第一章　绪 论

1.1 最优化问题

所谓最优化问题，就是在满足一定的约束条件下，寻找一组参数值，以使某些最优性度量得到满足，即使系统的某些性能指标达到最大或最小。最优化问题的应用可以说遍布工业、社会、经济、管理等各个领域，其重要性是不言而喻的。

最优化问题根据其目标函数、约束函数的性质以及优化变量的取值等可以分成许多类型，每一种类型的最优化问题根据其性质的不同都有其特定的求解方法。

不失一般性，设所考虑的最优化问题为：

$$\begin{aligned} &\min \sigma = f(X) \\ &s.t.\ \ X \in S = \{X \mid g_i(X) \le 0 \quad j = 1,\ldots,m\} \end{aligned} \tag{1.1}$$

其中，$\sigma = f(X)$ 为目标函数，$g_i(X)$ 为约束函数，S 为约束域，X 为 n 维优化变量。通常最大化问题很容易转换为最小化问题($\sigma = -f(X)$)，对于 $g_i(X) \ge 0$ 的约束和等式约束也可转换为 $-g_i(X) \le 0$ 的约束，所以(1.1)式所描述的最优化问题不失一般性。

当 $f(X)$、$g_i(X)$ 为线性函数，且 $X \ge 0$ 时，上述最优化问题即为线性规划问题，其求解方法有成熟的单纯形法和 Karmarc 方法。

当 $f(X)$、$g_i(X)$ 中至少有一个函数为非线性函数时，上述问题即为非线性规划问题。非线性规划问题相当复杂，其求解方法多种多样，但到目前仍然没有一种有效的适应所有问题的方法。

当优化变量 X 仅取整数值时，上述问题即为整数规划问题，特别是当 X 仅能取 0 或 1 时，上述问题即为 0－1 整数规划问题。由于整数规划问题属于组合优化范畴，其计算量随变量维数的增长而指数增长，所以存在着“ 维数灾难 ”问题。

当 $g_i(X) \le 0$ $(j = 1,\ldots,m)$ 所限制的约束空间为整个n维欧氏空间，即R^n时，上述最优化问题为无约束优化问题，即

$$\begin{aligned} &\min \sigma = f(X) \\ &s.t.\ \ X \in S \subset R^n \end{aligned} \tag{1.2}$$

非线性规划问题（包括无约束优化问题和约束优化问题），由于函数的非线性，使得问题的求解变得十分困难，特别是当目标函数在约束域内存在多峰值时。常见的求解非线性问题的优化方法，其求解结果与初值的选择关系很大，也就是说，一般的约束或无约束非线性优化方法均是求目标函数在约束域内的近似极值点，而非真正的最小点。

1.1.1 局部优化算法

定义 1.1　如果存在 $X_B^* \in B$，使得对 $\forall X \in B$ 有

$$f(X_B^*) \le f(X), \quad X \in B \tag{1.3}$$

成立，其中 $B \subset S \subseteq R^n$，$S$ 为由约束函数限定的搜索空间，则称 X_B^* 为 $f(X)$ 在 B 内的局部极小点，$f(X_B^*)$ 为局部极小值。

常见的优化方法大多为局部优化方法，都是从一个给定的初始点 $X_0 \in S$ 开始，依据一定的方法寻找下一个使得目标函数得到改善的更好解，直至满足某种停止准则。

成熟的局部优化方法很多，如 Newton-Raphson 法、共轭梯度法、Fletcher-Reeves 法、Polar-Ribiere 法、Davidon-Fletcher-Power(DFP)法、Broyden-Fletcher-Goldfarb-Shann(BFGS)方法等等，还有专门为求解最小二乘问题而发展的 Levenberg-Marquardt(LM)算法。所有这些局部优化算法都是针对无约束优化问题而提出的，而且对目标函数均有一定的解析性质要求，如 Newton-Raphson 法要求目标函数连续可微，同时要求其一阶导数连续。

对于约束非线性优化问题，除了根据一阶最优化必要条件直接将最优化问题转换为非线性代数方程组，然后采用非线性代数方程组的数值解法进行求解外，还有序列线性规划法、可行方向法、拉格朗日乘子法等等。最常用的方法是将约束问题通过罚函数法转换为无约束优化问题，然后再采用无约束优化方法进行求解。这些具体方法请读者参阅有关最优化理论与方法方面的文献。

1.1.2 全局优化算法

定义 1.2　如果存在 $X^* \in S$，使得对 $\forall X \in S$，有：

$$f(X^*) \le f(X), \quad X \in S \tag{1.4}$$

成立，其中 $S \subseteq R^n$ 为由约束条件限定的搜索空间，则称 X^* 为 $f(X)$ 在 S 内的全局极小点，$f(X^*)$ 为其全局极小值。

前面指出，已发展成熟的最优化方法大多为局部优化方法，其求解结果与初始值相关。对于目标函数为凸函数、约束域为凸域的所谓凸规划问题，局部最优与全局最优等效。而对于非凸问题，由于在约束域内目标函数存在多峰值，因而其全局最优与局部最优相差甚远。

目前为止，全局优化问题也已存在了许多算法，如填充函数法等，但比起局部优化问题的众多成熟方法，其间还有很大差距。

另外，解析性优化方法对目标函数及约束域均有较强的解析性要求，对于诸如目标函数不连续、约束域不连通、目标函数难以用解析函数表达或者难以精确估计(如仿真优化问题)等问题时，解析确定性优化方法就难以适应。

为了可靠解决全局优化问题，人们试图离开解析确定型的优化算法研究，转而探讨对函数解析性质要求较低甚至不作要求的随机型优化方法。最早的随机型优化方法是基于 Monte-Carlo 方法的思想，针对具体问题性质的特点，构造以概率 1 收敛于全局最小点的随机搜索算法。真正有效且具有普遍适应性的随机全局优化方法，是近十多年来人们模拟自然界的一些自然现象而发展起来的一系列仿生型智能优化算法，如模拟退火方法、进化类算法、群体智能算法等等。

1.1.3 无免费午餐定理(No Free Lunch Theorem)

在最优化理论研究领域中，最值得一提的是 Wolpert 和 Macready 于 1997 年在 IEEE Transactions on Evolutionary Computation 上发表了题为“ No Free Lunch Theorems for Optimization ”

的论文，提出并严格论证了所谓的无免费午餐定理，简称 NFL 定理。

NFL定理的简单表述为：对于所有可能的问题，任意给定两个算法A，A'，如果A在某些问题上表现比A'好(差)，那么A在其它问题上的表现就一定比A'差(好)，也就是说，任意两个算法A、A'对所有问题的平均表现度量是完全一样的。

有关 NFL 定理的推导与证明请参阅文献[Wolpert 1997]。

自从 NFL 定理提出以来，有关定理本身及其相关结论的争论在学术界一直持续未断，因为 NFL 定理本身涉及到了优化算法最基本的问题，而且其结论多少有点出人意料。

NFL定理的主要价值在于它对研究与应用优化算法时的观念性启示作用。虽然NFL定理是在许多假设条件下得出的，但它仍然在很大程度上反映出了优化算法的本质。当我们所面对的是一个大的而且形式多样的适应值函数类时，就必须考虑算法间所表现出的NFL效应。即若算法A在某些函数上的表现超过算法A'，则在这类的其它适应值函数上，算法A'的表现就比A要好。因此，对于整个函数类，不存在万能的最佳算法，所有算法在整个函数类上的平均表现度量是一样的。

有了上述讨论，关于优化算法的研究目标就应该从寻找一个大的函数类上的优化算法转变为：

(1) 以算法为导向，从算法到问题。对于每一个算法，都有其适用和不适用的问题；给定一个算法，尽可能通过理论分析，给出其适用问题类的特征，使其成为一个“ 指示性 ”的算法。

(2) 以问题为导向，从问题到算法。对于一个小的特定的函数集，或者一个特定的实际问题，可以设计专门适用的算法。

实际上，大多数在进化算法方面的研究工作可以看作是属于这一范畴的，因为它们主要是根据进化的原理设计新的算法，或者将现有算法进行部分改进，以期对若干特定的函数取得好的优化效果。

读者可能会产生这样的疑问，既然 NFL 定理表明，进化算法并不比一般随机搜索算法好，那么为什么还要去研究进化算法呢？实际上，NFL 定理只是否定了去寻找一个万能的最佳算法的可能性，但对于某些小的函数集合，NFL 定理则认为存在一个在该集合上的好算法。在求解某些或某个特定的复杂优化问题时，可能会出现现有大多数优化方法都不适用、而适用的少数方法效果又不理想的情况。这时，进化算法就大有用武之地。由于所有进化类算法均具有很强的通用性，对目标函数的解析性质几乎没有要求，因此将进化类算法作为求解优化问题的一个候选算法将是非常有意义的。

1.2 进化计算

近十余年来，遗传算法(Genetic Algorithm，GA)、进化策略(Evolutionary Strategy，ES)、进化规划(Evolutionary Programming，EP)和遗传程序设计(Genetic Programming，GP)等进化类算法在理论和应用两方面发展迅速、效果显著，并逐渐走向了融合，形成了一种新颖的模拟进化的计算理论，统称为进化计算(Evolutionary Computation，EC)，进化计算的具体实现方法与形式称为进化算法(Evolutionary Algorithm，EA)。进化算法是一种受生物进化论和遗传学等理论启发而形成的求解优化问题的随机算法，虽然出现了多个具有代表性的重要分支，但它们各自代表了进化计算的不同侧面，各具特点。

1.2.1.进化算法的一般框架

下面给出求解优化问题(1.1)的进化算法的一般框架。首先将待优化的目标函数在保持整体极值点不变的前提下变换为适应值函数(fittness-value function) $\Phi : s \to R$。用 A 表示包含附加的搜索状态信息的适当表示空间，通常 A 是需要自适应调整的策略参数。在进化算法 EA 中，搜索群

体中的任一个体(X,a)均是笛卡尔集 S×A 中的一个元素，其中$X \in S$， $a \in A$。个体(X,a)的适应值由优化变量X的适应值$\Phi(X)$给出。μ≥1 表示父代群体的大小，λ≥1 表示子代群体的大小。将进化过程中第 t 代的群体记为

$$P(t) = \{(X_{t,1}, a_{t,1}), (X_{t,2}, a_{t,2}), \ldots, (X_{t,u}, a_{t,u})\} \tag{1.5}$$

则用 EA 求解问题(1.1)的过程如下：

Begin

确定编码的形式并生成搜索空间，选择遗传算子的类型和所有控制参数值；

设置代数 t=0；

随机生成初始化群体

$P(0) = \{(X_{0,1}, a_{0,1}), (X_{0,2}, a_{0,2}), \ldots, (X_{0,n}, a_{0,n})\}$

计算每个个体的适应值$\Phi(X_{0,i})$；

while (终止条件不满足) do

t=t+1；

对 P(t-1)进行重组操作生成群体 P’(t)；

对 P’(t)进行变异操作生成群体 P’’(t)，并计算每个个体的适应值$\Phi(X_{t,i})$；

对(Q∪P’’(t))进行选择操作生成群体

$P(t) = \{(X_{t,1}, a_{t,1}), (X_{t,2}, a_{t,2}), \ldots, (X_{t,u}, a_{t,u})\}$（其中 Q 表示 P(t-1)的某个子集）；

END while

END Begin

上述进化算法的一般框架基本上包括了所有标准的进化类算法，但仍有一些 EA 的变形或扩展无法包含在内，如稳定态(steady-state)EA 等。下面我们分别就进化计算的几个重要分支进行简单的介绍。

1.2.2.遗传算法(Genetic Algorithm，GA)

遗传算法最早是由美国 Michigan 大学的 J.Holland 博士提出的。1975 年，他出版了其开创性的著作“ Adaptive in Natural and Artificial Systems ”[Holland 1975]，标志着遗传算法的正式诞生。

遗传算法的操作对象是一组二进制串，即种群(Population)。每一个二进制串称为染色体(Chromosome)或个体(Individual)，每个染色体或个体都对应于问题的一个解。从初始种群出发，采用基于适应值比例的选择策略在当前种群中选择个体，并使用交叉(Crossover)和变异(Mutation)来产生下一代种群，如此一代一代进化下去，直至满足期望的终止条件。

遗传算法可以形式化描述为：

$$GA=(P(0), N, l, S, g, p, f, t) \tag{1.6}$$

其中：

$P(0)=(a_1(0), a_2(0), \ldots, a_N(0)) \in I^N$，表示初始种群；

$I=B^l=\{0, 1\}^l$表示长度为l的二进制串全体，称为位串空间；

N 表示种群中含有的个体数目，即群体规模；

l 表示二进制串的长度；

S：$I^N \to I^N$表示选择策略；

g表示遗传算子，通常它包括复制算子O_r：I→I，交叉算子O_c：I×I→I×I和变异算子O_m：I→I

p表示遗传算法的操作概率，包括复制概率p_r、交叉概率p_c和变异概率p_m

f：I→R^+是适应值函数

t：I^N→ {0，1} 是终止条件。

有关遗传算法的研究成果已相当丰富，包括编码方案、遗传算子的设计以及操作概率的自适应性控制策略等等，各种各样的改进算法层出不穷，其应用范围几乎可以涉及优化问题的所有领域，有关遗传算法方面的专著在国内外也出现了很多。

1.2.3.进化策略(Evolutionary Strategies，ES)

进化策略是由德国柏林工业大学的 I.Rechenberg 和 H.-P.Schwefel 等在 20 世纪 60 年代初提出的。早期的演化策略，其种群中只包含一个个体且只使用变异操作。具体在每一操作代中，变异后的个体与其父代个体进行比较，从中择优选取作为新一代个体，这就是所谓的(1＋1)策略。它所使用的变异算子主要是基于正态分布的变异操作[Schwefel 1981]。

由于(1＋1)策略难以收敛到最优解，且搜索效率相对较低。其改进的方法就是增加种群内个体的数量，即(μ＋1)进化策略。此时种群内含有μ个个体，随机抽取一个个体进行变异，然后取代群体中最差的个体。为了进一步提高搜索效率，后来又提出了(μ＋λ)进化策略和(μ，λ)进化策略。(μ＋λ)进化策略是根据种群内的μ个个体采用变异和重组产生λ个个体，然后将这μ＋λ个个体进行比较选择μ个最优者；而(μ，λ)进化策略则是在新产生的λ(≥μ)个个体中比较选择μ个最优者。

进化策略与遗传算法相比，其主要不同之处在于：遗传算法是将原问题的解空间映射到位串空间上，然后再进行遗传操作，它强调的是个体基因结构的变化对其适应度的影响，而进化策略则是直接在解空间上进行遗传操作，它强调的是父代到子代行为的自适应性和多样性。

1.2.4.进化规划(Evolutionary Programming，EP)

进化规划方法最初是由美国科学家 Lawrence J.Fogel 等人在 20 世纪 60 年代提出的[Fogel 1966]。它在求解连续参数优化问题时与 ES 的区别很小。进化规划仅使用变异与选择算子，而绝对不使用任何重组算子。其变异算子与进化策略的变异相类似，也是对父代个体采用基于正态分布的变异操作进行变异，生成相同数量的子代个体。即μ个父代个体总共产生μ个子代个体。EP 采用一种随机－竞争选择方法，从父代和子代的并集中选择出μ个个体构成下一代群体。其选择过程如下：对于由父代个体和子代个体组成的大小为 2μ的临时群体中的每一个个体，从其它 2μ－1 个个体中随机等概率地选取出 q 个个体与其进行比较。在每次比较中，若该个体的适应值不小于与之比较的个体的适应值，则称该个体获得一次胜利。从 2μ个个体中选择出获胜次数最多的μ个个体作为下一代群体。

1.2.5.遗传程序设计(Genetic Programming，GP)

遗传程序设计是由 Stanford 大学的 J.R.Koza 在 20 世纪 90 年代初提出的，并于 1992 年出版了专著“ Genetic Programming ”[Koza 1992]。GP 采用遗传算法的基本思想，使用一种更为灵活的分层结构来表示解空间。这种分层结构的叶结点是问题的原始变量，中间结点则是组合这些原始变量的函数。它们很类似于 LISP 语言中的 S－表达式。这样的每一个分层结构对应问题的一个解，也可以理解为求解该问题的一个计算机程序。遗传程序设计是使用一些遗传操作动态地改变这些结构以获得解决该问题的可行的计算机程序。

遗传程序设计所采用的主要遗传操作是选择与交叉算子，GP 的选择操作是自然选择，适者生存的基本动因，其操作对象是父代 S－表达式，产生的结果是子代 S－表达式，选择操作方法与遗

传算法相同。

GP 交叉操作的对象是两个父代 S－表达式，交叉操作的结果产生两个子代 S－表达式，其中每一个体(S－表达式)都含有来自两个父代个体的部分基因。对每一个父代个体，采用均匀分布随机选择交叉点，然后相互交换交叉点以下的子树。由于 GP 染色体结构的形状和大小不是固定的，因而参与交叉的两个父代个体的大小一般是不相同的。

为了动态修改染色体的结构，GP 提供了变异操作、排列操作、编辑操作、封装操作、＋中抽－等辅助算子。

1.3. 群体智能算法(Swarm Intelligence Algorithm)

群体智能算法的研究开始于上世纪 90 年代初，其基本思想是模拟自然界生物的群体形为来构造随机优化算法。典型的方法有 M.Dorigo 提出的蚁群算法和 J.Kennedy 与 R.Eberthart 提出的微粒群算法。

1.3.1. 蚁群算法(Ant Colony Algorithm)

蚁群算法也称蚂蚁算法，是在 20 世纪 90 年代初由意大利学者 M.Dorigo 提出的，它是根据蚂蚁觅食原理而设计的一种群体智能算法。下面首先简单介绍一下蚂蚁觅食的方法，然后以 TSP 问题为例给出求解其最短回路蚁群算法。

据研究，当蚂蚁找到食物并将它搬回来时，就会在它经过的路上留下一种“ 外激素 ”，其它蚂蚁闻到这种激素的“ 味道 ”，就沿该路线去觅食，而且还会沿着最短的路径奔向食物。下面以 n 个城市的 TSP 问题为例，简要地介绍一下人们根据蚂蚁觅食原理设计出的求解最短回路的蚁群算法。

设共有m个蚂蚁，第i城市在t时刻有蚂蚁$a_i(t)$个，在t时刻在第i，j两城市间的道路上留下的外激素量为$b_{ij}(t)$。假设每个蚂蚁在未完成一个回路时，不重复已走过的城市，则第k个蚂蚁从城市i到城市j的概率为：

$$p_{ij}^{k} = \frac{b_{ij}(t)}{\sum b_{ij}(t)} \tag{1.7}$$

其中外激素量$b_{ij}(t)$可定义为$b(t) = e^{-ct}$，c>0；或定义为：

$$d_{ij} = \sum_{k=1}^{m} \frac{e}{L_k} a_{ij}^{k}(t) \tag{1.8}$$

其中 L_k 是第 k 只蚂蚁求得的回路长度。

$$b_{ij}(t+1) = c \bullet b_{ij}(t) + d_{ij} \tag{1.9}$$

$$a_{ij}^{k}(t) = \begin{cases} 1 & \text{第}t\text{轮第}k\text{只蚂蚁经过边}(i, j) \\ 0 & Else \end{cases} \tag{1.10}$$

这样，每只蚂蚁经过 n 次迁移后就得到一条回路，其长度为 L_k。再利用(1.7)~(1.10)重新计算各条路径的外激素浓度，进行下一步搜索。

蚁群算法自提出以来，已成功应用于许多领域，如 TSP、重建通讯路由、连续系统优化等许

多领域。

1.3.2.微粒群算法(Particle Swarm Optimization, PSO)

微粒群算法是在 1995 年由美国社会心理学家 James Kennedy 和电气工程师 Russell Eberhart 共同提出的[Kennedy 1995]，其基本思想是受他们早期对鸟类群体行为研究结果的启发，并利用了生物学家 Frank Heppner 的生物群体模型。

有关微粒群算法的详细介绍是本书后续各章节的内容，在这里就不作阐述，下一节将对微粒群算法的发展历史及研究方向作一简单介绍。

1.4. 微粒群算法的发展

自微粒群算法提出以来，由于它的计算快速性和算法本身的易实现，引起了国际上相关领域众多学者的关注和研究，其研究大致可以分为：算法的改进、算法的分析以及算法的应用。下面就这三个方面的研究情况作一简单的介绍。

1.4.1．微粒群算法综述

在微粒群算法的改进方面，首先是由 Kennedy 和 Eberhart 在 1997 年提出的二进制 PSO 算法[Kenney 1997]，为 PSO 算法与遗传算法的性能比较提供了一个有用的方式，该方法可用于神经网络的结构优化，其具体方法参见第三章。

其次，为了提高算法的收敛性能，Shi 和 Eberhart 于 1998 年对 PSO 算法的速度项引入了惯性权重 w[Shi 1998]，并提出在进化过程中动态调整惯性权重以平衡收敛的全局性和收敛速度，该进化方程已被相关学者称之为标准 PSO 算法。Clerc 于 1999 年在进化方程中引入收缩因子以保证算法的收敛性[Clerc 1999]，同时使得速度的限制放松。有关学者已通过代数方法对此方法进行了详细的算法分析，并给出了参数选择的指导性建议。

Angeline 于 1999 年借鉴进化计算中的选择概念，将其引入 PSO 算法中。通过比较各个微粒的适应值淘汰掉差的微粒，而将具有较高适应值的微粒进行复制以产生等数额的微粒来提高算法的收敛性。而 Lovbjerg 等人进一步将进化计算机制应用于 PSO 算法，如复制、交叉等，给出了算法交叉的具体形式，并通过典型测试函数的仿真实验说明了算法的有效性。

为了提高 PSO 算法收敛的全局性，保证微粒的多样性是其关键。为了保证进化过程中群体中微粒的多样性，Suganthan 在标准 PSO 算法中引入了空间邻域的概念，将处于同一个空间领域的微粒构成一个子微粒群分别进行进化，并随着进化动态地改变选择阈值以保证群体的多样性；Kennedy 引入邻域拓扑的概念来调整邻域的动态选择，同时引入社会信念将空间邻域与邻域拓扑中的环拓扑相结合以增加邻域间的信息交流，提高群体的多样性。Lovbjerg 等人于 2001 年将遗传算法中的子群体概念引入 PSO 算法中，同时引入繁殖算子以进行子群体的信息交流。

在 PSO 算法的行为分析和收敛性分析方面进行了大量的研究工作。首先是采用代数方法对几种典型的 PSO 算法的运行轨迹进行了分析，给出了保证收敛性的参数选择范围。在收敛性方面，Frans van den Bergh 引用 Solis 和 Wets 关于随机性算法的收敛准则，证明了标准 PSO 算法不能收敛于全局最优解，甚至于局部最优解。证明了保证收敛的 PSO 算法能够收敛于局部最优解，而不能保证收敛于全局最优解。

在PSO算法的应用方面，PSO算法最早应用于人工神经网络的训练方法，Kennedy和Eberhart成功地将PSO算法应用于分类XOR问题的神经网络训练。随后，PSO算法在函数优化、约束优化、极大极小问题、多目标优化等问题中均得到了成功的应用。特别是在电力系统、集成电路设计、系统辨识、状态估计等问题中的应用均有报导。有兴趣的读者可参考相关文献[Yoshida 1999, Fukuyama 2001]。

1.4.2.微粒群算法的研究方向

根据作者对国内外关于微粒群算法研究的相关文献以及进化算法领域的发展趋势的分析，认为目前主要有以下几个研究方向：

1) 微粒群算法的改进。

标准微粒群算法主要适用于连续空间函数的优化问题，如何将 PSO 算法应用于离散空间优化问题，特别是一类非数值优化问题，将是微粒群算法的主要研究方向。另外充分吸引其它进化类算法的优势，以改进 PSO 算法存在的不足也是值得研究的问题。

2) 微粒群算法的理论分析

到目前为止，PSO 算法的分析方法还很不成熟和系统，存在许多不完善和未涉及到的问题。如何利用有效的数学工具对 PSO 算法的运行行为、收敛性及计算复杂性进行分析是目前的研究热点之一。

3) 微粒群算法的生物学基础

如何根据群体进化行为完善 PSO 算法，同时分析群体智能行为，如何将其引入 PSO 算法中，以充分借鉴生物群体进化规律和进化的智能性也是目前的研究方向之一。

4) 微粒群算法与其它类进化算法的比较研究

5) 微粒群算法的应用

算法研究的目的是应用，如何将 PSO 算法应用于更多领域，同时研究应用中存在的问题也是值得关注的热点。

由于 PSO 算法提出的时间并不长，存在的问题很多，本书的目的在于对目前的研究状况作一比较系统的介绍，肯定存在挂一漏百的现象。同时，作者们从事这一研究也仅仅两年多时间，掌握的资料有限，难免有许多不当之处，敬请读者提出宝贵意见。