高级人工智能
Advanced Artificial Intelligence

汤云波
yunbotang@fzu.edu.cn
福州大学计算机与大数据学院

- **先修课程：**

  高级语言程序设计、数据结构、算法设计与分析，
  高等数学、线性代数、统计学

- **后续课程：**

  深度学习，大数据分析与挖掘，毕业设计

- **建议教材**

  书名：《计算智能》书号：ISBN 9787115534767 作者：毕晓君 出版社：人民邮电出版社出版时间：2020-06

- **考核方式**

考核环节	权重（%）
期末考试	70
点名/平时作业	30

- **课程特点**

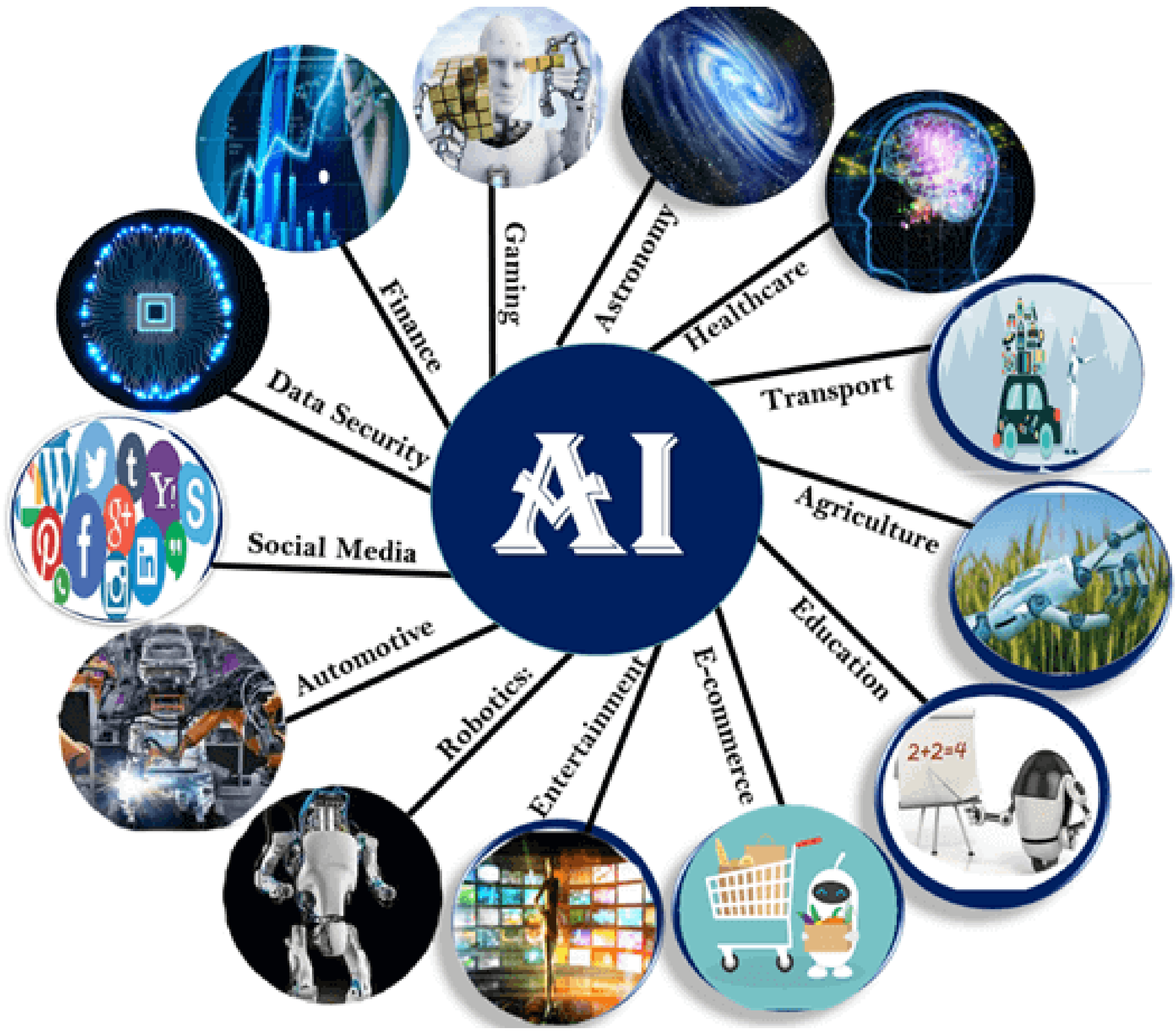
理论性、抽象性较强，实用价值高

知识回顾

人工智能的应用

1. 计算机视觉
2. 自然语言处理
3. 语音识别
4. 泛数据挖掘

等等



知识回顾

什么是人工智能？

- 人工智能（Artificial Intelligence），是研究、开发用于模拟、延伸和扩展人的智能的理论、方法、技术及应用系统的一门新的技术科学。

- 人工智能是计算机科学的一个分支，研究如何将人的智能转化为机器智能，或者是用机器来模拟或实现人的智能。

- 人工智能是包含广泛、极富挑战性的科学
  - 思维基础：人们长期以来探索研制能够进行计算、推理和其它思维活动的智能机器的必然结果
  - 理论基础：信息论、控制论、系统工程论、计算机科学、心理学、神经学、认知科学、数学和哲学等多学科相互渗透的结果
  - 技术基础：电子计算机和电子技术得到广泛应用的结果

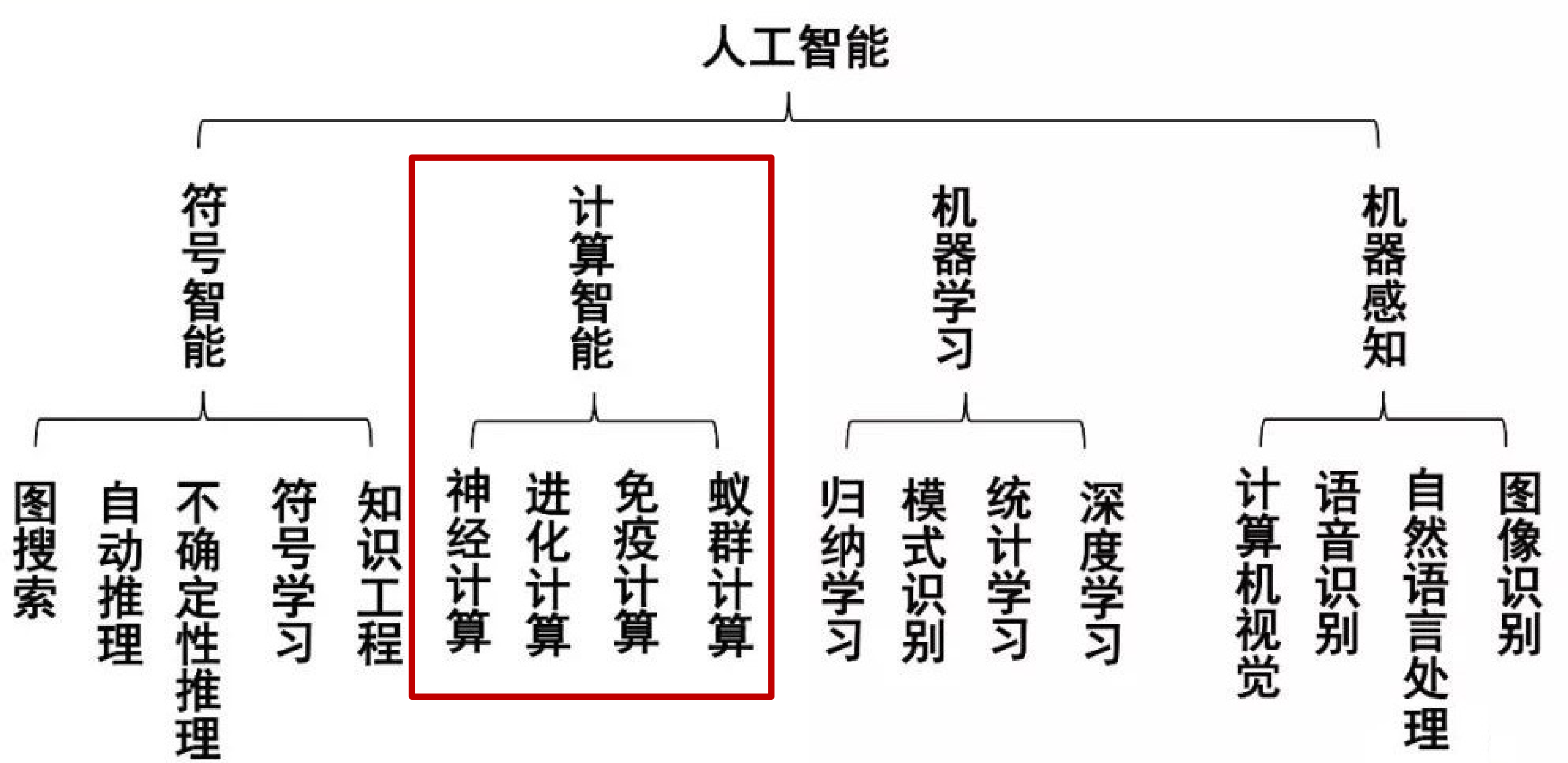
知识回顾

人工智能的目标

- **近期目标：**研究如何使计算机去做那些靠人的智力才能做的工作
- **最终目标：**探讨智能形成的基本机理，研究利用自动机模拟人的思维过程

知识回顾

人工智能的研究领域

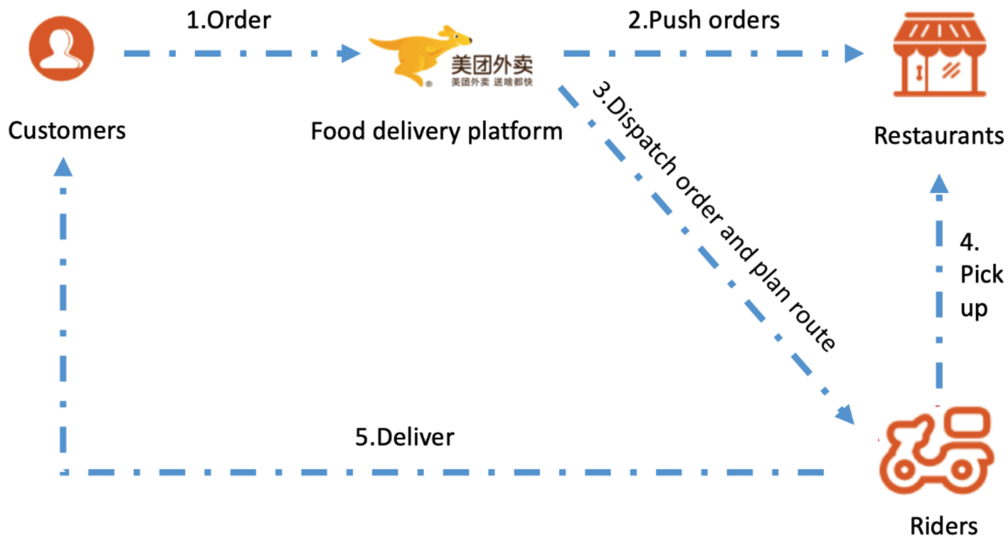


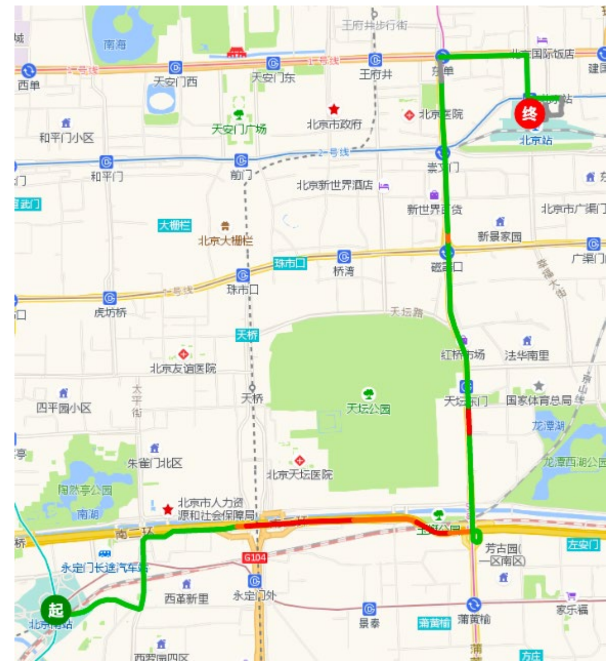
第一章 计算智能方法概述

计算智能概述

- 生物群体的演化、社会集群现象的规律所产生的智能优化算法
- 理论上不如传统优化方法（一阶、二阶梯度）完善, 但全局优化解获得的可能性较大
- 对目标函数和约束条件不要求连续性、可微、凸性等，甚至可以无解析式
- 对数据的不确定性有较强适应性
- 应用于信息处理、调度优化、工程控制、经济管理等

最优化问题典型案例


1.Order
2.Push orders
3.Dispatch order and plan route
4.
Pick
up
5.Deliver
Customers
美团外卖
Food delivery platform
Restaurants
Riders



最优化问题的数学模型

- 在一定的环境下，根据某种判定条件寻找最佳方案，以获得最大实际价值（最小风险损失）

$$\min_{x\in R^n} F(x)=[f_1(x), f_2(x),\cdots, f_h(x)]$$

$$s.t.\begin{cases} g_i(x)\ge 0, i=1,2,\cdots,m \\ h_j(x)=0, j=1,2,\cdots,p \end{cases}$$

- 变量的定义域是离散的，即为离散优化问题；变量的定义域是连续的，即为连续优化问题

最优化问题的分类

- **按目标的数量：单目标和多目标**

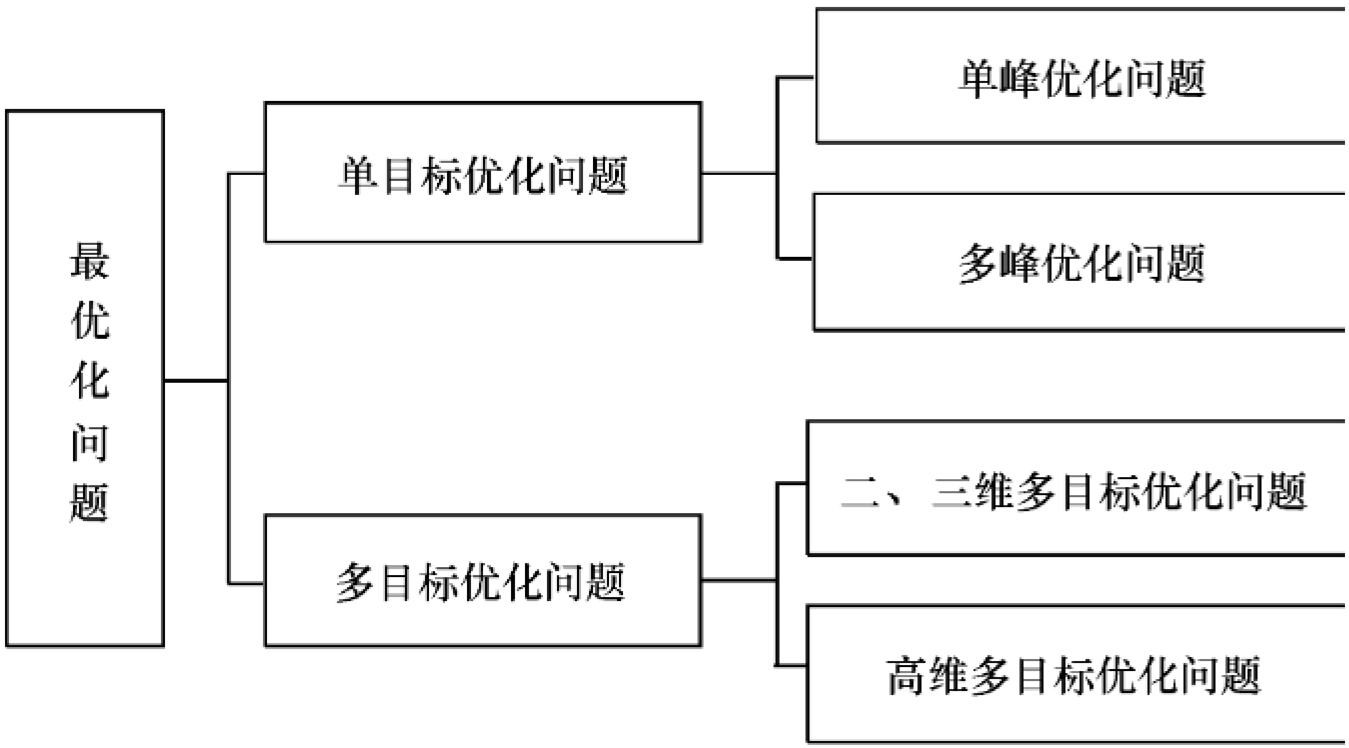


图 1.1　根据目标数量进行的最优化问题分类

最优化问题的分类

- 按变量是否连续：离散变量优化问题和连续变量优化问题
- 在离散变量优化问题中最常见的是组合优化问题
- 在连续变量优化问题最常见的是函数优化问题
- 例：九宫重排，最优排课，旅行推销商，最佳路由等为离散（优化）问题；模式识别（如图像识别、语音识别）和工程结构优化设计等为连续（优化）问题

最优化问题的分类

- 按目标函数或变量是否受到约束：无约束优化问题和约束优化问题
- 约束条件会使得可行解空间变得不连贯，一般较无约束问题求解难度大

一个图像去噪问题的最优化表达

- 目标J(A)=(A的主体与原图尽量接近)+α·(A尽可能光滑)
- 求目标的极小值（有时有约束）

- 合理的目标形式设计（建模）
- 求目标极小值的方法（优化）

机器学习与计算智能（最优化）之间的联系

- 建立了许多学习模型模拟人的学习过程，还能完成人难以完成的大数据信息处理工作；但实际上大部分只是样本学习
- 典型：任务（降维、回归、分类、聚类等），模型（主成分分析、k均值聚类、支持向量机、决策树、隐马尔科夫链等）
- 结构各异，一般都具有大量待定参数。通过学习，模型能实现输入数据输出抽象后的结果（类别、估值等）
- 需要提供大量样本供模型学习，才能逐步调整获得模型最终的参数；
- 模型学习设定的最优化目标函数（以分类问题为例）：模型的分类错误率尽量小

组合优化问题的局部搜索方法介绍

算法的时间复杂度

- 当组合优化问题的规模较小时，总能通过枚举法获得问题的最优解；但规模较大时，可能产生组合爆炸，难以找到全局最优。
- 一些典型的组合优化问题：TSP问题、背包问题、装箱问题……
- 常见的算法复杂度函数

$$O(\log n), O(n), O(n\log n), O(n^2),$$

$$O(2^n), O(n^{\log n}), O(n!), O(n^n)$$

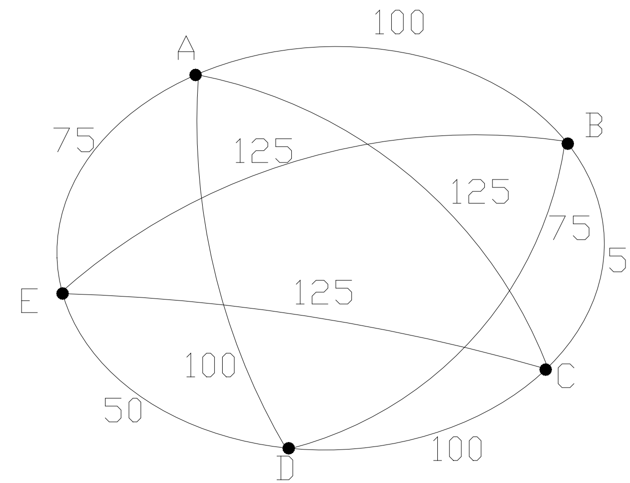
时间复杂性函数比较（10亿次/秒）

复杂度 \ n	10	20	30	40	100
n	10ns	20ns	30ns	40ns	100ns
$n\log n$	10ns	26.0ns	44.3ns	64.1ns	200ns
n^2	100ns	400ns	900ns	1.6μs	10 μ s
2^n	1.0 μ s	1.0ms	1.1s	18.3min	4.0世纪
$n!$	3.6ms	77.1年	8.4×10^{13} 世纪	2.6×10^{29} 世纪	3.0×10^{139} 世纪

$$1ns(\text{纳秒}) = 10^{-9}s; 1\mu s(\text{微秒}) = 10^{-6}s; 1ms(\text{毫秒}) = 10^{-3}s$$

TSP问题

Traveling Salesman Problem



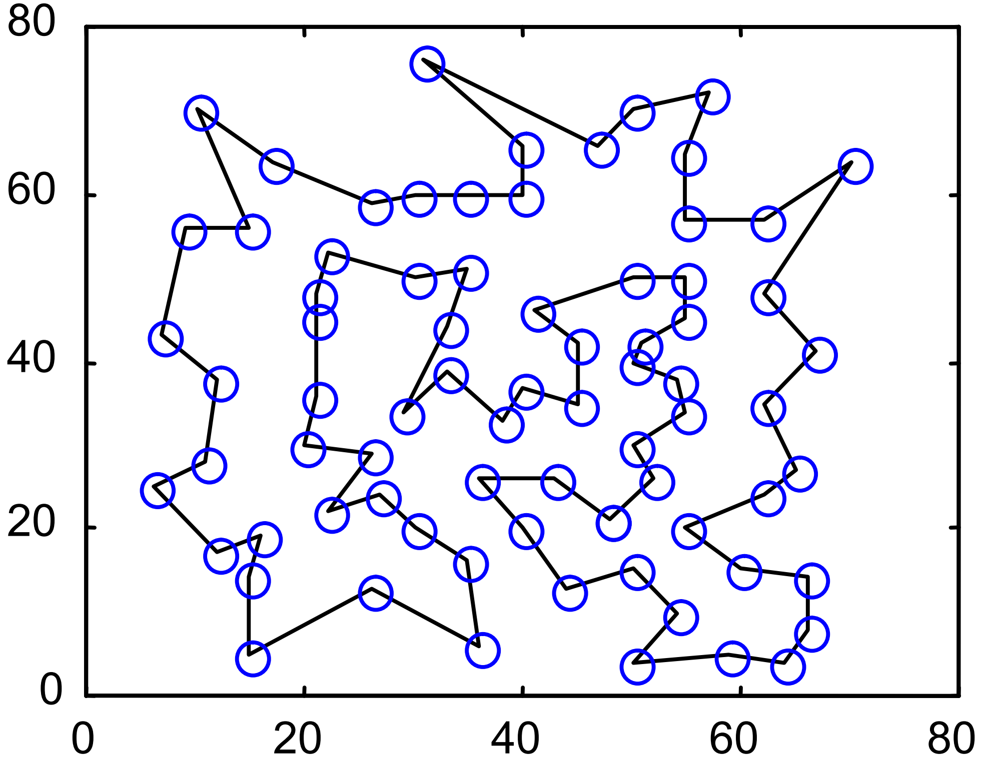
从某城市出发遍历所有n个城市一遍且仅一遍再回到出发地，求最短路径

- 没有通用的求解方法
- NP－hard(Karp,1972)
- 有效的启发式算法（1973）
- 完全多项式近似方案（1998）
- ……

n个城市，
路线的可能性(n-1)!；
O(n!);

寻求在可接受的时间内得到满意解的方法

组合爆炸问题—TSP为例



N个城市，路线的可能性（N-1）！

N＝21，小规模问题

（N-1）！>10^{18}

PC，1亿（10^8）次/秒！

花费 10^{10} 秒

1天86400秒（< 10^5 秒）

需要10^5 天，约300年！

邻域的概念

- 邻域:一个点附近其它点的集合
- 在距离空间，邻域就是以某一点为中心的距离域
- 组合优化问题的邻域定义：设D是问题的定义域，若存在一个映射N，使得：

$$N: S \in D \rightarrow N(S) \in 2^{D}$$

某种序或关系

则称$N(S)$为S的邻域。

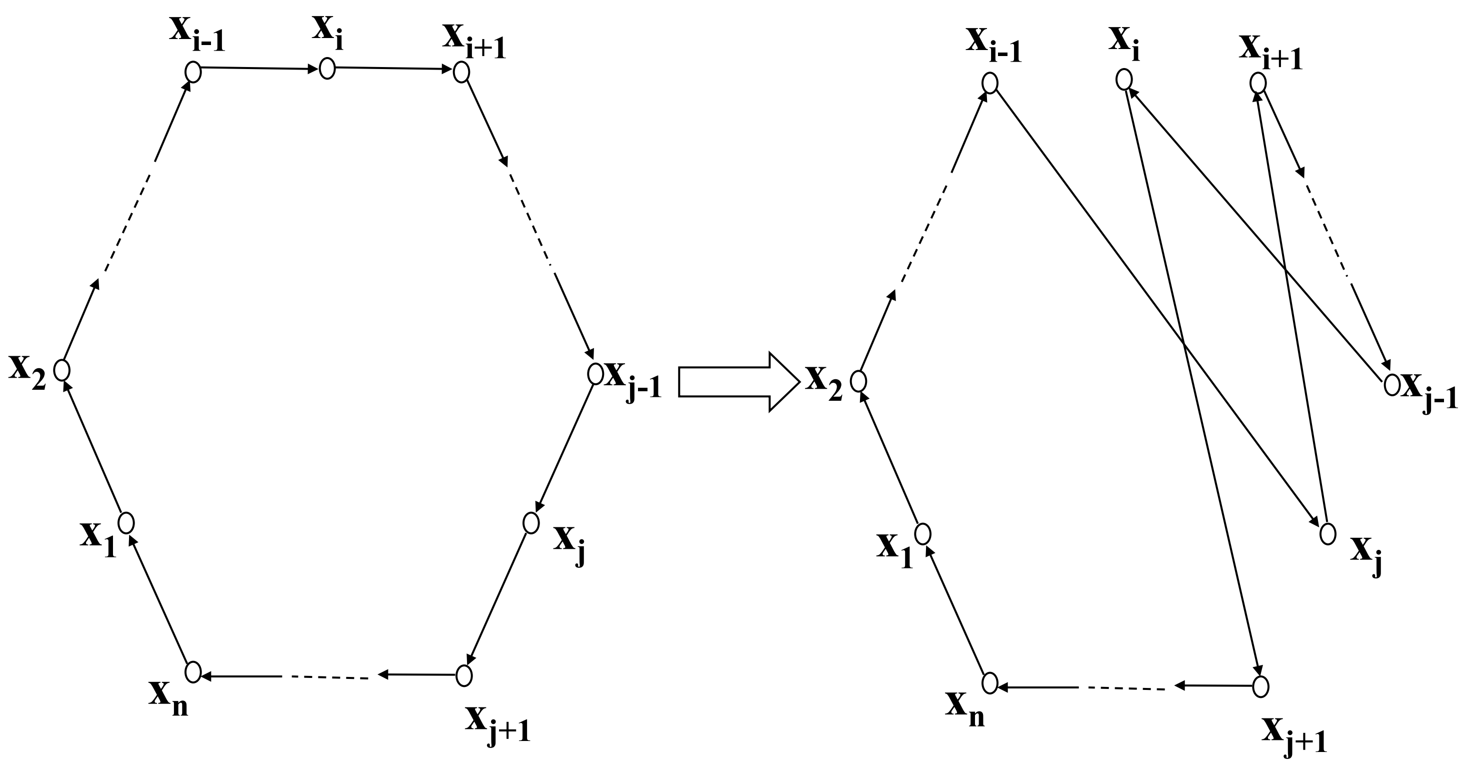
2^{D}：D的幂集，所有子集构成

TSP问题的邻域定义示例

- 可能解S：城市序列; 邻居S’：交换两城市位置获取
- 例1：简单交换方法

$$S=(x_1,x_2,\cdots,x_{i-1},x_i,x_{i+1},\cdots,x_{j-1},x_j,x_{j+1},\cdots,x_n)$$

$$S'=(x_1,x_2,\cdots,x_{i-1},\boxed{x_j},x_{i+1},\cdots,x_{j-1},\boxed{x_i},x_{j+1},\cdots,x_n)$$

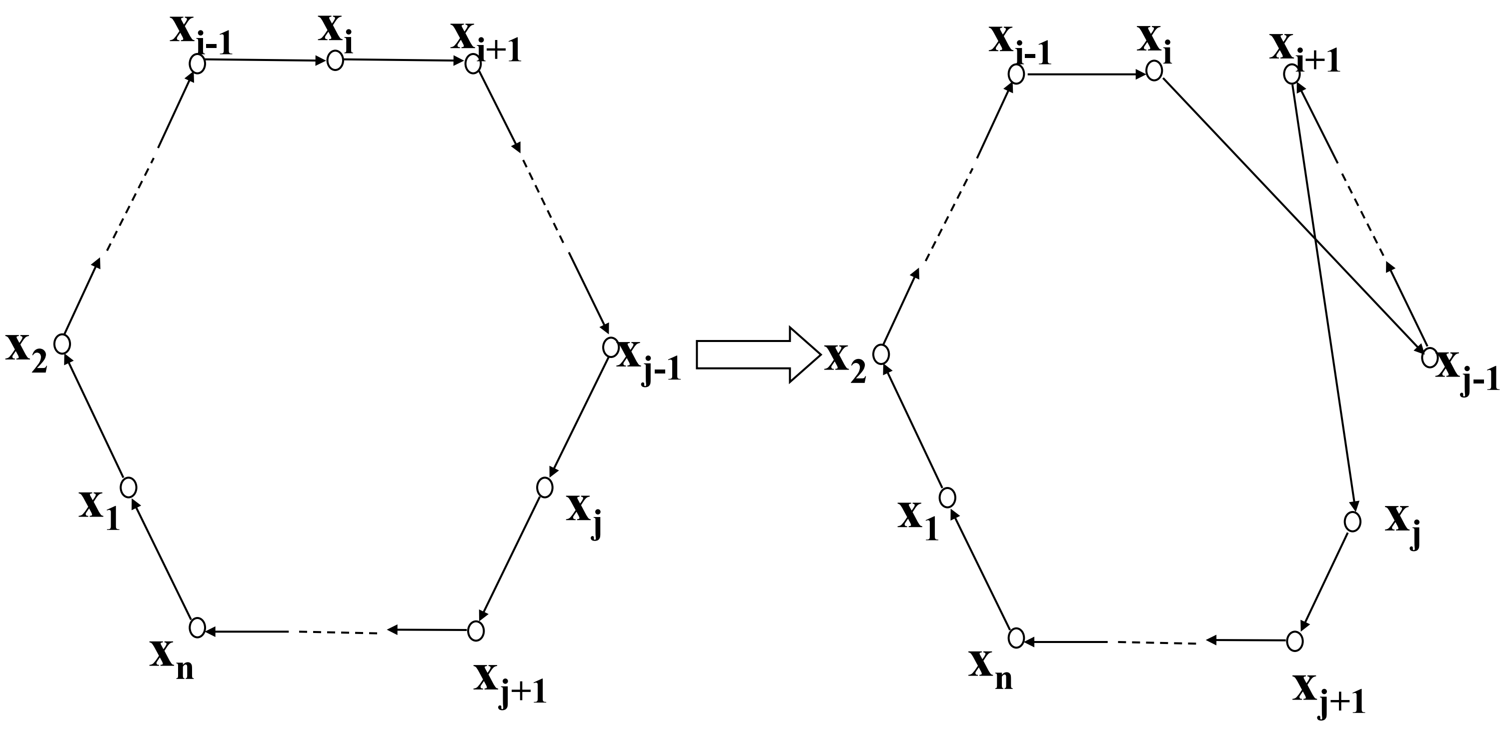


例2：逆序交换方法

邻居S’：选取两城市x_i、x_j，逆转x_i、x_j之间的城市次序

设：$S = (x_1, x_2, \ldots x_{i-1}, x_i, x_{i+1}, \ldots, x_{j-1}, x_j, x_{j+1}, \ldots, x_n)$

则：$S' = (x_1, x_2, \ldots x_{i-1}, x_i, x_{j-1}, x_{j-2} \ldots, x_{i+1}, x_j, x_{j+1}, \ldots, x_n)$



局部搜索算法

- 基本思想：在当前点的邻域中搜索，且始终向着离目标最接近的方向搜索。
- 目标可以是最大值，也可以是最小值。
- 若无特殊说明，均假定是最小值。

局部搜索算法（Local Search）

1）随机选择一个初始可能解 $x_0 \in D$，$x_b \leftarrow x_0$，

$P=N(x_b)$；定义指标函数 $f(x)$

2）若不满足结束条件，则

Begin

选择P的一个子集P'，x_n为P'中的最优解

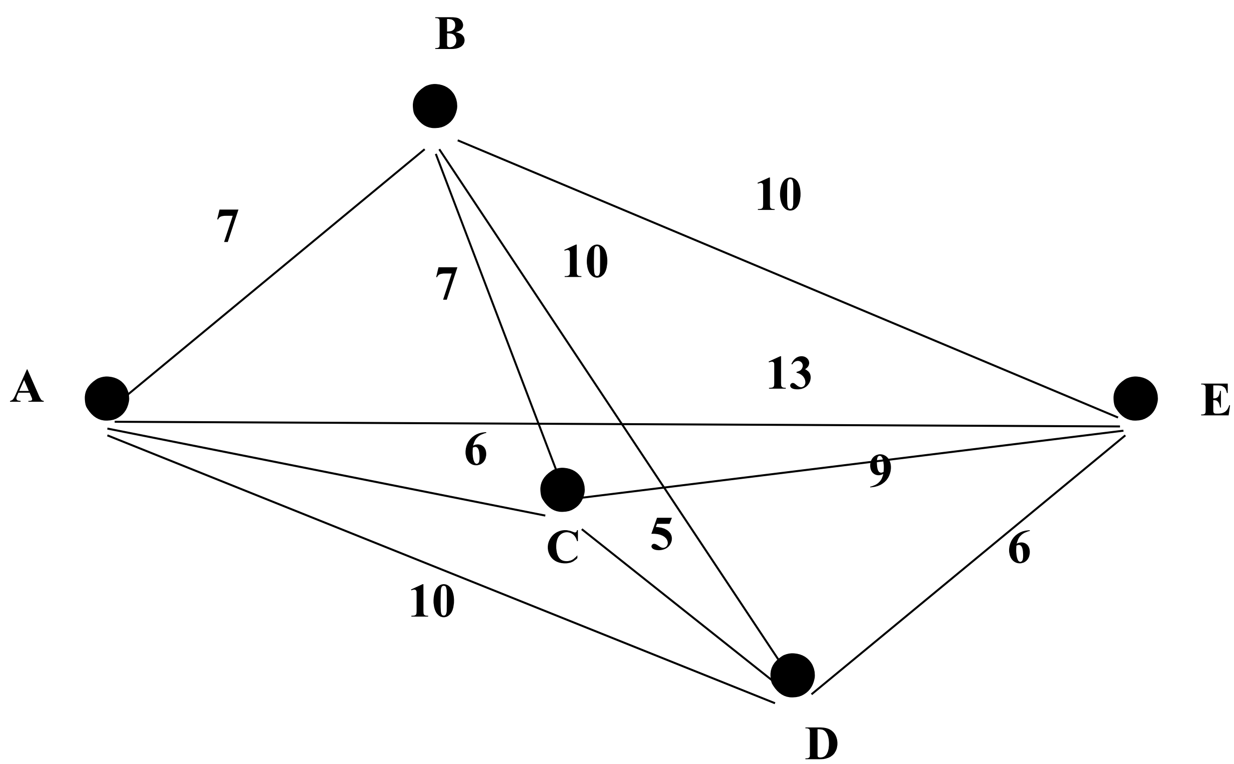
如果$f(x_n) < f(x_b)$，则$x_b \leftarrow x_n$，$P \leftarrow N(x_b)$，转2)；

否则$P = P - P'$，转2)。

End

3）输出计算结果，结束

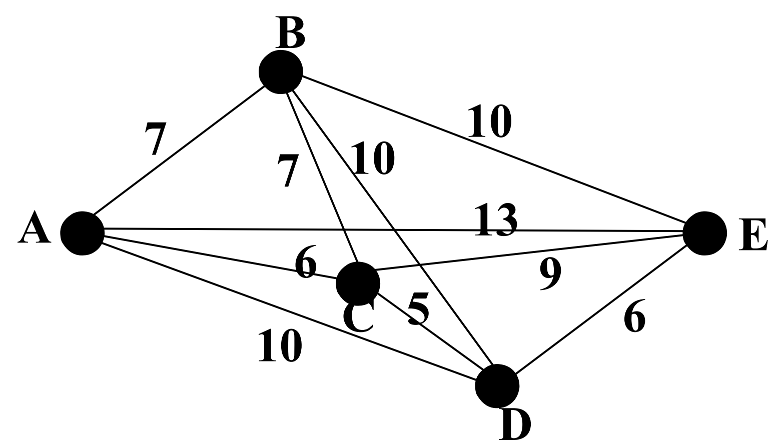
例：5城市旅行商问题



初始可能解：$x_0 = (a, b, c, d, e)$
指标函数：路径长度，$f(x_b) = f(x_0) = 38$
通过交换两个城市获得邻域

$P = \{(a, c, b, d, e), (a, d, c, b, e), (a, e, c, d, b),$
$(a, b, d, c, e), (a, b, e, d, c), (a, b, c, e, d)\}$

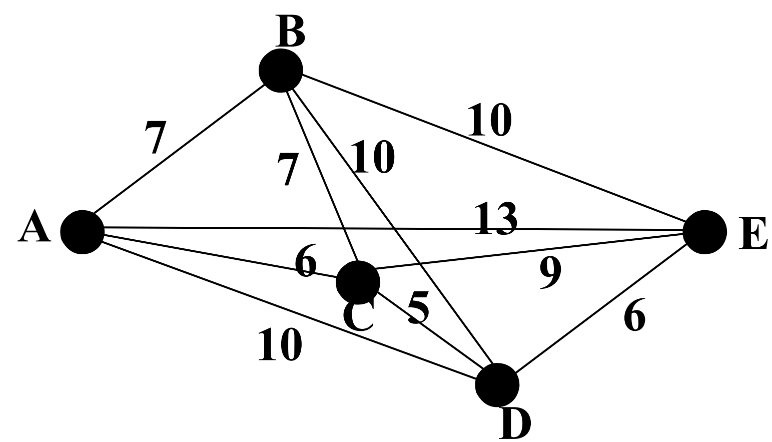
设每次随机从P中选择一个邻居。



第一次循环

P = {(a, c, b, d, e), (a, d, c, b, e), (a, e, c, d, b), (a, b, d, c, e), (a, b, e, d, c), (a, b, c, e, d)}

从P中选择一个元素，不妨为x_n = (a, c, b, d, e),



$f(x_n) = 42$,

$f(x_n) > f(x_b)$,

$P = P - \{x_n\}$

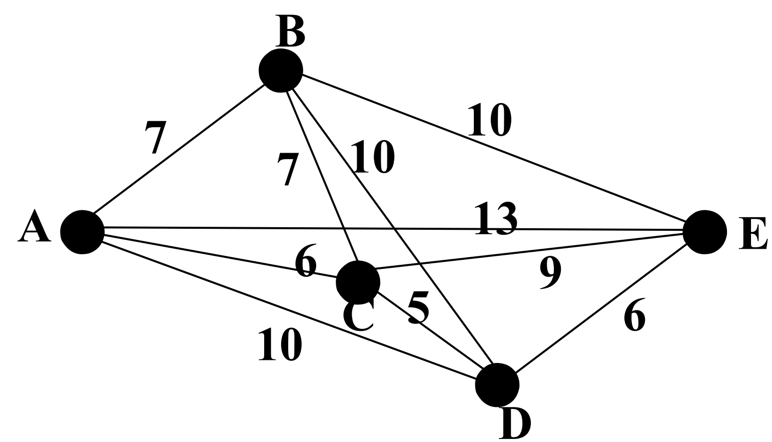
= {(a, d, c, b, e), (a, e, c, d, b), (a, b, d, c, e), (a, b, e, d, c), (a, b, c, e, d)}

第二次循环

$P = \{(a, d, c, b, e), (a, e, c, d, b), (a, b, d, c, e),$

$(a, b, e, d, c), (a, b, c, e, d)\}$

从P中选择一个元素，不妨为$x_n = (a, d, c, b, e)$,



$f(x_n) = 45$,

$f(x_n) > f(x_b)$,

$P = P - \{x_n\}$

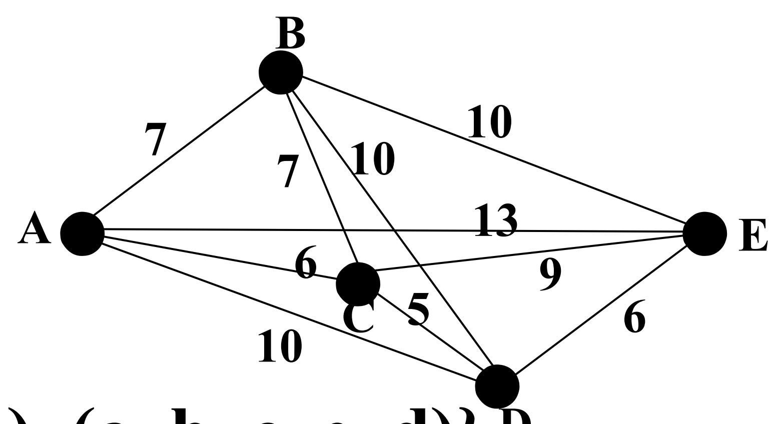
$= \{(a, e, c, d, b), (a, b, d, c, e), (a, b, e, d, c),$

$(a, b, c, e, d)\}$

第三次循环

P ={(a, e, c, d, b), (a, b, d, c, e), (a, b, e, d, c),
(a, b, c, e, d)}

从P中选择一个元素，不妨为x_n = (a, e, c, d, b),



$f(x_n) = 44$,

$f(x_n) > f(x_b)$,

$P = P - \{x_n\}$

= {(a, b, d, c, e), (a, b, e, d, c), (a, b, c, e, d)}

第四次循环

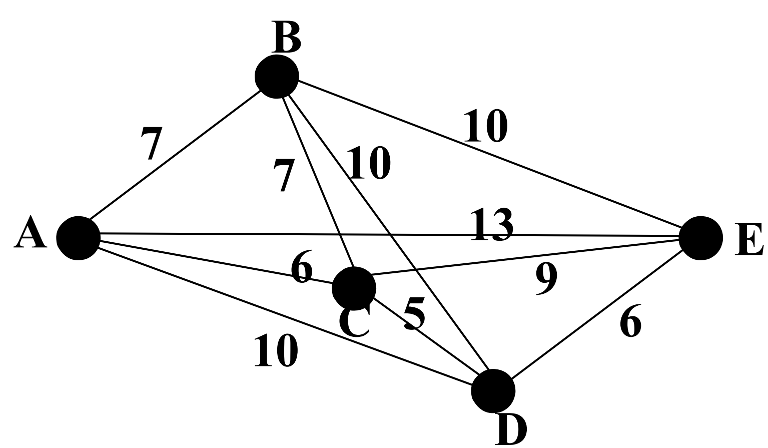
P ={(a, b, d, c, e), (a, b, e, d, c), (a, b, c, e, d)}

从P中选择一个元素，不妨为x_n = (a, b, d, c, e),

$f(x_n) = 44$,

$f(x_n) > f(x_b)$,

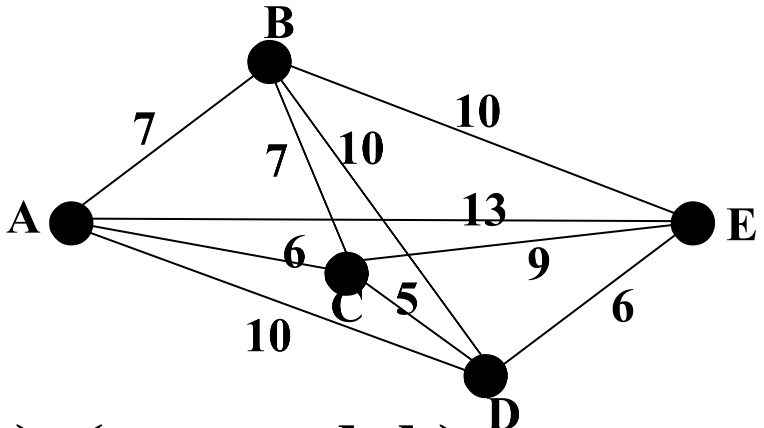
$P = P - \{x_n\} = \{(a, b, e, d, c), (a, b, c, e, d)\}$



第五次循环

$P = \{(a, b, e, d, c), (a, b, c, e, d)\}$

从P中选择一个元素，不妨为$x_n = (a, b, e, d, c)$,



$f(x_n) = 34$,

$f(x_n) < f(x_b)$,

$x_b = x_n = (a, b, e, d, c)$,

$P = \{(a, e, b, d, c), (a, d, e, b, c), (a, c, e, d, b),$

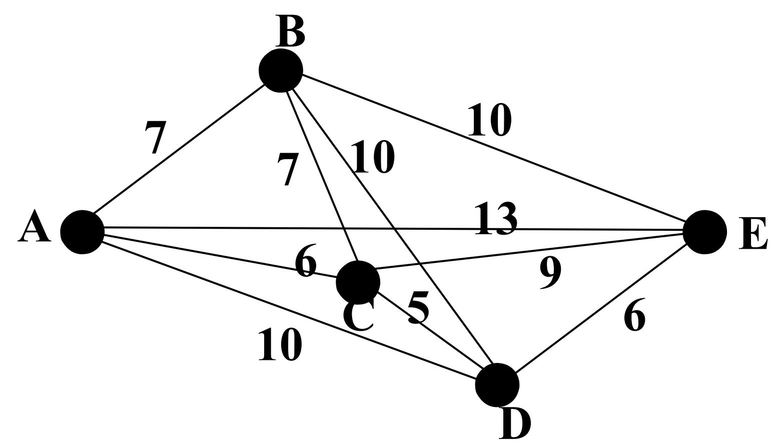
$(a, b, d, e, c), (a, b, c, d, e), (a, b, e, c, d)\}$

第六次循环

$P = \{(a, e, b, d, c), (a, d, e, b, c), (a, c, e, d, b),$

$(a, b, d, e, c), (a, b, c, d, e), (a, b, e, c, d)\}$

从P中选择一个元素，不妨为$x_n = (a, e, b, d, c)$,



$f(x_n) = 44$,

$f(x_n) > f(x_b)$,

$P = P - \{x_n\}$

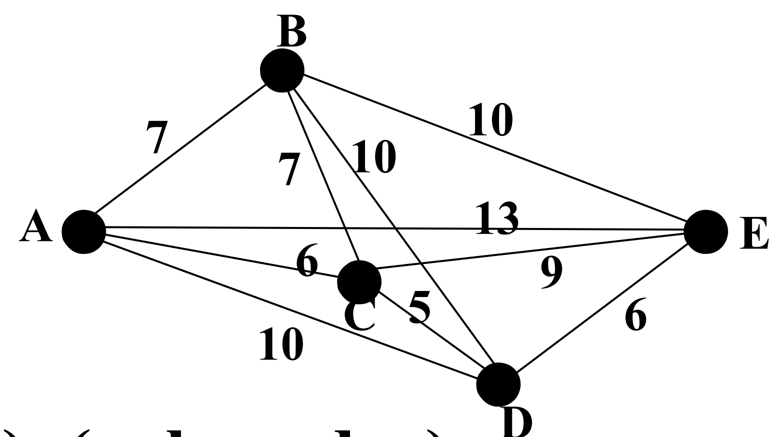
$= \{(a, d, e, b, c), (a, c, e, d, b), (a, b, d, e, c),$

$(a, b, c, d, e), (a, b, e, c, d)\}$

第七次循环

P = {(a, d, e, b, c), (a, c, e, d, b), (a, b, d, e, c),
(a, b, c, d, e), (a, b, e, c, d)}

从P中选择一个元素，不妨为x_n = (a, d, e, b, c),



$f(x_n) = 39$, $f(x_n) > f(x_b)$,

$P = P - \{x_n\}$

= {(a, c, e, d, b), (a, b, d, e, c), (a, b, c, d, e),
(a, b, e, c, d)}

第八次循环

P ={(a, c, e, d, b), (a, b, d, e, c), (a, b, c, d, e),
(a, b, e, c, d)}

从P中选择一个元素，不妨为x_n = (a, c, e, d, b),

$f(x_n) = 38$,

$f(x_n) > f(x_b)$,

$P = P - \{x_n\}$

= {(a, b, d, e, c), (a, b, c, d, e), (a, b, e, c, d)}

第九次循环

$P = \{(a, b, d, e, c), (a, b, c, d, e), (a, b, e, c, d)\}$

从P中选择一个元素，不妨为$x_n = (a, b, d, e, c)$,

$f(x_n) = 38$,

$f(x_n) > f(x_b)$,

$P = P - \{x_n\} = \{(a, b, c, d, e), (a, b, e, c, d)\}$

第十次循环

$P = \{(a, b, c, d, e), (a, b, e, c, d)\}$

从P中选择一个元素，不妨为$x_n = (a, b, c, d, e)$,

$f(x_n) = 38$,

$f(x_n) > f(x_b)$,

$P = P - \{x_n\} = \{(a, b, e, c, d)\}$

第十一次循环

$P = \{(a, b, e, c, d)\}$

从P中选择一个元素，不妨为$x_n = (a, b, e, c, d)$,

$f(x_n) = 41$,

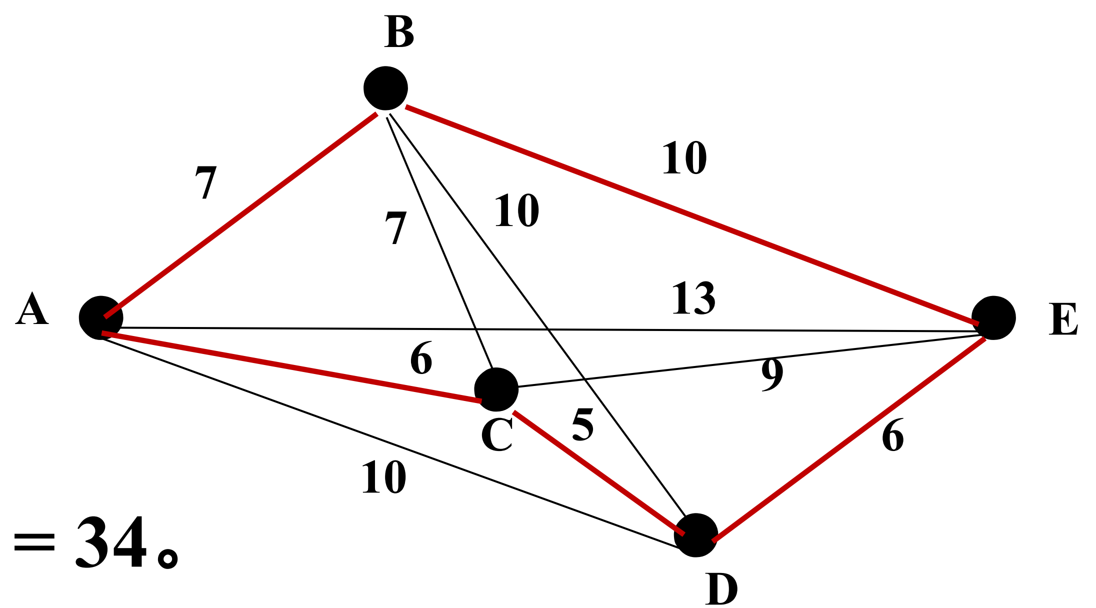
$f(x_n) > f(x_b)$,

$P = P - \{x_n\} = \varnothing$

P等于空，算法结束，

得到结果为

$x_b = (a, b, e, d, c)$, $f(x_b) = 34$。



局部搜索结果恰好为问题的全局最优解；

一般情况下，未必得到全局最优解。

连续优化问题复习回顾

无约束最优化方法回顾

问题： $\min_{x\in R^n} f(x)$

1）$f\in C^1$，若x^*为$f(x)$的极小点，则有：$\nabla f(x^*)=0$ （必要条件）

2）满足$\nabla f(x)=0$的点x，称为稳定点

$$\text{稳定点可能是}\begin{cases}\text{极大值点}\\ \text{极小值点}\\ \text{两者皆不是}\end{cases}$$

$$\boxed{\nabla f(x)=\left(\frac{\partial f(x)}{\partial x_1}\quad \frac{\partial f(x)}{\partial x_2}\quad \cdots\quad \frac{\partial f(x)}{\partial x_n}\right)^T}$$

3）$f\in C^2, \nabla f(x^*)=0,$ 且 $\nabla^2 f(x^*)$ 是正定的，

则：x^* 是极小值点 （充分条件）

说明：

$$a)\nabla^2 f(x)=\begin{pmatrix} \frac{\partial^2 f(x)}{\partial x_1^2} & \frac{\partial^2 f(x)}{\partial x_1\partial x_2} & \cdots & \frac{\partial^2 f(x)}{\partial x_1\partial x_n} \\ \frac{\partial^2 f(x)}{\partial x_2\partial x_1} & \frac{\partial^2 f(x)}{\partial x_2^2} & \cdots & \frac{\partial^2 f(x)}{\partial x_2\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f(x)}{\partial x_n\partial x_1} & \frac{\partial^2 f(x)}{\partial x_n\partial x_2} & \cdots & \frac{\partial^2 f(x)}{\partial x_n^2} \end{pmatrix}$$ 海色矩阵($Hessian$)

$b)$一元函数，充分条件：$f''(x^*)>0$

$c)\ f(x)=\frac{1}{2}x^T Ax+b^T x+c,$ 有：$\nabla^2 f(x)=A$

求解方法概述：

1）解方程组：$\nabla f(x)=0 \Leftrightarrow \begin{cases} \dfrac{\partial f(x)}{\partial x_1}=0 \\ \vdots \\ \dfrac{\partial f(x)}{\partial x_n}=0 \end{cases}$

$\begin{cases} f(x)\text{为二次函数时，线性方程组可解} \\ \text{一般情况下为非线性方程组，难解。常用数值方法近似求解} \end{cases}$

有时考虑与其用数值方法求解方程组，不如用数值方法直接求极值

2）迭代法求解的基本步骤：

a)选择初始点x_0

b)确定迭代方向P_k，迭代公式：$x_{k+1} = x_k + \lambda_k P_k$

c)在射线$x_k + \lambda P_k (\lambda \geq 0)$上选步长

$\lambda = \lambda_k$满足：$f(x_k + \lambda_k P_k) < f(x_k)$

（保证是下降算法，也可用一维搜索找到下降最多的λ值）

d)停机条件（通常）：$\|\nabla f(x_{k+1})\| \leq \varepsilon$

3）一个迭代方法通常要考虑其收敛性：迭代序列中某元素为x^*或序列的极限趋向于x^*

$$\lim_{k\to\infty}\left\|x_k - x^*\right\| = 0$$

4）算法的收敛速度

$$\lim_{k\to\infty}\frac{\left\|x_{k+1}-x^*\right\|}{\left\|x_k - x^*\right\|} = \alpha\begin{cases}\in(0,1), & \text{线性收敛}\\ =0, & \text{超线性收敛}\end{cases}$$

$$\lim_{k\to\infty}\frac{\left\|x_{k+1}-x^*\right\|}{\left\|x_k - x^*\right\|^2} = \beta\begin{cases}>0, & \text{二次收敛}\\ =0, & \text{超二次收敛}\end{cases}$$

一维搜索方法：

问题：$\min\limits_{-\infty<x<\infty} f(x)$　　思路：先求出稳定点 $f'(x)=0$

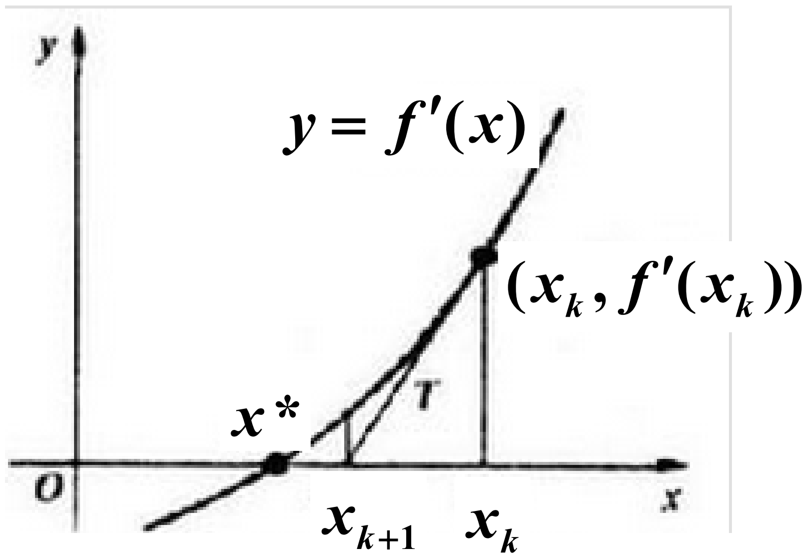
牛顿法：

1）最早用于方程求根 $\varphi(x)=0$，$x_{k+1}=x_k-\dfrac{\varphi(x_k)}{\varphi'(x_k)}$

2）$f(x)\approx f(x_k)+f'(x_k)(x-x_k)+\dfrac{1}{2}f''(x_k)(x-x_k)^2 \xrightarrow{\text{记为}} \varphi(x)$

$$\varphi'(x)=f'(x_k)+f''(x_k)(x-x_k)\xrightarrow{\text{令}} 0$$

$$x_{k+1}=x_k-\frac{f'(x_k)}{f''(x_k)}$$



性质：

1）收敛速度较快

2）二阶导数存在且需计算

3）初始点应足够靠近极小值点，否则可能不收敛

平分法：

1）先找到区间$[a,b]$满足：$f'(a)$与$f'(b)$异号，区间中有零点$f'(x)=0$用平分法搜索

2）$x=\dfrac{a+b}{2}$，计算$f'(x)$

3）$f'(x)\begin{cases}<0, a\Leftarrow x\\ >0, b\Leftarrow x\end{cases}$

1）收敛速度较牛顿法慢
2）计算量较小
3）总能收敛到局部极小点

0.618法：

假设$f(x)$在$[a,b]$中是单峰函数（另外计算方案）,$\alpha=0.618033988$

1）求$x_1=a+(1-\alpha)(b-a), x_2=a+\alpha(b-a), f(x_1)=f_1, f(x_2)=f_2$

2）若$|b-a|\le\varepsilon$,求出近似最优解$x^*=\dfrac{a+b}{2}$;否则转3）

3）$f_1<f_2: a\Rightarrow a, x_2\Rightarrow b, x_1\Rightarrow x_2, f_1\Rightarrow f_2$转4);

$f_1>f_2: x_1\Rightarrow a, b\Rightarrow b, x_2\Rightarrow x_1, f_2\Rightarrow f_1$转5);

$f_1=f_2: x_1\Rightarrow a, x_2\Rightarrow b_1$转1);

4）求$x_1=a+(1-\alpha)(b-a)$, $f(x_1)\Rightarrow f_1$转2）

5）求$x_2=a+\alpha(b-a), f(x_2)\Rightarrow f_2$转2）

1）要求函数有较高的解析性质（单峰）
2）不用求导，只需要函数值本身

n维极值的解析方法： **问题：** $\min\limits_{x\in R^n} f(x)$

解法：解析法采用一阶、二阶导数；直接法不用导数只用函数值

梯度法： $f(x)=f(x_k)+(x-x_k)^T\nabla f(x_k)+o(\|x-x_k\|)$

记： $x-x_k=\lambda P, \|P\|=1, \lambda>0;$

有： $f(x)=f(x_k)+\lambda P^T\nabla f(x_k)+o(\lambda)$

若： $P^T\nabla f(x_k)<0, \lambda_k$**足够小，取：** $x_{k+1}=x_k+\lambda_k P$

就有： $f(x_{k+1})<f(x_k)$

说： P**是点**x_k**的局部下降方向**

$$P^T\nabla f(x_k)=\|P\|\cdot\|\nabla f(x_k)\|\cdot\cos\theta\xrightarrow{\min}-\|\nabla f(x_k)\|\ (\theta=180°)$$

$f(x)$**在点**x_k**的梯度方向**$\nabla f(x_k)$**是函数值局部上升最快 的方向；**

$-\nabla f(x_k)$**是函数值局部下降最快 的方向。**

最速下降法：$x_{k+1}=x_k-\lambda_k\nabla f(x_k)$

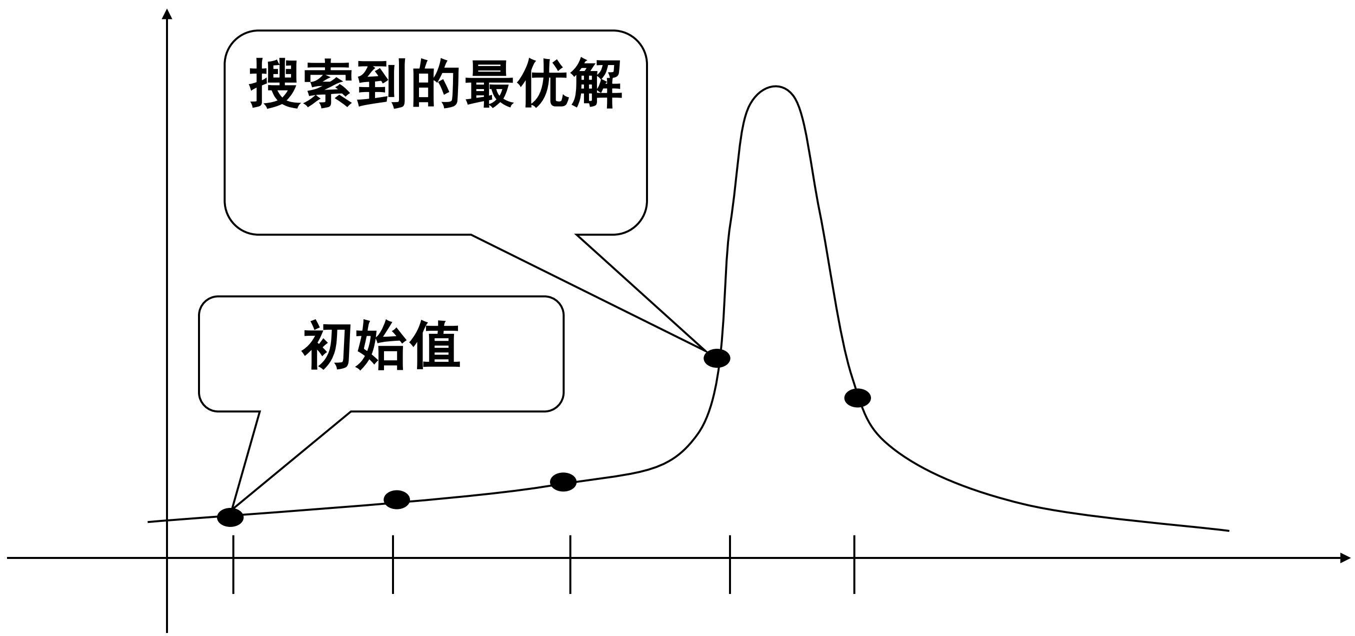
$\boxed{\text{最速下降法：} x_{k+1} = x_k - \lambda_k \nabla f(x_k)}$

步长λ_k的选取问题：

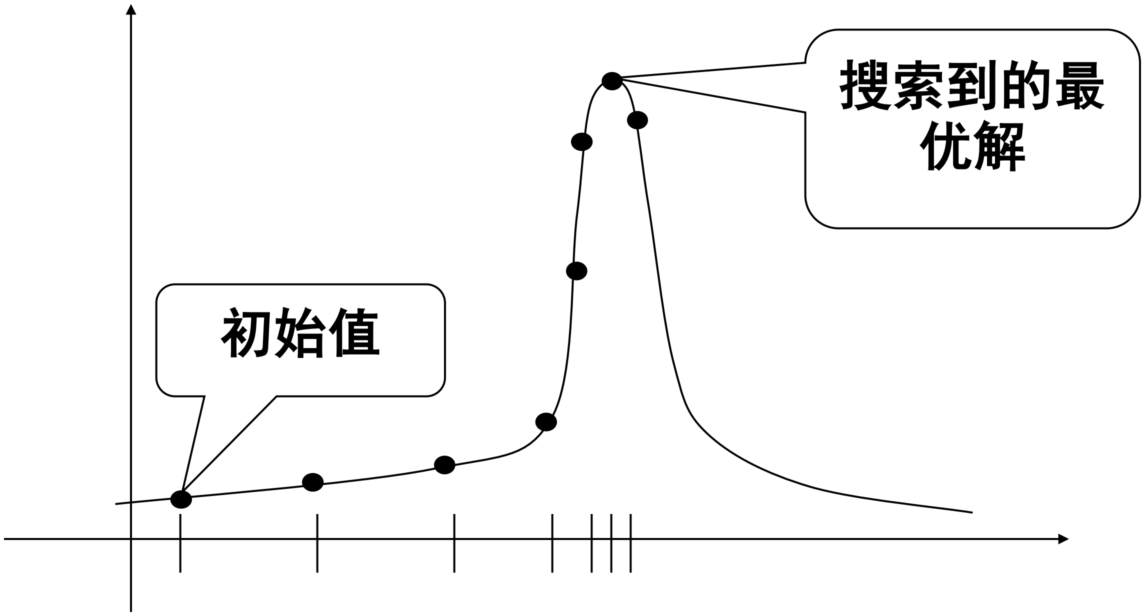
1）定步长：只能保证一定的精度，接近极小点时，
步长跨度可能越过极小点；

2）变步长：按一维搜索方法，沿方向$-\nabla f(x_k)$搜索
局部极小值点获得λ_k;但在接近x^*时，发生锯齿效应，
造成收敛速度变慢。

- **定步长存在的问题**



- **变步长**



牛顿法（n维）

1） *Taylor*展开至二阶：

$$f(x)\approx f(x_k)+(x-x_k)^T\nabla f(x_k)+\frac{1}{2}(x-x_k)^T\nabla^2 f(x_k)(x-x_k)$$

$$\xrightarrow{\text{记}}\varphi(x)$$

2） $\varphi(x)$的极小值点作为x_{k+1}：

$$\nabla\varphi(x)=\nabla f(x_k)+\nabla^2 f(x_k)(x-x_k)\underline{\underline{令}}0$$

$$x_{k+1}=x_k-[\nabla^2 f(x_k)]^{-1}\nabla f(x_k)$$

3）任何二次函数，用牛顿法，只需迭代一次就可得到极小点；但需要计算海色矩阵的逆；

4）对于非二次函数，若$\|x_0-x^*\|<1$，可达二次收敛；否则不能保证收敛

阻尼牛顿法（n维）

$1)0 \Rightarrow k$

2）求：$\nabla f(x_k),[\nabla^2 f(x_k)]^{-1}$，$P_k = -[\nabla^2 f(x_k)]^{-1}\nabla f(x_k)$

3)求：$x_{k+1} = x_k + \lambda_k P_k$,其中$\lambda_k$满足：

$$f(x_k + \lambda_k P_k) = \min_{\lambda} f(x_k + \lambda P_k)$$

4)若$\left\| x_{k+1} - x_k \right\| \le \varepsilon$，则求出最优解$x^* = x_{k+1}$;否则$k+1 \Rightarrow k$转2）

特性：

1）保留了牛顿法的二次收敛特性，但对初始点的要求不苛刻；

2）仍然要计算$[\nabla^2 f(x_k)]^{-1}$，因而要求$\nabla^2 f(x_k)$可逆；

3）即使$\nabla^2 f(x_k)$可逆，但若是非正定的，也有可能不收敛。

带约束最优化方法回顾

问题1: $\min\limits_{x\in R^n} f(x)$

$$s.t.\begin{cases} g_i(x)\geq 0, i=1,2,\cdots,m \\ h_j(x)=0, j=1,2,\cdots,p \end{cases}$$

定理($Kuhn-Tucker$最优性必要条件):

设x^*为问题1的最优解，如果在x^*处诸起作用约束的梯度向量$\nabla g_i(x^*), i\in I(x^*)$和$\nabla h_j(x^*)(j=1,2,\cdots,p)$线性无关，则存在向量$\lambda^*$、$\mu^*$使下述条件成立:

$$\begin{cases} \nabla f(x^*)-\sum_{i=1}^{m}\lambda_i^*\nabla g_i(x^*)-\sum_{j=1}^{p}\mu_j^*\nabla h_j(x^*)=0 \\ \lambda_i^* g_i(x^*)=0,\ \ \lambda_i^*\geq 0,\ \ i=1,2,\cdots,m \end{cases}$$

拉格朗日广义函数与乘子：

$$L(x,\lambda,\mu)=f(x)-\sum_{i=1}^{m}\lambda_i g_i(x)-\sum_{j=1}^{p}\mu_j h_j(x)$$

设x^*为问题1的最优解，则存在向量$\lambda^*\geq 0$与μ^*使(x^*,λ^*,μ^*)是$L(x,\lambda,\mu)$的稳定点。λ^*与μ^*称为原问题的拉格朗日乘子。

对凸规划问题(f是凸函数，g_i为凹函数，h_j为线性函数)，$K-T$条件是最优解的充要条件。

求解

- 局部逼近法
  - 用线性规划
  - 用二次规划
- 可行方向法
  - 线性规划求解可行方向
  - 投影梯度法等
- 罚函数法
  - 内罚函数法：从内点出发，每步在内点搜索，边界设置壁垒（罚函数）
  - 外罚函数：不限初始点，但外点惩罚，逐步趋内
- 通过求对偶问题

外罚函数法：

给定初始点x_0,可以是内点或外点；设误差容许值$\varepsilon > 0$,罚因子$M_0 > 0, c > 1$为给定实数；令$k = 0$

1)令$P(x) = \sum_{i=1}^{m} \left|\min(0, g_i(x))\right|^{\alpha} + \sum_{j=1}^{p} \left|h_j(x)\right|^{\beta}, \alpha \geq 1, \beta \geq 1$（通常取2）

$F_M(x) = f(x) + MP(x)$ 罚函数

2）求无约束极小问题$\min\limits_{x} F_{M_k}(x)$，得极小值点$x_{k+1}$

3)若$M_k P(x_{k+1}) < \varepsilon$,则取$x_{k+1}$为问题（1）的最优解，停机；否则

$M_{k+1} \Leftarrow cM_k, k \Leftarrow k+1$,转2）。

特性：

1）对初始点的要求不苛刻；

2）近似最优解x_k可能不是可行解，实际不可用；

3）M_k随迭代逐渐增大，趋于收敛时，可能增大到引起计算误差。

内罚函数法：

给定初始点x_0满足：$g_i(x_0)>0, i=1,2,\cdots,m$；设误差容许值$\varepsilon>0$,内罚因子$r_0>0, c>1$为给定实数；令$k=0$

1)令$B(x)=\sum_{i=1}^{m}\frac{1}{g_i(x)}$; $F_r(x)=f(x)+rB(x)$ 内罚函数

2）求无约束极小问题$\min_x F_{r_k}(x)$，得极小值点x_{k+1}

3)若$r_kB(x_{k+1})<\varepsilon$,则取$x_{k+1}$为问题（2）的最优解，停机；否则

$r_{k+1}\Leftarrow r_k/c, k\Leftarrow k+1$,转2）。

特性：

1）要求初始点必须是内点；

2）只能处理不等式约束；

3）每次迭代点都是可行解，实用有效；

4）随着r_k减小，罚函数越来越病态，无约束极小难求。

$$\boxed{\begin{aligned}\text{问题2:}\ &\min_{x\in R^n} f(x)\\ &s.t.\ g_i(x)\ge 0, i=1,2,\cdots,m\end{aligned}}$$

乘子法： $\boxed{\text{问题3:} \min_{x\in R^n} f(x),\ \ h_j(x)=0, i=1,2,\cdots,p}$

给定初始点x_0；取常数$c>0, 0<\gamma<1, \alpha>1$和$\varepsilon>0$，给出$\mu_1$，

令$k=1$

1)以x_{k-1}为初始点，求以下函数的无约束极小，得极小值点x_k；

$$M(x,\mu^{(k)})=f(x)-\sum_{j=1}^{p}\mu_j^{(k)}h_j(x)\boxed{+\frac{c}{2}\sum_{j=1}^{p}(h_j(x))^2}$$

2）若$\|h(x_k)\|<\varepsilon$，则以x_k为最优解；否则计算比值$\|h(x_k)\|/\|h(x_{k-1})\|$，

若该比值$\geq\gamma$，则$\alpha c\Rightarrow c$

3)$\mu_j^{(k+1)}=\mu_j^{(k)}-ch_j(x_k), j=1,2,\cdots,p; k\Leftarrow k+1$，转1）。

特性：

1）无约束函数不会发生病态；

2）可以从等式约束推广到不等式约束。

问题1: $\min\limits_{x\in R^n} f(x)$

$$s.t.\begin{cases} g_i(x)\ge 0, i=1,2,\cdots,m \\ h_j(x)=0, j=1,2,\cdots,p \end{cases}$$

乘子法公式如下：

$$M(x,\lambda,\mu)=f(x)+\frac{1}{2c}\sum_{i=1}^{m}\{[\max(0,\lambda_i^{(k)}-cg_i(x))]^2-(\lambda_i^{(k)})^2\}$$

$$-\sum_{j=1}^{p}\mu_j^{(k)}h_j(x)+\frac{c}{2}\sum_{j=1}^{p}h_j^2(x)$$

$$\lambda_i^{(k+1)}=\max(0,\lambda_i^{(k)}-cg_i(x_k)), i=1,2,\cdots,m$$

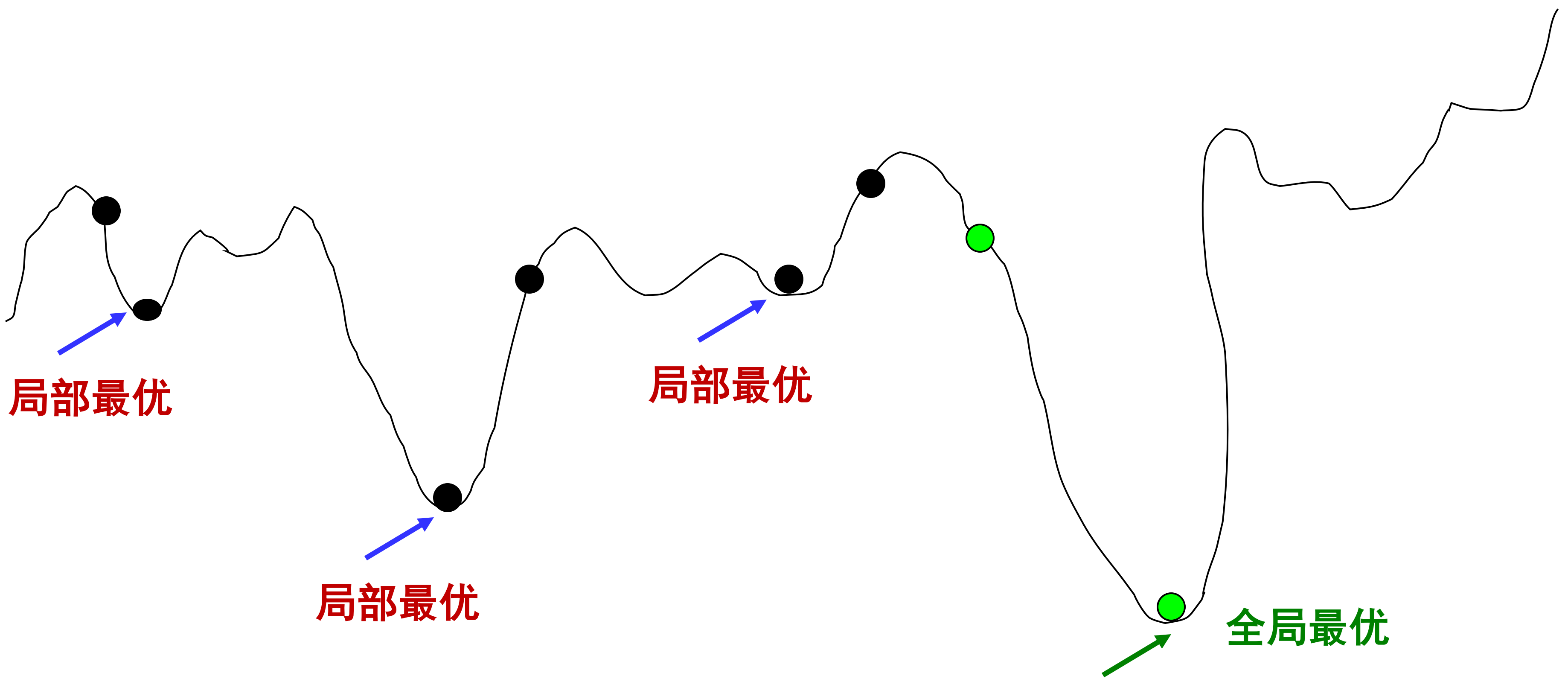
$$\mu_j^{(k+1)}=\mu_j^{(k)}-ch_j(x_k), j=1,2,\cdots,p$$

传统优化方法的局限性

- 单点方式限制了计算效率的提高
- 易陷入局部最优
- 对目标函数和约束函数的性质有一定的要求，限制了算法的适用范围

局限性之一图示

常导致局部最优，而非全局最优



函数优化

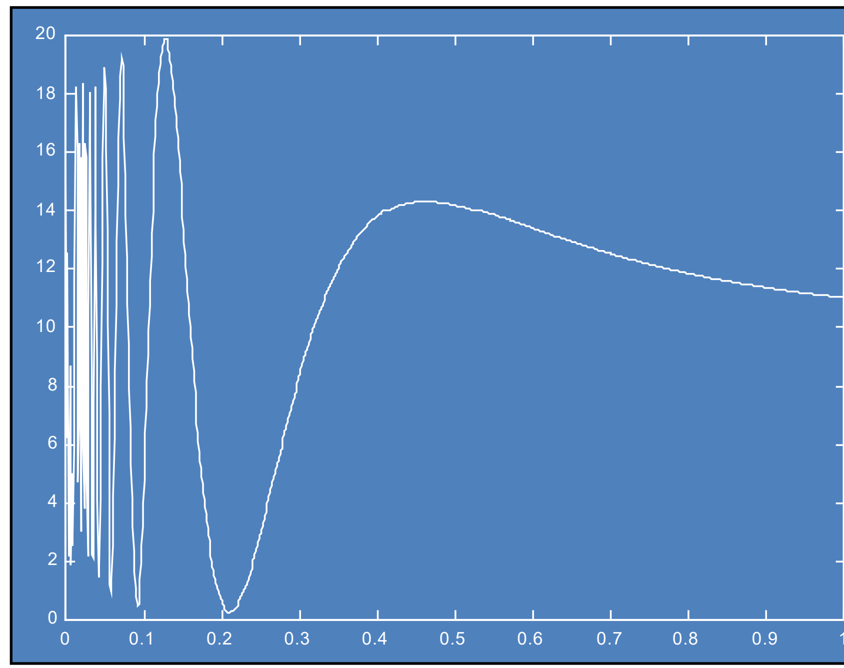
$$f(x)=10+\frac{\sin\left(1/x\right)}{(x-0.16)^2+0.1}$$

用于测试的
连续函数极
值求解问题

问题:

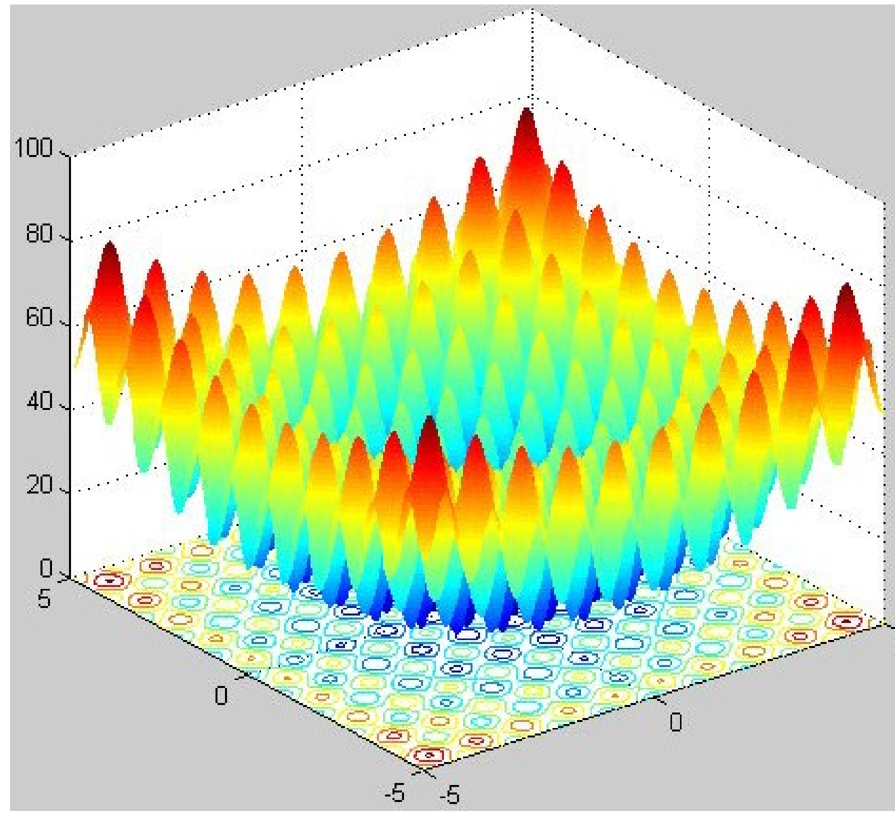
在(0,1)内寻找 x_{max} 使下式成立:

$$f(x_{max})\geq f(x),\forall x\in(0,1)$$



Rastrigin 函数：一个多峰值函数，其局部最优位置随着正弦波动，其全局最优点在 $\boldsymbol{x}^*=\boldsymbol{0}$,目标函数最优值在 $f(\boldsymbol{x}^*)=\boldsymbol{0}$，各变量之间独立。

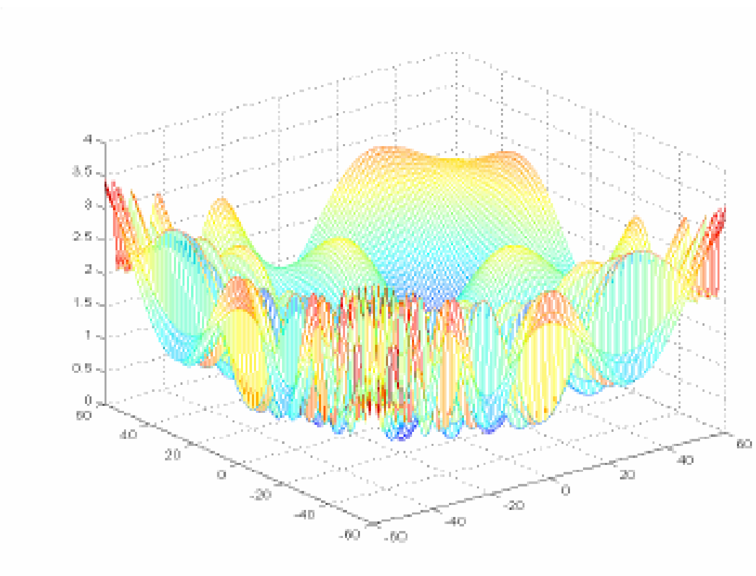
$$f_4(\boldsymbol{x})=\sum_{i=1}^{n}(x_i^2-10\cos(2\pi x_i)+10)$$



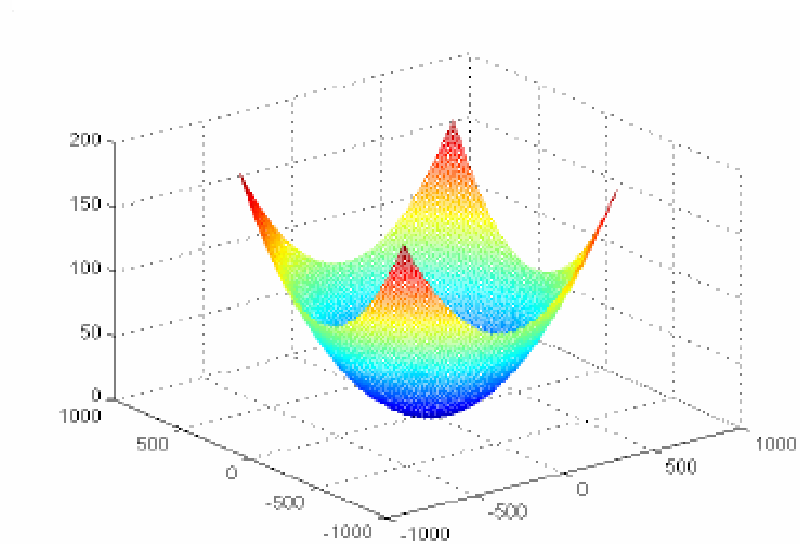
多维函数求极值。
有兴趣学习了后续
算法可编程求解

Griewank 函数：一个多峰值函数，变量之间有相互关系，其全局最优点在 $\boldsymbol{x}^* = \boldsymbol{0}$,目标函数最优值在 $f(\boldsymbol{x}^*) = \boldsymbol{0}$，各变量之间独立。二维 Griewank 函数如图所示。

$$f_5(\boldsymbol{x}) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$$



(a) 局部



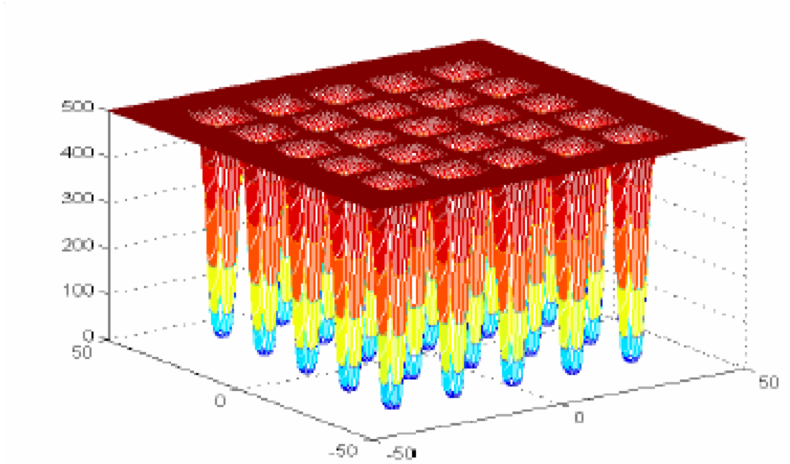
(b) 全局

Foxhole 函数：一个二维的多峰函数，其全局最优点在 $\boldsymbol{x}^*=(-32,-32)$，目标函数最优值在 $f(\boldsymbol{x}^*)=0.998$。

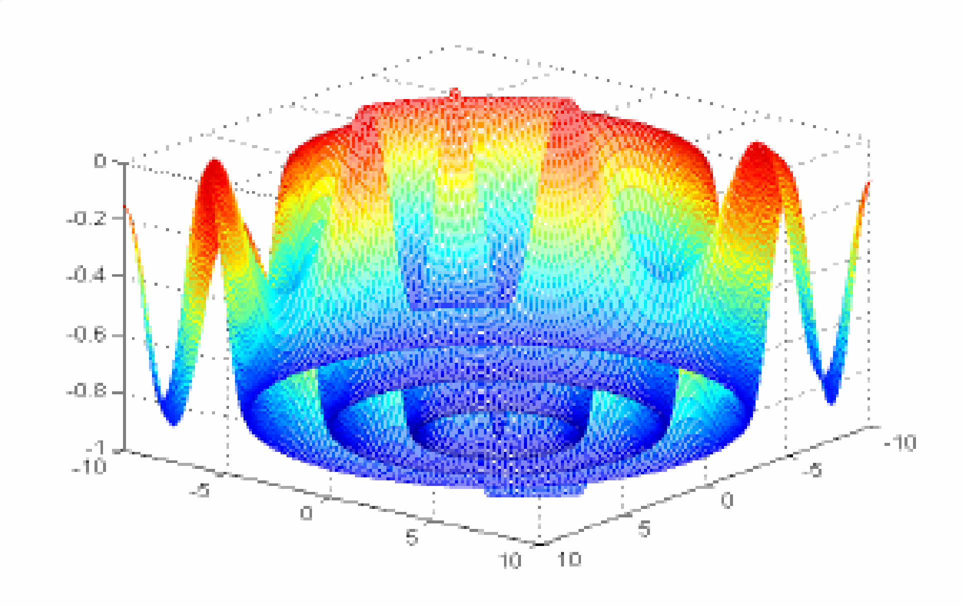
$$f_8(\boldsymbol{x}) = 1\Bigg/\left(0.002+\sum_{j=1}^{25}\frac{1}{j+\sum_{i=1}^{2}(x_i-a_{ij})^6}\right)$$

其中

$$\left[a_{i,j}\right]=\begin{bmatrix} -32 & -16 & 0 & 16 & 32 & -32 & -16 & 0 & \cdots & 0 & 16 & 32 \\ -32 & -32 & -32 & -32 & -32 & -16 & -16 & -16 & \cdots & 32 & 32 & 32 \end{bmatrix}$$



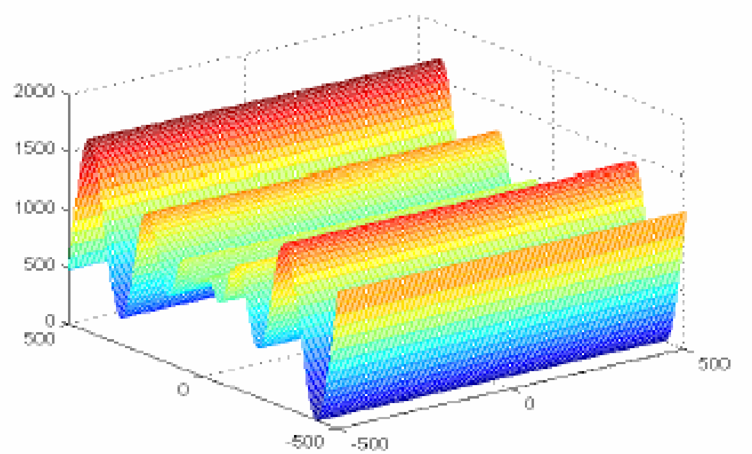
$$f_9(\boldsymbol{x}) = -0.5 + \frac{(\sin\sqrt{x^2+y^2}) - 0.5}{(1.0 + 0.001(x^2+y^2))^2}$$

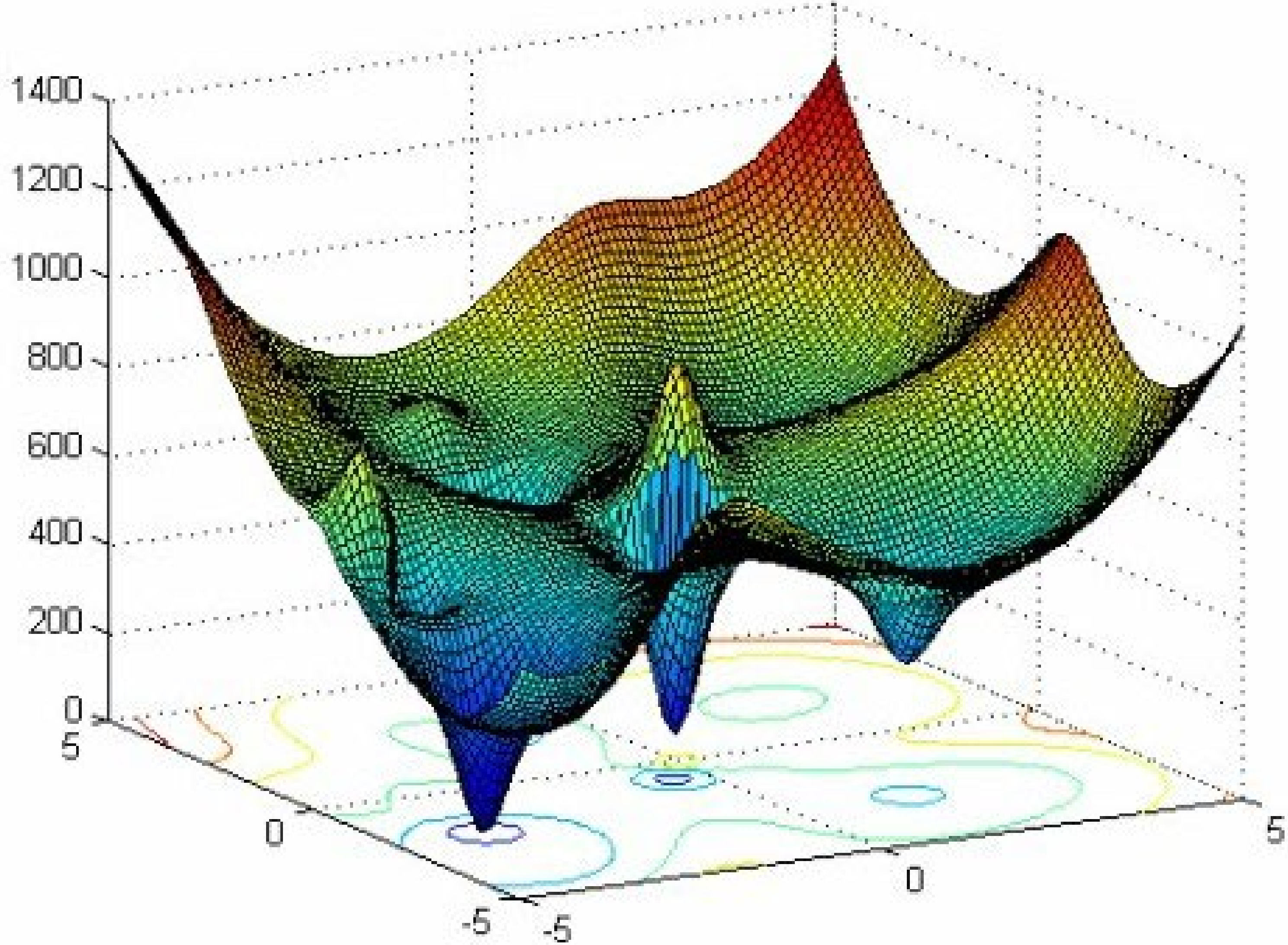


Shaffer's f6 函数三维图

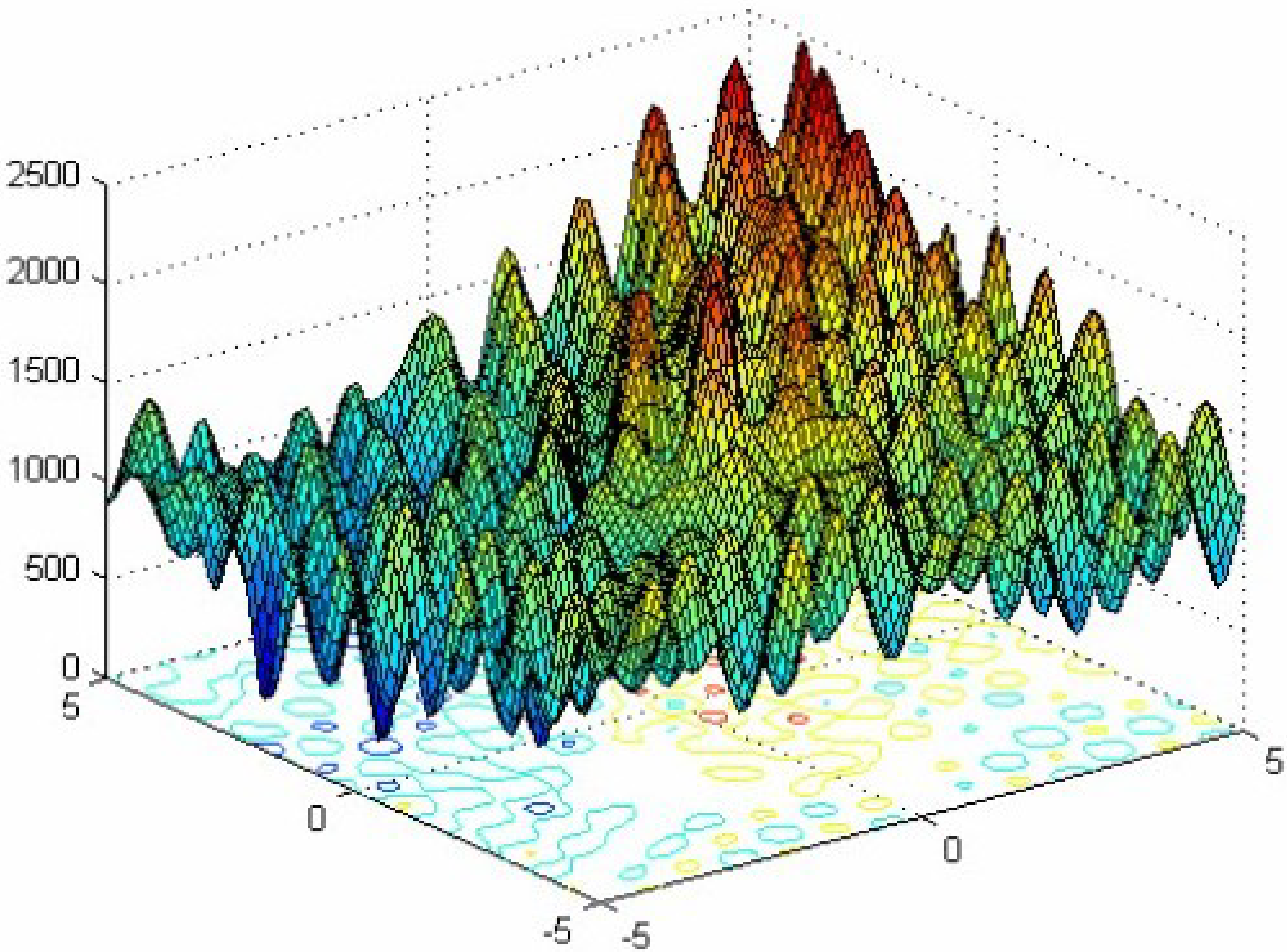
Schwefel 函数：一个多峰函数，局部最优随正弦波动，最优点位于搜索空间的一个角落中，$\boldsymbol{x}^* = (-420.9687, -420.9687, \cdots, -420.9687)$，$f(\boldsymbol{x}^*) = \mathbf{0}$。

$$f_6(X) = 418.9829n + \sum_{i=1}^{n} x_i \sin(\sqrt{|x_i|})$$





现实问题中也存在地形极值问题，虽然未给出解析式


2500
2000
1500
1000
500
0
5
0
-5
-5
0
5

权宜性解决方法一

- 每次不只接受邻域内较优的点，而是依据一定的概率，从邻域内随机选择一个点作为接受点

- 原则是：指标函数优的点，被选中的概率较大，而指标函数差的点，被选中的概率较小。

选择概率的计算

- 目标为求最大值时：

$$P_{\max}(x_i)=\frac{f(x_i)}{\sum\limits_{x_j\in N(x)} f(x_j)}$$

- 目标为求最小值时：

$$P_{\min}(x_i)=\frac{1-P_{\max}(x_i)}{\sum\limits_{x_j\in N(x)}(1-P_{\max}(x_j))}=\ldots=\text{（略）}$$

$$=\frac{1}{|N(x)|-1}\left(1-P_{\max}(x_i)\right)$$

局部搜索改进算法1（依概率接受新点）

1）随机选择一个初始可能解 $x_0 \in D$，$x_b = x_0$，$P = N(x_b)$

2）如果不满足结束条件，则

Begin

对于所有的 $x \in P$ 计算指标函数 $f(x)$，

并计算每一个点x的概率 $P_{\max}(x_i)$ 或 $P_{\min}(x_i)$

从P中依上述概率值选择一个点 x_n，

$x_b = x_n$，$P = N(x_b)$，转2）

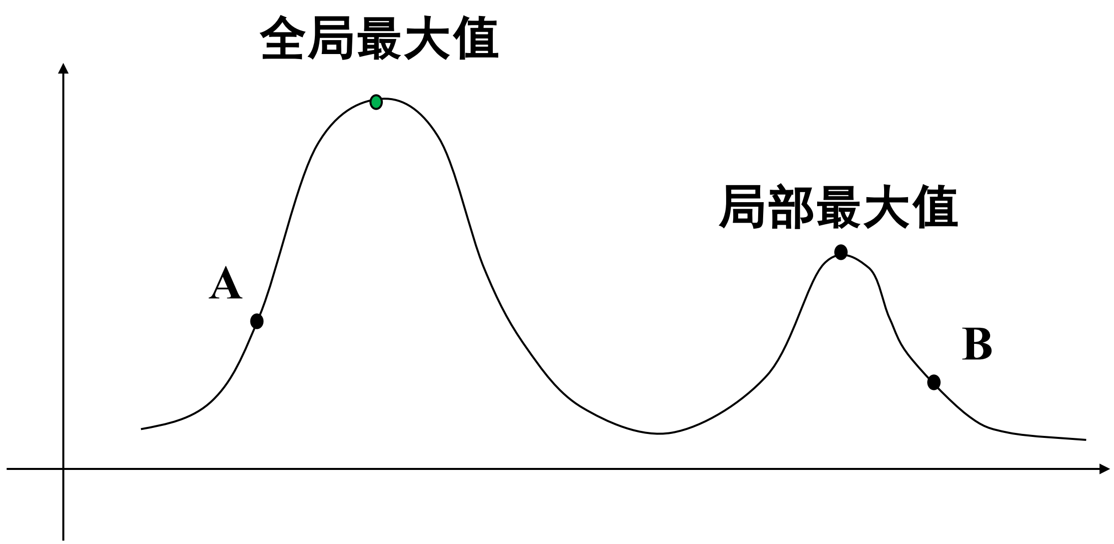
End

7）输出计算结果，结束

轮盘赌方法

权宜性解决方法二

- 随机生成一些初始点，从每个初始点出发进行搜索，找到各自的最优解。再从这些最优解中选择一个最好的结果作为最终的结果。



局部搜索改进算法2（多个初始点选优）

1) $k = 0$

2) 随机选择一个初始的可能解$x_0 \in D$，$x_b = x_0$，$P = N(x_b)$

3) 如果不满足结束条件，则

Begin

选择P的一个子集P‘，x_n为P’中的最优解

若$f(x_n) < f(x_b)$，则$x_b = x_n$，$P = N(x_b)$，转3）

否则$P = P - P'$，转3）。

End

4) $k = k+1$

5) 如果k达到了指定的次数，则从k个结果中选择一个最好的结果输出，否则转2）

6) 输出结果，结束

局部搜索改进算法3（变步长）

1）随机选择一个初始可能解$x_0 \in D$，$x_b=x_0$，
确定一个初始步长后计算$P=N(x_b)$

2）如果不满足结束条件，则

3）Begin

选择P的一个子集P'，x_n为P'中的最优解

如果$f(x_n) < f(x_b)$，则$x_b = x_n$

按照某种策略改变步长，计算$P = N(x_b)$，
转2）

否则$P = P - P'$，转2）。

End

4) 输出计算结果，结束

多种方法的集成

- 以上几种解决方法可结合使用，比如第一、第三种方法的结合，就产生了后面介绍的模拟退火方法。

计算智能方法的产生及其特点

- 1943人工神经网络，非线性数学模型的网络建立，全新的迭代逼近求解方法
- 1975遗传算法，模拟大自然生物进化规则
- 寻优与生物进化和社会集群现象极其相似
- 模拟热力学、模拟蚂蚁、模拟人体免疫机制……

计算智能方法的特点

- 隐并行性、协同进化、全局搜索能力强、健壮性佳、适合大规模数据
- 不以达到某个最优条件或理论上的最优解为要旨，更注重计算效率
- 对目标函数和约束条件的性质要求更宽松，无需满足可微、连续、凸性等条件
- 算法的基本思想都是来自于对某种自然规律的模仿，具有人工智能的特点
- 算法以包含多个进化个体的种群为基础，寻优过程就是种群的进化过程
- 算法的理论基础较薄弱，一般不能保证一定收敛到最优解