

摘要

从人工智能跨越到计算智能是计算机科学发展史上的一次重大变革。以进化算法和神经网络为核心的计算智能理论得到了长足的发展。最近十几年来，各种新的进化算法也纷纷被提出。这些算法都具有与问题无关、通用性好、收敛速度快、容易跳出局部最优等优点，因此在许多领域都得到了广泛的应用。

进化算法是一种具有高度计算复杂性的智能算法。当求解问题较复杂时，其计算所需的时间复杂度也比较高，同时，由于解集的庞大和局部最优问题的存在，在其中筛选出精确解所花费的过程也比较复杂。因此我们希望有一种能够有效降低进化算法时间复杂度的策略，以节约进化算法的计算时间。

目前存在着大量的进化算法模型，研究它们会发现有一些共同的规律，比如都存在一组优劣程度不等的候选解，经过一轮轮的反复迭代计算，每轮新生成的候选解总体上比上一轮精度更好，且存在着一定的相关性。每轮迭代所产生的最好候选解之间也存在着一定的相关性。那么能否研究这些算法的共同规律，优化这些规律，就能使计算智能算法模型的计算时间和性能都会得到整体上的提高。

本文介绍了目前主流的一些进化算法模型，并在总结它们共同规律的基础上，提出了一种分析解集进化规律来预测新的解集的算法思想，被称做附加统计-预测算法。它能够智能猜测最优解的取值范围，跳过进化算法中的适应值函数计算。通过这种方法可以对解集的演变规律进行分析，以诱使新的解集启发式的生成在最优解可能出现的范围内。从而大大减少了计算时间。具体内容如下：

(1) 对进化算法的基本理论、基本思想和主流算法进行了详细的综述。论述了进化算法的发展历史、应用领域和发展趋势，并着重介绍了本文所阐述算法需要用到的进化算法的共同规律。

(2) 介绍了本文所论述的附加统计-预测算法。首先分析该算法的来源和实现原理，然后阐述了其对应的基本思想和算法实现过程，并依次分析了适合应用该算法的一些进化算法模型。

(3) 研究了使用附加统计-预测算法改进了一些进化算法模型。结合了当今主

流的几种进化算法，即遗传算法、粒子群优化算法和分布估计算法。列举附加统计-预测算法的嵌入位置和附加效果，以阐明本算法的普适性和可行性。

(4) 将使用了附加统计-预测算法的进化算法用于最优化求解等问题进行对比实验。这些实验包括最优化求解等问题，对比这几种进化算法结合附加统计-预测算法后在计算时间和性能上的改进，以阐述本算法的有效性和可推广性。

关键字：附加统计-预测算法；进化算法；计算智能；计算复杂性

Abstract

Across from the artificial intelligence to computational intelligence in computer science is the history of the development of a major change. A genetic algorithm and neural network as the core of computational intelligence theory has been developed by leaps and bounds. The past decade, a variety of new evolutionary algorithms have been proposed. These algorithms have nothing to do with the problem, common good, fast convergence, easy to jump out of local optimum, etc., so in many areas has been widely used.

Evolutionary algorithm is a highly computational complexity of the intelligent algorithm. When the solving problem more complicated, the calculation of the required time complexity is also relatively high, at the same time, as the solution set of the large and local optimization problem, in which the exact solutions of selected spent more complicated process. Therefore, we hope to have an algorithm can effectively reduce the time complexity of the strategy to reduce the computation time of evolutionary algorithms.

There are a large number of evolutionary algorithm model, they will find a number of common law, for example, there are advantages and disadvantages of a group of candidate solutions of varying degrees, after repeated rounds of an iterative calculation, each round of the new candidate solution generated the overall accuracy of better than the last round, and there is a certain degree of correlation. Generated by each round of iteration the best solution among the candidates there is a certain degree of correlation. These algorithms can then study the common law, and optimize these laws, we will be able to model computational intelligence algorithm for calculating the time

and overall performance will be improved.

This article describes some of the current mainstream model of evolutionary algorithms, and to take stock of their common law, based on the analysis of a solution set to predict the evolution of the law of the solution set of the new thinking of the algorithm, known as the Additional statistics - prediction algorithm. It can guess the smart range of optimal solution, skip the evolutionary algorithm in the calculation of fitness function. This method can be adopted for the solution set of an analysis of the evolution of the law in order to induce a new generation heuristic solution set of the optimal solution in the framework of the possible. The computation time are reduced.

As follows:

(1) Of evolutionary algorithm's basic theory, thought and mainstream algorithm summarized in detail. Discuss the history of the development of evolutionary algorithms, application areas and development trends, with an emphasis on the algorithm in this paper required by the evolutionary algorithm used in the common pattern.

(2) This article discusses the Additional Statistic Prediction Algorithms. The algorithm first analyzes the sources and principles of implementation, and then their counterparts on the basic idea and algorithm of the implementation process, and followed by an analysis of the algorithm is suitable for a number of evolutionary algorithm models.

(3) Research some evolutionary algorithm model with the Additional Statistic Prediction Algorithms. Combination the several of today's mainstream evolutionary algorithm, Such as GA, PSO and EDA. Additional Statistic Prediction Algorithm listed embedded position and the additional

effects, to clarify the algorithm and the feasibility of the universal.

(4) Additional Statistic Prediction Algorithms for optimization of evolutionary algorithms to solve issues such as comparative experiment. These experiments include optimization problem solving, etc., compared to these types of evolutionary algorithms combined with additional statistical prediction algorithm in the computation time and performance improvements to the effectiveness of this algorithm.

Keywords: Additional Statistic Prediction Algorithms; Evolutionary Algorithm; Computational Intelligence; Computational Complexity

原创性声明

本人郑重声明：所呈交的学位论文，是本人在导师的指导下，独立进行研究所取得的成果。除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的科研成果。对本文的研究作出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律责任由本人承担。

论文作者签名：_____ 日 期：_____

关于学位论文使用授权的声明

本人完全了解济南大学有关保留、使用学位论文的规定，同意学校保留或向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借鉴；本人授权济南大学可以将学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或其他复制手段保存论文和汇编本学位论文。

(保密论文在解密后应遵守此规定)

论文作者签名：_____ 导师签名：_____ 日期：

第一章 绪论

1.1 引言

从人工智能跨越到计算智能是计算机科学发展史上的一次重大变革[1-5]。以进化算法和神经网络为核心的计算智能理论得到了长足的发展。计算智能以生物进化的观点认识和模拟智能，以数据为基础，通过训练建立联系而进行问题求解。按照这一观点，智能是在生物的遗传、变异、生长以及外部环境的自然选择中产生的。在用进废退、优胜劣汰的过程中，适应度高的结构保存下来，智能水平也随之提高^[1-5]。

进化算法是依靠模拟生物种群进化的过程来实现解集向精确解逐步演化的。正如大自然演化过程的漫长，大部分进化算法求解的精确化过程也需要消耗大量机器运算时间。例如使用进化算法训练神经网络有时需要数天之久。为了降低程序的计算复杂度，节约计算时间，我们希望能够找到一种在解集中智能地发现最优解的快速算法。而这种算法还必须具有相当的普适性和推广性，以应付形形色色的进化计算模型。这就需要我们研究分析现有和将来可能出现的进化计算模型的共性，从共性中找到一种通用的方法。

如果我们分析现有的和将来可能出现的这些进化计算模型，则会发现其有共同规律——即目前几乎所有的进化算法，均是由一组解集经过若干代演算后逐步逼近更精确的解集。那么如果我们找到一种分析解集变化规律的方法，则能有效地预测出更精确的解集的取值可能，从而达到上述目的。这就是本论文所要阐述的主题。

1.2 研究背景

进化算法作为计算智能理论的核心问题之一。最近十几年来，各种新算法也纷纷被提出。比如遗传算法(Genetic algorithm, GA)^[6-10]、免疫算法(Immune Programming, IP)^[11-14]、遗传编程(Gene Programming, GP)^[15-19]、基因表达式编程(Gene Expression Programming, GEP)^[20-22]、概率增强式程序进化算法(Probabilistic incremental program evolution, PIPE)^[23]、人工蚁群算法(ant colony optimization、ACAO)^[24-26]、粒子群

优化算法 (Particle Swarm Optimization , PSO) [27-31]、分布估计算法 (Estimation of Distribution Algorithms, EDA) [32-33]、入侵草算法(Invasive Weed Optimization, IWO). [40]、区域再生算法(region reproduction algorithms、 RRA)等。

这些算法都具有与问题无关、通用性好、收敛速度快、容易跳出局部最优等优点，因此在许多领域都得到了广泛的应用。研究这些新兴的计算智能算法模型，会发现它们都有如下基本步骤：

- 1) 都要求有一组随机数据作为初始解集。
- 2) 将这组初始解集代入该计算模型后，得到了一组近似的解集。并有专门的适应度函数来判断现存所有解集哪些比较精确，去粗存精保留较好的解集。
- 3) 利用步骤 2 的解集为初始数据，重做若干次步骤 2，每次得到的保留解集比前次循环更为精确，直到解集比较满意为止。

研究这些计算智能算法模型的基本步骤，我们会发现这些算法具有下面一些共同特征。

- A: 存有一组优劣程度不等的候选解。
- B: 经过若干轮的反复迭代计算，每轮新生成的候选解总体上比上一轮精度更好，且存在着一定的相关性。每轮迭代所产生的最好候选解之间也存在着一定的相关性。
- C: 利用步骤 B 的解集为初始数据，重做若干次步骤 B，每次得到的保留解集比前次循环更为精确，直到解集比较满意为止。

上面的步骤 B 所生成的每一轮的解集，虽然和上下轮的解集相比，总体上是不断演化的。但由于求解问题的高度复杂性，往往会导致算法收敛速度过于缓慢，需要几小时甚至几天的时间，来完成平时看起来或许很简单的收敛过程。这就需要我们找到一种新方法来智能地优化进化计算模型的收敛轨迹。这里，我们提出了一种方法来分析每轮循环中最优解产生的时间序列的变化过程，通过这个时间序列矩阵，来预测若干轮循环以后最优解的可能变化趋势，从而提前猜测出可能的最优解取值，减轻算法的计算量。

这个被称为附加统计-预测算法 (Additional Statistic- Prediction Algorithms, 以下简称为 ASPA) 的通用进化算法改进策略，就是本文所论述的主题。通过对它的研究。我们实现了下列一些目的：

A: 使用 ASPA 算法来优化几个有代表性的进化计算模型，包括遗传算法 (Genetic Algorithm 、 GA)、粒子群优化算法 (Particle Swarm Optimization , PSO)、分布估计算法 (Estimation of Distribution Algorithms, EDA)。

B: 使用经过 ASPA 改进的进化计算模型，来解决几个典型的最优化求解问题。这些问题包括 Schwefel' s Problem、Shifted Rastrigin' s Function、50 维最优化问题等。

C: 使用经过 ASPA 改进的进化计算模型，来解决几个典型的生物信息学问题。

1.3 论文内容安排

第一章 绪论，主要简介进化计算模型的共性，引出了本论文的研究背景。

第二章 进化算法与预测理论，主要介绍了进化计算的基本理论、基本思想和基本方法。以及一些典型的进化计算模型。和一些预测理论。

第三章 附加统计-预测算法，主要阐述了 ASPA 的设计思路，算法流程，和典型用法。系统地介绍了 ASPA 的基本理论基础以及其的实现方法。

第四章 附加统计-预测算法与其他算法结合，主要阐述了 ASPA 与几种典型的进化算法相结合的方法，以及其对典型测试用例的改进程度。

第五章 总结和展望，对附加统计-预测算法的研究和应用做出了总结，同时指出存在的问题，并对未来的研究和发展做出了展望。

第二章 进化算法与预测理论

2.1 进化算法概述

进化算法是计算智能理论的核心组成之一。进化算法是基于达尔文的进化论，在计算机上模拟生命进化机制而发展起来的一门新学科^[1]。近年来，由于进化计算在机器学习、过程控制、经济预测和工程优化等领域取得了巨大的成功，引起了数学、物理学、化学、生物学、计算机科学、社会科学、经济学及工程应用领域的科学家们的极大兴趣。自 20 世纪 80 年代中期以来，世界上许多国家就掀起了进化计算的研究热潮^[2]。

地球上的生物，部是经过长期进化而形成的。根据达尔文的自然选择学说，地球上的生物具有很强的繁殖能力。在繁殖过程中，大多数生物通过遗传，使物种保持相似的后代；部分生物由于变异，后代具有明显差别，甚至形成新物种。正是由于生物的不断繁殖后代，生物数目大量增加，而自然界中生物赖以生存的资源却是有限的。因此，为了生存，生物就需要竞争。生物在生存竞争中，根据对环境的适应能力，适者生存，不适者消亡。自然界中的生物，就是根据这种优胜劣汰的原则，不断地进行进化。进化算法就是借用生物进化的规律，通过繁殖—竞争—再繁殖—再竞争，实现优胜劣汰，一步一步地逼近问题的最优解。进化算法中的“进化”二字，就是由此而来的^[3]。

进化计算采用简单的编码技术来表示各种复杂的结构，并通过对一组编码进行简单的遗传操作和优胜劣汰的自然选择来指导学习和确定搜索的方向。由于它采用种群的方式组织搜索，可以同时搜索解空间的多个区域。而且用种群组织搜索的方式使得进化算法特别适合大规模并行计算。在赋予进化计算自组织、自适应和自学习等特征的同时，优胜劣汰的自然选择和简单的遗传操作使得进化计算具有不受其搜索空间限制性条件的约束及不需要其他辅助信息的特点。这些特点也使得进化计算不仅能获得较高的效率而且具有简单、易操作和通用的特性^[1-5]。

2.2 目前流行的部分进化算法

由于设计者对进化规则的不同理解和诠释，以致目前有许多种进化算法得到广泛的使用。在此，我们简述了大约 10 种进化计算算法，同时再从里面选择了三种具有代表性的进化算法加以详细的介绍，总结出它们的共同特征。同时作为实际用例来进行具体分析。

2.2.1 遗传算法

1975 年，Holland 首先提出了遗传算法 (Genetic algorithm, GA)^[8]。遗传算法是一种借鉴生物界自然选择和自然遗传学机理上的迭代自适应概率性搜索算法。遗传算法主要由选择、交叉和变异三个算子组成。它应用于实际问题时，要对优化问题进行编码，称为个体，个体的集合称为种群，每一个个体都表示问题的一个潜在解。遗传算法首先随机产生的初始种群，然后通过选择、交叉、变异使种群一代代向最优解进化。选择机制基于适者生存理论，它应用于群体中的每一个个体，使这些个体按照其适应值进化到下一代群体中。标准遗传算法利用交叉和变异算子对解空间进行搜索，交叉算子是两者中的主要算子。交叉算子组合了父辈个体特征，是产生新个体的主要途径。它和选择算子相结合，构成算法中信息交换的重要方法。

遗传算法是一种鲁棒的搜索方法，具有计算简单及功能强的特点，它通过目标函数来计算适应度，不需要其他的推导和辅助信息，对问题的依赖较小。遗传算法对待寻优函数基本无限制，它既不要求函数连续，也不要求可微。遗传算法从许多初始点开始并行操作，而不是从一个点开始，因而可以有效地防止搜索过程收敛于局部最优，而且有较大的可能求得全部的最优解，同时也可通过大规模并行计算来提高计算速度，更适合复杂问题的优化。

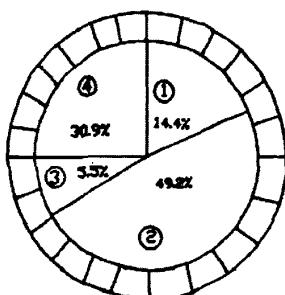
遗传算法在解空间内不是盲目穷举或完全随机的测试，而是一种启发式搜索，它不是简单的随机走动，它可以有效地利用已经有的信息处理来搜索那些有希望改善解质量的串，类似于自然进化，遗传算法通过作用于染色体上的基因，寻找好的染色体来求解问题。与自然界相似，遗传算法对待求解问题本身一无所知，它所需要的仅是对算法所产生的每个染色体进行评价，并基于适应度值来改造染色体，使适用性好的

染色体比适应性差的染色体有更多的繁殖机会。

遗传算法一般按照以下步骤执行^[8]:

- 1) 对于待处理的问题确定一个表示某组可能解适应度的目标函数。
- 2) 将候选解的种群初始化为服从特定的约束。一般情况下，将每组试验解编码为向量x，称为染色体，其中的元素称为基因，特定位置的不同值称为等位基因。
- 3) 种群中每个染色体 x_i , $i=1, \dots, P$ 解码为一个适于估价的形式，算出对于目标的适应度值 $\mu(x_i)$ 。
- 4) 算出每个染色体的繁殖概率 p_i , $i=1, \dots, P$ ，使得它被选择的可能性正比于它相对于种群中其他染色体的适应度。如果每个染色体的适应度取正数的值，通常使用轮盘赌选择。
- 5) 一个新的染色体种群从当前种群中的概率选择串取决于繁殖概率 p_i , $i=1, \dots, P$ 而产生。被选中的染色体通过交叉和比特突变等特定基因操作产生它的“后代”。交叉应用于两个(父代)染色体间，选择一些位置，把第一个串的某些位置的值和第二个串相应位置交换形成两个新的(后代)染色体。最简单的单点交叉操作是随机选择一个位置，把第二个串这个位置后的内容拼接到第一个串这个位置前的内容后，反之亦然(见图2)。比特突变指形成新解的时候有一定概率在某位发生突变。交叉的概率值通常取0.6到0.95范围，突变概率通常取0.001到0.01。

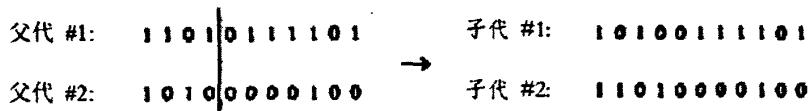
进化算法中的选择通常取决于不同后代的适应度。在典型的方法中，每个染色体给定一个复制进入下一代的概率，这个概率正比于它相对于种群中其他染色体的适应度。通过连续试验管理哪个染色体应当被选中，直到所有可能的位置都被遍历。适应度高的染色体比适应度低的有更多机会给下一代提供复本。



No.	String	Fitness	% of Total
1	01101	169	14.4
2	11000	576	49.2
3	01000	64	5.5
4	10011	361	30.9
Total		1170	100.0

(图1) 进化算法中的轮盘赌选择法

6) 如果找到一个可行解或者可供计算的时间已到则结束算法，否则记录新的染色体回到第(3)步重复执行过程



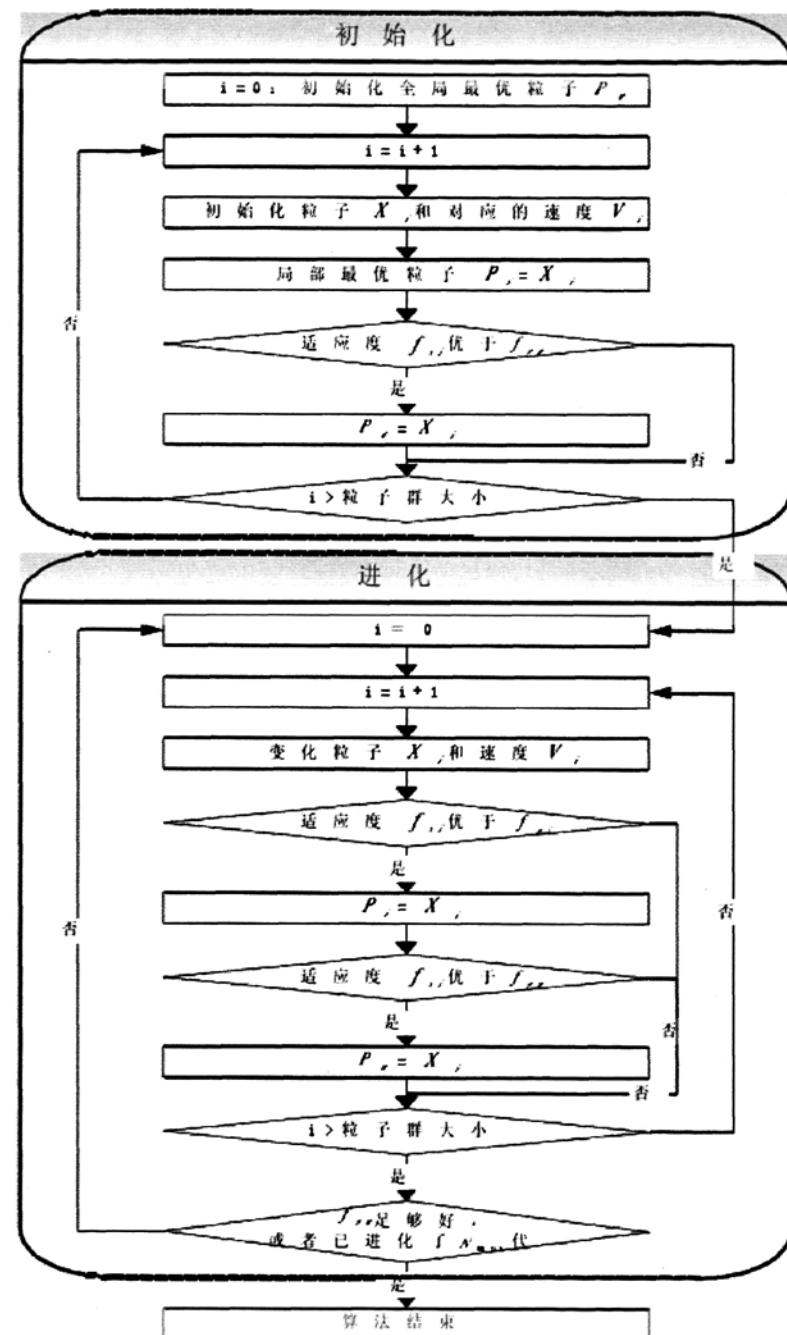
(图2) 单点交叉操作

2. 2. 2 粒子群优化算法

粒子群优化算法 (Particle Swarm Optimization , PSO)^[27]是近年来发展起来的一种新的进化算法，属于进化算法的一种，和遗传算法相似，它也是从随机解出发，通过迭代寻找最优解，它也是通过适应度来评价解的品质。但是它比遗传算法规则更为简单，它没有遗传算法的“交叉” (Crossover) 和“变异” (Mutation) 操作。它通过追随当前搜索到的最优值来寻找全局最优

PSO 同遗传算法类似，是一种基于叠代的优化工具。系统初始化为一组随机解，通过叠代搜寻最优值。但是并没有遗传算法用的交叉 (crossover) 以及变异 (mutation)。而是粒子在解空间追随最优的粒子进行搜索。同遗传算法比较，PSO 的优势在于简单容易实现并且没有许多参数需要调整。目前已广泛应用于函数优化，神经网络训练，模糊系统控制以及其他遗传算法的应用领域

如前所述，PSO 模拟鸟群的捕食行为。设想这样一个场景：一群鸟在随机搜索食物。在这个区域里只有一块食物。所有的鸟都不知道食物在那里。但是他们知道当前的位置离食物还有多远。那么找到食物的最优策略是什么呢。最简单有效的就是搜寻目前离食物最近的鸟的周围区域。PSO 从这种模型中得到启示并用于解决优化问题。PSO 中，每个优化问题的解都是搜索空间中的一只鸟。我们称之为“粒子”。所有的例子都有一个由被优化的函数决定的适应值 (fitness value)，每个粒子还有一个速度决定他们飞翔的方向和距离。然后粒子们就追随当前的最优粒子在解空间中搜索。



(图3) PSO算法流程

PSO 初始化为一群随机粒子(随机解)。然后通过叠代找到最优解。在每一次叠代中，粒子通过跟踪两个“极值”来更新自己。第一个就是粒子本身所找到的最优解。这

个解叫做个体极值 pBest。另一个极值是整个种群目前找到的最优解。这个极值是全局极值 gBest。另外也可以不用整个种群而只是用其中一部分最为粒子的邻居，那么在所有邻居中的极值就是局部极值。

PSO 的基本算法步骤描述如下^[27]:

- 1) 初始化粒子群，即随机设定每个粒子的初始位置 X_i 和初始速度 V_i ；
- 2) 计算各个粒子的适应度值；
- 3) 对每个粒子，比较它的适应度值和它经历过的最好位置 P_i 的适应度值，如果更好，更新 P_i ；
- 4) 对每个粒子，比较它的适应度值和群体经历过的最好位置 P_g 的适应度值，如果更好，更新 P_g ；
- 5) 根据公式 1 和公式 2 调整粒子的速度 X_i 和位置 V_i ；
- 6) 如果达到结束条件，则结束；否则转步骤 2。

2. 2. 3 分布估计算法

分布估计算法 (Estimation of Distribution Algorithms, EDA)^[32-33] 提出了一种全新的进化模式。在传统的遗传算法中，用种群表示优化问题的一组候选解，种群中的每个个体都有相应的适应值，然后进行选择、交叉和变异等模拟自然进化的操作，反复进行，对问题进行求解。而在分布估计算法中，没有传统的交叉、变异等遗传操作，取而代之的是概率模型的学习和采样。分布估计算法通过一个概率模型描述候选解在空间的分布，采用统计学习手段从群体宏观的角度建立一个描述解分布的概率模型，然后对概率模型随机采样产生新的种群，如此反复进行，实现种群的进化，直到终止条件。

根据概率模型的复杂程度以及不同的采样方法，分布估计算法发展了很多不同的具体实现方法，但是都可以归纳为下面两个主要步骤：

- 1) 构建描述解空间的概率模型。通过对种群的评估，选择优秀的个体集合，然后采用统计学习等手段构造一个描述当前解集的概率模型。
- 2) 由概率模型随机采样产生新的种群。一般的，采用蒙特卡罗方法，对概率模

型采样得到新的种群。

分布估计算法作为一种新型的进化算法，它的科学价值主要体现在以下几个方面。首先，从生物进化的数学模型上来看，分布估计算法与传统进化算法不同：传统进化算法是基于对种群中的各个个体进行遗传操作(交叉、变异等)来实现群体的进化的，是对生物进化“微观”层面上的数学建模；而分布估计算法则基于对整个群体建立数学模型，直接描述整个群体的进化趋势，是对生物进化“宏观”层面上的数学建模。其次，分布估计算法给人类解决复杂的优化问题提供了新的工具，它通过概率模型可以描述变量之间的相互关系，从而对解决非线性、变量耦合的优化问题更加有效，试验表明，分布估计算法能更加有效的解决高维问题，降低时间复杂性。最后，分布估计算法是一种新的启发式搜索策略，是统计学习理论与随机优化算法的结合，与其他智能优化算法的混杂设计，将极大丰富混杂优化算法的研究内容，给优化算法的研究提供了新的思路。

2.2.4 其他进化算法

为了充实对进化算法共性的研究，论文中还或多或少地提到了下列一些进化算法。它们是免疫算法(Immune Programming, IP)、基因表达式编程(Gene Expression Programming, GEP)、遗传编程(Gene Programming, GP)、概率增强式程序进化算法(Probabilistic Incremental Program Evolution, PIPE)、人工蚁群算法(Ant Colony Optimization、ACO)、多表达式编程(Multi Expression Programming, MEP)、区域再生算法(Region Reproduction Algorithms、RRA)等等。这些算法也是实际工作中相对常用的进化计算模型

2.3 预测的理论基础

在经济、工程、自然科学和社会科学等领域的实际工作者和研究人员，都不可避免地要和一系列的历史观察、统计数据打交道，我们把按时间顺序产生和排列的观察数据序列称为时间序列。而对时间序列进行预测则成为一个具有相当实际价值的应用研究领域。传统的时间序列分析是从纯数学的角度来分析的，多基于对数据的某种分

步假设和对预测模型的简化，但是由于实际应用中时间序列具有不规则、混沌等非线性特征，简化假设条件难以与实际相符，所以很难对系统建立理想的模型，预测更是无能为力，导致预测结果往往与实际偏差较大，计算智能技术的发展让研究者找到了新的问题解决思路，把它们用在这里可以解决在时间序列中难以解决或解决得不好的优化问题，发挥他们对目标函数要求不高的优势。利用它们优越的逼近性能来逼近序列，但由于其自身算法的一些缺陷，这些预测方法还在不断完善和发展之中。^[41-46]

随着社会经济系统日趋复杂，预测面临前所未有的困难。针对不同的环境及要求，预测理论的研究者提出了大量的方法。目前已有近 200 种预测方法。这些预测方法都是建立在如下的预测的基本原理之上的，它们是：可知性原理、可能性原理、连续性原理、相似性原理、反馈性原理、可控性原理、系统性原理^[46-52]。

2.3.1 可知性原理

又称规律性原理，是关于预测对象服从某种发展规律的原理。它认为，预测对象由于其发展规律可以被人们所掌握，因而其未来发展趋势和状况便可以被人们所知晓。人们的预测活动，不论其形式如何，都与这一原理有关。

2.3.2 可能性原理

预测对象未来发展的趋势和状况，它是在内因与外因的共同作用下出现的，它的结果具有不同的可能性，而常常不是只存在单一的可能性，但是预测对象演化到不同结果的可能性大小不同。对研究对象所作的预测，就是对它的未来发展的可能性进行预测。这一原理是建立在预测对象发展变化的结果与内外因共同作用有一定关系的基础上。

2.3.3 连续性原理

把预测对象的发展看成是一个连续的、统一的过程，其未来的发展是这个过程的继续。该原理强调，预测对象总是从过去发展到现在，再从现在发展到未来，如果没有预测对象的过去和现在，就没有它发展变化的未来。要对该对象的未来发展进行预

测，就必须了解它的过去和现在，预测对象发展过程中的过去、现在和未来，处于一种辩证的、统一的、连续的关系之中。连续性原理适用于那些发展极其缓慢、几乎没有变化的预测对象。

2.3.4 相似性原理

在许多情况下，被人们作为预测和研究对象的一个事物，其现在的发展过程和发展状况往往与一个已知的过去事物的一定阶段的发展过程和发展状况类似，相似性原理就是反映这种情况的原理。当甲事物现在的发展过程和发展状况与乙事物过去一定阶段的发展过程和发展状况相似时，乙事物后来阶段的发展过程和发展状况就可能相似于甲事物的未来发展过程。因此，预测者就可以根据乙事物后来的发展过程和状况，预测出甲事物的未来发展过程和状况。应用这样的原理来分析和预测研究对象，被称为类推法或历史类比预测方法。

2.3.5 可控性原理

作为人们预测对象的大量事物，其未来发展过程往往呈现出可以调节和控制的可能性。人们在考查事物发展的机制和过程时发现，把预测到的未来信息传递给可以影响、调节和控制研究对象未来发展的人或其它因素，就可以通过人的行为或其它因素的变化来达到调节和控制研究对象未来发展的目的。可控性原理就是关于预测对象的未来发展可以得到调节和控制的原理。可控性原理对预测活动具有重要指导意义。它强调，只要研究对象的未来发展趋势是可以调节、可以控制的，人类就可以通过对它的调节和控制，使它朝着符合人类目标和需要的方向发展。这正是有目的预测活动所希望达到的效果。

2.3.6 反馈性原理

人们在预测和研究事物的未来发展趋势和发展状况时，预测结果往往和预测目的有不同程度的差距。只有对预测依据进行反馈调节，才能缩小这种差距，做出符合活动目的的预测。依据这种原理，预测者要有效地进行反馈调节，必须抓住三个环节。首先，他必须认真地分析预测的结果与目的之间的差距，了解这种差距；其次，他必

须了解产生这种差距的原因；第三，根据已经查明的原因，对活动依据实施反馈调节。

2.3.7 系统性原理

这一原理，强调预测对象的未来发展是系统整体发展的连续，强调该对象内在与外在的系统作用，认为不考虑系统而进行的预测是一种顾此失彼的活动，将导致顾此失彼的决策。这一原理从下列三个方面指导预测活动：首先，它把预测对象分为内外两个系统，要求预测者在预测对象未来发展时，同时考虑它的内在系统和外在系统；其次，它把预测对象作为动态的、系统的发展过程来对待，预测对象的未来发展，是整个系统的动态发展，这种发展与系统内部子系统的相互作用和相互影响密切相关；第三，它强调预测对象的系统整体的属性与功能大于各子系统的总和，即整体最佳，要求做出整体最佳的预测。

2.3 本章小结

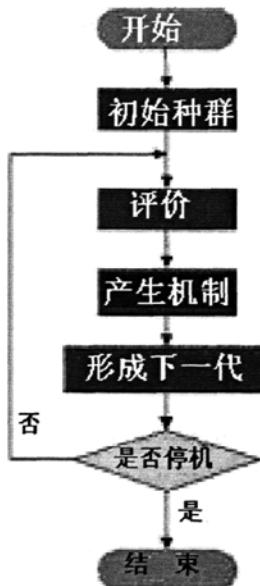
本章主要介绍了进化计算的概念与几种将要用到的典型的进化计算模型。包括遗传算法、粒子群优化算法和分布估计算法。这三种算法作为进化计算模型的代表，是近年来最为广泛使用的进化计算模型。对附加统计-预测算法的推广具有广泛意义。同时，本章也介绍了一些预测原理，用以对本文所介绍的附加统计-预测算法进行理论上的指导。

下一章里将论述附加统计-预测算法的设计思路和算法流程，以及如何将附加统计-预测算法嵌入到本章所提及的进化算法中去。

第三章 附加统计-预测算法

3.1 进化计算模型的共性

第二章论述了三种已有的进化算法模型。通过对其算法描述，我们可以知道，虽然它们都具有与问题无关、通用性好、容易跳出局部最优等特点。因此在许多领域都得到了广泛的应用。但是它们在实际应用的时候，并不具备我们希望看到的合理的收敛速度。即收敛效率不够高，收敛过程不够智能。为此我们需要找到进化计算模型的共通之处。让这个共通之处智能地向最优解靠近。那么，各种进化计算模型有哪些共通之处呢？



(图4) 进化算法流程

研究这些新兴的进化算法模型，会发现它们都有如下共同特点：

- 1) 都要求有一组随机数据作为初始解集。
- 2) 将这组初始解集代入该计算模型后，得到了一组近似的解集。并有专门的适应度函数来判断现存所有解集哪些比较精确，去粗存精保留较好的解集。
- 3) 利用步骤 2 的解集为初始数据，重做若干次步骤 2，每次得到的保留解集比

前次循环更为精确，直到解集比较满意为止。

通过研究第二章所列出的各种进化算法模型来看，它们都具备上述特点：

每轮遗传算法循环都有一次交叉——变异——选择过程。

每轮粒子群优化算法循环都有一次选择最优——追随最优过程。

每轮分布估计算法循环都会估计出一次新结果，并能统计出当前最优解，产生一系列基与概率的候选解。

每轮免疫算法循环都有一次克隆——无性繁殖——计算亲和性过程。

每轮基因表达式编程算法循环都有一次交叉——变异——选择过程。

每轮遗传编程算法循环都有一次交叉——变异——选择过程。

每轮概率增强式程序进化算法循环都有一次交叉——变异——选择过程。

每轮人工蚁群算法循环都可以额外统计每边信息素的分布信息。

每轮多表达式程序设计算法循环都有一次交叉——变异——选择过程，。

每轮区域再生算法循环也存在一次散布——选择过程。

既然上述三步骤是各类计算智能算法的共性，那么假如我们能找到一种方法来优化上面三个步骤的速度，就相当于找到了一种提高所有计算智能算法性能的新算法。这就是本文所提算法的基本思路。我们将寻找若干种算法，统计前几轮计算模型变化的规律。来智能预测出下一轮最优解的可能取值，并加入上述循环结尾。

3. 2 附加统计-预测算法的来源与设计思路

分析上述的几种计算智能算法模型，我们发现若要减轻它们的运算强度，减少循环的次数是最有效的方法之一。也就是设法使每次循环得到的解集收敛的更快且不会早熟。而要想使新产生的解集收敛加快，最直观的方法就是分析前几次循环时产生解集的变化趋势，把解集的变化过程看作一组时间序列，通过分析这组时间序列来推断下一次循环有可能产生什么样的解集。

对解集时序的分析也有两个思路，一个是统计法，通过统计历次循环产生的最优解集时序的分布趋势，来生成最符合历史分布规律的下一组可能解集。另一种是预测法，通过预测历次循环产生的最优解集的变化趋势，来生成最符合时序将来走势的下

一组可能解集。

无论是统计法还是预测法，生成的可能解集其实都不是很准的，因候选解集非常之多。所以还应根据可能解集再生成一组正态分布的候选解集。将这些候选解集一起代入上述步骤 B 的适应度函数，由该函数判断本算法生成的候选解集是否值得保留，而本文所提的算法自身不做适应度判断。

因此我们可以提出一种被称做附加统计-预测算法 (Additional Statistic-Prediction Algorithms, ASPA) 的计算智能算法改进策略，当进化算法进行若干轮后，统计一次解集的取值范围，使用某种模型分析其中的变化规律，并预测出下一轮的取值可能。

这个算法的基本思路来自粒子群算法 (Particle Swarm Optimization , PSO) 与分布估计算法 (Estimation of Distribution Algorithms, EDA)。

PSO 算法中，对全局最优解的记录和比较是该算法的一大亮点，PSO 的高效率很大程度归功于对全局最优解的追赶，而追赶的过程本身就是对全局最优解的历史变迁的再次分析过程。因此，单靠追踪全局最优解的运动轨迹，就可以很大程度上提高算法的性能。

PSO 算法通过大量粒子群来追踪全局最优解的运动历史轨迹来寻找最优解，这种统计分析全局最优解的历史变化规律的方法简单有之但并非高效。我们希望有更直观的方法对全局最优解的历史变化规律做出统计分析，从而对下一次全局最优解的可能出现位置作出预测。而 EDA 算法通过分析染色体分布规律的方法，则在估计染色体变化趋势方面具有独到的优势，因为 EDA 算法通过一个概率模型描述候选解在空间的分布，采用统计学习手段从群体宏观的角度建立一个描述解分布的概率模型，然后对概率模型随机采样产生新的种群。这种思路天然就可以预测出将来全局最优解的出现位置。

虽然本算法的设计思路脱胎于 PSO 和 EDA，但和两者具有本质的不同。和 PSO 相比，每次循环选取当前最优解作为分析目标是两个算法的共性。但 PSO 算法将当前最优解直接送如下一轮进行了正态分布的飞行路线选择，而本附加统计-预测算法则分析当前最优解在若干轮循环时变化的趋势，对下一轮变化进行预测后产生正态分布结果。

和 EDA 相比，两个算法都会在每次循环选取当前染色体概率分布情况。但 EDA 算法选择的是一轮循环时多个较优染色体的取值概率分布。是一次的静态选择，而附加统计-预测算法则是选择多轮循环时一个最优染色体的取值变化趋势，是多次的动态规划思想。两个算法都会进行正态分布的下一轮散布，但正态分布的最优解取值思想并非 EDA 的特色。

3.3 附加统计-预测算法流程

根据 3.1 部分所阐述，我们可以了解进化算法模型都可以描述为下面三个步骤：

- 1) 计算智能算法模型按照待解集问题的要求，随机生成一组初始解集。
- 2) 使用适应度函数评估生成解集的效果，取前面若干位好解集作为下一轮解集。
- 3) 按照主计算模型给出的变化规律，把步骤 2 生成的解集转成更精确的解。

这时，我们在每进行若干次步骤 3 后，用某种规则来分析计算智能算法模型最优解集的分布规律和变化趋势，生成一组可能解集，加入到候选解集中去参与竞争。这种思路就是我们要做的附加统计-预测算法。

附加统计-预测算法的思路就是分析每轮循环产生的最优解，分析前几次循环时产生解集的变化趋势，把解集的变化过程看作一组时间序列，通过分析这组时间序列来推断下一次循环有可能产生什么样的解集。有下列步骤：

- 1) 生成一个向量序列，记录每轮循环产生的最优解集的取值
- 2) 主计算模型进行循环时，记录最优解集的取值变化情况到向量序列
- 3) 主计算模型循环若干次后，统计和预测下一轮的可能取值
- 4) 按照步骤 3 所生成的统计和预测解集，以正态分布方法生成一系列候选解集

这里的关键问题在于步骤 3，使用什么样的方法来进行预测？总的来说是要计算步骤少，复杂度低，不能干扰主计算模型的性能。

为了形象地称呼步骤 3，我们把它叫做预测器。

目前来已经找到了四种方法进行时序统计-预测。

3.3.1 简单平均预测器

第一种是简单平均的统计-预测（简单平均预测器）。即将阶段最优解简单相加，

求它们的方差和数学期望，这个方法直观易用，但效果未必最佳，而且最优解变化幅度大时效果会显著降低，因此仅适用于简单场合。

简单平均预测器的基本算法步骤描述如下：

- 1) 主计算模型循环若干次后，将向量序列的每一列累加。
- 2) 将向量序列的累加和除以主计算模型循环的次数。即得到最优解分向量在这几轮循环的平均值。这个取值即我们要找的下一轮的可能取值。

3.3.2 线性加权预测器

第二种是线性加权的统计-预测（线性加权预测器）。即将不同阶段的最优解按照时序次序乘以一定的权值再相加，求它们的方差和数学期望，这个方法使得时序中较近的数据能够产生更重要的作用。效果要比简单平均的统计-预测好的多。

简单加权预测器的基本算法步骤描述如下：

- 1) 主计算模型循环 m 次后，从第一行依次读取向量组最优解乘以对应的权值。
第一行权值为 $1/m$ 。第二行元素的权值为 $2/m$ …依次类推，直到最后一行的最优解和权值相乘结束。
- 2) 将向量序列的每一列累加的时候乘以权值。
- 3) 再带权累加的向量和除以向量的总权值，结果即我们要找的下一轮的可能取值。

线性加权预测器的效果要比简单平均法好很多，最近最好解将占有较大的权重，而早期的次优解则占有较小的比重，而且计算量也没有增加多少。但这个方法仍然相对机械一些，无法反应出最优解的动态变化，尤其无法反映程序突然跳出局部最优解的过程。

3.3.3 HMM 预测器

第三种是基于隐马尔科夫模型的统计-预测（Hidden Markov Model。HMM 预测器）。将最优解的变化过程看作一个隐马尔科夫过程，利用已知样本的特征作为先验知识，

来对 HMM 进行训练，从而获取产生字符序列的状态序列的最大概率。

隐马尔科夫模型统计的并非当前取值，而是由当前取值状态到下一取值状态的概率转移矩阵（这个状态可以表现为遗传算法、EDA 算法染色体的变化、PSO 算法当前位置的变化等等）。它由两个符号序列组成，一个是不可观察的（隐含的）状态变化序列，另一个是由该不可观察的状态基于某些函数一对一（状态一对一，但产生的可选符号会有多个）产生的可观察的符号序列，利用已知样本的特征作为先验知识，来对 HMM 进行训练，从而获取产生字符序列的状态序列的最大概率。本论文把可观察序列作为演算对象，求出最佳可观察序列作为最优解集，再利用 HMM 将其“翻译”成隐含的状态变化序列，即所需答案。

隐马尔科夫模型预测器的基本算法步骤描述如下：

- 1) 主计算模型循环 m 次后，将向量组每列的取值的变化再生成一个状态变换矩阵。
- 2) 按照状态变换矩阵的变迁概率，依照 Viterbi 算法选择变迁概率最大的一条似然路径。
- 3) 再取向量组的第一行结果，依照上述最大似然路径生成最大可能取值。

隐马尔科夫模型预测器能够动态地反映每步最优解的变迁过程，无论大的跳跃还是小的跳跃都能够及时地发现并加以逼近，是一种相对较准的统计-预测算法。但这个方法的缺点是方法比较复杂，计算量较大。

上面提到了 Viterbi 算法是一种利用动态规划从后向前逐步反推取最优解的算法。在某个最佳路径上的任意节点出发，则到终点的路径也是该点到终点的最佳路径。因此就能从后到前逐步求得最佳路径。

动态规划的优势在于不仅仅考察前后两个观察状态的关系，而是全面考察整个观察序列，得出一个“最大似然”的结果。即使序列中有一两个误码，还是能够根据整个序列给出正确的解码。

Viterbi 算法的基本算法步骤描述如下：

- 1) 根据状态变换矩阵的变迁概率，填写一个包含所有可能状态的二维矩阵。
- 2) 从初态开始逐一取条件转移序列，求出条件概率下的最大可能状态，并打分存入数组。反复执行该操作直到序列处理完毕。

3) 统计该数组里分值最高的状态，即需要的状态转换过程。

3.3.4 EDA 预测器

第四种是基于 EDA 的统计-预测（EDA 预测器）。由于本算法的设计思路本身就有一部分来自 EDA，因此使用 EDA 做生成时序统计-预测就显得顺理成章。EDA 算法通过一个概率模型描述候选解在空间的分布，采用统计学习手段从群体宏观的角度建立一个描述解分布的概率模型，然后对概率模型随机采样产生新的种群。这种思路天然就可以预测出将来全局最优解的出现位置。

EDA 预测器虽然方式和 HMM 预测器差别很大，但基本原理类似于 HMM 预测器。算法步骤描述如下：

1) 主计算模型循环 m 次后，将当前最优解向量组看作一个染色体单元，对该向量组做分布估计。

2) 将经过分布估计后的最优解看作下一轮的可能取值。

事实上 EDA 预测器本质上其实是 HMM 预测器的一种等价形式，都是统计向量组里某组取值概率最高的候选解。这个解通过 HMM 预测器中的最大似然概率，和 EDA 里分布估计最大的取值反应出来。所不同的是，EDA 不需要再建立状态变换矩阵，而是把数据直接拿来做分布估计，这个过程得到的解和 HMM 做的统计最大似然概率是一样的，即稳定少动的染色体保持原有取值，变化较大的染色体取可能性高的取值。EDA 算法易于理解，编程方便，是一种较好的附加预测器。

3.3.5 最优解的高斯散布

通过上述步骤 3（即预测器）生成的最优解，仅仅是一种基于前面若干轮最优解的一种大致预测，无论是统计法还是预测法，生成的可能解集其实都不是很准的。因此还应根据可能解集再生成一组正态分布的候选解集。将这些候选解集一起代入原进化计算模型的适应度函数，由该函数判断本算法生成的候选解集是否值得保留，而本文所提的算法自身不做适应度判断。

至于正态分布中的 μ 与 σ 的取值，则可取决于前面向量组所统计出来最优解的波动程度。

根据最优解生成最优解集的正态分布的基本算法步骤描述如下：

- 1) 根据前面步骤生成的向量表，依据其波动程度求其 μ 与 σ
- 2) 以前面求出的最优解，使用正态分布随机函数，播撒出一系列的解集。参与到主计算模型中让评估函数判断其适应度，适者生存不适者淘汰。

3. 4 在进化计算模型中嵌入附加统计-预测算法

我们回头看 3.1 部分所论述的这些进化计算模型，尤其是 GA、PSO 和 EDA 这三种进化计算模型，均有上节所论述的插入循环位置。例如遗传算法里每一轮遗传算法循环都有一次交叉——变异——选择过程，此时会有一批随机个体加入到种群。粒子群优化算法里每一轮粒子群优化算法循环都有一次选择最优——追随最优过程。分布估计算法每一轮循环都会估计出一次新结果，并能统计出当前最优解。

那么我们可以在这个每轮循环结束的地方，都可以记录下当前全局最优解的取值，经过若干轮循环后，就得到了一组由最优解取值所组成的时间序列。然后再通过上节所给出的几种预测器来分析下一轮循环中最优解的最可能取值，产生一系列候选解加入到种群里来，并利用进化计算模型的评估函数，对参与进来的候选解与已有的保留解集一起进行优胜劣汰。而附加统计-预测算法本身并不做适应度判断。

3. 5 本章小结

在这一章里，我们论述了进化算法的共性，和基于这种共性，如何推导出附加统计-预测算法的思路。接下来，则论述该算法的具体流程，并重点论述了简单平均预测器、线性加权预测器、HMM 预测器、EDA 预测器四种附加预测器的算法步骤。和最优解的正态分布散布算法。以及如何将该算法附加到现有的通用进化算法模型上去。

下面一章将详细论述遗传算法、PSO 算法、EDA 算法结合了附加统计-预测算法之后，解决一些最优化问题时，对计算时间的节省，以及结果的改善。

第四章 附加统计-预测算法与其他算法结合

4.1 测试用例与测试规则

进化算法的主要用途有三种，即求最优解、分类和预测。其中求最优解是进化算法最实用的方向之一。为此这里提供三个典型的测试用例，测试附加统计-预测算法的所改善的性能。

这三个测试用例均为相对复杂的最优化问题。后面将分别使用标准遗传算法、标准 PSO 算法、标准 EDA 算法与采用附加统计-预测的遗传算法、采用附加统计-预测的 PSO 算法、采用附加统计-预测的 EDA 算法做对比。看看它们性能与精度上的改变。

测试用机配置为联想扬天 M4600V，CPU 为双核 E2140，内存 1G。

测试函数的前两个来自参考文献，后一个测试函数为平时常用。

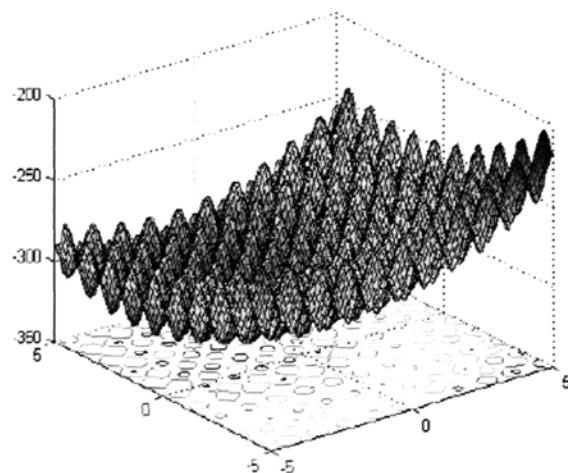
代号为 F9 的测试函数为 Shifted Rastrigin's Function，

公式如下：

$$F_9(\mathbf{x}) = \sum_{i=1}^D (z_i^2 - 10 \cos(2\pi z_i) + 10) + f_bias_9, \quad \mathbf{z} = \mathbf{x} - \mathbf{o}, \quad \mathbf{x} = [x_1, x_2, \dots, x_D]$$

D : dimensions

$\mathbf{o} = [o_1, o_2, \dots, o_D]$: the shifted global optimum



(图5) Shifted Rastrigin's Function

测试函数 F9 的 x 取值为 $[-5, 5]$ ，取 40 维。极值为每个偏移量 f_bias_9 处。极

小值点时 F9 取值为 -390。

F9 问题是一个比较容易求解的函数，无论是 GA 还是 PSO 或 EDA，求出 40 维的可接受最优解所花的计算时间均可忽略不计。

代号为 F12 的测试函数为 Schwefel's Problem，公式如下：

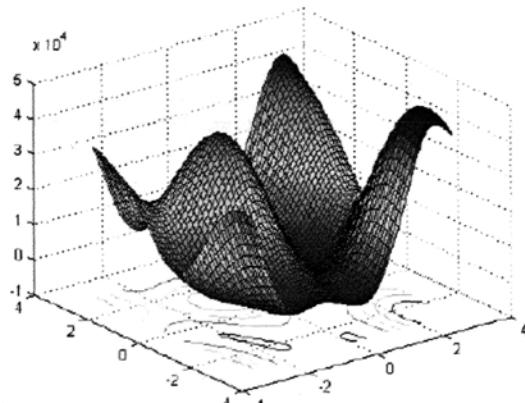
$$F_{12}(\mathbf{x}) = \sum_{i=1}^D (\mathbf{A}_i - \mathbf{B}_i(\mathbf{x}))^2 + f_bias_{12}, \mathbf{x} = [x_1, x_2, \dots, x_D]$$

$$\mathbf{A}_i = \sum_{j=1}^D (a_{ij} \sin \alpha_j + b_{ij} \cos \alpha_j), \mathbf{B}_i(\mathbf{x}) = \sum_{j=1}^D (a_{ij} \sin x_j + b_{ij} \cos x_j), \text{ for } i = 1, \dots, D$$

D: dimensions

A, B are two D*D matrix, a_{ij}, b_{ij} are integer random numbers in the range [-100, 100]

$\alpha = [\alpha_1, \alpha_2, \dots, \alpha_D], \alpha_j$ are random numbers in the range $[-\pi, \pi]$.



(图6) Schwefel's Problem

测试函数 F12 仅为 3 维函数。取值为 $[-\pi, \pi]$ ，极值为每个偏移量 $\alpha_1, \alpha_2, \alpha_3$ 处。极小值点时 F12 取值为 -460。

F12 问题是一个计算比较复杂的函数，3 维循环 10000 代的时候，标准 GA 求出误差小于 1 的最优解所花时间大约 12 秒，PSO 约 7 秒，EDA 也约 7 秒。

代号为 F50 的测试函数为 50 维最优化问题，公式如下：

$$F(\mathbf{x}) = \sum (x_i - b_i) \quad (i=1..50)。 \text{ 其中 } b_i \text{ 为偏移量。} X = [-10, 10]。$$

测试函数 F50 为 50 维函数。取值为 $[-10, 10]$ ，极值为每个偏移量 b_i 处。极小

值点时 F1 取值为 0。

F12 问题是一个计算比较复杂的函数，标准 GA 求出误差小于 0.1 的最优解所花时间大约 31 秒，PSO 约 24 秒，EDA 约 22 秒。

测试规则为：

对遗传算法来说，每个测试用例均用标准遗传算法、附加简单平均统计-预测的遗传算法、附加线性加权统计-预测的遗传算法、附加 HMM 统计-预测的遗传算法、附加 EDA 统计-预测的遗传算法。各自运行 10 遍，记录最优解的取值情况，并加以对比分析。

对 PSO 算法来说，每个测试用例均用标准 PSO 算法、附加简单平均统计-预测的 PSO 算法、附加线性加权统计-预测的 PSO 算法、附加 HMM 统计-预测的 PSO 算法、附加 EDA 统计-预测的 PSO 算法。各自运行 10 遍，记录最优解的取值情况，并加以对比分析。

对 EDA 算法来说，每个测试用例均用标准 EDA 算法、附加简单平均统计-预测的 EDA 算法、附加线性加权统计-预测的 EDA 算法、附加 HMM 统计-预测的 EDA 算法、附加 EDA 统计-预测的 EDA 算法。各自运行 10 遍，记录最优解的取值情况，并加以对比分析。

4.2 附加统计-预测算法与遗传算法结合

首先对比的是测试用例 F9。

在 F9 问题中，X 取了 40 维。进化代数为 30 代。

原代码库里 F91G..CPP 为标准遗传算法求 F9 问题。

F92G.CPP 为附加简单平均统计-预测的遗传算法求 F9 问题

F93G.CPP 为附加线性加权统计-预测的遗传算法求 F9 问题

F94G.CPP 为附加 HMM 统计-预测的遗传算法求 F9 问题

F95G.CPP 为附加 EDA 统计-预测的遗传算法求 F9 问题

经过各自运行 10 遍后，其运算结果如下：

程序编号	程序名称	取值(10 次平均)	较标准算法所优化程度(10 次平均)	误差小于%1 的循环代数(10 次平均)	计算时间(10 次平均)
标准遗传算法	F91G..CPP	390.00	0	15.6	可忽略
简单平均附加统计-预测	F92G.CPP	390.00	0	14.2	可忽略
线性加权附加统计-预测	F93G.CPP	390.0	0	13.3	可忽略
HMM 附加统计-预测	F94G.CPP	390.0	0	13.9	可忽略
EDA 附加统计-预测	F95G.CPP	390.0	0	13.6	可忽略

(表1) 使用ASPA遗传算法与标准遗传算法
解Shifted Rastrigin's Function的效果对照

其次对比的是测试用例 F12。

在 F12 问题中，X 取了 3 维。进化代数为 10000 代。

原代码库里 F121G..CPP 为标准遗传算法求 F9 问题。

F122G.CPP 为附加简单平均统计-预测的遗传算法求 F12 问题

F123G.CPP 为附加线性加权统计-预测的遗传算法求 F12 问题

F124G.CPP 为附加 HMM 统计-预测的遗传算法求 F12 问题

F125G.CPP 为附加 EDA 统计-预测的遗传算法求 F12 问题

经过各自运行 10 遍后，其运算结果如下：

程序编号	程序名称	取值(10 次平均)	较标准算法所优化程度(10 次平均)	误差小于%1 的循环代数(10 次平均)	计算时间(10 次平均)
标准遗传算法	F121G..CPP	-458.2	0	4275	12
简单平均附加统计-预测	F122G.CPP	-459.8	90%	3250	12
线性加权附加统计-预测	F123G.CPP	-459.7	85%	3125	12

HMM 附加统计-预测	F124G.CPP	-459.6	80%	3150	12
EDA 附加统计-预测	F125G.CPP	-459.4	70%	3100	12

(表2) 使用ASPA遗传算法与标准遗传算法
解Schwefel's Problem的效果对照

第三对比的是测试用例 F50。

在 F12 问题中，X 取了 50 维。进化代数为 50000 代。

原代码库里 F501G..CPP 为标准遗传算法求 F9 问题。

F502G.CPP 为附加简单平均统计-预测的遗传算法求 F50 问题

F503G.CPP 为附加线性加权统计-预测的遗传算法求 F50 问题

F504G.CPP 为附加 HMM 统计-预测的遗传算法求 F50 问题

F505G.CPP 为附加 EDA 统计-预测的遗传算法求 F50 问题

经过各自运行 10 遍后，其运算结果如下：

程序编号	程序名称	取值(10 次平均)	较标准算法所优化程度(10 次平均)	误差小于%1 的循环代数(10 次平均)	计算时间(10 次平均)
标准遗传算法	F501G..CPP	1.230	0	45000	30
简单平均附加统计-预测	F502G.CPP	0.296	76%	30000	30
线性加权附加统计-预测	F503G.CPP	0.305	75%	27500	30
HMM 附加统计-预测	F504G.CPP	0.254	80%	27500	30
EDA 附加统计-预测	F505G.CPP	0.278	77%	27500	30

(表3) 使用ASPA遗传算法与标准遗传算法
解 50 维最优化问题的效果对照

4.3 附加统计-预测算法与 PSO 传算法结合

首先对比的是测试用例 F9。

在 F9 问题中，X 取了 40 维。进化代数为 30 代。

原代码库里 F91P..CPP 为标准 PSO 算法求 F9 问题。

F92P.CPP 为附加简单平均统计-预测的 PSO 算法求 F9 问题

F93P.CPP 为附加线性加权统计-预测的 PSO 算法求 F9 问题

F94P.CPP 为附加 HMM 统计-预测的 PSO 算法求 F9 问题

F95PCPP 为附加 EDA 统计-预测的 PSO 算法求 F9 问题

经过各自运行 10 遍后，其运算结果如下：

程序编号	程序名称	取值(10 次平均)	较标准算法所优化程度(10 次平均)	误差小于%1 的循环代数(10 次平均)	计算时间(10 次平均)
标准 PSO 算法	F91P..CPP	381.2	0	0	可忽略
简单平均附加统计-预测	F92P.CPP	374.7	0	0	可忽略
线性加权附加统计-预测	F93P.CPP	375.9	0	0	可忽略
HMM 附加统计-预测	F94P.CPP	384.1	33%	0	可忽略
EDA 附加统计-预测	F95PCPP	382.4	14%	0	可忽略

(表 4) 使用 ASPA-PSO 算法与标准 PSO 算法解 Shifted Rastrigin's Function 的效果对照

其次对比的是测试用例 F12。

在 F12 问题中，X 取了 3 维。进化代数为 10000 代。

原代码库里 F121P.CPP 为标准 PSO 算法求 F12 问题。

F122P.CPP 为附加简单平均统计-预测的 PSO 算法求 F12 问题

F123P.CPP 为附加线性加权统计-预测的 PSO 算法求 F12 问题

F124P.CPP 为附加 HMM 统计-预测的 PSO 算法求 F12 问题

F125P.CPP 为附加 EDA 统计-预测的 PSO 算法求 F12 问题

经过各自运行 10 遍后，其运算结果如下：

程序编号	程序名称	取值(10 次平均)	较标准算法所优化程度(10 次平均)	误差小于%1 的循环代数(10 次平均)	计算时间(10 次平均)
标准 PSO 算法	F121P..CPP	-458.9	0	4500	7
简单平均附加统计-预测	F122P.CPP	-459.8	82%	7000	7
线性加权附加统计-预测	F123PCPP	-459.5	54%	3125	7
HMM 附加统计-预测	F124P.CPP	-459.6	74%	3150	7
EDA 附加统计-预测	F125PCPP	-459.4	46%	3100	7

(表 5) 使用 ASPA-PSO 算法与标准 PSO 算法
解 Schwefel's Problem 的效果对照

第三对比的是测试用例 F50。

在 F12 问题中，X 取了 50 维。进化代数为 50000 代。

原代码库里 F501G..CPP 为标准 PSO 算法求 F9 问题。

F502P.CPP 为附加简单平均统计-预测的遗传算法求 F50 问题

F503P.CPP 为附加线性加权统计-预测的遗传算法求 F50 问题

F504P.CPP 为附加 HMM 统计-预测的遗传算法求 F50 问题

F505P.CPP 为附加 EDA 统计-预测的遗传算法求 F50 问题

经过各自运行 10 遍后，其运算结果如下：

程序编号	程序名称	取值(10 次平均)	较标准算法所优化程度(10 次平均)	误差小于%1 的循环代数(10 次平均)	计算时间(10 次平均)
标准 PSO 算法	F501P..CPP	0.975	0	47500	24
简单平均附加统计-预测	F502P.CPP	0.426	48%	42500	24

线性加权附加统计-预测	F503P.CPP	0.425	48%	45000	24
HMM 附加统计-预测	F504PCPP	0.474	42%	37500	24
EDA 附加统计-预测	F505PCPP	0.808	18%	45000	24

(表 6) 使用 ASPA-PSO 算法与标准 PSO 算法
解 50 维最优化问题的效果对照

4.4 附加统计-预测算法与 EDA 算法结合

首先对比的是测试用例 F9。

在 F9 问题中, X 取了 40 维。进化代数为 30 代。

原代码库里 F91E.CPP 为标准 EDA 算法求 F9 问题。

F92E.CPP 为附加简单平均统计-预测的 EDA 算法求 F9 问题

F93E.CPP 为附加线性加权统计-预测的 EDA 算法求 F9 问题

F94E.CPP 为附加 HMM 统计-预测的 EDA 算法求 F9 问题

F95E.CPP 为附加 EDA 统计-预测的 EDA 算法求 F9 问题

经过各自运行 10 遍后, 其运算结果如下:

程序编号	程序名称	取值(10 次平均)	较标准算法所优化程度(10 次平均)	误差小于%1 的循环代数(10 次平均)	计算时间(10 次平均)
标准 EDA 算法	F91E.CPP	390.0	0	15.4	可忽略
简单平均附加统计-预测	F92ECPP	390.0	0	15.3	可忽略
线性加权附加统计-预测	F93E.CPP	390.0	0	15.3	可忽略
HMM 附加统计-预测	F94E.CPP	390.0	0	15.7	可忽略
EDA 附加统计-预测	F95E.CPP	390.0	0	15.5	可忽略

(表 7) 使用 ASPA-EDA 算法与标准 EDA 算法

 解 Shifted Rastrigin's Function 的效果对照

其次对比的是测试用例 F12。

在 F12 问题中，X 取了 3 维。进化代数为 10000 代。

原代码库里 F121E.CPP 为标准遗传算法求 F9 问题。

F122E.CPP 为附加简单平均统计-预测的遗传算法求 F9 问题

F123E.CPP 为附加线性加权统计-预测的遗传算法求 F9 问题

F124E.CPP 为附加 HMM 统计-预测的遗传算法求 F9 问题

F125E.CPP 为附加 EDA 统计-预测的遗传算法求 F9 问题

经过各自运行 10 遍后，其运算结果如下：

程序编号	程序名称	取值(10 次平均)	较标准算法所优化程度(10 次平均)	误差小于%1 的循环代数(10 次平均)	计算时间(10 次平均)
标准 EDA 算法	F121E..CPP	-433.6	0	4500	7
简单平均附加统计-预测	F122E.CPP	-459.3	95%	3000	7
线性加权附加统计-预测	F123E.CPP	-459.5	95%	3000	7
HMM 附加统计-预测	F124E.CPP	-459.6	95%	3000	7
EDA 附加统计-预测	F125E.CPP	-458.1	90%	3250	7

(表 8) 使用 ASPA-EDA 算法与标准 EDA 算法
解 Schwefel's Problem 的效果对照

第三对比的是测试用例 F50。

在 F12 问题中，X 取了 50 维。进化代数为 50000 代。

原代码库里 F501E..CPP 为标准遗传算法求 F9 问题。

F502E.CPP 为附加简单平均统计-预测的遗传算法求 F9 问题

F503E.CPP 为附加线性加权统计-预测的遗传算法求 F9 问题

F504E.CPP 为附加 HMM 统计-预测的遗传算法求 F9 问题

F505E.CPP 为附加 EDA 统计-预测的遗传算法求 F9 问题

经过各自运行 10 遍后，其运算结果如下：

程序编号	程序名称	取值(10 次平均)	较标准算法所优化程度(10 次平均)	误差小于%1 的循环代数(10 次平均)	计算时间(10 次平均)
标准 EDA 算法	F501E..CPP	0.83	0	42500	24
简单平均附加统计-预测	F502E.CPP	0.74	11%	40000	24
线性加权附加统计-预测	F503E.CPP	0.75	10%	40000	24
HMM 附加统计-预测	F504E.CPP	0.29	66%	27500	24
EDA 附加统计-预测	F505E.CPP	0.61	27%	37500	24

(表 9) 使用 ASPA-EDA 算法与标准 EDA 算法解 50 维最优化问题的效果对照

4.5 结果分析

根据上面的测试结果我们可以进行如下对比。

就优化精度来看，各种附加预测器对原始优化算法的优化精度改进如下：

程序类型	遗传算法解 F9 问题	遗传算法解 F12 问题	遗传算法解 F50 问题	PSO 算法解 F9 问题	PSO 算法解 F12 问题	PSO 算法解 F50 问题	EDA 算法解 F9 问题	EDA 算法解 F12 问题	EDA 算法解 F50 问题	总进度
标准算法	0	0	0	0	0	0	0	0	0	0
简单平均附加统计 - 预测	0	90%	76%	0	82%	48%	0	95%	11%	44%
线性加权附加统计 - 预测	0	85%	75%	0	54%	48%	0	95%	10%	40%
HMM 附加统计 - 预测	0	80%	80%	33%	74%	42%	0	95%	66%	52%

EDA 附加统计 -预测	0	70%	77%	14%	46%	18%	0	90%	27%	38%
-----------------	---	-----	-----	-----	-----	-----	---	-----	-----	-----

(表 10) 使用 ASPA 算法与标准计算模型
在求解精度方面的效果对照

由表 10 我们可以认为单就预测精度而言, HMM 预测器的预测精度是最好的。但各预测器对预测精度的大小差别并不太大。从前面的分析我们知道 HMM 和 EDA 具有相似的运作机理, 但在该表中并未严格地体现出来。

就误差小于%1 的循环代数来看, 各种附加预测器对原始优化算法的快速收敛改进如下 (该数值代表算法快速收敛的效率, 因此是越小越好):

程序 编号	遗传 算法 解 F9 问题	遗传 算法 解 F12 问题	遗传 算法 解 F50 问题	PSO 算法 解 F9 问题	PSO 算法 解 F12 问题	PSO 算法 解 F50 问题	EDA 算法 解 F9 问题	EDA 算法 解 F12 问题	EDA 算法 解 F50 问题	总改 进程 度
标准 算法	15.6/ 30	4275/ 10000	45000/ 5000 0	0	4500/ 10000	47500/ 5000 0	15.4/ 30	4500/ 10000	42500/ 5000 0	56%
简单 平均 附加 统计- 预测	14.2/ 30	3250/ 10000	30000/ 5000 0	0	7000/ 10000	42500/ 5000 0	15.3/ 30	3000/ 10000	40000/ 5000 0	50%
线性 加权 附加 统计- 预测	13.3/ 30	3125/ 10000	27500/ 5000 0	0	3125/ 10000	45000/ 5000 0	15.3/ 30	3000/ 10000	40000/ 5000 0	46%
HMM 附加 统计- 预测	13.9/ 30	3150/ 10000	27500/ 5000 0	0	3150/ 10000	37500/ 5000 0	15.7/ 30	3000/ 10000	27500/ 5000 0	42%
EDA 附加 统计- 预测	13.6/ 30	3100/ 10000	27500/ 5000 0	0	3100/ 10000	45000/ 5000 0	15.5/ 30	3250/ 10000	37500/ 5000 0	46%

(表 11) 使用 ASPA 算法与标准计算模型
在收敛速度方面的效果对照

由表 11 我们可以发现。HMM 预测器的收敛速度的改进最快。比不使用附加统计预测算法的原始模型快了 14%。其次是 EDA 预测器和线性加权预测器。简单平均预测器所带来的收敛速度改进并不是很快，只有 6%。

就对不同的计算模型的优化程度来看，各种附加预测器对三种优化算法的求解精度改进如下：

程序类型	简单平均 附加统计 -预测	线性加权 附加统计- 预测	HMM 附加统 计-预测	EDA 附加统 计-预测	合计
遗传算法	52%	53%	53%	49%	51%
PSO 算法	40%	33%	53%	26%	39%
EDA 算法	35%	35%	53%	39%	40%

(表 12) ASPA 算法对不同进化计算模型
在求解精度方面的效果对照

由表 12 我们可以认为该算法对遗传算法的改进程度较好，对 PSO 和 EDA 的改进程度相对较差。这可能是因为遗传算法的性能本身一般的原因。这种改进程度或许是不同进化计算模型算法效率的一个大致估计。

综合对比表 10 表 11 和表 12 的数据，我们可以得出如下结论：

简单平均预测器在精度方面的改进较好，但在收敛速度方面的改进最差。

HMM 预测器无论是在精度和收敛速度方面的效果都是最好的。但这个方法本身相对比较复杂。

线性加权预测器和 EDA 预测器在精度方面表现效果一般，但收敛速度相对较快。

虽然这个描述能大概说明四种方法的优劣，但四种方法所带来的性能上的改进均一般，没有过于明显的差距。考虑到程序设计上的复杂性，简单平均预测器仍有它应有的可取之处。

4.6 通过测试得到的结论

根据上面的测试结果分析我们可以得到如下结论：

1. 这个算法用来改进遗传算法的收敛速度比较有效，而改进 PSO 或 EDA 等算法的效果是不如对遗传算法的效果好的。考虑到在进化计算模型中，遗传算法属于效率相对低下的入门算法，而 PSO、EDA 等则是相对高效的高性能算法。那么，ASPA 对不同进化计算模型所带来的不一样的改善程度，或许可以作为一种衡量不同架构进化计算模型的通用标尺。一个算法使用 ASPA 后改进的好，说明这个算法目前效果还不是最好，尚有很多潜力可挖，而一个算法使用 ASPA 后改进效果不好，则可能说明该算法已经效果很好，难以再改进了。
2. 即使对效果一般的进化计算模型而言，其消耗的计算时间也可以忽略不计。因此它的设计思想其实类似赌博，是一种以小博大的启发式策略。一旦预测准了，对结果会有较大的改善，而即使预测不准也没关系。原因是其不怎么消耗计算时间，另外预测是一轮轮连续的，只要其中有一轮预测获取成功就足以补偿前面消耗的时间。所以 ASPA 可以作为通用的进化计算改进模型，附加到任何一个现有的进化计算模型上，从而进一步增加该模型的实际效果。它是一种不必消耗太多成本，实际效果却可以接受的普及型计算模型。
3. 通过观察这些测试用例我们会发现附加统计-预测算法的关键在于统计历史数据并进行分析。至于怎么统计怎么分析，最终结果其实是相差不是特别大。这说明只要有一种说的过去的预测方法所存在就可以大大改善进化计算模型的性能，而不在于它用什么精巧的方法来预测可能解集。这也许导致 ASPA 向两个不同的方向发展，一个方向是使用复杂的计算模型，追求更高的预测精度，另一个方向是尽量使用快速的计算模型，在结果可以接受的情况下，追求更快的计算速度，而不是盲目地为了小数点后的几位数而消耗大量 CPU 时间。
4. 作为时序预测的各种主流计算模型而言，其更多追求的是一种预测精度，

通过使用大规模长时间的计算力求使预测结果相对准确一些。而本文所提出的附加统计-预测算法所希望的预测方法，则和这种思路截然相反。它希望有一种快速而准确率不要求太高的时序预测方法。通过多次不太准确的预测，形成一条预测链逐步接近最优解。它的每一次预测，结果都不是很准，但到最后总会逐步靠近所想要的结果。由于它的预测由多个小型计算组成，因此，降低其计算复杂性，要比提高其预测精度更为有效的多。

4.7 本章小结

本章所做的系列测试实验，由 45 个小程序组成，这些小程序 5 个一组共 9 组，分别对应遗传算法、PSO 算法和 EDA 算法这三个算法的各自三个测试用例。通过观察这些测试用例可以得出预测过程重于预测方法的结论，同时，我们还可以分析出一些算法的改进余地。

在下章本文将使用进化计算最优化竞赛中的典型比赛题目，分别比较不使用附加统计预测算法和使用附加统计算法进行效果比较，来说明附加统计预测算法的有效性。

第五章 使用 ASPA 求高维 Ackley 函数最优解

5.1 CEC'05 测试函数集

进化计算的最重要应用就是逼近复杂函数的最优解。那么为了证明附加统计预测算法的性能，自然需要测试本方法对高维复杂函数的求解效果。为了实际测试附加统计预测算法的性能。本章将采用实际的进化计算国际比赛题中的几道典型测试题，来对比使用附加统计预测算法和不使用附加统计预测算法的不同效果。题目来自 IEEE Proceedings of Congress on Evolutionary Computation (CEC2005) 的最优化问题求解竞赛。这里选择的测试函数要比第四章的测试函数复杂而且难度加大很多。

本处实际应用选取了代号 F8 的移动旋转 Ackley 函数。该函数的特点为：

- 1) 大部分局部最优解均一般，仅有一个非常小的解空间拥有一个高度突变的最小值。考虑到函数的高维度和 x 的取值范围，这个解空间几乎不能随机取到。
- 2) 可以设置非常高维的解集和干扰空间。本次测试 x 的维数为 50，同时还配备了一个 50X50 的线性变换矩阵作为干扰空间。

该函数的描述如下：

$$F_8(\mathbf{x}) = -20 \exp\left(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D z_i^2}\right) - \exp\left(\frac{1}{D} \sum_{i=1}^D \cos(2\pi z_i)\right) + 20 + e + f_bias_8, \quad \mathbf{z} = (\mathbf{x} - \mathbf{o})^* \mathbf{M},$$

$\mathbf{x} = [x_1, x_2, \dots, x_D]$, D : dimensions

$\mathbf{o} = [o_1, o_2, \dots, o_D]$: the shifted global optimum;

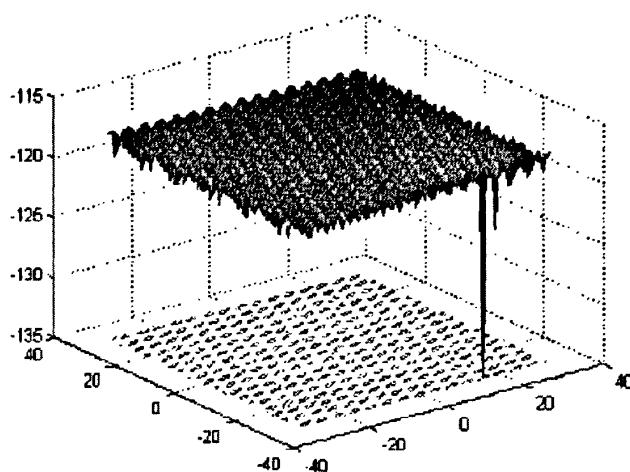
After load the data file, set $o_{2j-1} = -32$ o_{2j} are randomly distributed in the search range, for $j = 1, 2, \dots, \lfloor D/2 \rfloor$

\mathbf{M} : linear transformation matrix, condition number=100

其中， D 为 x 的维数，本次测试取 50.0 为 50 维全局最优解取值数组，并且 $o_{2j-1}=-32$ ， o_{2j} 为随机值，线性变换矩阵每个元素取值均固定为 100。

x 的取值范围为 $[-32, 32]$ 。全局最优解的 x 的取值即 0 的取值。全局最优解时， $F8(x) = -140$ 。

该函数图形如下。测试用机配置为联想扬天 M4600V，CPU 为双核 E2140，内存 1G。



(图6) Shifted Rotated Ackley' s Function

5.2 测试过程与结果分析

测试过程为使用标准遗传算法和第四章选用的附加 HMM 统计-预测的遗传算法做对比。各自运行 10 遍，记录最优解的取值情况，并加以对比分析。进化代数为 100000 代。

算法名称	取值(10 次平均)	误差小于%1 的循环代数(10 次平均)
标准遗传算法	-97	77500
HMM 预测器遗传算法	-113	65000

(表 12) Ackley 函数使用标准遗传算法与 ASPA 遗传算法
在求解精度方面的效果对照

根据测试结果来看，均存在一定程度的找不到最优解现象。而且 100000 代的收敛次数使得运算时间达到数十分钟。但采用了附加预测器后。算法在求解成功率，和找到最优解的收敛速度方面，均有了一定程度的提高。但是受求解函数复杂性的影响，其效果仍然是有限的，并且不阻碍其发展潜力。因此我们可以认为。附加统计预测这

种方法，对简单中等复杂度的优化函数效果相对比较明显，而对高维复杂函数的优化效果仍然相对有限。为此我们寄托新的附加预测器的出现，来提高其优化效果和找到最优解的收敛速度。

5.3 小结

本章所做的实验，来自 CEC2005 测试题。难度比第 4 章的题目要大的多。通过观察这个测试用例子我们发现附加统计预测算法对高维复杂函数的优化效果仍然相对有限。

在下章本文将对附加预测统计算法的改进余地、不足问题和发展前景做出最后的总结。

第六章 总结和展望

6.1 总结

本文的主要工作是围绕附加统计-预测算法在最优化的研究和应用展开的。其主要研究了如何利用附加统计-预测算法来加速进化算法的收敛速度，减轻进化算法的计算复杂性，让进化算法在尽可能少的循环时保持足够的精度。

绝大多数进化计算模型，都对应着自然界里实际存在的智能群体的集体行为活动，附加统计-预测算法也不例外。如果把其他进化算法看作大自然里自由竞争的生物种群，那么附加统计-预测算法就是指导生物快速进化的外力智慧生物。它通过观察种群里最优个体的取值变化，来有意识地直到整个种群向这个合理范围加速演化，这个过程可以看作是人工把野生动物培育成家畜的过程。自然界天然进化的过程过于缓慢，而人工驯养的速度则快的多，但存在着驯养不成的风险。

附加统计-预测算法总的来说其实是一种分析向量（如 GA 的染色体，ANN 的隐含层）变化时可能的取值的思想，类似赌博，是一种以小博大的策略。它对主计算模型的计算复杂性影响很小，一般仅增加了染色体位数的平方阶，且可以在主计算模型进行多次循环后使用一次。我们有了附加统计-预测算法的思想，就可以寻找的各种统计或预测方法，来进一步研究采用何种具体的统计与预测方法效果更好，可以得出一系列后续成果，其算法改进的余地仍然较大。

由于预测本身是一种猜想，因此附加统计-预测算法尚有许多不足。如有可能会出现无意义的解集；局部最优的解集出现太多；仅能有一定概率提高主计算模型的收敛速度，并非每次都能取得满意效果。

最主要的一个问题是附加统计-预测算法仅能优化现有进化计算模型，无法独立使用。原因是相比现有的进化计算模型，本算法缺少了关键步骤——适应度函数。附加统计-预测算法本身没有适应度函数一说，而是由主计算模型的适应度函数去做适应判断，因此附加统计-预测算法的宗旨是优化现有的进化计算模型，这也扩大了它的使用范围，因任何进化计算模型都可以使用本附加统计-预测算法使其收敛速度更快。这也是本算法得名附加统计-预测算法的原因，可以附加在任何一个现有的进化

计算模型后面进行附加的统计和预测。

6.2 展望

由于算法的广泛适用性，我们希望附加统计-预测算法能够做如下工作：

- 1) 由于该算法针对现有各种计算智能算法模型的共性，因此它可以优化目前存在的各种计算智能算法的计算量，具有广泛的使用前景。
- 2) 由于可以用不同的方法分析时间序列。通过不同的统计-预测器和不同计算智能算法模型的组合，可以产生多种多样的研究方向。
- 3) ASPA 算法对现有计算智能模型的优化程度不一，这种优化程度也许是一种衡量新算法的潜在改进能力的评价指标
- 4) 由于 ASPA 算法优化的直观性和其他计算智能算法的广泛结合性，或许可以用 ASPA 的优化程度来评价不同计算智能算法性能高低的比较工具。同时也可作为衡量两个异构计算智能算法相似度的参数指标。

在程序设计时也取得了一些初步的经验，我们发现如果在程序设计中注意下列问题，算法性能会相对提高许多：

- 1) 要优化的主计算模型应为计算智能算法，即需要一组解集合作计算，而不是靠一个解慢慢搜索。因此传统的数学优化算法如牛顿迭代法、欧拉两步法无法使用附加统计-预测算法。
- 2) 要优化的向量组状态要尽可能多，取值可能也要多且为可数状态。
- 3) 主计算模型循环多了，有助于提高一次 HMM 预测的准确性（但也相对提高了整体运算时间）
- 4) 多个 HMM 同时主计算模型投票的效果，似乎不如一个 HMM 对该主计算模型运算多次

6.3 进一步的工作

附加统计-预测算法主要是一种分析向量（如 GA 的染色体，ANN 的隐含层，PSO 的步进长度，EDA 的概率分布）变化时可能的取值的思想，类似赌博，可取得以小博大的收益。因此，采用什么方法进行向量变化的时序分析，哪些方法最好哪些方法不好，好的原因是什么不好的原因是什么，都是下一步研究要考虑的问题。为此还有如

以下几个问题需要后续研究进一步讨论：

- 1) 有无衡量预测准确程度的数学方法？
- 2) 所生成的时间序列矩阵预测仍然精度较低，需要散布出一系列可能的取值（比如利用正态分布方法），如何提高散布的准确性？
- 3) 如何摆脱局部最优的预测解集？
- 4) 预测中有可能会生成毫无意义的解集，如何降低这种解集的出现可能？

参考文献

- [1] 周春光, 梁艳春. 计算智能. 吉林大学出版社, 2001. 11
- [2] 丁永生 编著。计算智能——理论、技术与应用。科学出版社, 2004
- [3] 黄席樾, 张著洪, 何传江, 胡小兵, 马笑萧, 著。现代智能算法理论及应用。科学出版社, 2005
- [4] Jiawei Han, Micheline Kamber. Data mining concepts and techniques [M]. 北京: 机械工业出版社, 2003. 188-190.
- [5] Quinlan J R. C4.5 programs for machine learning [M]. San Mateo, 1993. 170-247.
- [6] Back B., Laitinen T., and Sere K.: Neural networks and genetic algorithms for bankruptcy predictions. Expert Systems with Applications, vol. 11 (1996), pg. 407-413.
- [7] Venkatesan R. and Kumar V. 'A genetic algorithms approach to growth phase forecasting of wireless subscribers.' International Journal of Forecasting, vol. 18 (2002), pg. 625-646.
- [8] 陈国良、王煦栋, 遗传算法原理及其应用, 人民邮电出版社
- [9] 刘雄, 陈严, 叶枝全。遗传算法在风力机风轮叶片优化设计中的应用。太阳能学报, 2006,27(2)
- [10] 汪冰, 席平原。应用遗传算法的塔机起升机构模糊优化设计。机械传动, 2006,(1)
- [11] 工磊, 免疫进化计算理论及应用(博士论文), 西安电子科技大学, 2001.9
- [12] Petr Musilek., Adriel Lau, Marek Reformat: Immune programming. Information Sciences 176 (2006) 972-1002
- [13] 焦李成、杜海峰、刘若辰, 人工免疫系统进展与展望, 西安电子科技大学雷达信号处理国家重点实验室与只能信息处理研究报告, 2002
- [14] 杜海峰、焦李成, 免疫策略, 西安电子科技大学雷达信号处理国家重点实验室与只能信息处理研究报告, 2002
- [15] Kaboudan M. 'Forecasting stock returns using genetic programming in C++.' Proceedings of 11th Annual Florida Artificial Intelligence International Research

Symposium, vol. 1 (1998), pg. 502-511.

- [16] Wagner N. and Michalewicz Z. 'Genetic programming with efficient population control for financial times series prediction.' 2001 Genetic and Evolutionary Computation Conference Late Breaking Papers, vol. 1(2001), pg. 458-462.
- [17] Iba H. and Sasaki T. 'Using genetic programming to predict financial data.' Proceedings of the Congress of Evolutionary Computation, vol. 1(1999), pg. 244-251.
- [18] Iba H. and Nikolaev N. 'Genetic programming polynomial models of financial data series.' Proceedings of the 2000 Congress of Evolutionary Computation, vol. 1 (2000), pg. 1459-1466.
- [19] Iba H., de Garis H., and Sato T. 'Genetic programming using a minimum description length principle.' Advances in Genetic Programming, vol. 1(1994), pg. 265-284.
- [20] Ferreira, C.: Gene Expression Programming: a New Adaptive Algorithm for Solving Problems. Complex Systems (2001)13(2): 87-129.
- [21] Ferreira, C.: Gene Expression Programming: Mathematical Modeling by an Artificial Intelligence. Angra do Heroismo, Portugal(2002)
- [22] Heitor S.Lopes., Wagner R.Weinert.: A Gene Expression Programming System For Time Series Modeling.Proceedings of XXV Iberian Latin American Congress on Computational Methods in Engineering(CILAMCE),Recife(Brazil),10-12,november,2004.
- [23] R Salustowicz,J Schmidhuber. Probabilistic incremental program evolution[J].Evolutionary Computation, 1997 ;5 (2):123~141
- [24] Colorni A,Dorigo M,Maniezzo V,et al.Distributed optimization by ant colonies[C]//Proceedings of the 1st European Conference on Artificial Life,1991:134-142.
- [25] Dorigo M.Optimization,learning and algorithms[D].Department of Electronics,Politecnico diMilano,Italy,1992.
- [26] Dorigo M,Maniezzo V,Colorni A.Ant system:optimization by a colony of cooperating agents[J].IEEE Transaction of Systems,Man, and Cybernetics-Part B,1996,26(1):29-41.
- [27] Kennedy, J., Mendes, R.,: Population structure and particle swarm performance. In:

- Proc. the Congress on Evolutionary Computation. NanJing: IEEE Press (2002) 1671-1676
- [28] Chau, K.W.: Particle swarm optimization training algorithm for ANNs in stage prediction of Shing Mun River. Journal of Hydrology. Vol. 329. Elsevier, Amsterdam (2006) 363-367
- [29] Zhang, Q.L., Li, X., Tran, Q.A.: A modified particle Swarm optimization algorithm. In: Machine Learning and Cybernetics. Institute of Electrical and Electronics Engineers Computer Society, Piscataway (2005) 2993-2995
- [30] 施彦, 黄聪明, 侯朝桢。基于改进的 PSO 算法的神经网络集成。复旦学报(自然科学版), 2004(10)。
- [31] 周驰, 高海兵, 高亮,等. 粒子群优化算法[J]. 计算机应用研究, 2003, (12): 7-12
- [32] Larrañaga P, Lozano J A. Estimation of Distribution Algorithms. A New Tool for Evolutionary Computation. Boston: Kluwer Academic Publishers, 2002
- [33] Endika Bengoetxea, Pedro Larrañaga, Isabelle Bloch, et al. Inexact graph matching by means of estimation of distribution algorithms[J]. Pattern Recognition, 2002, 35 (12): 2867 ~ 2880.
- [34] Yuehui Chen, Lizhi Peng, Ajith Abraham, "Exchange Rate Forecasting using Flexible Neural Trees", Lecture Notes on Computer Science, Vol. 3973, pp. 518-523, 2006.
- [35] Yuehui Chen, Lizhi Peng, Ajith Abraham, Gene Expression Profiling Using Flexible Neural Trees, 7th International Conference on Intelligent Data Engineering and Automated Learning (IDEAL2006), Burgos, Spain, Lecture Notes on Computer Science, Vol. 4224, pp. 1121-1128, 2006.
- [36] Yuehui Chen, Ajith Abraham, Bo Yang, Feature Selection and Classification using Flexible Neural Tree, Neuro computing, Vol.70, Issues 1-3, pp.305-313, 2006.
- [37] Yuehui Chen, Bo Yang, Jiwen Dong and Ajith Abraham, Time-series Forecasting Using Flexible Neural Tree Model, Information Science, Vol.174, Issues 3/4, pp.219-235, 2005
- [38] Yuehui Chen, Bo Yang and Jiwen Dong, Evolving Flexible Neural Networks using Ant Programming and PSO algorithm, International Symposium on Neural Networks

(ISNN'04), LNCS 3173, pp.211-216, 2004.

- [39] 张建伟,张颖江,王宗跃,林志毅,黄樟灿 多表达式编程 武汉理工大学学报（信息与管理工程版）[J] Vol.2, 2007
- [40] A.R. Mehrabian, C. Lucas. A novel numerical optimization algorithm inspired from weedcolonization[J]. ECOLOGICAL INFORMATICS, 2006(1):355-365
- [41]陈玉祥, 章汉亚. 预测与技术与应用. 机械工业出版社, 1985
- [42] 马超群, 高仁祥. 现代预测理论与方法. 湖南大学出版社, 1999
- [43] 李学伟, 关忠良, 陈景艳. 经济数据分析预测学. 中国铁道出版, 1998
- [44] 孙明玺, 现代预测学. 浙江教育出版社, 1998
- [45] 冯文权. 经济预测与决策技术（3 版）. 武汉大学出版社, 1994.4
- [46] 杨东升. 经济分析中的预测方法评述. 预测. 1998, 1: 45~50
- [47] 马志民, 陈锡林. 微型计算机在预测决策中的应用. 四川科学技术出版社, 1985
- [48] 陈玉祥. 回归预测方法. 预测. 1991, 1: 64~67
- [49] 暴奉贤, 陈立宏. 经济预测与决策方法. 暨南大学出版社, 1991
- [50]茆诗松, 王静龙, 史定华等. 统计手册. 北京: 科学出版社. 2003. 184-562
- [51]王小林. 统计学. 北京: 经济科学出版社, 2001. 297-310
- [52]吴喜之. 非参数统计. 北京: 中国统计出版社, 2999. 53-66

致谢

本文是在作者的导师陈月辉教授的悉心指导下完成的，从算法的设计到论文的写作均得到陈教授的巨大指导和支持。陈教授不仅仅为论文的诞生提供了大量的学习资料，更重要的是提供了高瞻远瞩的设计思想，在多次的探讨中，陈教授提供的指导总是切中要害，准确及时。这种指导让我在三年的研究生学习生活中收益非浅。在此，谨向陈教授致以衷心的感谢，您辛苦了。

在三年的研究生学习期间，通过和计算智能实验室的老师和同学的广泛交流。让自己得到了无数无私的帮助。尤其是贾光峰、李伟、宋乐辉、潘萌等同学。在和大家互相学习探讨交流的过程中，自己的学术水平有了很大的提高。在此一齐表示深深的谢意。

本文参考了许多前辈的著作，没有这些学者的作品，本文无法完成，感谢你们为计算机科学的发展所创造的基础。

最后，希望为我评审论文的诸位老师严格审查，多多提出宝贵意见。

攻读硕士期间发表论文目录

- [1] 周泉, 陈月辉.附加统计-预测算法[J].济南大学学报研究生创新论坛, 2007, 2(21): 5-8.