

北京邮电大学
硕士学位论文
关于快速变化环境中简单进化策略的理论分析
姓名：王蔚
申请学位级别：硕士
专业：计算机应用
指导教师：马严
20031124

摘要：使用进化算法是解决动态优化问题的一条有效途径。但由此产生的问题是：动态问题本身实际上是独立于算法的，且随时可能发生变化，而现有的算法都假设环境只在相邻两世代间才发生变化。本文首次关注这样一个课题：进化算法如何应付世代过程中的快速环境变化？

以Stanhope和Daida的相关工作为主要基础，重点对用 $(1+1)$ 、 $(1,2)$ 和 $(1,4)$ 这几个简单EA解决动态比特匹配问题进行理论研究。我们首先推导出 $(1,2)$ 繁殖方案的理论模型，并且在世代间环境变化的前提下将它的性能与 $(1+1)$ 繁殖方案作了比较。然后转向世代内环境变化的论题，比较了采用 $(1,2)$ 繁殖方案的两条思路：始终用最新的适应度函数还是整个世代都用旧的适应度函数。其中为解决重复使用同一个随机变量进行概率计算的相关性问题，提出了两阶段变异的分析方法。该方法的正确性被统计试验证明，并被进一步用来分析 $(1,4)$ 问题。研究表明，在一定条件下，暂时忽略世代内的环境变化，能够比使用最新的适应度函数评估个体获得更好的性能。本文最后探讨了进化算法在下一代网络传输控制中的应用，提出了解决动态QoS组播路由选择问题的进化策略，并用仿真实验证实了该策略的有效性。

关键词：进化算法 进化策略 QoS 组播路由

Abstract. Using Evolutionary Algorithms (EAs) is one of the effective approaches to dynamic optimization problems. Another problem occurs that: Dynamic problems themselves are actually independent from the algorithms, thus can change at any time. However, all current algorithms have an assumption that the environment changes only *between* two neighbouring generations. This thesis takes such a topic into account for the first time: How EAs deal with environments changing quickly *during* a generation? Based mainly on related work done by Stanhope & Daida, we focus on a theoretical investigation of a simple $(1+1)$, $(1,2)$ and $(1,4)$ EA on the dynamic bit matching problem. We derive the theoretical model for the $(1,2)$ reproduction scheme and compare its performance to the $(1+1)$ reproduction scheme, assuming the environment changes *between* generations. Then we turn to the issue of changes *during* a generation and compare the ideas of always using the up-to-date fitness function or using an old fitness function for the whole generation for the $(1,2)$ reproduction scheme. To deal with the difficulty of correlation between twice counting on the same variable, analysis method of two-step mutation is proposed. The method is validated by empirical trials, and used on the $(1,4)$ analysis as well. Results shows that on certain conditions it's better to ignore the environmental changes until the end of the generation, than to evaluate each individual with the most up-to-date fitness function. Finally we discuss the issue of applying EAs to transmission control of the next generation network. We propose a new strategy of EA for a dynamic QoS-based multicast routing problem. The simulation results suggest that the scheme is feasible and effective.

Keywords: Evolutionary algorithm Evolution strategy QoS Multicast routing

第1章 绪论

1.1 进化计算简介

自然界始终是人类灵感的重要来源。仿生学直接模仿生物界的现象和原理，而另外一些研究方向则起源于对自然现象或过程的模拟，如控制论，人工神经网络，模拟退火算法，元胞自动机等。遗传算法（Genetic Algorithm, GA）也是其中之一，它是一种借鉴生物界自然选择和进化机制发展起来的高度并行、随机、自适应搜索算法。它最先由John Holland于1975年提出。从那以后，逐渐发展成为一种通过模拟自然进化过程解决最优化问题的计算模型，也称为进化计算（Evolutionary Computation）。

1.1.1 算法工作原理及特点

最优化问题通常可归结为极大化问题，可用公式描述为：

$$\max_{x \in S} f(x) \quad (1.1)$$

其中 $f(x)$ 为目标函数， S 为可行域，它们是由工程实际问题的具体条件决定的。

计算的基本过程是：首先采用某种编码方式将解空间映射到编码空间（可以是二进制位串、实数、有序串、树或图，Holland最初的遗传算法是基于二进制串的，类似于生物染色体结构，易于用生物遗传理论解释，各种遗传操作也易于实现。另外，可以证明，采用二进制编码式，算法处理的模式最多。但是，在具体问题中，直接采用解空间的形式进行编码，可以直接在解的表现型上进行遗传操作，从而易于引入特定领域的启发式信息，可以取得

比二进制编码更高的效率。实数编码一般用于数值优化,有序串编码一般用于组合优化),每个编码对应问题的一个解,称为染色体(chromosome)或个体(individual)。然后在可行域中随机挑选一些编码组成作为进化起点的第一代编码组(称为种群, population)。接着在种群中根据适应度或某种竞争机制选择个体(适应度fitness代表解的满意程度,可以由外部显式适应度函数计算,也可以由系统本身产生,如由协同演化时不同对策的博弈确定,或者由个体在群体中的存活量和繁殖量确定),接下来使用各种遗传操作算子(包括杂交,变异,倒位等等)繁殖产生下一代(generation)。下一代既可以完全替代原种群,也可以部分替代原种群中一些较差的个体。……如此重复上述选择和繁殖过程而不断进化,直到满足期望的终止条件。进化过程最后一代中的最优解就是用进化算法解最优化问题所得到的最终结果。

从上面的原理可以看出,进化算法就像自然界中优胜劣汰一样,利用选择机制从编码组中随机挑选编码作为繁殖过程前的编码样本。选择机制保证了适应度较高的解能够保留较多的样本;而适应度较低的解则保留较少的样本,甚至被淘汰。算法因而具有如下一些特点:

1. 进化算法从问题的解集开始搜索,而不是从单个解开始。这是进化算法与传统优化算法的极大区别。传统优化算法是从单个初始值迭代求最优解的,容易误入局部最优解。进化算法从串集开始搜索,如同在搜索空间上覆盖的一张网,覆盖面大,利于全局择优。

2. 进化算法求解时使用特定问题的信息极少,容易形成通用算法程序。由于进化算法使用适应度这一信息进行搜索,并不需要问题导数等与问题直接相关的信息。进化算法只需适应值和串编码等通用信息,不必要求连续性、可导或单峰等。对领域知识依赖程度低,不受搜索空间限制性假设的约束。故几乎可处理任何问题。

3. 进化算法有极强的容错能力 进化算法的初始串集本身就带有大量与最优解甚远的信息;通过选择、交叉、变异操作能迅速排除与最优解相差极大的串;这是一个强烈的滤波过程;并且是一个并行滤波机制。故而,进化算法有很高的容错能力,能很快适应问题和环境的变化。

4. 进化算法中的选择、交叉和变异都是随机操作，而不是确定的精确规则。这说明进化算法是采用随机方法进行最优解搜索，选择体现了向最优解逼近，交叉体现了最优解的产生，变异体现了全局最优解的覆盖。

5. 进化算法具有隐含的并行性(Implicit Parallelism)，非常适合于并行计算。实践表明，遗传算法解最优化问题的计算效率比较高、适用范围相当广。为了解释这一现象，Holland给出了图式定理。所谓图式，就是某些码位取相同值的编码的集合。图式定理说明在进化过程的各代中，属于适应度高、阶数低且长度短的图式的编码数量将随代数以指数形式增长。另外，Holland还发现遗传算法具有隐含的并行计算特性。最近的研究则表明，上述遗传算法经适当改进后对任意优化问题以概率1收敛于全局最优解。

由于上述特点，进化算法特别适合于处理传统搜索算法解决不好的复杂的和非线性问题。80年代后，遗传算法得到了迅速发展，以遗传算法为核心的进化算法已与模糊系统理论、人工神经网络等一起成为计算智能研究中的热点，受到许多学科的共同关注。不仅理论研究十分活跃，而且在越来越多的应用领域中得到应用。

1.1.2 进化算法的应用和研究方向

随着对进化算法研究的不断深入完善，其应用范围也越来越广泛，已涉及机器学习、模式识别、图像处理、神经网络、工业优化控制和社会科学等方面。特别是在解决旅行商问题、煤气管道的最优控制、铁路运输计划优化、喷气式收音机涡轮机的设计等问题上进化算法都取得了很大的成功。列举如下。

- 控制：煤气管道控制，防避导弹控制，机器人控制
- 规划：生产规划，并行机任务分配
- 设计：VLSI版面设计，背包问题，图划分问题，键盘排列优化
- 图像处理：模式识别，特征抽取
- 信号处理：滤波器设计

- 机器人：路径规划
- 人工生命：生命的遗传进化
- 人工神经网络：权值训练和网络结构生成。

在通信领域，进化计算理论同样应用广泛。例如，在信息网络设计方面，它可用于移动网络频率优化、通信网设计优化（参考文献[11]）；在网络智能控制方面，可完成优化路由选择、容量分配、时延和流量控制（文献[12]、[13]、[14]）；此外，可用进化算法建立通信系统可靠性最优分配模型（文献[15]），等等。

目前在进化计算的研究中主要有三类研究方向：1. 研究进化算法本身的理论基础；2. 用进化算法作为工具解决工程问题，主要是进行优化，关心的是是否能在传统方法上有所提高；3. 用进化算法研究演化现象，一般涉及到人工生命和复杂性科学领域。

1.2 本课题的背景、目的和内容

许多优化问题，如通信系统中的受控参数所处的周边环境，都是随时间动态变化的。因此，为了能反映这些变化，好的优化算法必须不断调整最优解使之适应变化的目标函数。进化算法（Evolutionary Algorithm, EA）从自然界的进化过程获得灵感，自然进化本质上是在随机动态环境（dynamic and stochastic environment）中的适应过程。如此看来，EA自然可应用于动态优化（dynamic optimization）问题。尽管近来的研究表明，标准的EA由于收敛过快而无法适应持续变化的环境，但是科学家们已经提出了许多办法来克服这个难题，并随之出现了一些很有希望的变种EA（例如可参见[2]）。

EA通过多次循环以实现优化，每次循环（在EA中称为“世代”，generation，下同）根据决策规则（繁殖方案，reproduction scheme）产生一定数量的解（个体，individual），对其进行评估（evaluate，即计算该个体的适应度，fitness），然后把它们加入到当前解集（种群，population）中。迄今为止，至少是基于作者的知识，所有的EA方法都基于这样的假

设：环境（environment，即适应度函数，fitness function）仅在两个相邻的世代之间发生变化。然而，比上述假设更普遍的情形是：环境是独立于进化算法本身的，因而可能在一个世代过程当中的任何时刻发生变化。

在本文中，我们专注于这样一个问题：EA如何应付一个世代过程中的环境变化。

在科技文献中，人们为EA提出了各种繁殖方案。在下面的列表中，前两个常见于关于遗传算法的文献中，而后两个则来源于进化策略（evolution strategy）文献。

- **整代繁殖**（generational reproduction）每个世代老种群中的 n 个（ n 是种群个体总数（population size））母体（parent）被相等数目的后代（child）完全取代。该方案通常与精英策略（elitism）结合使用，即老种群中最佳的 e 个（称为“精英”，elite）个体被保留下来，与产生的 $n-e$ 个新个体组成后代。

如果适应度函数在两个世代之间发生变化，则需要重估（reevaluate）精英以保证它们与新个体具有一致的评估标准。但如果适应度函数在对后代的评估期间发生变化，则至少有以下策略可供考虑：

- 重估所有已生成的后代。当然，如果适应度函数频繁变化，这是一种代价高昂的办法。
- 忽略该变化，继续使用旧的适应度函数评估当前世代的所有后代。这种做法忽视所有环境变化的信息，直到该世代结束为止。
- 使用变化后的适应度函数评估以后的个体，同时保留已被评估的那些后代的适应度。这里的问题是，参与相互竞争的个体们被用新、旧两个不同的适应度函数进行评估。
- 临时增加精英个体的数目。这种情况时，不再产生后代，而是用老种群的个体“填充”当前种群。
- 临时减少种群个体的总数。结束当前的世代，并以已生成的后代为基础生成下世代种群（个体总数仍为 n ）。

- **定态繁殖** (steady state reproduction) 这是一种粒度更小、更有连续性的繁殖方案。在每个世代仅产生一个新个体，用它取代当前种群中适应度最差的那一个体。这种方案也可以被看作是整代繁殖方案在精英数目多达 $n - 1$ 时的特例。

由于每个世代仅生成一个后代，无须区分环境变化是发生在世代间还是在世代内。所以，该繁殖方案不在我们的研究之列。尽管如此，还是让我们简单讨论一下它在动态环境中的应用。在动态环境中使用定态繁殖的问题在于：一些个体因此而可能永久地生存下去，而且为保证评估标准一致，每当适应度函数变化后，必须重估整个种群。[10]给出了一种折衷办法，后代要取代的不是种群中适应度最差的，而是最老的那一个体，并且沿用所有旧适应度评估值。这时，该方法变得与整代方法非常相似，因为一个个体最多只能存活 n 个世代。和整代繁殖遇到的问题一样，相互竞争的个体们被不同的适应度函数所评估。另一个派生方法叫做**保守选择** (conservative selection)，出现于[7]。后代还是用于取代最差个体，但以锦标赛形式选拔母体，且最老个体总是参加选拔。关于此问题的其他派生解法，请参阅[4, 5]。

- **(μ, λ) -繁殖** 这种情况下，母体种群的个体数为 μ ，每个世代都从这些母体产生 $\lambda > \mu$ 个后代，其中最佳的 μ 个后代组成新的母体种群。

如果从个体寿命有限的角度考虑，这种方法在动态环境中的行为和整代方法是相似的。然而，相互竞争的个体们基于不同适应度函数的问题却变得更严重了，因为在 λ 个后代中只有 μ 个所谓“最佳的”个体才有机会繁殖。不过面对环境的变化，该方法所采取的可能的对策是与整代繁殖相似的。

- **$(\mu + \lambda)$ -繁殖** 同样地，母体数量为 μ ，而后代数量为 λ 。但这次，后代和母体们都参与生存竞争，新的母体种群由旧母体种群和所有后代中的 $\mu + \lambda$ 个最佳个体组成。

这里，正如定态繁殖那样，个体们能够无限期地生存下去，而不管事实上自从被评估以来它的实际适应度或许已经明显地下跌了。所以，为避免这

种“一好永好”，当环境变化后他们必须被重估。但是如果每个世代创造 $\lambda > 1$ 个后代，还会有环境在世代过程中变化的可能。下面是一些合理的抉择：

- 重估所有母体和产生的后代。当变化频繁时这当然是代价高昂的。
- 忽略该变化并继续使用旧的适应度函数直到评估完本世代的所有后代。这忽略了该世代结束前的环境变化信息。
- 使用变化了的适应度函数评估所有后面的个体，同时保留已被评估的那些后代的适应度。这里的问题是，参与相互竞争的个体们被用新、旧两个不同的适应度函数进行评估。
- 临时减少后代的数目 λ 。结束当前的世代，并从 μ 个老个体和已生成的后代中选拔最佳的 μ 个组成新种群。

注意上述对付变化的可能中有些是假设了旧的评估函数在环境变化之后仍然能被使用，这应该是合理的。因为在通常情况下，适应度是通过在计算机上模拟的环境加以评估的。所以，继续使用旧适应度函数仅仅意味着延迟修改环境的计算机模型而已。

本文中，我们重点对用 $(1+1)$ 、 $(1,2)$ 和 $(1,4)$ 这几个简单的EA解决动态比特匹配问题（dynamic bit matching problem）进行理论研究。动态比特匹配问题是众所周知的onemax问题的动态变种：优化目标是再现一个时变的二进制目标样板（object template），目标函数取值为当前解与样板相等的比特数。

本文的组织结构如下：下一章将用形式化方法定义基本概念并描述所研究的问题。接下来第3章简要提及和我们的研究工作相关的文献，其中特别有一篇论文是我们工作的基础，将较详细地讨论。第4章将推导出 $(1,2)$ 繁殖方案的理论模型，并且在世代间环境变化的前提下将它的性能与 $(1+1)$ 繁殖方案作了比较。第5章我们转向世代内环境变化的论题，比较了采用 $(1,2)$ 繁殖方案的两条思路：始终用最新的适应度函数与整个世代都用旧的适应度函数。最后是总结和今后工作的展望。

第2章 基本定义

本章从进化计算的角度定义了动态比特匹配问题和以其为优化目标的进化算法，并讨论了它们的各种组合情况，作为后继讨论的基础。

2.1 (d, e^{th}) 动态环境

进化计算用**动态环境** (Dynamic Environment, DE) 来统称动态优化目标函数 (简称目标)。动态比特匹配问题目标函数的特征集中体现为一个称做**样板** (template) 的时变二进制串 t 。

- 有关数据的定义

- 样板长度: $L \in \mathbb{N}$ 为样板所含比特数。今后结果报告中默认 $L = 100$
- 时间标记: $\tau \in \mathbb{Z}^0$ 代表相邻两次样板变化之间的任意一个时刻
- τ 时刻样板: $t_\tau = \{(t_\tau)_i | (t_\tau)_i \in \{0, 1\}, i \in I = \{1, \dots, L\}\}$

- 有关运算的定义

- 初始化 (initialize): 令 $\tau = 0$, 随机化 t_0 的每个比特
- 评估 (evaluate): 以**适应度函数**反映个体对环境的适应程度。
 $f(b, t_\tau) = f_\tau(b) = |\{(b)_i | (b)_i = (t_\tau)_i, i \in I\}|$ 称为个体 b 在 τ 时刻环境中的**适应度**。为便于阅读, 常简写为 $f(b)$
- 环境变化通过定期更新样板实现: 每 e 次评估后, 将原样板中任意 d 比特翻转而得到新样板, 记为 $m_d(t_\tau)$
已知 $d \in I$, 任选 $\forall D \subseteq I, |D| = d, t_{\tau+1} = m_D(t_\tau) = \{(t_\tau)_j | j \notin D\} \cup$

$\{\neg(t_\tau)_j | j \in D\}$ 称为**变化后的环境**，这里 \neg 表示二进制取反。今后在不引起歧义的前提下，我们将 $m_D(t_\tau)$ 与 $m_d(t_\tau)$ 混用。 $f(b, t_{\tau+1}) = f_{\tau+1}(b) = |\{(b)_i | (b)_i = (t_{\tau+1})_i, i \in I\}|$ 称为个体 b 在（环境）变化后的**适应度**，即 b 与下一个目标比较而得的适应度。

2.2 仅含变异的进化算法

我们所要研究的**进化算法**（Evolutionary Algorithm, EA）是一种仅含变异操作和简单进化策略的循环算法。每次循环称为一个世代，根据繁殖方案，从老种群的母体个体经过变异而产生一定数量的后代个体，然后择优加入组建新种群。当新种群的个体总数和老种群相等时，新种群取代老种群而发生世代交替，以新种群中的个体为母体开始下一世代。

- 有关数据的定义

- 个体用小写字母表示，例如个体 a 定义为： $a = \{(a)_i | i \in I, (a)_i \in \{0, 1\}\}$
- 种群个体总数： μ

- 有关运算的定义

- 初始化：算法开始时，随机化 μ 个个体的每一比特
- 变异（mutate）：已知变异率（mutation rate） $r \in I$ 为变异的比特数，任选 $\forall R \subseteq I, |R| = r, m_R(a) = \{(a)_j | j \notin R\} \cup \{\neg(a)_j | j \in R\}$ 为个体 a 发生了 r -比特变异而得到的结果，记为 $m_r(a)$ 。今后将 $\{(a)_j | j \in R\}$ 简写为 a_R ；在不引起歧义的前提下，我们将 $m_R(a)$ 与 $m_r(a)$ 混用
- 选择（select）： ψ 个个体 a_1, a_2, \dots, a_ψ 中具有最大适应度的最佳个体 a_{max} 被定义为

$$a_{max} = \arg \max(f_{a_1}(a_1), \dots, f_{a_\psi}(a_\psi))$$

$$= \arg \max_{b \in \{a_1, \dots, a_\psi\}} f_b(b)$$

如果对于 $\forall b \in \{a_1, a_2, \dots, a_\psi\}$, 都有 $f_{a_{max}}(a_{max}) \geq f_b(b)$

此处 $f_b(b)$ 、 $f_{a_{max}}(a_{max})$ 分别表示个体 b 和 a_{max} 在其对应环境下的适应度, 且该适应度被用于参与本次选择竞争。 $\arg\max_{\Omega} f$ 代表集合 Ω 中使函数 f 取得极大值的元素。由于多个个体可能同时具有最大适应度, 所以 a_{max} 的选择要视具体算法而定。

• 进化策略 (evolution strategy)

- (μ, λ) -繁殖: 母体种群个体总数为 μ , 每个世代从这些母体变异产生 $\lambda > \mu$ 个后代。选择其中最佳的 μ 个组成新母体种群
- $(\mu + \lambda)$ -繁殖: 这里 $\lambda \geq 1$ 。与上面策略不同在于, 该策略选择最佳的 μ 个个体的范围不仅包含所有后代, 而且包括母体, 共有 $\mu + \lambda$ 个候选对象
- 重估 (reevaluate): 为了跟上环境的变化, EA策略可以在最新的环境下重估个体。我们把这叫做**显式重估** (explicit reevaluation)。与正常评估一样, 显式重估被环境变化触发计数器记录。另外一方面, 为了分析的目的, 即使在EA策略中没有定义 (当然EA实际上也不会执行重估), 我们也必须在最新的环境下重估被选中的个体。我们称之为**隐式重估** (implied reevaluation)。隐式重估仅仅是一个分析步骤, 它不是EA的运行步骤, 所以不会触发环境变化。

2.3 DE与EA的不同组合情形

为了在一个具体的动态环境中分析一个具体的遗传算法, 需要考虑可能出现的不同情形。例如 $(d, 2^{th})$ DE中的 $(1, 2)$ EA, 我们采用类似C和C++的伪码分别描述算法如下。

情形 2.1 一个在两世代之间变化的 $(d, 2^{th})$ DE中的 $(1, 2)$ EA

```

 $\tau = 0$ ;
initialize( $t_0$ );
initialize(parent);
fitness =  $f_0$ (parent);
while (NOT termination condition) do
    child1 =  $m_r$ (parent);
    child2 =  $m_r$ (parent);
    fitness1 =  $f_\tau$ (child1);
    fitness2 =  $f_\tau$ (child2);
    if (fitness1  $\geq$  fitness2){
        parent = child1;
        fitness = fitness1;
    }
    else{
        parent = child2;
        fitness = fitness2;
    }
     $t_{\tau+1} = m_d(t_\tau)$ ;
     $\tau = \tau + 1$ ; //环境变化后需要隐式重估入选后代
enddo

```

情形 2.2 一个在世代内变化的 $(d, 2^{th})$ DE 中的 $(1, 2)$ EA

```

 $\tau = 0$ ;
initialize( $t_0$ );
initialize(parent);
fitness =  $f_0$ (parent);
while (NOT termination condition) do
    child1 =  $m_r$ (parent);
    child2 =  $m_r$ (parent);
    fitness1 =  $f_\tau$ (child1);
     $t_{\tau+1} = m_d(t_\tau)$ ;
     $\tau = \tau + 1$ ;
    fitness2 =  $f_\tau$ (child2);
    if (fitness1  $\geq$  fitness2){
        parent = child1;
        fitness = fitness1; //需要隐式重估入选个体（后代一）
    }
    else{
        parent = child2;
        fitness = fitness2;
    }
enddo

```

正如我们从以上情形中看到的，它们的差异来自于环境变化的不同开始点。当 $\lambda \geq 2$ 且 $1 \leq e \leq \lambda/2$ 时，甚至仅仅用“世代内变化”也无法准确地描

述环境变化，因为环境可能在新母体种群生成过程中的多个时间点发生变化。所以始终有必要给出程序代码来详细描述被分析的具体情形。

如果确定了环境参数和进化策略，并取世代作为时间间隔，取母体适应度作为状态，则在当前世代母体适应度已知的情况下，下一世代母体适应度转移概率仅和当前世代的变异率有关，而与以前的状态无关，从而具有无后效性。可以证明，对 (d, e^{th}) DE中简单进化策略EA适应度变迁的控制过程实质是一个离散时间马尔可夫决策过程 (Markovian decision process)。决策优化目的在于寻求能使下个世代母体适应度数学期望最高的当前世代**最优变异率**，其关键前提是求出个体适应度转移概率的解析式。

第3章 相关工作

3.1 Stanhope与Daida所做的前导工作

我们以Stanhope和Daida发表在[8, 9]中的研究成果为基础, 并对其做了较显著的开拓工作。他俩以动态比特匹配问题为优化目标, 分析推导出了 $(1+1)$ EA个体适应度转移概率公式。在描述我们的工作前, 有必要简要回顾一下他们的成果, 以其作为我们扩展工作的基线。下面是他们所研究的EA-DE组合情形。

情形 3.1 对入选个体作显式重估的 $(1+1)$ EA位于在世代间变化的 $(d, 2^{th})$ DE中

```
 $\tau = 0;$   
 $initialize(t_0);$   
 $initialize(parent);$   
 $fitness = f_0(parent);$   
 $while (NOT\ termination\ condition) do$   
     $child = m_r(parent);$   
     $if (f_\tau(child) > fitness) //这里使用 > 还是 \ge 不会影响最终适应度$   
         $parent = child;$   
     $t_{\tau+1} = m_d(t_\tau);$   
     $\tau = \tau + 1;$   
     $fitness = f_\tau(parent); //显式重估$   
 $enddo$ 
```

他们首先证明了以下引理:

引理 3.1 (变异的有效比特数) 假定 $\forall R \subseteq I, |R| = r$, 定义 $a_R = \{(a)_i | i \in R\}$, 设 $f(m_R(a)) = x$, 定义变异的有效比特数 r_n 并计算如下:

$$r_n = |\{(a)_i | i \in R, (a)_i \neq (t_\tau)_i\}|$$

$$\begin{aligned}
&= |\{(a)_j | (a)_j \in (m_R(a))_R, (a)_j = (t_\tau)_j\}| \\
&= f((m_R(a))_R, (t_\tau)_R) \\
&= \frac{x - f(a) + r}{2}
\end{aligned} \tag{3.1}$$

为便于分析比较，他们从一个具有确定适应度 $f(a)$ 的母体个体 a 着手，证明了以下定理。

定理 3.1 (变异后适应度概率分布) 假定母体个体 a 的适应度为 $f(a)$ ，从 a 变异产生一个后代，变异率为 r ，则其适应度为 x 的概率可以计算如下：

$$P(f(m_r(a)) = x) = \frac{\binom{L-f(a)}{r_n} \binom{f(a)}{r-r_n}}{\binom{L}{r}} \tag{3.2}$$

接着他们给出了在 a 和 $m_r(a)$ 之间进行选择的结果概率。(1+1) EA中，两个个体（母体和后代）中适应度较高的那个才能生存下来，存活个体的适应度为 x 概率计算如下：

$$P(f(\arg \max_{b \in \{a, m_r(a)\}} f(b)) = x) = \begin{cases} 0 & : x < f(a) \\ \sum_{i=0}^x P(f(m_r(a)) = i) & : x = f(a) \\ P(f(m_r(a)) = x) & : x > f(a) \end{cases} \tag{3.3}$$

然后他们分析了入选个体的显式重估。为了计算环境变化对适应度的影响，首先注意到，从个体适应度概率分布的角度来说，目标上的 d 比特变化等于个体上的 d -比特变异。

定理 3.2 (环境变化后的适应度概率分布)

$$\begin{aligned}
\forall d \in I, P(f_{\tau+1}(a) = x) &= P(f(a, m_d(t_\tau)) = x) \\
&= P(f(m_d(a), t_\tau) = x) \\
&= P(f(m_d(a)) = x)
\end{aligned} \tag{3.4}$$

这意味着在已变化环境下可以如下重估一个个体：将个体中与环境变化比特位置相同的比特做同样的变化，并在旧环境下对其评估。所以他们找到了一

个可以将动态环境变换成静态环境的办法，可以如下来计算入选个体在环境变化后的适应度概率分布函数：

$$\begin{aligned} & P(f_{\tau+1}(\arg \max_{b \in \{a, m_r(a)\}} f(b)) = x) \\ &= \sum_{i=0}^L P(f(\arg \max_{b \in \{a, m_r(a)\}} f(b)) = i) P(f(m_d(c_i)) = x) \end{aligned} \quad (3.5)$$

这里 c_i 表示任一适应度为 i 的个体。这种假想方式不仅不会影响概率计算结果，而且有助于简化概率分析过程，所以今后我们将反复采用。

在分析上述研究成果的基础上，我们还得出以下关于选择的更一般的结论，用于指导后面的研究。

推论 3.1 （无优先权的选择结果概率分布）

设 $\arg \max_{b \in \{a_1, \dots, a_\psi\}} f_b(b) = a_{max}$ ，则

$$P(f_{a_{max}}(a_{max}) = x) = \prod_{i=1}^{\psi} P(f_{a_i}(a_i) \leq x) - \prod_{i=1}^{\psi} P(f_{a_i}(a_i) < x) \quad (3.6)$$

对于情形3.1，将 $\psi = 2, a_1 = a, a_2 = m_r(a), P(f_{a_1}(a_1) = f(a)) = 1$ 代入上式，可得公式(3.3)。

推论 3.2 （某一具有优先权的个体因其适应度不小于其它个体而入选的概率）

已知 $\omega \in \{1, \dots, \psi\}$ ，设 $\arg \max_{b \in \{a_1, \dots, a_\psi\}} f_b(b) = a_{max}$ 除非有其它个体的适应度大于 $f_{a_\omega}(a_\omega)$ ，否则将优先选择 a_ω ，则

$$\begin{aligned} & P(f_{a_{max}}(a_{max}) = f_{a_\omega}(a_\omega) = x) \\ &= P(f_{a_\omega}(a_\omega) = x) \prod_{i=1, \dots, \omega-1, \omega+1, \dots, \psi} P(f_{a_i}(a_i) \leq x) \end{aligned} \quad (3.7)$$

3.2 其它相关工作

Droste [6]的工作和我们的关系接近。其推导出了在解决动态比特匹配问题时 $(1+1)$ EA首次达到最优解所需时间的期望值。与我们的工作略有不同的是，该模型假定的是每个比特上的变异概率而不是固定的翻转比特数。

还有其他文献如[1]介绍了通过EA解决动态优化问题的概观，在[2]中可以找到涉及该研究领域各方面的彻底处置办法。

第4章 将(1,2)及(1+1)用于世代间变化的环境的比较

关于第10页情形2.1所描述的 (1, 2) 繁殖方案，现在让我们推导与前文相似的公式。图4.1直观地比较了两个选择方案 (1 + 1) 和 (1, 2)。注意，虽然现在每次循环生成了两个新个体，但是在每个世代结束之后环境发生变化时，世代中的评估总次数仍然和 (1 + 1) 繁殖方案相等，因为在 (1 + 1) 中已经规定在每个世代重估入选个体，以保证下个世代中选择的公平性。

如果暂时忽略适应度函数的变化，则根据推论5.1，两个后代中的较好者的适应度分布可以描述为：

$$P\left(\max_{b \in \{m_{r_1}(a), m_{r_2}(a)\}} f(b) = x\right) = P^2(f(m_r(a)) \leq x) - P^2(f(m_r(a)) < x) \quad (4.1)$$

它可通过定理3.1的公式3.2来计算。

结合上式及定理3.2，下面的公式考虑了每个世代结束后的变化：

$$\begin{aligned} & P(f_{\tau+1}(\arg \max_{b \in \{m_{r_1}(a), m_{r_2}(a)\}} f(b)) = x) \\ &= \sum_{i=0}^L P(f(\arg \max_{b \in \{m_{r_1}(a), m_{r_2}(a)\}} f(b)) = i) P(f(m_d(c_i)) = x) \end{aligned} \quad (4.2)$$

4.1 最优变异率

图4.2比较了 (1 + 1) 和 (1, 2) 基于当前母体适应度的最优变异率。注意，在实际算法中我们可以利用 (1 + 1) 的最优变异率，而对 (1, 2) 来说，因为不知道母体在当前环境中的适应度，所以更确切地说，有关数据是猜想的。

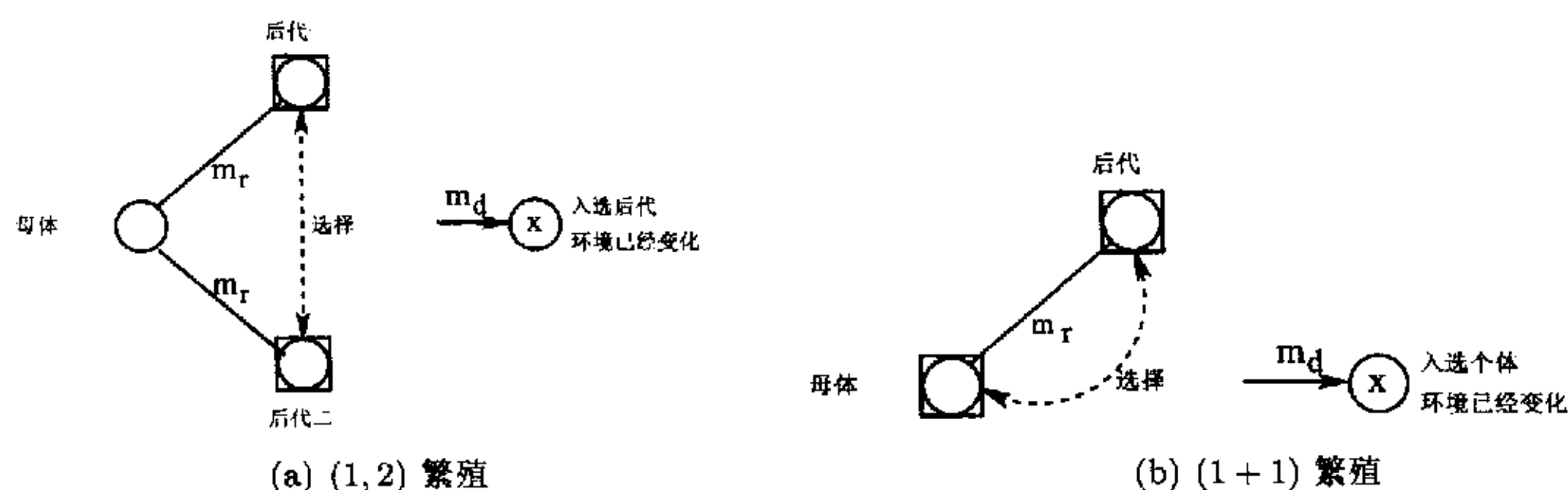


图 4.1: (a) (1,2) 繁殖 与 (b) (1+1) 繁殖

其中圆圈代表个体；框圈代表评估过的个体

已知母体适应度，求标有“x”的个体的适应度分布

显然在两个方案中，随着当前适应度的减少，最优变异率都上升得很快，因为此时变异改进适应度的机会增多而破坏有效比特的机会减少。最佳适应度与适应度函数变化的激烈程度无关，因为适应度函数在变异和选择之后才发生变化且该变化以同样方式影响所有的入选个体。

由于 (1+1) 从不接受一个比当前母体更差的个体，所以即使当母体适应度高时，允许一些变异也不会出问题。但是对 (1,2) 来说情况就不一样了，它得冒由变异造成适应度重大损失的风险，所以只要母体适应度大于等于73，最优变异率就等于0，而且在整个适应度取值区间上它的最佳适应度都比 (1+1) 要低。

有趣的是，至少对 (1+1) 来说，从未有偶数比特的最优变异率。原因在于：只有当后代的适应度的确大于母体适应度时，才会改进优化结果。偶数比特翻转（比奇数比特翻转）更有可能相互抵消从而根本没有改变个体适应度（即未改进优化结果）。而翻转奇数比特“强迫”变异改变个体适应度，这导致了更多更好的个体，同时导致了更多更差的个体（没关系，因为只有那些更好的个体才会被接受）。

4.2 收敛曲线

根据下个世代母体适应度的概率分布，我们还可进一步计算出各世代母体适应度期望。本节中为便于比较，我们为两个方法选用各自的最优变异率。

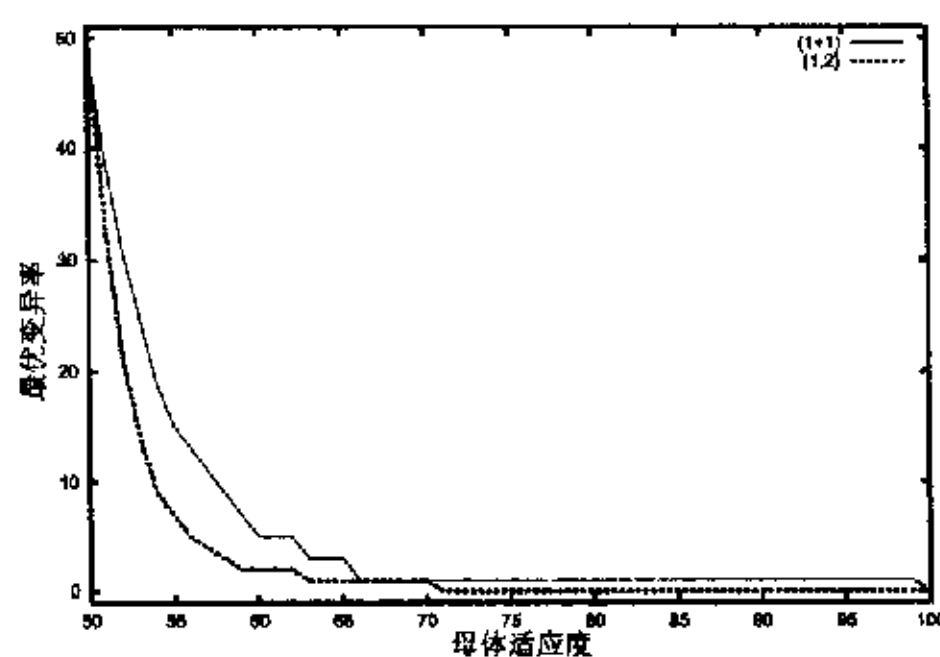


图 4.2: $(1,2)$ 和 $(1+1)$ 繁殖基于当前母体适应度的最优变异率
整个串长度假定为100

图4.3 画的是分别根据 $d = 0, 1, 3$ 推导出的收敛曲线。

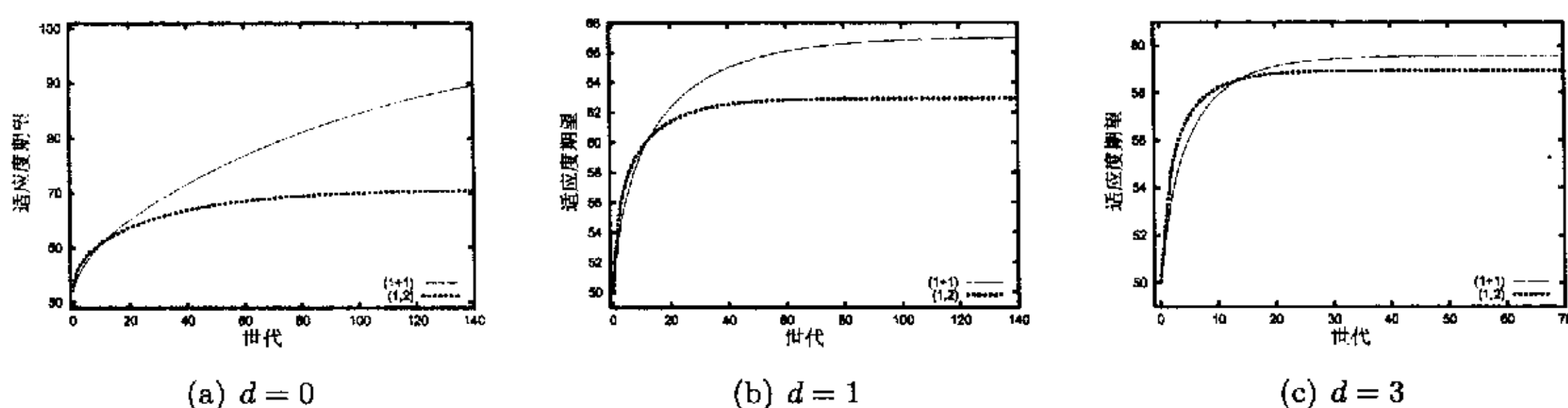


图 4.3: 在不同变化程度 d 影响下 $(1+1)$ 与 $(1,2)$ 收敛曲线之比较
假定整个串长度为100

正如期望的那样，在静态环境中， $(1+1)$ 明显优于 $(1,2)$ 。onemax适应度函数没有局部极值，因而一个局部优化算法如 $(1+1)$ 工作得很顺利。比较将这两个繁殖方案用于起伏的适应度地形将会是有趣的，那里 $(1+1)$ 会被局部最优解“绊住”，不过此处我们的分析仅限于简单的动态比特匹配测试基准。

当环境开始变化后，随着动态越来越激烈，带有探险（exploratory）性质的 $(1,2)$ 繁殖方案越来越靠近具有开采（exploitatory）性质的 $(1+1)$ 繁殖方案。

上述任何一种情况下，当母体适应度还很小时， $(1,2)$ 均优于 $(1+1)$ ，因此在开始阶段使用 $(1,2)$ 稍后再切换到 $(1+1)$ 的做法是有益的。

第5章 世代内的环境变化

在本章中，我们将要考虑第11页情形2.2所描述的 $(1, 2)$ 繁殖方案。适应度函数将在世代内，即在第一个后代被评估之后发生变化。基于这些简单设置，我们将分析比较以下两个简单策略：

- 两个个体分别使用当时的适应度函数评估。该方法使用最新的适应度信息，但无法回避这样一个带来困扰的事实：选择是以基于两个不同适应度函数的评估值为基础的。
- 人为地延迟适应度函数的变化，即使用旧的适应度函数评估两个后代。尽管该方法故意忽略新的环境信息，但在被相同适应度函数评估的个体间进行选择。该方法假设通过模拟现实环境来实现评估，这便允许我们考虑决定何时加入环境变化。这种做法对于几乎所有实际应用来说应该是合理的。

显然，上述方法各有利弊，而我们想知道在特定情况下究竟哪种更好。让我们考虑以下例证：令 x_1 和 x_2 为产生的两个后代，令它俩在环境变化前的适应度分别为 $f(x_1) = 10$ 和 $f(x_2) = 12$ ，在环境变化后的适应度分别为 $f_{\tau+1}(x_1) = 11$ 和 $f_{\tau+1}(x_2) = 9$ 。采用最新适应度的方法会在 $f(x_1) = 10$ 与 $f_{\tau+1}(x_2) = 9$ 之间做选择，即正确地选择 x_1 作为最大值问题的解；采用延迟变化的方法会在 $f(x_1) = 10$ 及 $f(x_2) = 12$ 之间做比较后选择 x_2 ，而事实上它劣于 x_1 。

反过来，当 $f_{\tau+1}(x_1) = 8$ 且 $f_{\tau+1}(x_2) = 9$ 时，采用最新适应度的方法产生错误的结果，而采用延迟变化的方法会正确地选择 x_2 。

更一般地说，只要变化不影响两个后代的大小关系，延迟变化的方法就会

得到正确的结果。对于动态比特匹配问题来说,这种情况有可能在比特串很长而变异率很小时出现,因为那样的话 x_1 和 x_2 的绝大多数比特是相等的,样板的变化很有可能对它俩施加同样的影响。特别地,如果环境变化比变异更加剧烈,则用不同适应度函数评估个体所造成的不利影响将“掩盖”来自变异的适应度差异,从而更有可能出现错误。

下面我们将分析比较在解决动态比特匹配问题时的这两种策略。对于延迟变化策略来说,其适应度分布将等于第4章中讨论过的环境在世代结束时变化的情形。

使用另一方法的情况见图解5.1:首先,产生并评估一个后代,然后是环境变化,接着产生并评估第二个后代,并选择适应度较高的那个后代。

为便于分析比较,我们需要知道入选后代的真实的适应度,也就是它在新环境中的适应度。如果第二个后代入选,则不会有问题,因为它已在新环境中被评估过。但是如果第一个后代入选,则指派给它的适应度是过时的,于是它必须在新环境中被隐式重估(注意该重估仅是理论研究所必需的,它并不是EA实现的一部分)。这图5.1比较了这两种情形。

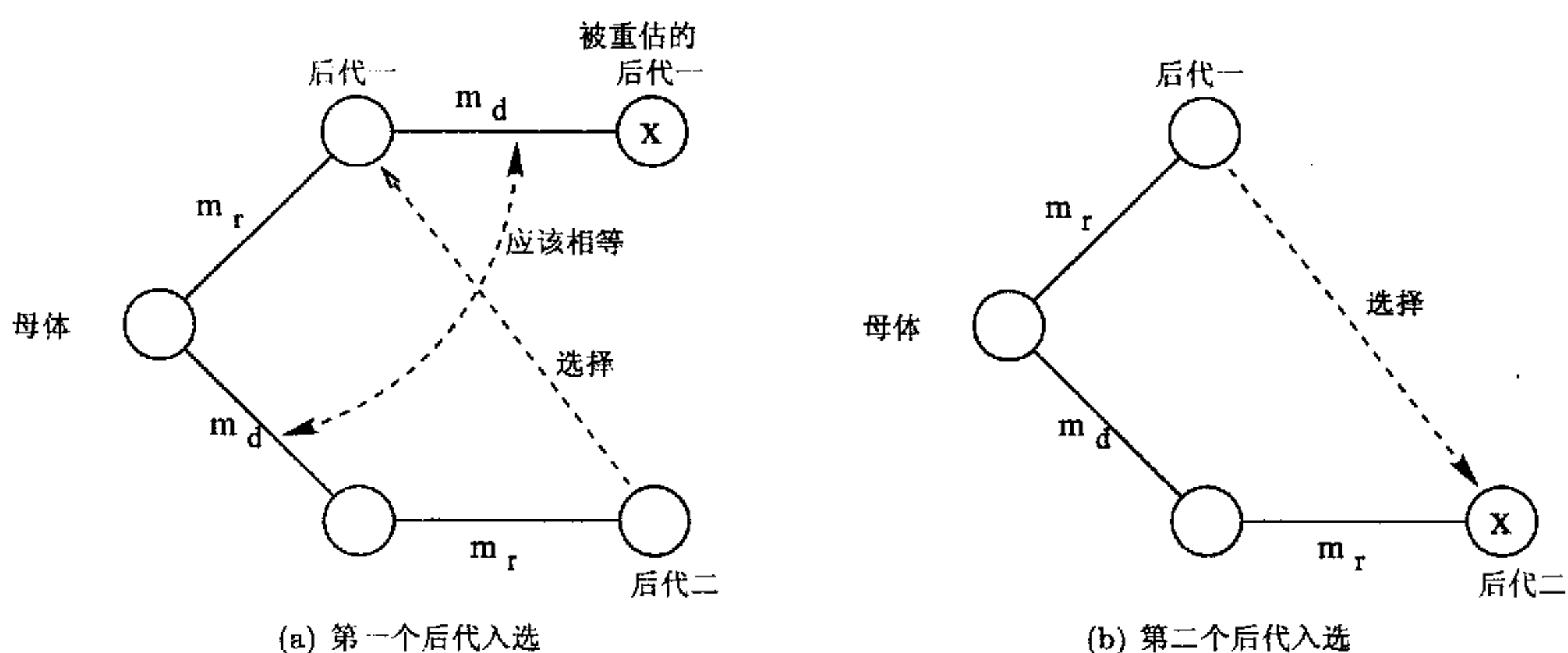


图 5.1: 当适应度函数在第一个后代被评估后变化时,分析过程取决于 (a) 第一个后代入选还是 (b) 第二个后代入选,求标有“x”的个体的适应度概率分布。因为环境只有一次变化,所以(a)中两个变化必须相等

难点是用于重估第一个后代的变化必须与评估第二个后代前发生的那个变

化相等。我们必须能够在分析计算中复制该变化，同时还希望沿用前面章节中的高级分析方法，以免在比特级穷举所有变化。

5.1 一个分析难题

考虑情形2.2, $\forall R \subseteq I, |R| = r, \forall D \subseteq I, |D| = d$, 设母体 a 变异产生的后代为 $m_R(a)$, 变化后的样板 $t_{\tau+1} = m_D(t_\tau)$ 。如果该后代入选，则它应该在新环境下被隐式重估。求该后代入选并被隐式重估为 x 的概率

$$P(f(\arg \max(f(m_R(a), t_\tau), f(a, m_D(t_\tau)))) = m_R(a), m_D(t_\tau)) = x)$$

首先，该后代入选的概率可通过以下公式计算：

$$\begin{aligned} & P(\arg \max(f(m_R(a), t_\tau), f(a, m_D(t_\tau)))) = m_R(a)) \\ &= \sum_{i=0}^L P(f(m_r(a)) = i) P(f(m_d(a)) \leq i) \end{aligned}$$

注意，根据该情形所对应的程序代码，此处应当使用推论而不是推论来计算概率。

但接下来，我们不能将上式乘以 $P(f(m_d(c_i)) = x)$ 来完成工作，因为 m_d 已经被前面的选择步骤用过，所以这里它已不再是一个独立的步骤了。

注意到该问题还可以用另一种形式描述：

$$\begin{aligned} & P(f(\arg \max(f(m_R(a), t_\tau), f(a, m_D(t_\tau)))) = m_R(a), m_D(t_\tau)) = x) \\ &= P(\arg \max(f(m_R(a), t_\tau), f(a, m_D(t_\tau)))) = m_R(a)) P(f(m_D(m_R(a)), t_\tau) = x) \end{aligned}$$

而公式右侧最后一个因式提醒我们另辟蹊径——对变异自身做更深入的研究。

下面让我们首先引入两阶段变异（two-step mutation）的概念。

5.2 两阶段变异

从本质上说，个体 a 上的任何一个 r -比特变异可被分解成两个独立的子变异（sub-mutation），分别是 s -比特变异和 $(r - s)$ -比特变异。即首先变异

$s < r$ 比特, 随后在一个更短的、不包含已变异的 s 比特 (避免将相同的比特翻转两次) 的子串上变异剩余的 $r - s$ 比特。

定义 5.1 (两阶段变异)

$$\forall R \subseteq I, |R| = r, \exists S \subseteq R, |S| = s, m_R(a) = m_{R-S}(m_S(a))$$

该概念在图5.2中被说明。

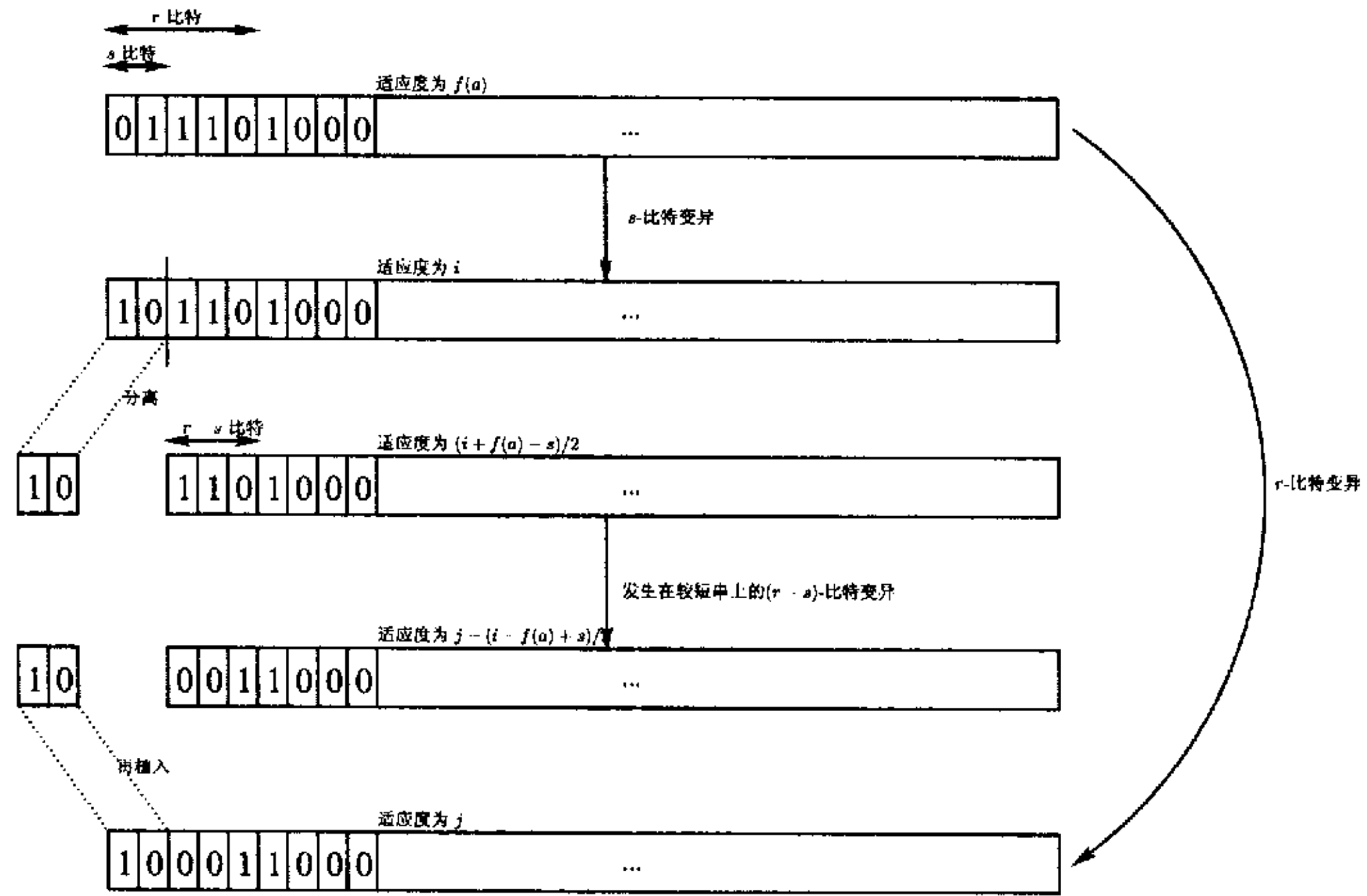


图 5.2: 两阶段变异

由于比特的前后次序是无关紧要的, 所以图中不失一般性地假设前面的比特发生变异

首先, 我们在整个串上执行一个普通的 s -比特变异。让我们假设变异得到的个体适应度为 i , 第一子变异步骤是容易分析的。由于第二子变异和第一个没有交叉比特, 所以它可以被设想成发生在 a 的一个子串 $a_{I-S} = \{(a)_i | i \notin S, i \in I\}$ 中, 我们于是可以推导出该子串的所有特征:

- 子串长度: $|a_{I-S}| = L - s$
- 初始适应度: $f(a_{I-S}) = f(a) - s + s_n = \frac{f(m_S(a)) + f(a) - s}{2} = (i + f(a) - s)/2$
- 变异率: $r - s$

- 子变异后的适应度: $j - s_n = j - (i - f(a) + s)/2$ (假设 $f(m_R(a)) = j$, 即整个两阶段变异之后个体适应度等于 j)

我们把发生在长度为 $L - s$ 、适应度为 $(i + f(a) - s)/2$ 的串上的这样一个 $(r - s)$ -比特变异表示为 $m_{r-s}(c_{(i+f(a)-s)/2}^{L-s})$ 。

这样, 第二个子变异也转化为可用引理3.1及定理3.1描述的基本变异操作了。整个两阶段变异的概率分布可以通过下面的定理计算。

定理 5.1 (两阶段变异概率分布)

$$\begin{aligned}
 & P(f(m_r(a)) = j) \\
 &= \sum_{\substack{i=\max\{0, f(a)-s\} \\ i \leq \min\{L, f(a)+s\}}} P(f(m_s(a)) = i) \\
 & \quad \times P\left(f(m_{r-s}(c_{(i+f(a)-s)/2}^{L-s})) = j - \frac{i - f(a) + s}{2}\right)
 \end{aligned} \tag{5.1}$$

与包含许多展开项的解析证明方式相比, 我们更愿意用统计方式证明该定理 (参见5.6节)。试验结果表明, 两阶段变异与单步变异在相同条件下的行为完全吻合。

我们相信使用两阶段变异能够解决上述难题, 因为该方法允许我们精确地记录下相关的变异的子步骤, 以便在以后进行“重放”。

5.3 使用两阶段变异建立世代内变化的模型

现在能用上面提到的两阶段变异恢复在第二个后代被评估之前发生的那个变化, 并且同样将它作用到第一个后代上。基本思路是将变异分成两个步骤: 将 r -比特变异与 d 比特目标变化中相同的 s 比特首先进行变异。然后将剩余的 $r - s$ 个需要变异的比特以及 $d - s$ 个环境变化的比特进行翻转 (参见图解5.3)。接下来, 如果我们想要对第一个后代实施同样的环境变化, 我们可以通过消除第一步 s -比特变异的效果, 并在一个长度为 $L - r$ 的串上实施 $(d - s)$ -比特变异 (最后这个变异与 r -比特变异没有交叉比特) 来达到目的。

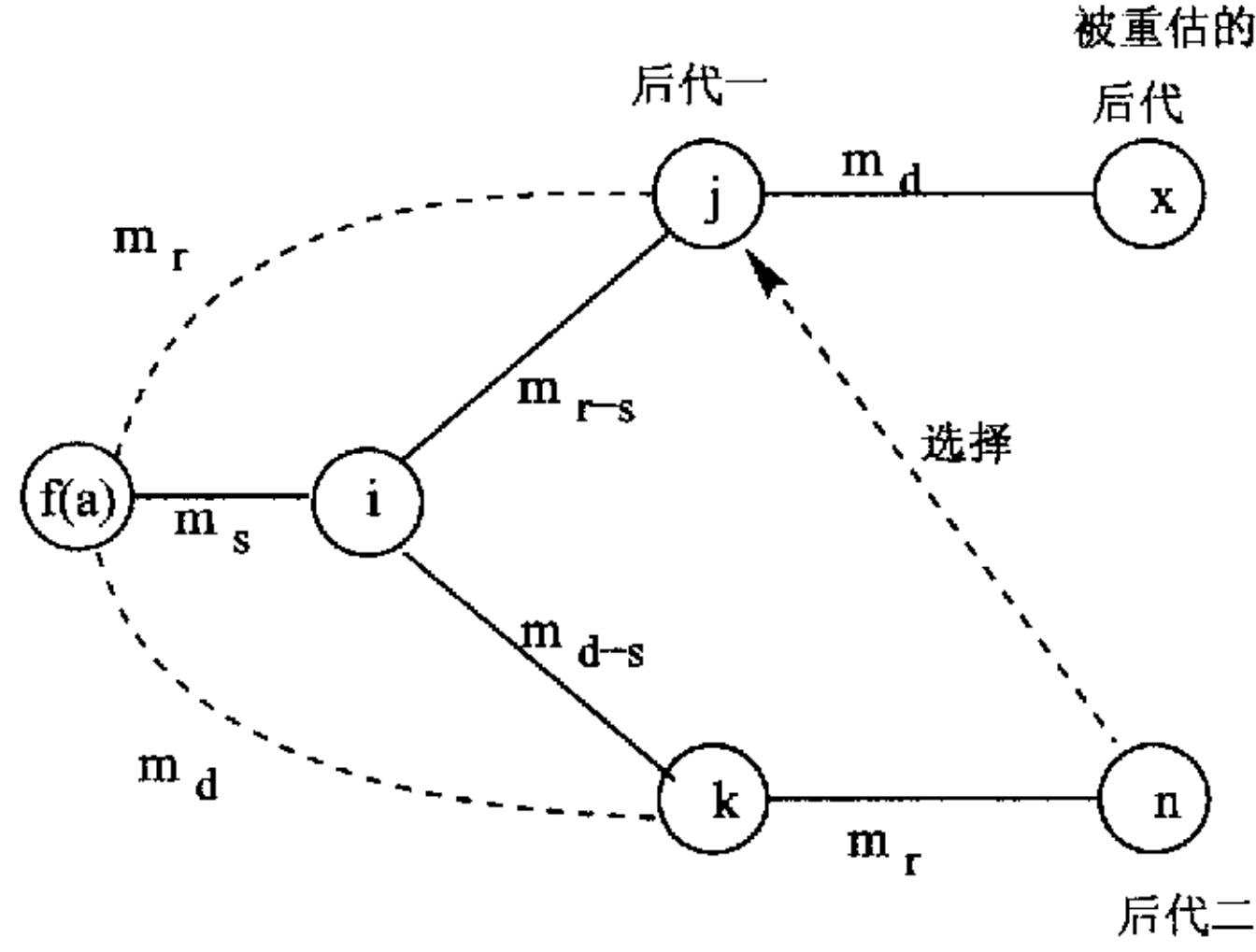


图 5.3: 使用两阶段变异分析(1,2)EA
环境在第一个后代被评估后变化，假定第一个后代入选
圆圈中的字母表示相应个体的适应度

s -比特变异及 $(r-s)$ -比特变异已经在上面讨论过了。对 $(d-s)$ -比特变异来说，串长度是 $L-r$ （因为它与 s -比特变异及 $(r-s)$ -比特变异都没有交叉比特），初始适应度为 $(j + f(a) - r)/2$ ，变异后的适应度应为 $k - i + (j + f(a) - r)/2$ （假定在整个两阶段变异之后个体适应度等于 k ）。

总的说来，

$$\begin{aligned} & P((f(m_d(a)) = k) | (f(m_r(a)) = j) \wedge f(m_s(a)) = i)) \\ &= P\left(f(m_{d-s}(c_{(j+f(a)-r)/2}^{L-r})) = k - i + \frac{j + f(a) - r}{2}\right) \end{aligned} \quad (5.2)$$

让我们首先考虑第一个后代的适应度大于或等于第二个后代的情形，根据程序伪码此时第一个后代入选。按照惯例，我们想要计算后代（此处为第一个后代）适应度为 x 的概率。

第一个后代入选的概率计算如下

$$\begin{aligned} & \sum_{j=\max\{0, f(a)-r\}}^{\min\{L, f(a)+r\}} P(f(m_r(a)) = j) P(f(m_r(m_d(a))) \leq j) \\ &= \sum_{j=\max\{0, f(a)-r\}}^{\min\{L, f(a)+r\}} P(f(m_r(a)) = j) \sum_{k=\max\{0, f(a)-d\}}^{\min\{L, f(a)+d\}} P(f(m_d(a)) = k) \end{aligned}$$

$$\times P(f(m_r(c_k)) \leq j) \quad (5.3)$$

使用两阶段变异，上式可重新表达为

$$\begin{aligned} & \sum_{s=0}^{\min\{r,d\}} P(v(r,d) = s) \sum_{i=\max\{0,f(a)-s\}}^{\min\{L,f(a)+s\}} P(f(m_s(a)) = i) \\ & \times \sum_{j=\max\{0,i-r+s\}}^{\min\{L,i+r-s\}} P\left(f(m_{r-s}(c_{(i+f(a)-s)/2}^{L-s})) = j - \frac{i - f(a) + s}{2}\right) \\ & \times \sum_{k=\max\{0,i-d+s\}}^{\min\{L,i+d-s\}} P\left(f(m_{d-s}(c_{(j+f(a)-r)/2}^{L-r})) = k - i + \frac{j + f(a) - r}{2}\right) \\ & \times P(f(m_r(c_k)) \leq j) \end{aligned} \quad (5.4)$$

此处 $P(v(r,d) = s)$ 代表 d -比特变异和 r -比特变异共有 s 个交叉比特的概率，可计算如下

$$P(v(r,d) = s) = \frac{\binom{L}{s} \cdot \binom{L-s}{r-s} \cdot \binom{L-r}{d-s}}{\binom{L}{r} \cdot \binom{L}{d}} \quad (5.5)$$

第一个后代被重估后的适应度由公式 $x = f(a) + j + k - 2i$ 决定。若 x 已知，则 $k = x - f(a) + 2i - j$ 。

于是第一个后代入选且被重估为 x 的概率可计算如下

$$\begin{aligned} & \sum_{s=0}^{\min\{r,d\}} P(v(r,d) = s) \sum_{i=\max\{0,f(a)-s\}}^{\min\{L,f(a)+s\}} P(f(m_s(a)) = i) \\ & \times \sum_{j=\max\{0,i-r+s\}}^{\min\{L,i+r-s\}} P\left(f(m_{r-s}(c_{(i+f(a)-s)/2}^{L-s})) = j - \frac{i - f(a) + s}{2}\right) \\ & \times P\left(f(m_{d-s}(c_{(j+f(a)-r)/2}^{L-r})) = x + i - \frac{j + f(a) + r}{2}\right) \\ & \times P(f(m_r(c_{x-f(a)+2i-j})) \leq j) \end{aligned} \quad (5.6)$$

第二种情形，即第二个后代因适应度高而入选的情况较容易分析。因为我们无须重估该个体，仅需计算它的适应度为 x 且第一个后代的适应度小于 x

的概率即可。可用公式表示如下

$$\sum_{k=\max\{0, f(a)-d\}}^{\min\{L, f(a)+d\}} P(f(m_d(a)) = k) P(f(m_r(c_k)) = x) P(f(m_r(a)) < x) \quad (5.7)$$

综合上述两种可能，新母体适应度为 x 的概率就等于第一个后代入选并重估为 x 的概率(公式5.6)与第二个后代入选且适应度为 x 的概率(公式5.7)之和。

在实际的研究工作中，为了简化问题，我们曾在分析解决上述情形之前，首先尝试将两阶段变异用于以下看上去更简单的情形。

情形 5.1 $(1+1)$ EA位于世代内变化的 $(d, 2^{\text{th}})$ DE中

```

 $\tau = 0;$ 
 $initialize(t_0);$ 
 $initialize(parent);$ 
 $fitness = f_0(parent);$ 
while (NOT termination condition) do
     $child = m_r(parent);$ 
     $fitness1 = f_\tau(child);$ 
     $t_{\tau+1} = m_d(t_\tau);$ 
     $\tau = \tau + 1;$ 
     $fitness = f_\tau(parent);$  //显式重估
    if ( $fitness1 \geq fitness$ ) { //把  $\geq$  换成  $>$  将导致不同的隐式评估值
         $parent = child;$ 
         $fitness = fitness1;$  //需要隐式重估
    }
enddo

```

求后代入选且被重估后的概率分布。

我们可以使用两阶段变异解决上述问题。分别将 r -比特变异和 d -比特变异分成两个独立的子变异： s -比特变异是它们相同的部分，然后分别是 $(r-s)$ -比特变异和 $(d-s)$ -比特变异：假设 $I = \{1, \dots, L\}, \forall a = \{(a)_i | i \in I, (a)_i \in \{0, 1\}\}, \forall R \subseteq I, |R| = r, \forall D \subseteq I, |D| = d, \exists S = R \cap D, |S| = s$, 再设 $f(m_r(a), m_d(a)) = s, P(s) = \frac{\binom{L}{s} \binom{L-s}{r-s} \binom{L-r}{d-s}}{\binom{L}{r} \binom{L}{d}}, \exists S_N = \{(a)_i | (a)_i \in a_S, (a)_i \neq (t_\tau)_i\}, |S_N| = s_n = \frac{f(m_S(a)) - f(a) + s}{2}$

$$|(R-S)_N| = (r-s)_n$$

$$\begin{aligned}
&= \frac{f(m_{R-S}(a_{I-S})) - f(a_{I-S}) + r - s}{2} \\
&= \frac{f(m_R(a)) - f(m_S(a)) + r - s}{2} \\
m_R(a) &= m_{R-S}(m_S(a)), \quad m_D(a) = m_{D-S}(m_S(a)), \\
m_D(m_R(a)) &= m_{D-S}(m_S(m_{R-S}(m_S(a)))) = m_{D-S}(m_{R-S}(a))
\end{aligned}$$

因为上述三个子变异没有交叉比特，它们的次序可以调换，且不影响最终的变异结果。我们想象每个步骤发生在一个 a_I 减去前面步骤中已经变异比特元素的子串中。于是这些子串的所有特征都可以求得。

• 关于 $m_{R-S}(a_{I-S})$:

- 子串长度: $|a_{I-S}| = L - s$
- 初始适应度: $f(a_{I-S}) = f(a) - s + s_n$
- 变异率: $r - s$
- 子变异后的适应度: $j - s_n$ (假设 $f(m_R(a)) = j$)

• 关于 $m_{D-S}(a_{I-R})$:

- 子串长度: $|a_{I-R}| = L - r$
- 初始适应度: $f(a_{I-R}) = f(a) - s + s_n - (r - s) + (r - s)_n$
- 变异率: $d - s$
- 子变异后的适应度: $k - s_n - (r - s) + (r - s)_n$ (假设 $f(m_D(a)) = k$)

后代入选的概率可用另一种方式书写:

$$\begin{aligned}
&\sum_{s=0}^{\min(r,d)} P(s) \sum_{i=\max(0,f(a)-s)}^{\min(L,f(a)+s)} P(f(m_S(a)) = i) \sum_{j=\max(0,i-r+s)}^{\min(L,i+r-s)} P(f(m_R(a)) = j) \\
&\times \sum_{k=0}^j P(f(m_D(a)) = k)
\end{aligned} \tag{5.8}$$

上述公式中，若给定 $f(a), L, r, d, s, i, j$ 和 k ，假定 $f(m_{D-S}(m_{R-S}(a))) = x$ ，则

$$x = f(a) + j + k - 2i \quad (5.9)$$

因为后代入选，故有 $j \geq k$ ， $\frac{x - f(a) + 2i}{2} \leq j \leq x - f(a) + 2i$

入选后代的重估适应度等于 x 的概率为：

$$\begin{aligned} & \sum_{s=0}^{\min(r,d)} P(s) \sum_{i=\max(0,f(a)-s)}^{\min(L,f(a)+s)} P(f(m_S(a)) = i) \sum_{j=\frac{x-f(a)+2i}{2}}^{x-f(a)+2i} P(f(m_R(a)) = j) \\ & \times P(f(m_D(a)) = x - f(a) + 2i - j) \end{aligned} \quad (5.10)$$

对最后两项使用两阶段变异，得：

$$\begin{aligned} & \sum_{s=0}^{\min(r,d)} P(s) \sum_{i=\max(0,f(a)-s)}^{\min(L,f(a)+s)} P(f(m_S(a)) = i) \\ & \times \sum_{j=\frac{x-f(a)+2i}{2}}^{x-f(a)+2i} P(f(m_{R-S}(a_{I-S})) = j - s_n) \\ & \times P(f(m_{D-S}(a_{I-R})) = x - f(a) + 2i - j - s_n - (r - s) + (r - s)_n) \end{aligned} \quad (5.11)$$

在 $(1+1)$ 分析结果基础上，为了计算 $(1,2)$ 后代一入选后隐式重估适应度概率分布，仅需修改用于描述

$$f(m_R(a)) \geq f(m_D(a))$$

的那些公式。具体分析步骤如下：

第一个后代入选的概率可以如下表达：

$$\begin{aligned} & \sum_{s=0}^{\min(r,d)} P(s) \sum_{i=\max(0,f(a)-s)}^{\min(L,f(a)+s)} P(f(m_S(a)) = i) \sum_{j=\max(0,i-r+s)}^{\min(L,i+r-s)} P(f(m_R(a)) = j) \\ & \times \sum_{k=0}^L P(f(m_D(a)) = k) P(f(m_{r_2}(m_D(a))) \leq j) \end{aligned} \quad (5.12)$$

注意到 $m_{r_2}(m_D(a))$ 是一个发生在整个串上的简单变异：

- 子串长度: $|m_D(a)| = L$
- 初始适应度: $f(m_D(a)) = k = x - f(a) + 2i - j$
- 变异率: r
- 变异后的适应度: n (设 $f(m_r(m_D(a))) = n$)

而且公式(5.9)仍然适用。适用两阶段变异, 上式于是变成:

$$\begin{aligned}
 & \sum_{s=0}^{\min(r,d)} P(s) \sum_{i=\max(0,f(a)-s)}^{\min(L,f(a)+s)} P(f(m_S(a)) = i) \\
 & \times \sum_{j=\max(0,i-r+s)}^{\min(L,i+r-s)} P(f(m_{R-S}(a_{I-S})) = j - s_n) \\
 & \times P(f(m_{D-S}(a_{I-R})) = x - f(a) + 2i - j - s_n - (r - s) + (r - s)_n) \\
 & \times \sum_{n=0}^j P(f(m_r(m_D(a))) = n) \tag{5.13}
 \end{aligned}$$

该式化简后和公式5.6相符。

通过与 (1,2) 分析的比较后我们才知道, 这样的“(1,1) 简化问题”其实并未降低分析难度。不过, 正是通过上述情形分析, 证实了两阶段变异不仅是可行的, 而且具备和已知分析方法的结合能力。它不仅可以紧接着两个体变异分开之后, 而且可以在几个进化操作步骤之后被“重演”。正是出于研究其结合能力的考虑, 我们从一开始分析上述两个情形时就假设 i 、 j 和 k , 即尽量多地使用整个串的适应度而不是子串适应度。

5.4 若干比较

上述 (1,2) 的结论公式允许我们依据下个世代母体适应度的期望值来比较这两个策略, 即使用旧的适应度函数评估两个后代, 或总是使用最新的适应度函数。

在下面的结果报告中, 我们仍然假设比特串长度为100。

图5.4比较了受当前母体适应度及环境变化激烈程度 d 影响的下个世代母体适应度期望的差值。正如我们所看到的, 对 $d=1$ 来说, 使用最新的信息评估个体总是表现较好。这也可从另一条途径推理得出: 在环境变化前, 两个后代的适应度差值从不等于1, 因为在这两个变异中翻转了相同的比特数。如果适应度差异为0, 知晓环境的影响对于正确选择较好个体来说就显得非常重要。如果适应度差异 ≥ 2 , 则单个比特的变化无法颠倒两后代的大小次序, 于是两个适应度评估方案将得到同样的判定结果。

但是对于 $d > 1$ 的情况, 不可能再做这样简单的分析。依照图5.4, 在 $d=2$ 或 $d=3$ 情况下, 除非母体适应度很低, 否则使用旧环境来评估第二个后代的效果显然更好。在母体适应度很低时, 最优变异率很高, 因而两个后代的实际适应度很有可能相差显著。但随着母体适应度的增大以及最优变异率的减小, 两个后代的实际适应度相差变小, 而当一个相对剧烈的变化($d > 1$) 仅仅作用在其中一个个体上时, 实际适应度差异就有被该变化“掩盖”的危险, 从而导致错误的选择。不同适应度评估方案对收敛曲线的影响效果可通过图5.5看出。

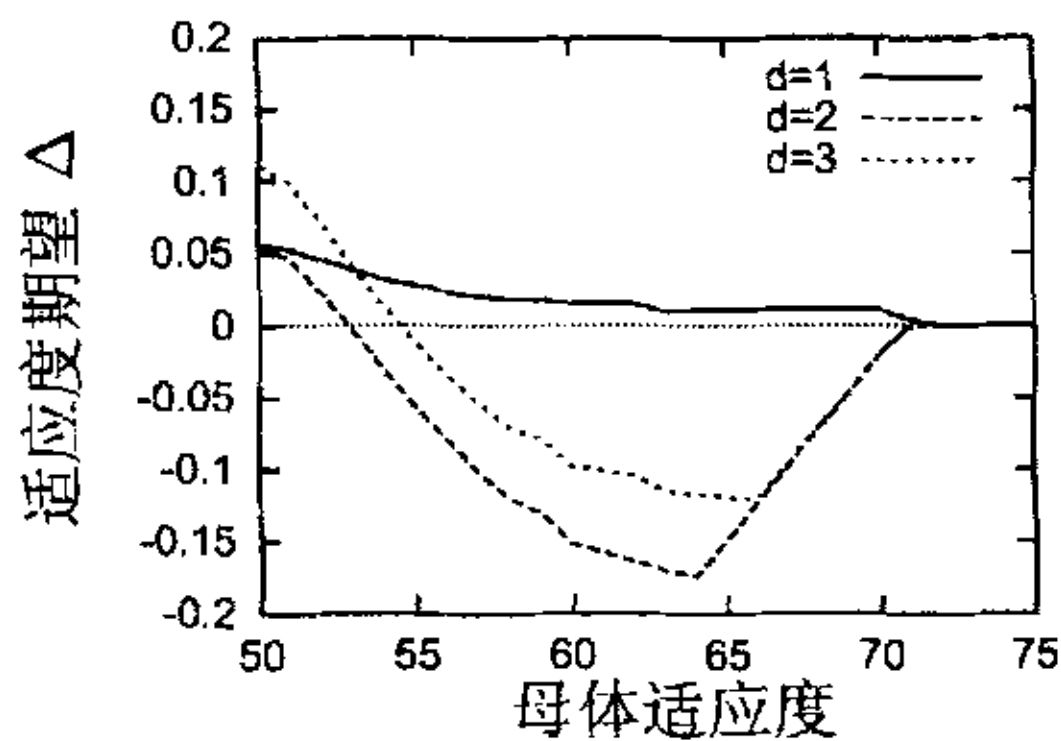


图 5.4: 在不同激烈程度 d 变化的环境中, 对第二个后代分别用新、旧适应度函数评估的下个世代母体适应度期望的差值。它随当代母体适应度的不同而变化。如差值大于0则用新适应度函数效果较好, 反之道理亦然。

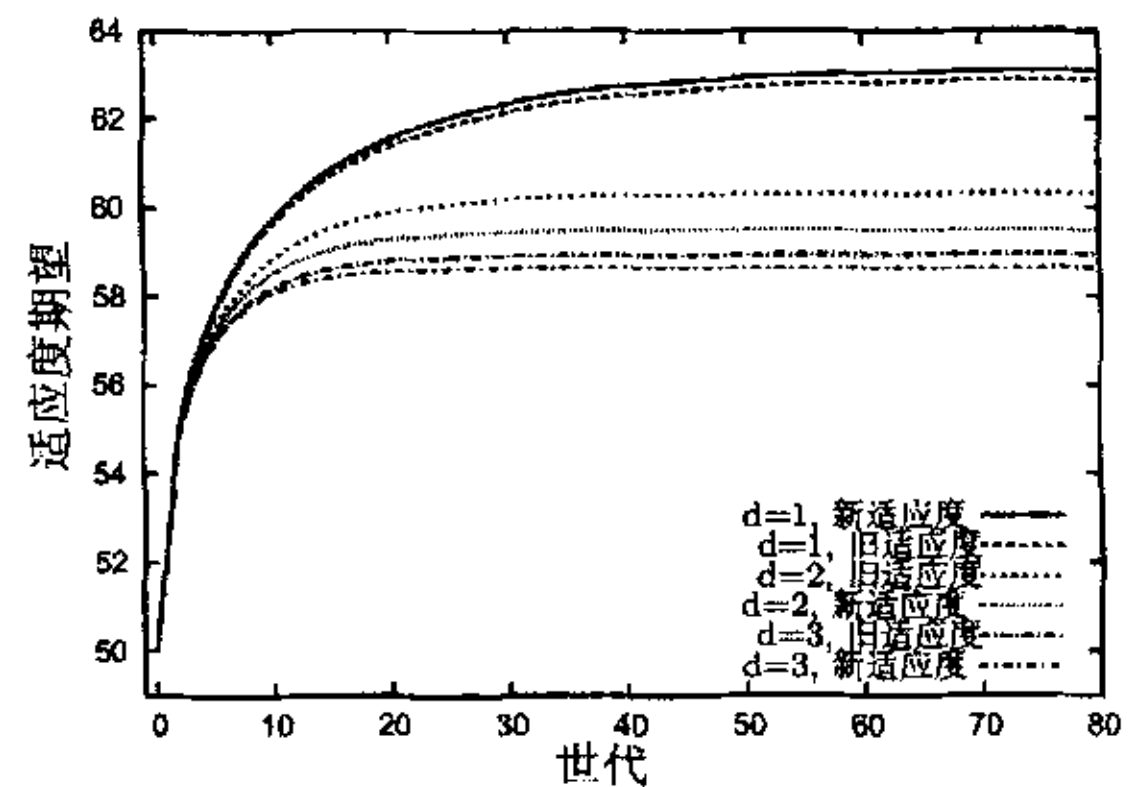


图 5.5: 在不同激烈程度 d 变化的环境中, (1,2) 繁殖策略的对第二个后代分别使用新、旧适应度函数的收敛曲线。线条按照其在图中的次序进行标注。

5.5 种群个体总数更多的情形

现在让我们来考虑一个在前两个后代被评估之前和之后变化的 $(d, 2^{th})$ DE中的 $(1, 4)$ EA:

情形 5.2 在一个世代内变化的 $(d, 2^{th})$ DE中的 $(1, 4)$ EA

```

 $\tau = 0$ ;
initialize( $t_0$ );
initialize(parent);
fitness =  $f_0$ (parent);
while (NOT termination condition) do
    child1 =  $m_r$ (parent);
    child2 =  $m_r$ (parent);
    child3 =  $m_r$ (parent);
    child4 =  $m_r$ (parent);
     $t_{\tau+1} = m_d(t_\tau)$ ;
     $\tau = \tau + 1$ ;
    fitness1 =  $f_\tau$ (child1);
    fitness2 =  $f_\tau$ (child2);
     $t_{\tau+1} = m_d(t_\tau)$ ;
     $\tau = \tau + 1$ ;
    fitness3 =  $f_\tau$ (child3);
    fitness4 =  $f_\tau$ (child4);
    switch (max(fitness1, fitness2, fitness3, fitness4)){
        case fitness1 :
            parent = child1;
            fitness = fitness1; //需要隐式重估
            break;
        case fitness2 :
            parent = child2;
            fitness = fitness2; //需要隐式重估
            break;
        case fitness3 :
            parent = child3;
            fitness = fitness3;
            break;
        case fitness4 :
            parent = child4;
            fitness = fitness4;
    }
enddo

```

我们想要计算最新环境中入选后代适应度概率分布。

首先让我们计算第一或第二个后代入选并被隐式重估为 x 的概率。

我们用 p 代表母体。为了将环境的第一次变化转换成为个体的变化，我们用 a 代表从母体经过 d -比特变异而产生的个体，且这 d 个比特恰好和环境变化的那些比特的位置相同，因此 $f_{\tau+1}(p) = f(m_d(p)) = f(a)$ 。设 $f(a) = m$ ，则前两个后代中的最佳评估值等于 j 的概率可描述如下：

$$P(f(m_r(c_m)) = j)(P(f(m_r(c_m)) = j) + 2P(f(m_r(c_m)) < j)) \quad (5.14)$$

已知上述 j 以及母体在（再次变化后的）最新环境中的评估值 $f_{t+2}(p) = f_{\tau+1}(a) = f(m_d(a)) = k$ ，则前两个后代之一入选的概率等于后两个后代的最佳评估值小于等于 j 的概率：

$$\sum_{n=0}^j (P^2(f(m_r(c_k)) = n) + 2P(f(m_r(c_k)) = n)P(f(m_r(c_k)) < n)) \quad (5.15)$$

已知 $f(p)$ ，我们能够使用上面的两个公式计算前两个后代之一入选且其初始评估值为 j 的概率：

$$\begin{aligned} & \sum_{m=0}^L P(f(m_d(c_{f(p)})) = m)P(f(m_r(c_m)) = j)(P(f(m_r(c_m)) = j) \\ & + 2P(f(m_r(c_m)) < j)) \sum_{k=0}^L P(f(m_d(c_m)) = k) \sum_{n=0}^j (P^2(f(m_r(c_k)) = n) \\ & + 2P(f(m_r(c_k)) = n)P(f(m_r(c_k)) < n)) \end{aligned} \quad (5.16)$$

如果第一/第二个后代入选，它应该在最新环境中被重估。接下来我们将使用两阶段变异分析来自前两个后代的入选个体的重估适应度的概率分布。

由于前两个后代有相同的统计特征，后两个后代也是，这使我们可以只关注前两个后代的最佳者以及后两个后代的最佳者，而且不会影响发生在前两个后代之间以及后两个后代之间的局部选择，于是我们可以将(1,4)问题简化成为已被解决的(1,2)问题。

来自前两个后代的入选个体被重估后的适应度 $x = m + j + k - 2i$ ，于是 k 可被 $k = x - m - j + 2i$ 替换。我们可以重写前式的某些部分，同时应两阶段变异的要求而加入新的部分到其中，它于是变成：

$$\sum_{m=0}^L P(f(m_d(c_{f(p)})) = m) \sum_{s=0}^{\min(r,d)} P(v(r,d) = s)$$

$$\begin{aligned}
& \times \sum_{i=\max(0,m-s)}^{\min(L,m+s)} P(f(m_s(c_m)) = i) \\
& \times \sum_{j=\max(0,i-r+s)}^{\min(L,i+r-s)} P(f(m_{r-s}(c_{(m-s+i)/2}^{L-s})) = j - (i - m + s)/2) \\
& \times (P(f(m_r(c_m)) = j) + 2P(f(m_r(c_m)) < j)) \\
& \times P(f(m_{d-s}(c_{(m-r+j)/2}^{L-r})) = x + i - (m + r + j)/2) \\
& \times \sum_{n=0}^j (P^2(f(m_r(c_{x-m-j+2i})) = n) + 2P(f(m_r(c_{x-m-j+2i})) = n) \\
& \times P(f(m_r(c_{x-m-j+2i})) < n))
\end{aligned} \tag{5.17}$$

其次, 已知 $f(p)$, 则来自后两个后代的适应度为 x 的个体入选的概率 (当且仅当 x 大于前两个后代的最佳初始适应度) 容易写出:

$$\begin{aligned}
& \sum_{m=0}^L P(f(m_d(c_{f(p)})) = m) \sum_{j=0}^{x-1} (P^2(f(m_r(c_m)) = j) + 2P(f(m_r(c_m)) = j) \\
& \times P(f(m_r(c_m)) < j)) \sum_{k=0}^L P(f(m_d(c_m)) = k) (P^2(f(m_r(c_k)) = x) \\
& + 2P(f(m_r(c_k)) = x)P(f(m_r(c_k)) < x))
\end{aligned} \tag{5.18}$$

当前问题的答案, 即入选后代在最新环境下的适应度概率分布, 就是上面公式5.17和公式5.18之和。

5.6 验证

为了验证两阶段变异的概念, 我们使用下列条件的所有组合比较了分别采用单阶段变异和两阶段变异计算生成的适应度分布理论值: $L = 100, r = [0, L], f_0 = [0, L], x = [0, L], s = [0, r]$ 事实显示两阶段变异结果和单阶段变异完全吻合。

为验证前面定义的EA适应度变迁模型, 我们比较了适应度分布的理论值和100,000次蒙特·卡罗试验 (Monte-Carlo trial) 的统计结果。

下面, 图5.6和图5.7分别为情形5.1和情形2.2提供了蒙特·卡罗法生成的入选个体适应度分布与预估的适应度分布之间的比较, 比较基于下列条件: 个体 a 长度 $L = 100$ 比特, 初始适应度 $f(a) = 65$, 变异率 $r = 10$ 比特, 动态适应度函数变化率 $d = 10$ 比特。EA变迁模型显然已经成功地模拟了实际变迁过程。

EA变迁模型的精确度还可进一步用Kolmogorov-Smirnov统计表(Kolmogorov-Smirnov statistics)加以衡量, 该统计表对蒙特·卡罗法生成的适应度分布与预估适应度分布的相似程度进行比较。考虑到蒙特·卡罗法的比较结论已足够明确地验证了EA变迁模型的正确性, 所以我们略去该统计表。

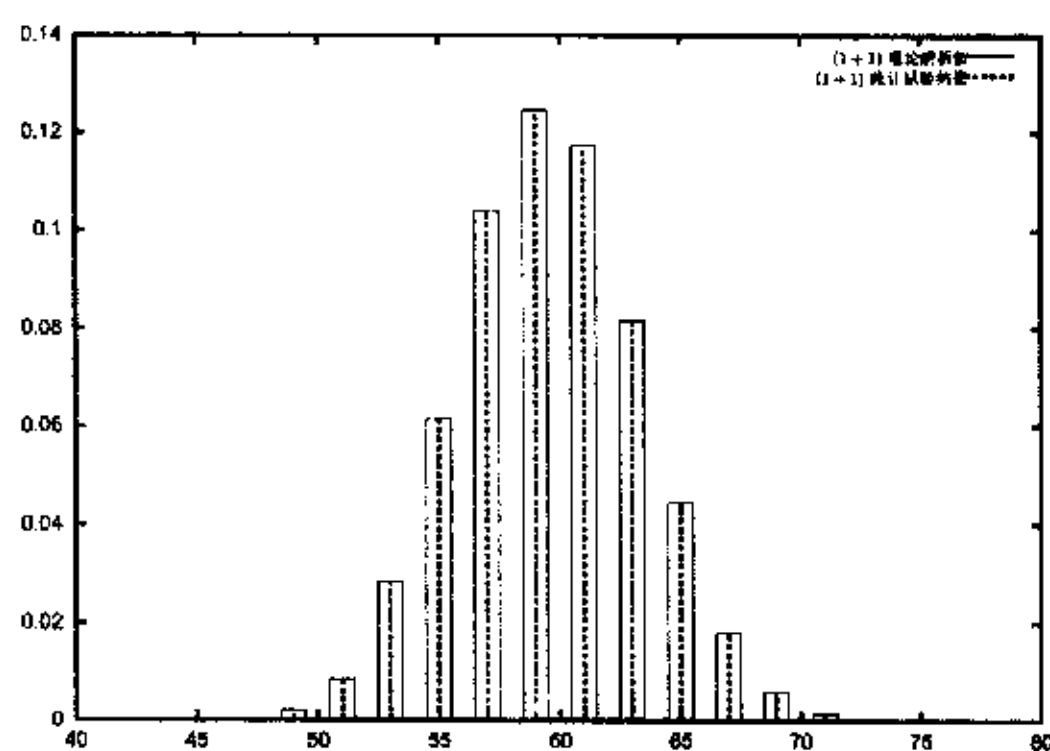


图 5.6: 情形5.1中后代带优先权入选且被重估后的适应度概率分布理论值和试验值的比较

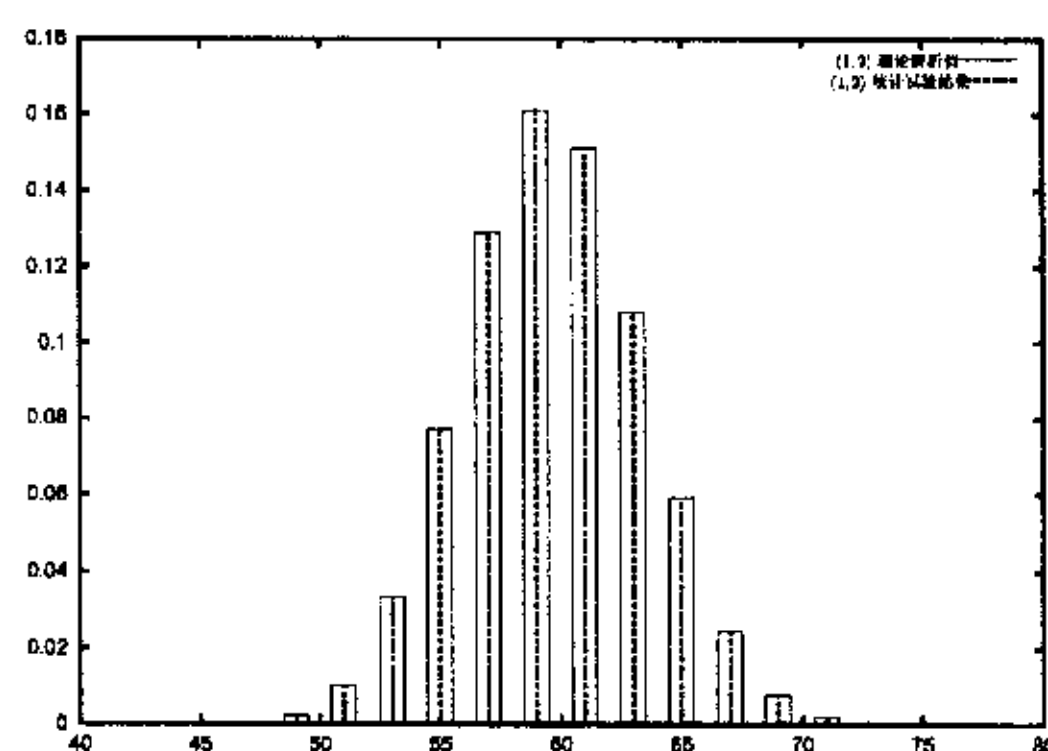


图 5.7: 情形2.2中后代一带优先权入选且被重估后的适应度概率分布理论值和试验值比较

第6章 进化计算在新一代网络传输控制中的应用

在本章中，我们将首先分析新一代网络传输控制的特点，探讨其中进化计算的地位和作用。然后就网络传输控制中应用进化算法提出几点建议。最后运用快速变化环境中简单进化策略的理论分析结果，提出了解决动态QoS组播路由选择问题的进化策略，并用实验证明该策略的有效性。

6.1 进化计算在新一代网络传输控制中的地位和作用的思考

经过20多年的发展，Internet已经在全球取得了巨大的成功。然而，当今的计算机网络仍然存在许多技术上的难题与挑战，网络基础研究仍然面临诸多关键性课题。例如，对于网络传输的模型分析，传统的方法是将网络中分组或连接的到达假设为泊松过程(Poisson process)，在理论分析上，泊松到达对于网络传输的性能评价具有简单、有效等显著特点。但是，自1993年以来，大量的证据显示实际网络中的数据传输是长范围相关的(long-range dependent, LRD)、自相似的(self-similar)、分形的(fractal)或者称为重尾分布的(heavy-tailed, or power-tailed)。这是90年代高速网络性能评价领域的一个重大发现。其挑战性在于以前使用的泊松模型无法描述这类现象。换言之，传统的基于Markov模型的性能评价结果对于自相似传输已不再适用，需要研制新的模型与工具。

作为未来网络发展的基石，面向新一代网络的基础研究已经起步，并提出了各种网络传输控制模型与机制。主要有：基于排队理论的信息源模型、基于网络/链路状态的管理与控制模型等，它们从不同的角度反映了网络数据传输

控制的本质:

一、动态性。动态模型的最显著特征是其动态特性,即系统必须在运行过程中实时地处理所出现的异常、随机事件和局部故障,实现动态重构,而且,动态模型应该能够对分布实时资源进行管理,及时监测外部环境和内部状态引发的分布应用负载变化,并依据性能需求参数和当时的系统资源状况实时地调整资源分配,以保证分布实时任务集中可靠实时地活动。

二、自适应性。自适应模型的研究目标是通过引入自适应资源管理策略及其实现机制,为网络应用提供可靠、高效、公平的系统服务。

三、不确定性。不确定模型是针对网络状态信息的不精确而提出的。由于在真实的网络中,不可能做到随时更新网络状态信息,加之网络规模的不断扩展和网络本身的分层结构,导致网络状态信息必然存在不精确性。解决的途径有:采用模糊理论进行不精确描述和概率分析;鉴于神经网络不需要精确的系统描述,只需训练学习,能够实现联想推理,容错性强,且具有并行结构,实时性强,因此可以采用神经网络对不精确模型进行建模;在网络系统的路由、调度、接纳控制等算法中引入随机性方法与策略等。

四、复杂性。复杂系统的性能分析所要解决的问题包括网络系统的有效模型、模型分析的状态爆炸及模型复杂性问题(NP问题)。系统状态数量随系统规模指数性增长,这是性能数量分析技术所面临的重要问题。采用模拟模型与方法可以分析、求解复杂的系统,但构造和使用模型的费用较高。目前仍没有普遍有效的技术和方法,复杂和大规模系统模型的分析求解仍然是一个极大的挑战。

通过上述分析可以看出,网络传输控制诸多难题的性质,恰好都是进化算法的特长所在。因为进化算法高度并行、随机、自适应等特征,所以特别适合于处理传统搜索算法解决不好的复杂的和非线性问题,可以在庞杂空间搜索优化解。目前的事实证明,基于启发、探索算法的进化计算的确是一种有效的求解技术。我们针对新一代网络传输控制中运用进化算法提出以下建议。

6.2 在网络传输控制中运用进化算法的原则

一、简化系统分析复杂性。例如：性能界限求解技术可以快速地给出系统模型的性能界限，即对系统模型进行分解、压缩，通过模型相似逼近、等价与迭代，进行模型的近似求解，从而突破某些复杂模型不能分析求解的难点。众所周知，进化算法的简单性和鲁棒性表现在算法可以没有问题的预先知识，这些知识在不断的选择和进化过程中自动获得。然而，我们能够针对具体问题的固有特性，给予一定的预先知识。这些预先知识可以明显地简化问题规模和适应度函数的复杂性，从而大大减少算法的计算时间。此外，在不影响优化效果的前提下，仅保留最重要的约束条件，减少次要的约束条件，也能大大降低计算复杂度。

二、提高进化算法收敛速度。在单个算法的动态模型中，当环境剧烈变化时使用开拓性的策略（如增加变异率，减少精英数量等），而当环境变化轻微时使用开采性的策略。根据网络控制具有分布式的特点，可以采用并行分布式进化算法，即采用多Agent协同进化策略完成相关结点间的配合。由于Agent具有自适应性、自主性、学习性和社会性等特征，因此它是实现自适应分布实时资源管理的一种理想机制。

三、把握好提高算法收敛速度和维持个体多样性之间的分寸。进化算法将适应度高的个体的编码信息迅速传播到下一代，使得整个种群朝着该最佳个体“同质化”，这对于维持个体多样性以应付可能出现的环境变化是个不利因素。所以，我们建议将预处理和在线计算机制相结合：除了保证最佳个体信息的传播外，在备用存储器中保存过去计算的历史记录。当发现环境变化时试图从这些记录恢复一部分旧个体信息。相关研究表明，采用扩大存储的方式有利于提高算法在动态环境中的鲁棒性，更易于全局优化。

下面，我们将结合具体的QoS组播路由选择应用实例，说明如何运用上述原则。

6.3 QoS组播路由选择应用实例

伴随网络技术、多媒体技术的飞速发展, Internet已逐步由单一的数据传送网向数据、语音、图像等多媒体信息的综合传输网演化。但Internet中现有的传输模式仍为单一的尽量做好服务, 无法满足多媒体应用和各种用户对网络传输质量的不同要求。因此, 为用户提供服务质量(QoS)成为Internet发展的重要挑战, 网络QoS技术作为公认的新一代网络的核心技术之一, 是当前网络研究和开发的热点。目前IETF已经提出了许多服务模型与机制来满足QoS需求, 典型的有: 集成服务(IntServ)/资源预留协议(RSVP)模型、区分服务(DiffServ)模型、多协议标签交换(MPLS)、流量工程(Traffic Engineering)和QoS路由(QoS-based routing)。QoS路由是根据网络上可用资源和流的QoS需求决定流的路由的机制。QoS路由应能达到以下目标: (1)动态确定可行路径; (2)优化资源利用; (3)对性能影响尽可能小。大部分的交互式多媒体应用是多方应用(multi-party applications), 需要网络的组播(multicast)支持。近年来的研究表明, 进化算法因为具有原理简单、鲁棒性强、易于并行分布处理等优点, 有效地避免了其它各种路由选择方案存在的缺陷, 能够较快较好地解决QoS路由选择问题。文献[17]提出了一种基于遗传算法的QoS路由选择策略。我们在它的基础上做了改进, 增加了动态环境下的应对策略。

6.3.1 问题描述

通常将QoS组播路由选择问题描述为一个有向图 $G(N, L)$, 其中, 有向图的顶点表示网络结点, 有向图的边表示网络链路, N 是网络结点集, L 是网络链路集。 $M \subseteq \{(s, D) | D \subset N - \{s\}\}$ 是网络中组播组集合。QoS组播路由选择即为在有向图 $G(N, L)$ 中, 为特定的 $m \in M$ 找出代价cost最小, 同时满足组播服务所需QoS的路由。

我们采用图6.1所示的网络拓扑结构, 它由13个结点组成。每个结点由三元组 $[nd, lr, nj]$ 表示, 元组中的元素分别表示结点延迟、结点丢失率和结点抖

动(图中 $1e-2$ 表示 1×10^{-2} , 余类同), 每一条边由三元组 (c, bw, ld) 表示, 元组中的元素分别表示代价、可用带宽和链路延迟。

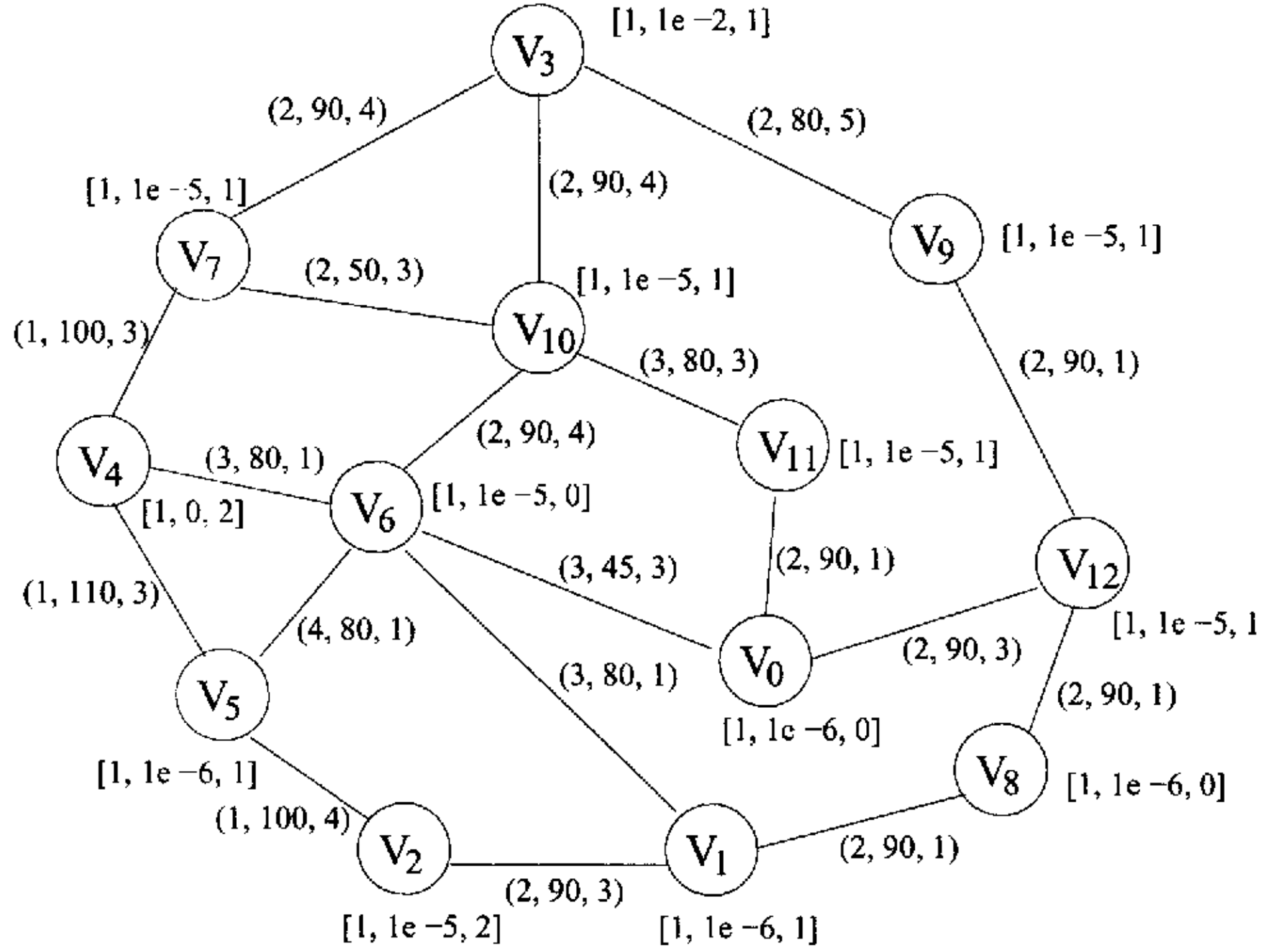


图 6.1: 网络拓扑结构及其QoS参数

这里采用基于源路由的路由选择机制, 即网络中的每一个源结点都维护一个全局的网络拓扑及其状态参数, 基于这些信息在源端计算并确定路由。拓扑结构及其参数值将随着网络的变化而变化, 这是通过结点之间的相互通信实现的。

6.3.2 算法描述

我们沿用了文献[17]中的编/解码机制、交叉变异策略、适应度函数及算法, 该算法首先对可用带宽小于需求带宽的链路进行过滤以简化网络拓扑。以此为基础我们做了如下改进: 在繁殖过程中检测适应度函数的变化, 然后根据环境变化的剧烈程度决定是否采取延迟策略。

改进后的繁殖方案描述如下:

```

reproduction(){
    copy elites from old population into new population;
    reevaluate elites;
    if (sum of elites fitness slightly changed)
        generate and evaluate children using newest fitness function;
    else
        generate and evaluate children using old fitness function;
    copy children into new population;
    replace old population with the new one;
    renew fitness function;
}

```

在上述算法中，我们采用已被评估的最佳个体适应度之和作为环境变化的标志。在将当前种群中的最佳个体（精英）复制到下一代种群时对其进行重估，如果这些个体适应度之和发生了变化，那么只可能有一个原因：适应度函数发生了变化。采用这种方法可以同时检测到环境变化的剧烈程度。延迟策略使用旧的适应度函数评估整个新种群，这意味着全局的网络拓扑及其状态参数被延迟更新。显然，这种做法在实际情况下是完全可行的。

6.3.3 算法收敛性证明

本文提出的进化算法有以下特征：(1)以概率 $p_c \in (0, 1)$ 进行交叉；(2)变异概率 $p_m \in (0, 1)$ ；(3)在选择前保存最优解且按比例选择。根据文献[18]定理2.7：具有交叉概率 $p_c \in [0, 1]$ ，变异概率 $p_m \in (0, 1)$ ，且采用比例选择法在选择前保留当前最优解的算法可以收敛到全局最优解。所以本文提出的算法可以收敛到全局最优解。

由于QoS组播路由选择是NP-hard问题，当网络规模巨大时，得到全局解的时间复杂度很大，可以通过限定进化世代数的办法求出一个次优解作为替代。

6.3.4 仿真结果及分析

举例：采用如图??所示的网络拓扑，假设结点2有组播请求，目的结点个数 $N_d = 5$ ， $D = \{6, 5, 8, 7, 11\}$ ，QoS需求带宽 $B_m = 60$ ，延迟 $D_m = 16$ ，丢失率

$L_m = 10^{-3}$, 抖动 $J_m = 3$ 。采用原算法中改进编码机制的进化算法参数设定为: 编码长度为34, 群体规模为100, 交叉概率为0.9, 变异概率为0.05, 其他参数与原文中相同。环境的变化通过每次评估后各个参数值分别从不变化、(0,3%)、(5%,10%) 3个变化率区间中随机偏移产生。新算法中环境变化剧烈程度门限设为4%。

通过仿真计算, 在环境变化率为0 (静态环境) 的情况下, 两个算法都可得到全局最优解, 结果见图6.2, 经过多次仿真, 得到平均迭代次数均为2018。算法在持续变化的动态环境中只可得到稳态次优解。当环境变化率为 (0,3%) 的情况下, 我们的算法与原算法到达稳态次优解所需的平均迭代次数均为1839。当环境变化率为 (5%,10%) 的情况下, 新算法到达稳态次优解所需的平均迭代次数为1573, 而原算法所需的平均迭代次数为1662。可见在较剧烈变动的环境中, 对原算法使用延迟变化的策略可获得更好的性能。

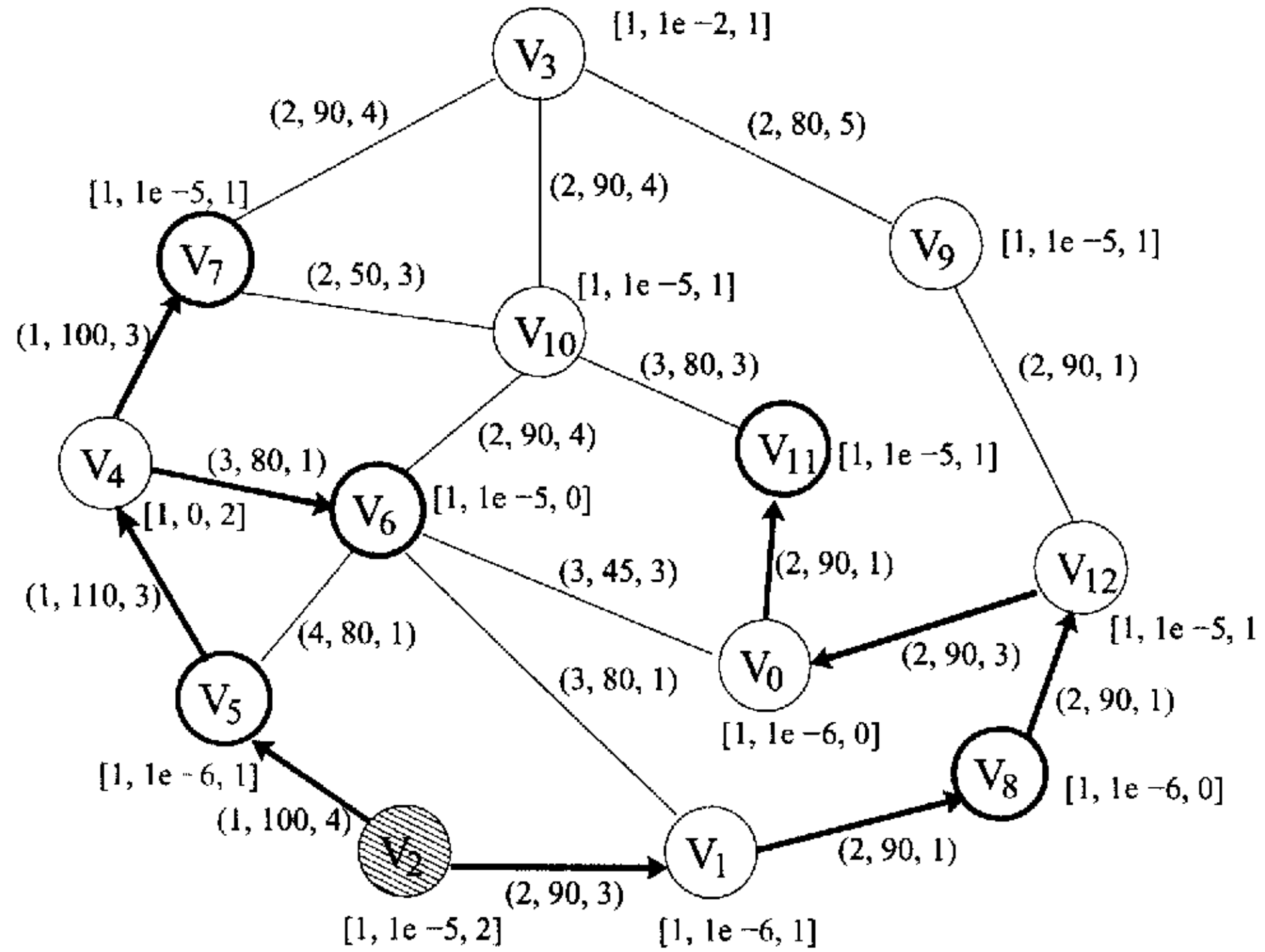


图 6.2: 基于进化算法的组播最佳路由

第7章 结论

7.1 结论

在本文中，我们讨论了快速变化环境的问题，特别是如何处理适应度函数在某个进化算法的世代内发生变化的话题。除了对本论点以及一些应付这些环境变化的思路做一般性描述以外，我们还推导了可用来计算在动态比特匹配问题中(1,2)EA下个世代母体适应度概率分布的解析式。

研究结果表明，在一定条件下，暂时忽略世代内的环境变化，能够比使用最新的适应度函数评估个体获得更好的性能。

实验证明，我们提出的两阶段变异分析方法不仅能够紧接着两个体因变异而分开之后，而且能在若干进化步骤之后被重演。该方法具备与已知分析方法无缝结合的能力，所以它可以被推广到更一般情况的分析。

本文最后探讨了进化算法在下一代网络传输控制中的应用，提出了解决动态QoS组播路由选择问题的进化策略，并用仿真实验证实了该策略的有效性。

7.2 今后的工作

对进化算法的理论分析目前尚处于初级阶段，仅限于变异、选择操作和简单进化策略，而含有交叉操作以及个体数目较多的实际算法只能靠统计试验方法获得经验结果。

除了继续从事理论研究外，我们目前正同样地从实验角度考察世代内变化的论点，这允许我们着眼于远更复杂的问题和绪论中提到的EA变种。

致谢

生物进化以适应自然，适宜的环境促成生命之美。本文是作者在2001年10月至2002年10月间赴联邦德国访问进修研究成果的基础上完成的。作者谨向对本文有贡献的以下项目、组织和个人表达衷心的感谢。

资助本研究工作的项目、合同单位、提供有关支持的组织和个人：2000年度巴登符腾堡—江苏友好省州交流项目——选拔并资助作者完成了为期一年的赴德访问进修；江苏省教育厅——《留学协议书》甲方一；南京邮电学院——《留学协议书》甲方二；郭胜英先生——江苏省国际教育交流与合作处工作人员、交流项目中方协调人；Kuno Egle教授——任职于卡尔斯鲁厄技术大学统计学和数学经济理论学院（Institut für Statistik und Mathematische Wirtschaftstheorie, SMW），是交流项目德方协调人；Annette Heininger女士——卡尔斯鲁厄技术大学学生、交流项目德方工作人员。她为初到德国的我提供了许多帮助；卡尔斯鲁厄技术大学外事办公室——负责提供交流项目规定的每月生活津贴。

指导、合作、协助完成本研究工作的组织和个人：卡尔斯鲁厄技术大学应用信息学和形式描述方法学院（Institut für Angewandte Informatik und Formale Beschreibungsverfahren, AIFB）——我访问进修的德方单位。那里安静，舒适，有很好的科研环境（炎热的夏天除外：-）；Hartmut Schmeck教授——AIFB学院负责人之一。他接受了我的访问申请，是我访问研究工作的负责人，并给予我适当的关心和指导；Jürgen Branke博士——AIFB学院研究助理。他既是我的指导教师，也是本研究的主要合作者；Christopher Ronnewinkel先生——在SAP公司工作。他在该研究的早期参与了有益的讨论。

提供便利条件的组织和个人：卡尔斯鲁厄技术大学SMW学院——2002年1月当AIFB房间紧张时，我搬进它的一间与AIFB位于同一楼层的办公室，并在那里一直工作到回国；Diethard Pallaschke教授——SMW负责人之一。他热情友善、平易近人的态度使我对他尊敬有加；Manfred Gehann先生以及Stefan Liede先生——AIFB学院技术支持。他们曾帮我安装电脑、打印机，设置网络，排除网络故障；Mohammad Salavati先生——AIFB学院行政主管。他做了一些后勤工作。

曾经对本研究工作提出建议和提供帮助的个人：Martin Middendorf教授——现任职于莱比锡大学信息学院（Institut für Informatik, IfI），是该领域的国际学术权威。他高瞻远瞩的建议使我最终选择动态规划这一研究课题；Daniel Merkle博士——现为IfI研究助理，是一位计算机高手。他帮我绕过了网络安全检查，使我在炎热的夏季能从凉爽的地下室机房通过远程登录方式潜心完成最关键的测试程序。他还教我使用TexPower演示软件；Bernd Scheuermann先生——AIFB学院博士生。他是个沉默寡言的孤独者，但我每次都能从他那里得到切实的帮助；Michael Stein先生——AIFB学院博士生。他曾试图帮我实现远程登录过程的自动化；还有同一课题组的德国同事们——他们的真诚、友善和言行一致的可贵品质令我难忘。

其他应该感谢的个人：马严教授——任职于北京邮电大学信息网络中心。他作为我的硕士导师始终关心着我的论文工作，并在关键时刻提供了宝贵的帮助；张宗橙教授——南京邮电学院教育技术系主任。作为我的部门领导，他大力支持我的论文工作；秦军副教授——南京邮电学院教育技术系副主任。她十分关心我的成长，曾给予我多方面的照顾。

最后，特别感谢我的父亲王志华、母亲姚学梅，他们用无私的爱给予我巨大的精神鼓励和物质资助；还有我的爱妻顾生蓉，她是我生活中的良师益友，是我心灵的好伴侣。

参考文献

- [1] J.Branke. Evolutionary approaches to dynamic optimization problems - updated survey. In *CECCO Workshop on Evolutionary Algorithms for Dynamic Optimization Problems*, pages 27-30, 2001.
- [2] J.Branke. *Evolutionary Optimization in Dynamic Environments*. Kluwer, 2001.
- [3] J.Branke. and W.Wang. Theoretical analysis of simple evolution strategies in quickly changing environments. Technical Report 423, Institute AIFB, University of Karlsruhe, 76128 Karlsruhe, Germany, 2002.
- [4] G.Dozier. Distributed steady-state neuro-evolutionary path planning, in non-stationary environments using adaptive replacement. In D. Whitley, D. Goldberg, E. Cantu-Paz, L. Spector, I. Parmee, and H.-G Beyer, editors, *Genetic and Evolutionary Computation Conference*, pages 58-65. Morgan Kaufmann, 2000.
- [5] G.Dozier. Steady-state evolutionary path planning, adaptive replacement, and hyper-diversity. In M.Schoenauer, K.Deb, G. Rudolph, X. Yao, E. Lutton, J. J. Merelo, and H.-P. Schwefel, editors, *Parallel Problem Solving from Nature (PPSNVI)*, volume 1917 of *LNCS*, pages 562-570. Springer, 2000.
- [6] S. Droste. Analysis of the (1+1) EA for a dynamically changing onemax-variant. In *Congress on Evolutionary Computation*, pages 55-60, 2002.

- [7] J. E. Smith and F. Vavak. Replacement strategies in steady state genetic algorithms: Dynamic environments. *Journal of Computing and Information Technology*, Volume 7, No.1, page 49-59, 1999.
- [8] S. A. Stanhope and J. M. Daida. Optimal mutation and crossover rates for a genetic alorithm operating in a dynamic environment. In *Evolutionary Programming VII*, volume 1447 of *LNCS*, pages 693-702. Springer, 1998.
- [9] S. A. Stanhope and J. M. Daida. (1+1) Genetic alorighm fitness dynamics in a changing environment. In *Congress on Evolutionary Computation*, volume 3, pages 1851-1858. IEEE, 1999.
- [10] F. vavak and T. C. Fogarty. A comparative study of steady state and generational genetic algorithms for use in nonstationary environments. In T. C. Fogarty, editor, *AISB workshop on Evolutionary Computing*, volume 1143 of *LNCS*, page 297-304. Springer, 1996.
- [11] J. Arabas and S. Kozdrowski. Applying an evolutionary algorithm to telecommunication network design. *IEEE Transactions on Evolutionary Computation*, Volume 5, No.4, 2000
- [12] C. He. Route selection and capacity assignment in Computer. *IEEE ICIPS Papers Collection*, 1997.
- [13] C. He. Route selection in virtual circuit data networks based on genetic algorithms. *Journal of Electronics*, 1997.
- [14] 何翠红. A genetic algorithm to route selection in packet switched communication networks. *通信学报*, Volume 18, No.3, 1997.
- [15] 任庆生, 叶中行, 曾进. 基于实数型遗传算法的通信系统可靠性最优分配. *通信学报*, Volume 21, No.3, page 43-46, 2000.

- [16] 林闯, 单志广, 盛立杰. 新一代网络传输控制的模型与机制. 计算机世界, May 5, 2001.
- [17] 何小燕, 费翔, 罗军舟, 吴介一. Internet中一种基于遗传算法的Qos路由选择策略. 计算机学报, Volume 23, No.11, page 1171-1178, 2000.
- [18] 陈国良, 王煦法, 庄镇泉等. 遗传算法及其应用. 北京: 人民邮电出版社, 1996.

索引

$(\mu + \lambda)$ -繁殖, 6

(μ, λ) -繁殖, 6

$(d - s)$ -比特变异, 24-27

$(r - s)$ -比特变异, 22-27

d -比特变异, 14-33

r -比特变异, 9-27

s -比特变异, 22-27

DE, 参见 动态环境

EA, 参见 进化算法

EA变迁, 参见 适应度变迁

Kolmogorov-Smirnov统计 表

Kolmogorov-Smirnov statistics, 35

onemax问题, 7

保守选择 conservative selection, 6

变化后的环境, 9

变化后的适应度, 参见 环境变化后的适应度

变异 mutate, 9-43

定义, 9

变异率 mutation rate, 9-35

定义, 9

概率分布, 14

有效比特数, 13

重估 reevaluate, 5-35

定义, 10

初始化 initialize

个体, 9

样板, 8

当前解集, 4

定理3.1, 参见 变异概率分布

定理3.2, 参见 环境变化后的适应度
概率分布

定理5.1, 参见 两阶段变异概率分布

定态繁殖 steady state reproduction,
6

定义5.1, 参见 两阶段变异定义

动态比特匹配问题 dynamic bit
matching problem, 7-43

定义, 8

动态环境 Dynamic Environment,
DE, 4-32

定义, 8

- 动态优化 dynamic optimization, 4-16
- 繁殖方案 reproduction scheme, 4-20
定义, 5
- 个体 individual, 4-43
定义, 9
- 后代 child, 5-35
定义, 9
- 环境 environment, 5-43
环境变化, 5-43
定义, 8
环境变化后的适应度, 9-15
定义, 9
概率分布, 14
- 解, 4-20
- 进化策略 evolution strategy, 5-43
定义, 10
- 进化算法 Evolutionary Algorithm, EA, 4-43
定义, 9
- 精英 elite, 5
精英策略 elitism, 5
- 决策规则, 4
- 老种群, 5-9
定义, 9
- 两阶段变异 two-step mutation, 22-43
定义, 23
概率分布, 24
图解, 23
验证, 34
- 马尔可夫决策过程 Markovian decision process, 12
- 蒙特·卡罗试验 Monte-Carlo trial, 34
- 母体 parent, 5-43
定义, 9
母体种群, 参见 老种群
- 目标, 参见 目标函数
目标变化, 参见 环境变化
目标函数, 4-8
目标样板 object template, 参见 样板
- 评估 evaluate, 4-32
定义, 8
评估值, 参见 适应度
- 世代 generation, 4-43
定义, 9
世代交替, 9
- 适应度 fitness, 4-43
定义, 8

- 变迁, 12–34
- 适应度函数 fitness function, 5–43
- 定义, 8
- 变化率, 35
- 转移概率, 12–13
- 随机动态环境 dynamic and stochastic environment, 4
- 推论5.1, 参见 选择结果概率分布 (无优先权)
- 推论5.1, 参见 选择结果概率分布 (有优先权)
- 显式重估 explicit reevaluation, 10–27
- 定义, 10
- 新母体种群, 参见 新种群
- 新种群, 7–9
- 定义, 9
- 选择 select, 6–43
- 定义, 9
- 结果概率分布 (无优先权), 15
- 结果概率分布 (有优先权), 15
- 样板 template, 7–22
- 定义, 8
- 遗传算法 genetic algorithm, 1
- 引理3.1, 参见 变异有效比特数
- 隐式重估 implied reevaluation, 10–32
- 定义, 10
- 整代繁殖 generational reproduction, 5
- 种群 population, 4–32
- 定义, 4
- 种群个体总数 population size, 5–32
- 定义, 9
- 子变异 sub-mutation, 22–28
- 定义, 22
- 最优变异率, 12–31
- 定义, 12