$\begin{array}{ll} \min & f(x) \\ s.t. & g_i(x) \ge 0, i=1,2,\cdots m \\ & h_j(x)=0, j=1,2,\cdots l \end{array}$$

其中 $f(x), g_i(x), h_j(x), i=1,2,\cdots m, j=1,2,\cdots l$ 是 R^n 上的的连续函数。

罚函数的基本思想是，利用目标函数和约束函数组成广义目标函数 $F(x,\sigma)$

$$F(x,\sigma) = f(x) + \sigma P(x)$$

非线性规划

而$F(x,\sigma)$具有这样的性质：当点x位于可行域以外时，$F(x,\sigma)$取值很大，而且离可行域越远其值越大；当点在可行域内时，函数$F(x,\sigma)=f(x)$。这样，可将原来问题转化成关于广义目标函数$F(x,\sigma)$的无约束优化问题

$$\min \quad F(x,\sigma)=f(x)+\sigma P(x)$$

再利用无约束优化问题数值迭代算法实现求解。在极小化过程中，若x不是可行点，则$\sigma P(x)$取很大正值，其作用是迫使迭代点尽量靠近可行域，通常将$\sigma P(x)$称为罚项，$F(x,\sigma)$又称为罚函数。

非线性规划

可以看出，这类方法的关键是构造$P(x)$，其一般方式为

$$P(x)=\sum_{i=1}^{m}\Phi(g_i(x))+\sum_{j=1}^{l}\Psi(h_j(x))$$

Φ和Ψ是满足下列条件的连续函数

$$\Phi(y)=\begin{cases}0 & y\geq 0\\ >0 & y<0\end{cases}$$

$$\Psi(y)=\begin{cases}0 & y=0\\ >0 & y\neq 0\end{cases}$$

非线性规划

通常，取下述函数：

$$\Phi(x)=[\max\{0,-g_i(x)\}]^{\alpha}$$
$$\Psi(x)=|h_j(x)|^{\beta}$$

其中 α ≥ 1，β ≥ 1 均为给定常数，通常α =β =2.

另外，实际计算中，罚因子σ 的选择很重要；σ 太小，则罚函数的极小点与原约束优化问题的极小点差距较大；σ 太大，给计算增加困难。一般是取一个趋向无穷大的严格递增函数列 $\{\sigma_k\}$。

非线性规划

对于只有不等式约束的非线性规划问题

$$\begin{aligned} &\min \quad f(x) \\ &s.t. \quad g_i(x) \geq 0, i = 1,2,\cdots m \end{aligned}$$

还可以采用障碍函数法，迭代过程总是从可行域的内点出发并保持在可行域内。

设 $f(x), g_i(x), i = 1,2,\cdots m$ 是连续函数，可行域为

$S = \{x \mid g_i(x) \geq 0, i = 1,2,\cdots m\}$。为保持迭代点含于可行域内部，定义障碍函数

$$F(x,r) = f(x) + rB(x)$$

非线性规划

其中 $B(x)$ 是连续函数，当点 x 趋于可行域边界时，$B(x)\to\infty$，r 是很小的正数。这样，当 x 趋向边界时，函数 $F(x,r)\to\infty$；否则，由于 r 很小，则函数 $F(x,r)$ 的取值近似于 $f(x)$，因此，前面问题近似等价于

$$\begin{array}{ll} \min & F(x,r) \\ s.t. & x\in \operatorname{int} S \end{array}$$

式中，int S意为S内部。

非线性规划

两种典型的 $B(x)$ 为

$$B(x)=\sum_{i=1}^{m}\frac{1}{g_i(x)} \qquad B(x)=-\sum_{i=1}^{m}\log(g_i(x))$$

由于 $B(x)$ 的作用，在可行域边界形成围墙，自动阻止迭代点趋向边界，因此相当于无约束优化问题。同样，r 的取值对问题的求解有很大影响，但 r 太小将带来较大的计算问题，合理的作法是取一个严格单调减且趋于零的障碍因子数列 $\{r_k\}$ 。

非线性规划

如果通过某种方法（如Taylor展开）将非线性规划问题的目标函数和约束函数在每个迭代点附近逼近为较简单的函数形式，如线性函数或二次多项式函数，则原非线性规划问题就简化为线性规划问题或较易求解的二次规划问题，大大降低了问题求解的难度。这一处理思想导致了逐次函数逼近类算法的产生。

逐次函数逼近思想最简单的是逐次线性规划算法。

非线性规划

$$\begin{array}{ll} \min & f(x) \\ s.t. & g_i(x) \geq 0, i = 1,2,\cdots m \\ & h_j(x) = 0, j = 1,2,\cdots l \end{array}$$

式中 $f(x), g_i(x), h_j(x)$ 是一阶可微的函数。设 $x^{(k)}$ 是第 k 步迭代点，则在 $x^{(k)}$ 附近对 $f(x), g_i(x), h_j(x)$ 作台劳展开

$$\begin{aligned} f(x) &= f(x^{(k)}) + \nabla f^T(x^{(k)})(x - x^{(k)}) + o(x - x^{(k)}) \\ &\approx f(x^{(k)}) + \nabla f^T(x^{(k)})(x - x^{(k)}) \end{aligned}$$

非线性规划

$$
\begin{aligned}
g_i(x) &= g_i(x^{(k)}) + \nabla g_i^T(x^{(k)})(x - x^{(k)}) + o(x - x^{(k)}) \\
&\approx g_i(x^{(k)}) + \nabla g_i^T(x^{(k)})(x - x^{(k)}) \\
& \quad i = 1, 2, \cdots m
\end{aligned}
$$

$$
\begin{aligned}
h_j(x) &= h_j(x^{(k)}) + \nabla h_j^T(x^{(k)})(x - x^{(k)}) + o(x - x^{(k)}) \\
&\approx h_j(x^{(k)}) + \nabla h_j^T(x^{(k)})(x - x^{(k)}) \\
& \quad j = 1, 2, \cdots l
\end{aligned}
$$

$$
\begin{aligned}
\min \quad & f(x^{(k)}) + \nabla f^T(x^{(k)})(x - x^{(k)}) \\
s.t. \quad & g_i(x^{(k)}) + \nabla g_i^T(x^{(k)})(x - x^{(k)}) \geq 0, i = 1, 2, \cdots m \\
& h_j(x^{(k)}) + \nabla h_j^T(x^{(k)})(x - x^{(k)}) = 0, j = 1, 2, \cdots l
\end{aligned}
$$

非线性规划

如果目标函数$f(x)$二次可微，则可通过对$f(x)$的二次泰勒逼近和对$g_i(x), h_j(x)$的一次线性逼近，将复杂的非线性规划问题化为简单易解的二次规划问题，这种方法称为逐次二次规划算法。

多目标规划

线性规划、非线性规划、整数规划等处理的都是单目标函数的最优化问题。然而，许多实际问题都有多个优化目标，而且有些目标函数之间是相互矛盾的，因此常需要处理多目标规划。

多目标规划

假设有一个姑娘要找一个对象结婚。根据目前流行的观念，她需要考虑帅，有钱，有能力，有家庭背景，会体贴人，爱她，等等。假设她找对象的范围是确定的，比如就在上海市。现在她希望找一个男同胞，争取上面所述的各方面都是最好的。这么一个例子在《笑林广记》记载了。不过那是在战国的齐国。候选者只有两个，标准也只有两个，帅，有钱。当然齐女的选择是明智的，她说：这样吧，我到有钱的那里吃饭穿衣服，到帅的那个家里睡觉。

多目标规划

例某厂生产两种产品A和B，其中A产品是畅销产品，供不应求，每月至少生产1万件，每件利润为1元。但因原料限制，A产品月产量不能超过3万件。生产一件B产品所需工时与生产一种A产品所需工时相同，但每件B产品利润3元。在工人不加班的正常情况下，每月两种产品产量总和不低于5万件，也不高于7万件，且产量超过5万件的部分叫做超产量。工厂希望：（1）A的产量要多；（2）为使工人不加班，超产量要少；（3）利润要多。

多目标规划

解 设 A的月产量 x_1万件，B的月产量为 x_2万件。此问题有三个目标：

$$\max \quad x_1$$
$$\min \quad x_1 + x_2 - 5$$
$$\max \quad x_1 + 3x_2$$

约束条件有：

总产量大于等于 5 万件， $x_1 + x_2 \geq 5$

总产量不能超过 7 万件， $x_1 + x_2 \leq 7$

A产品产量介于 1 万件与 3 万件之间

产量 x_1, x_2 具有非负属性

多目标规划

为表达清楚起见，第二个目标用极大值表示：$\max \quad 5-x_1-x_2$，则此问题的数学模型为

$$
\begin{aligned}
\max \quad & [f_1(x_1,x_2), f_2(x_1,x_2), f_3(x_1,x_2)] \\
s.t. \quad & x_1+x_2 \ge 5 \\
& x_1+x_2 \le 7 \\
& 1 \le x_1 \le 3 \\
& x_2 \ge 0
\end{aligned}
$$

其中，

$f_1(x_1,x_2)=x_1, f_2(x_1,x_2)=5-x_1-x_2, f_3(x_1,x_2)=x_1+3x_2$ 。显然，这是多目标规划问题。

多目标规划

多目标规划的标准形式为

$$\max \quad [f_1(x), f_2(x), \cdots f_p(x)]$$

$$s.t. \quad g_i(x) \le 0, i = 1, 2, \cdots m$$

在多目标规划中，由于目标函数之间相互矛盾特点，一般多个目标函数不能同时达到极值点，总体最优实际上是多个目标之间折衷、平衡的结果，为表达这一概念，需用到非劣解的定义。

多目标规划

设 $S=\{x\mid g_i(x)\le 0, i=1,2,\cdots m\}$ 为多目标规划问题的可行域。若 $x^{'}\in S$，且不存在另一可行点 $x\in S$，使 $f_i(x)\ge f_i(x^{'}), i=1,2,\cdots p$ 成立，且其中至少有一个严格不等式成立，则称 $x^{'}\in S$ 是该多目标优化问题的一个非劣解，所有非劣解构成的集合为非劣集。在非劣集中，使决策者满意的非劣解称为最终解。

一个多目标规划如果存在非劣解，往往存在无穷多个，形成非劣解集。在非劣解集中求解最终解常用两种方法：加权法和约束法。

多目标规划的解法：加权法

加权法将求解多目标规划的非劣解问题化为单目标规划问题。设多目标规划如（2-75），可行域为 S，加权法先给定权系数 $w_j \geq 0, j=1,2,\cdots p$，构造相应的单目标规划

$$\max \quad \sum_{j=1}^{p} w_j f_j(x)$$

$$s.t. \quad x \in S$$

多目标规划

在一定条件下，线性组合求得的最优解为原优化问题的非劣解，有目的地改变权系数 w_j， $j=1, 2\ldots\ldots p$，求解一系列的非劣解，构成非劣解集，然后按某种偏好选取最终解。

多目标规划的解法：约束法

约束法也是一种将多目标规划问题化为单目标规划求解的方法，与加权法不同的是，它在多个目标中选取一个最重要的作为主目标，将其余目标变为约束条件，构造下述单目标规划问题：

$$
\begin{aligned}
\max \quad & f_k(x) \\
s.t. \quad & f_j(x) \ge \varepsilon_j, j = 1, 2, \cdots p, j \ne k \\
& x \in S
\end{aligned}
$$

这里 $\varepsilon_1, \varepsilon_2, \cdots \varepsilon_{k-1}, \varepsilon_{k+1}, \cdots \varepsilon_p$ 是根据目标函数性质给定的常数。

多目标规划–混合法

有时，有多目标规划问题中，虽经反复统筹比较，也无法将问题简化为单主目标函数的情况。这时，首先尽可能减少主目标函数的数目，将次目标函数变为约束条件；然后对保留的多个主目标函数用加权法求解。这种将约束法和加权法联合使用的方法叫混合法。

多目标规划–理想点法

- 对于多目标问题如果决策者事先能够对每个目标 给出一个目的值（或叫目标值），$f_i^*(\mathbf{x})$ 使其满足：$f_i^*(\mathbf{x})=\min f_i(\mathbf{x})$
- $$F^*(\mathbf{x})=(f_1^*(\mathbf{x}),f_2^*(\mathbf{x}),\ldots,f_m^*(\mathbf{x}))^T$$
- 为理想点。在目标空间中，适当地引进某种范数，将多目标函数的优化问题，转化为求解与理想点之间的“距离”最小的问题。

多目标优化–非劣解（Pareto）

非劣解前沿
常规方法；
进化算法。

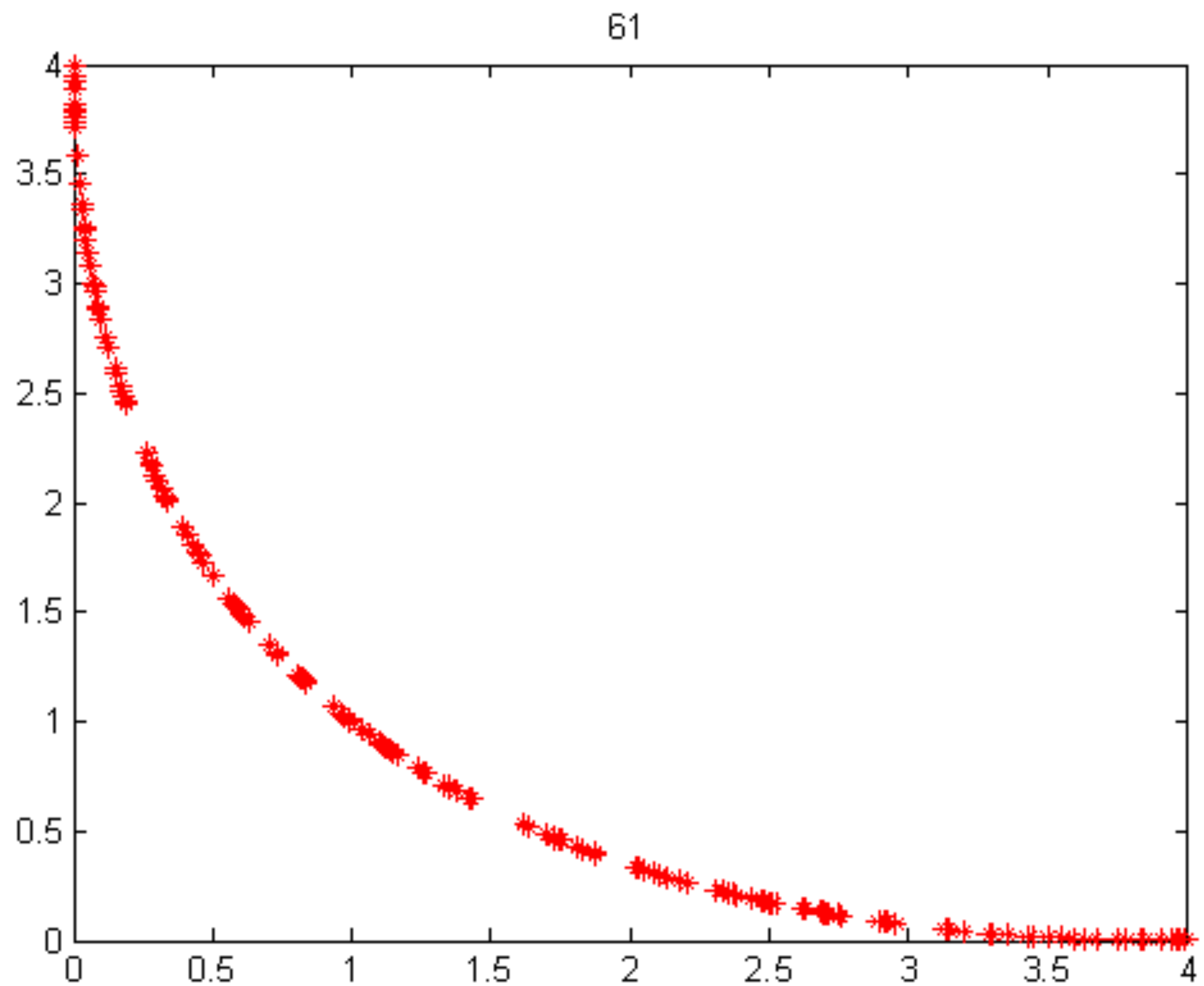
http://www.lania.mx/~ccoello/EMOO/EMOObib.html

2维多目标优化问题

$$\min \quad f_1(x) = x^2$$

$$\min \quad f_2(x) = (x-2)^2$$

$$x \in [-100, 100]$$

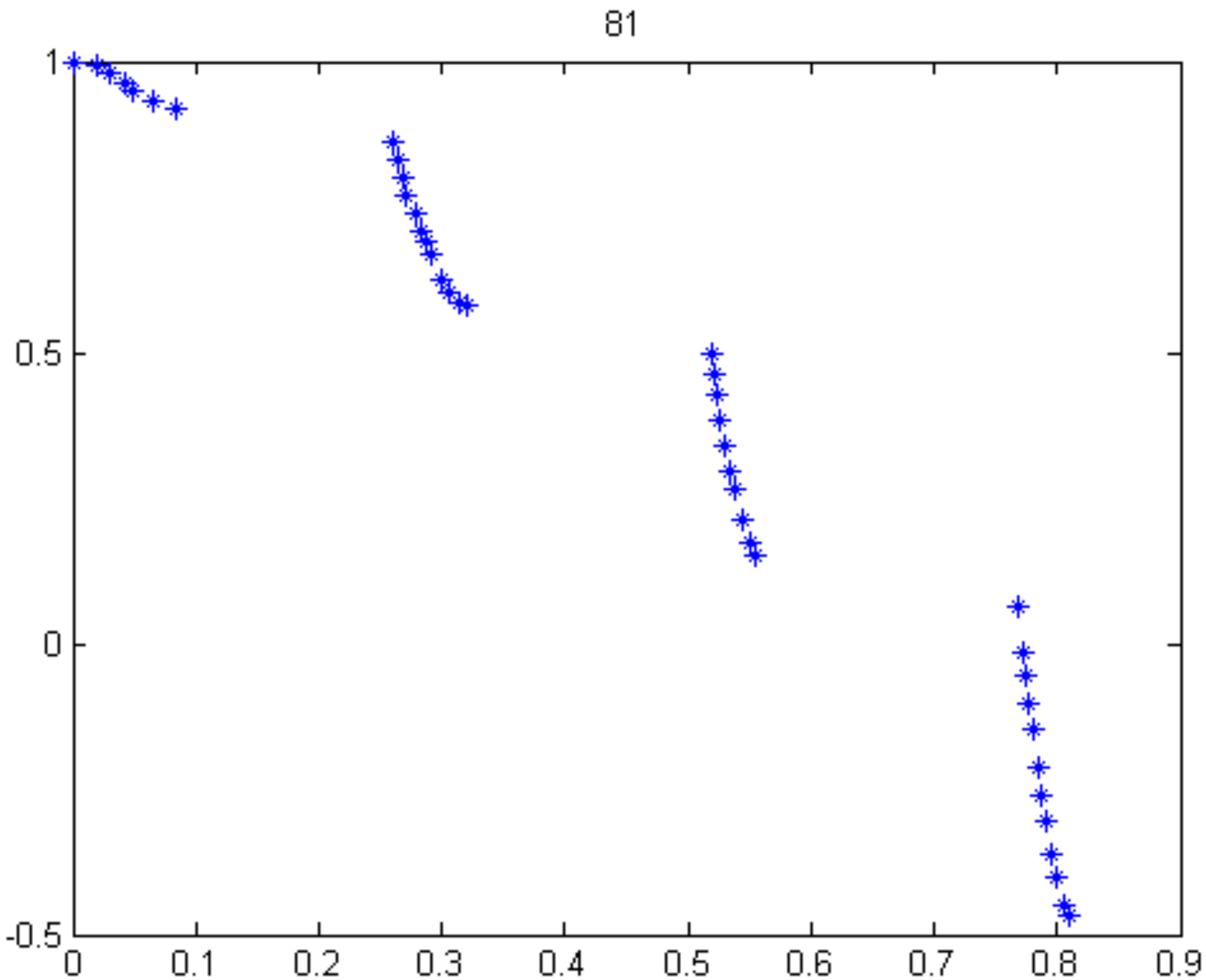


2维多目标优化问题

$$\min \quad f_1(\vec{x}) = x_1$$

$$\min \quad f_2(\vec{x}) = (1+10x_2)\left(1-\left(\frac{x_1}{1+10x_2}\right)^2 - \frac{x_1}{1+10x_2}\sin(8\pi x_1)\right)$$

$$x_1, x_2 \in [0,1]$$



控制论基础

大系统理论

大系统一般是指**规模庞大**（维数高）、**结构复杂**（多层次、多关联）、**目标众多**（目标间往往有冲突）、**时标各异**、**位置分散**，并常常具有**随机性**和**不确定性**的复杂系统。

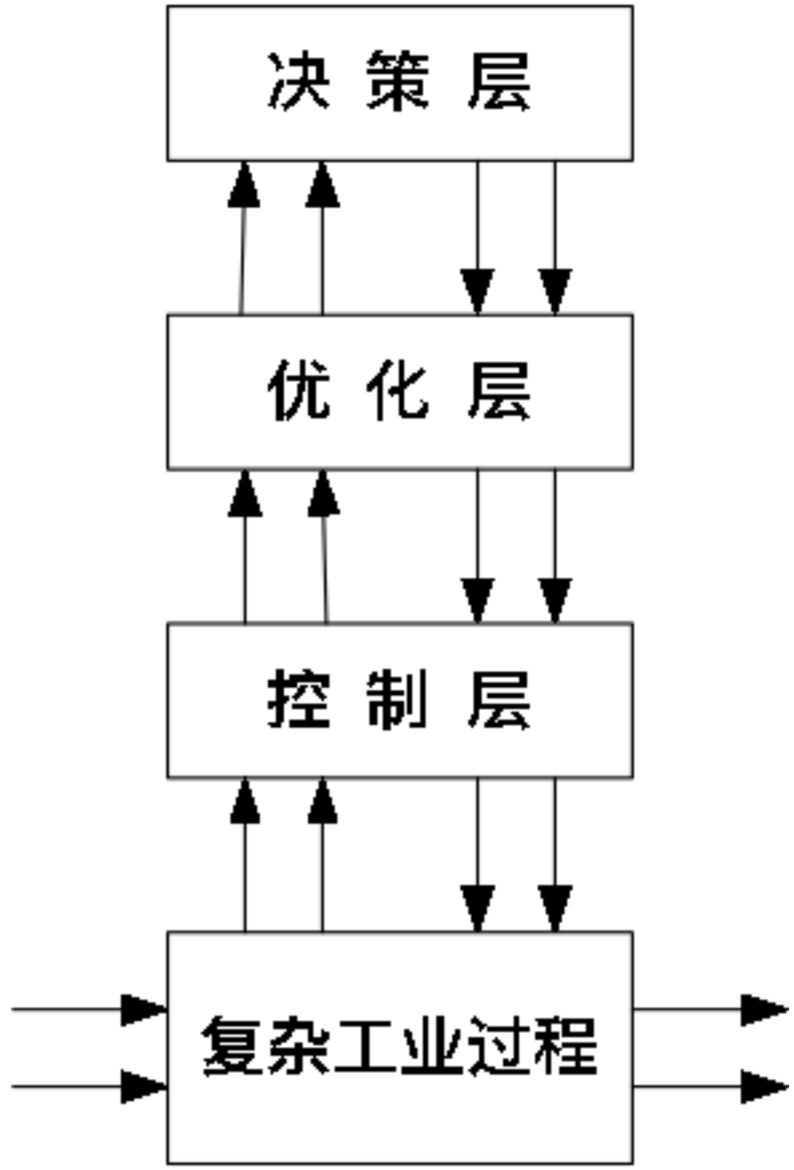
大系统的主要特征之一体现在其**结构复杂**上。大系统结构取决于组成大系统的子系统集合和各子系统之间的关联，并决定了大系统的功能，不同结构往往会产生不同的总体功能。

大系统理论

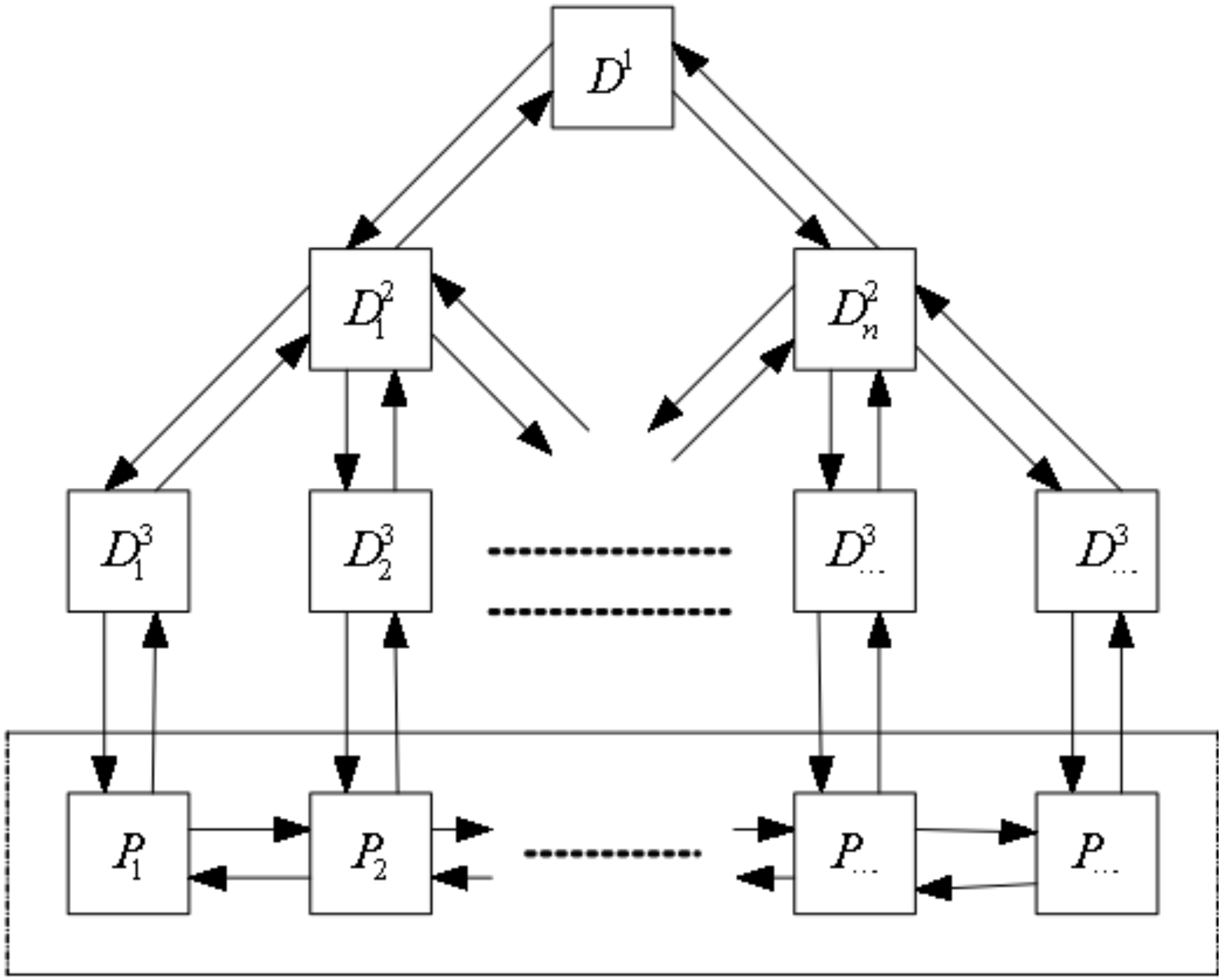
由于大系统的对象分散，变量数目众多，关联复杂，往往不宜采用集中式结构，而多为**递阶结构和分散结构**。

在递阶结构中，整个大系统分成独立平行处理的许多子系统，且用一个协调器来协调各子系统之间的关联，通过协调器（上级）和下级间往复的信息交换实现协调控制。原则上协调器可以拥有局部控制器所有的全部信息，所以递阶结构具有“经典信息模式”。从控制目的出发，递阶结构又可分为**多层递阶结构和多级递阶结构**。

多层递阶结构控制



多级递阶结构



多级递阶结构的决策单元处在不同级别，只有上下级才能交换信息，同级之间不交换信息，目标之间的冲突通过上一级协调解决。协调的最终结果应该为全局优化或近似全局优化的结果。

递阶结构

多层结构把一个复杂的决策问题进行纵向分解，按任务的复杂性分成若干子决策层，多级结构则考虑各子系统的关联，把决策问题进行横向分解，许多情况下，它们可同时存在于一个系统中。

递阶协调控制

大系统涉及许多相互关联的子系统，包含许多控制输入变量、输出变量和内部关联变量，这些变量还相互受到某些约束的限制。对于这种系统如果还是采用单一的控制器并按照常规的控制规律来控制整个系统和确定最优工作点，就会导致很大误差甚至失控，因此，常采用**大系统递阶协调控制**的方法解决问题。

递阶协调控制

递阶协调控制的基本思想是将大系统分解成若干相对独立的子系统，并构成控制系统的下层，而用上层的协调器来处理各子系统之间的关联作用。

为求解大系统递阶协调控制问题，常采用关联预测法和关联平衡法。

递阶协调控制

关联预测法是协调器预测各子系统的关联输入和输出变量，下层各决策单元按预测的关联值求解各自的决策问题，然后把达到的性能指标送给协调器，协调器再修正预测值，直到总体目标达到最优为止。由于在协调过程中引入了模型约束，也称为模型协调法。

关联平衡法在下层各决策单元求解自己的优化问题时，不考虑关联约束，协调器通过干预信息来修正各决策单元的优化目标，以保证最后关联约束得到满足，这时目标修正项的值也趋于零，达到原目标最优值。此方法也称目标协调法。

信息论基础

信息论基础

信息是系统工程的基本概念之一。

当前，信息的概念和方法已经广泛应用于通讯、物理、化学、生物、心理、经济、社会等学科中，由此而总结、归纳出的信息论也已形成并日益成熟，这些都离不开人们研究信息论之初所碰到的两个基本问题：**什么是信息和如何度量信息**。

信息论基础

信息论的创立者香农将信息定义为“两次不确定性之差”，即“不确定性的减少”。

从通信角度看，信息是数据、信号等构成的消息所载有的内容。消息是信息的“外壳”，信息是消息的“内核”。

从应用角度讲，信息是指能为人们所认识和利用的、但事先又不知道的消息、情况等，也就是说，信息对于收信者应该是有用的和未知的。

信息论基础

一旦信宿收到了信源发来的信息就消除了这种不确定性，所以通信系统中传送的正是这种能够消除不确定性的信息，从而增加了系统的确定性。然而，实际的通信过程可能完全消除了不确定性，也可能只是部分消除了不确定性，甚至完全没有消除不确定性，这取决于信息量的大小。因此将信息定义为不确定性的减少是完全合理的。

信息论基础

对某一变量的不确定性，数学上往往用概率加以描述。令 p 为某消息发生的概率，I 为该消息发生时可能携带的信息量，称为自信息，则如果对该消息本身内容没有更多的了解，该消息携带的信息量可以由其发生的概率决定，即信息量 I 是概率 p 的某个函数：

$$I = f(p)$$

并且概率大者信息量小，概率小者信息量大。

信息论基础

信息是消息发生的意外程度的度量，必然发生的事件（概率为 1）其信息量为零。另一方面，若 A和 B是两个独立发生的可能消息，则从物理意义上讲，A与 B同时发生的消息（称为联合消息 AB）的自信息满足

$$I(AB) = I(A) + I(B)$$

因此，单个可能消息的信息量（自信息）定义为：

$$I = \log_2(1/p) = -\log_2 p$$

信息量 I 的单位为比特。

信息论基础

信息交换中面对的往往不是单个消息，而是可能消息的集合，其中较为简单的情形是集合中包含有限个可能消息。针对这一情形，重要的是了解可能消息集合的整体信息能力，这需要用整体平均信息量 H来表示，又称为信息熵。设整个信源的各状态 $x_1, x_2, \cdots x_n$ 的发生概率分别为 $p_1, p_2, \cdots p_n$，则信息熵定义为：

$$H = -\sum_{i=1}^{n} p_i \log_2 p_i \qquad \sum_{i=1}^{n} p_i = 1$$

信息论基础

信息熵是信息论中的奠基性概念，是系统不确定性的定量化表征。一个系统不确定性越大，则系统越无序，信息熵越大。信息熵具有以下性质：

（1）**对称性** $H(p_1, p_2) = H(p_2, p_1)$

即信息熵关于变元是对称的，所有变元可以互换，不影响熵函数的值。

（2）**非负性**

$$H(p_1, p_2, \cdots p_n) = -\sum_{i=1}^{n} p_i \log_2 p_i \geq 0$$

信息论基础

(3) 确定性

$$H(1,0) = H(1,0,0) = \cdots = H(1,0,\cdots 0) = 0$$

即只要有确定性事件，信息熵为 0

(4) 最大熵原理

$$H(p_1, p_2, \cdots p_n) \le \log_2 n$$

其中等号成立，当且仅当 $p_i = \frac{1}{n}, i = 1, 2, \cdots n$ 。这表明，等概率场具有最大熵。

其他的理论

- 对策论（博弈论）
- 排队论

排队论

排队论(Queuing Theory)，是研究系统随机聚散现象和随机服务系统工作过程的数学理论和方法，又称随机服务系统理论。是通过对服务对象到来及服务时间的统计研究，得出这些数量指标（等待时间、排队长度、忙期长短等）的统计规律，然后根据这些规律来改进服务系统的结构或重新组织被服务对象，使得服务系统既能满足服务对象的需要，又能使机构的费用最经济或某些指标最优。

排队论

- **排队系统**由输入过程与到达规则、排队规则、服务机构的结构、服务时间与服务规划组成。
- 一般还假设到达间隔时间序列与服务时间均为独立同分布随机变量序列，且这两个序列也相互独立。

排队论

- 评价一个**排队系统**的好坏要以顾客与服务机构两方面的利益为标准。就顾客来说总希望等待时间或逗留时间越短越好，从而希望服务台个数尽可能多些但是，就服务机构来说，增加服务台数，就意味着增加投资，增加多了会造成浪费，增加少了要引起顾客的抱怨甚至失去顾客，增加多少比较好呢？顾客与服务机构为了照顾自己的利益对**排队系统中的3个指标**：队长、等待时间、服务台的忙期（简称忙期）都很关心。因此这3个指标也就成了排队论的主要研究内容。

博弈论(GAME THEORY)

- **博弈论考虑游戏中的个体的预测行为和实际行为，并研究它们的优化策略。**

博弈论

- 博弈论主要研究公式化了的**激励结构间**的相互作用。是研究具有**斗争或竞争性质现象**的数学理论和方法。博弈论考虑游戏中的个体的预测行为和实际行为，并研究它们的优化策略。生物学家使用博弈理论来理解和预测进化论的某些结果。
- 博弈论已经成为经济学的标准分析工具之一。在生物学、经济学、国际关系等很多学科有广泛的应用。

囚徒困境

- 假设有两个小偷A和B联合犯事、私入民宅被警察抓住。警方将两人分别置于不同的两个房间内进行审讯，对每一个犯罪嫌疑人，警方给出的政策是：如果两个犯罪嫌疑人都坦白了罪行，交出了赃物，于是证据确凿，两人都被判有罪，各被判刑8年；如果只有一个犯罪嫌疑人坦白，另一个人没有坦白而是抵赖，则以妨碍公务罪再加刑2年，而坦白者有功被减刑8年，立即释放。如果两人都抵赖，则警方因证据不足不能判两人的偷窃罪，但可以私入民宅的罪名将两人各判入狱1年。

囚徒困境

- 对A来说，尽管他不知道B作何选择，但他知道无论B选择什么，他选择“坦白”总是最优的。显然，根据对称性，B也会选择“坦白”，结果是两人都被判刑8年。但是，倘若他们都选择“抵赖”，每人只被判刑1年。在四种行动选择组合中，（抵赖、抵赖）是帕累托最优的，因为偏离这个行动选择组合的任何其他行动选择组合都至少会使一个人的境况变差。不难看出，“坦白”是任一犯罪嫌疑人的占优战略，而（坦白，坦白）是一个占优战略均衡。

A＼B	坦白	抵赖
坦白	8，8	0，10
抵赖	10，0	1，1

三个火枪手

- **说的是有三个心怀不共戴天之恨的枪手，某天在街头遭遇，三人同时拔枪开火，由此产生一场博弈学有名的论证。假设枪手甲的命中率为80%，枪手乙的命中率为60%，枪手丙的命中率为40%，三人都是非常理性的人。问题是第一阵枪响之后，谁最有可能活下来？谁最有可能死掉？**

三个火枪手

- **两种情况：**
  - **同时开枪；**
  - **每人轮流开枪。**

测试

- 每个人从1-100间任选一个数；
- 计算全班选择数据的平均值；
- 计算该平均值的2/3--A；
- 与A值误差最小的同学获胜。

- 全班30人，成绩分3档:5-3-1
- 与A误差最小的10人：5分；
- 与A误差最大的10人：1分。