

# 目录

|                          |           |
|--------------------------|-----------|
| <b>第一部分 基础理论</b>         | <b>3</b>  |
| <b>第一章 绪论</b>            | <b>7</b>  |
| 1.1 起源与动机                | 7         |
| 1.2 烟花算法属于群体智能优化算法研究范畴   | 8         |
| 1.3 烟花算法的组成与研究内容         | 9         |
| 1.4 优点与特色                | 11        |
| 1.5 研究历史与现状              | 12        |
| 1.6 未来发展方向               | 15        |
| 1.7 未来五年内需要研究的 25 个问题    | 15        |
| 1.8 小结                   | 16        |
| <b>第二章 烟花算法</b>          | <b>19</b> |
| 2.1 引言                   | 19        |
| 2.2 烟花算法的组成              | 21        |
| 2.3 烟花算法的实现              | 23        |
| 2.4 基本烟花算法特点分析           | 27        |
| 2.5 影响算法性能的因素分析          | 30        |
| 2.6 烟花算法与遗传算法和粒子群优化算法的比较 | 33        |
| 2.7 实验结果及分析              | 35        |
| 2.8 小结                   | 38        |
| <b>第三章 烟花算法的理论分析</b>     | <b>39</b> |
| 3.1 随机模型                 | 39        |
| 3.2 全局收敛性                | 40        |
| 3.3 时间复杂度的基本理论           | 42        |
| 3.4 时间复杂度分析              | 45        |
| 3.5 小结                   | 48        |

|                                |               |
|--------------------------------|---------------|
| <b>第四章 随机数对烟花算法性能的影响</b>       | <b>49</b>     |
| 4.1 引言 . . . . .               | 49            |
| 4.2 随机数生成器 . . . . .           | 50            |
| 4.3 实验设定 . . . . .             | 52            |
| 4.4 实验结果与分析 . . . . .          | 53            |
| 4.5 小结 . . . . .               | 55            |
| <br><b>第二部分 改进算法研究</b>         | <br><b>61</b> |
| <b>第五章 基于适应度函数值估计的烟花算法</b>     | <b>65</b>     |
| 5.1 进化计算算法加速策略 . . . . .       | 65            |
| 5.2 基于适应度函数值估计的烟花算法 . . . . .  | 68            |
| 5.3 实验 . . . . .               | 71            |
| 5.4 性能分析及讨论 . . . . .          | 72            |
| 5.5 小结 . . . . .               | 74            |
| <b>第六章 构造型烟花算法</b>             | <b>77</b>     |
| 6.1 对烟花算法各组成部分的改进 . . . . .    | 77            |
| 6.2 实验 . . . . .               | 80            |
| 6.3 小结 . . . . .               | 80            |
| <b>第七章 增强烟花算法</b>              | <b>87</b>     |
| 7.1 对基本烟花算法的分析 . . . . .       | 87            |
| 7.2 增强烟花算法 . . . . .           | 88            |
| 7.3 实验 . . . . .               | 95            |
| 7.4 小结 . . . . .               | 100           |
| <b>第八章 动态搜索烟花算法</b>            | <b>101</b>    |
| 8.1 引言 . . . . .               | 101           |
| 8.2 增强烟花算法 . . . . .           | 102           |
| 8.3 增强烟花算法最小爆炸半径检查策略 . . . . . | 104           |
| 8.4 动态搜索烟花算法 . . . . .         | 105           |
| 8.5 实验 . . . . .               | 110           |
| 8.6 小结 . . . . .               | 114           |

|                            |            |
|----------------------------|------------|
| <b>第九章 自适应烟花算法</b>         | <b>117</b> |
| 9.1 引言                     | 117        |
| 9.2 对 FWA 和 EFWA 中爆炸半径的分析  | 117        |
| 9.3 自适应爆炸半径                | 118        |
| 9.4 自适应烟花算法                | 125        |
| 9.5 实验                     | 126        |
| 9.6 讨论                     | 127        |
| 9.7 小结                     | 128        |
| <b>第十章 混合算法</b>            | <b>129</b> |
| 10.1 引言                    | 129        |
| 10.2 带差分变异的烟花算法            | 129        |
| 10.3 差分演化与烟花算法的混合算法        | 138        |
| 10.4 文化烟花算法                | 139        |
| 10.5 基于生物地理学优化的烟花算法        | 140        |
| 10.6 小结                    | 141        |
| <b>第三部分 高级主题</b>           | <b>143</b> |
| <b>第十一章 多目标烟花算法</b>        | <b>147</b> |
| 11.1 基本概念                  | 147        |
| 11.2 施肥问题                  | 148        |
| 11.3 多目标烟花算法               | 150        |
| 11.4 实验和讨论                 | 154        |
| 11.5 小结                    | 162        |
| <b>第十二章 求解 TSP 的离散烟花算法</b> | <b>167</b> |
| 12.1 旅行商问题                 | 167        |
| 12.2 离散烟花算法                | 169        |
| 12.3 实验结果及其分析              | 182        |
| 12.4 与传统算法比较               | 185        |
| 12.5 小结                    | 186        |

|                            |            |
|----------------------------|------------|
| <b>第十三章 基于 GPU 的并行烟花算法</b> | <b>187</b> |
| 13.1 引言                    | 187        |
| 13.2 GPU 通用计算              | 188        |
| 13.3 基于 GPU 的烟花算法及实现       | 205        |
| 13.4 GPU-FWA 的算法实现         | 210        |
| 13.5 实验分析                  | 213        |
| 13.6 小结                    | 215        |
| <br>                       |            |
| <b>第四部分 应用</b>             | <b>217</b> |
| <br>                       |            |
| <b>第十四章 非负矩阵分解</b>         | <b>221</b> |
| 14.1 引言                    | 221        |
| 14.2 相关工作                  | 222        |
| 14.3 低秩估计                  | 223        |
| 14.4 基于群体智能算法的非负矩阵计算算法     | 224        |
| 14.5 实验设置                  | 229        |
| 14.6 实验结果和讨论               | 230        |
| 14.7 小结                    | 237        |
| <br>                       |            |
| <b>第十五章 聚类和模式识别应用</b>      | <b>239</b> |
| 15.1 烟花算法在文档聚类中的应用         | 239        |
| 15.2 垃圾邮件检测算法参数优化          | 243        |
| 15.3 图像识别                  | 250        |
| 15.4 小结                    | 256        |
| <br>                       |            |
| <b>第十六章 群体机器人多目标搜索问题</b>   | <b>257</b> |
| 16.1 引言                    | 257        |
| 16.2 多目标搜索问题定义             | 260        |
| 16.3 分组爆炸策略                | 262        |
| 16.4 算法分析                  | 268        |
| 16.5 实验结果                  | 269        |
| 16.6 小结                    | 272        |

|                       |                |
|-----------------------|----------------|
| <b>第十七章 地学反演问题</b>    | <b>273</b>     |
| 17.1 引言 . . . . .     | 273            |
| 17.2 反演问题 . . . . .   | 273            |
| 17.3 实验 . . . . .     | 277            |
| 17.4 小结 . . . . .     | 279            |
| <br><b>第五部分 参考文献</b>  | <br><b>283</b> |
| 参考文献                  | 285            |
| <br><b>第六部分 附录</b>    | <br><b>311</b> |
| <b>附录 A 测试函数集</b>     | <b>313</b>     |
| A.1 测试函数集 1 . . . . . | 313            |
| A.2 测试函数集 2 . . . . . | 320            |
| A.3 测试函数集 3 . . . . . | 321            |
| A.4 测试函数集 4 . . . . . | 322            |
| <br><b>附录 B 资源</b>    | <br><b>323</b> |
| B.1 互联网资源 . . . . .   | 323            |
| B.2 国际组织 . . . . .    | 323            |
| B.3 期刊 . . . . .      | 324            |
| B.4 会议 . . . . .      | 324            |
| <br><b>附录 C 术语列表</b>  | <br><b>327</b> |
| <br><b>附录 D 符号列表</b>  | <br><b>330</b> |
| <br><b>附录 E 图表列表</b>  | <br><b>333</b> |
| <br><b>附录 F 索引</b>    | <br><b>339</b> |

# 第一部分

## 基础理论



本部分主要介绍烟花算法的提出和基本理论成果，共分成四章。第一章绪论介绍了烟花算法研究的起源和动机、研究范畴、内容和特色、未来的研究方向以及有待研究的问题。第二章给出了烟花算法的基础知识，包括烟花算法的框架、特点、实现和与其他算法对比的实验结果。第三章给出了烟花算法的随机模型，证明了烟花算法的全局收敛性，讨论了烟花算法的时间复杂度的理论，并分析了算法的时间复杂度。第四章给出了多种随机数产生方法，并详细研究了这些随机数对烟花算法性能的影响，给出了相应的实验结果。



# 第一章 绪论

## 1.1 起源与动机

记得小的时候在四川老家，每逢一年中最重要的节日春节到来时，我都会邀上几位要好的发小或同学们一起，到空旷的操场或人烟稀少的街道上，尽情燃放一种在空中爆炸的爆竹花炮。有时，几个小伙伴还会一起进行比赛，看看谁的爆竹扔得高、放得响，在空中燃放出最美丽的图像。在这些活动中，伴随着我们儿时的快乐和美好时光，在我儿时的脑海里留下了很深的印迹。

2006 年春节，是我来北京大学任教将近一年时间了。这段时间里，我对进化计算投入了较多的精力，进行了深入的研究。因此，在这段时间不管在干什么和遇到什么新鲜的事物都会看看它们是否与进化计算能联系上。正好是这 2006 年春节期间，北京将禁放烟花爆竹的规定改有限放。在经历了多年的禁放烟花爆竹后的首次开禁，首都市民都迫切地期待着除夕之夜的到来，盼望着过一个更加热闹和欢庆的春节。这年的除夕之夜，北京的天空尽情的开放，市民们都争相燃放，好像想把过去的压抑全部释放。人们燃放出了大量绚丽多彩的礼花，将漆黑的夜空照得亮如白昼，五彩斑斓的烟花，燃放出各种美丽的图像，激发了我内心深处的儿时印象，心情无比的畅快和愉悦。

此时，我的脑海里突然将烟花的爆炸图像与进化计算中随机搜索建立起了联系，产生了一种可以用像烟花爆炸图像一样的方式来对问题解空间进行有效搜索的新方式。通过模拟烟花爆炸的方式来进行多点同时爆炸式搜索，这也许是一种高效的搜索方式，是有别于现有其他方法的新型搜索方法，从而有了研究这种爆炸搜索方式的想法，当时为其取名烟花算法（Fireworks Algorithm，缩写为 FWA）。

虽然烟花算法这个名称比较直观和简洁，但是由于它没有直接与优化等求解问题建立直接的联系，此后有些研究人员有时也用其他别称来称呼我们的烟花算法，例如有：烟花优化算法、烟花爆炸算法、烟花爆炸优化算法、烟花爆炸搜索算法、爆炸搜索方法，等等。尽管有这些不同的别称，我们这里统一采用原始的名称烟花算法，缩写为 FWA，以免混淆。

我们对烟花算法的研究动机是希望寻求一种求解复杂问题的全局最优解的高效方法。尤其是对具有多模特性的复杂优化问题能找到高效求解的新途径。

正好，在 2006 年的夏季学期，我招收了来自吉林大学的朱元春同学为我的直博生，并安排他来我的实验室从事毕业设计工作。我就将我之前的所有想法和研究烟花算法任务交给他来具体实现，并将他的毕业设计题目拟定为“烟花算法的研究与实现”，从而带领他一起共同开始了对烟花算法的全面研究。

经过半年时间的深入研究，我们共同设计了烟花算法的各个主要要素、组成和基本框架，并以“爆炸算子”为基础搭建了基本的烟花算法。到 2007 年 5 月份就完成了对烟花算法的基本研究工作。

但是，在接下来的近两年时间里，由于我忙于主持一项国家 863 计划项目，就暂缓了对烟花算法的相关研究工作。直到 2009 年夏才又抽出一些精力来，重新展开对烟花算法的研究，并于 2010 年在首届国际群体智能大会上发表了题为“**Fireworks algorithm for optimization**”的开创性学术论文 [1]。从此以后，烟花算法的研究才开始受到业界的关注，对其的研究才开始在群体智能领域逐渐展开。

## 1.2 烟花算法属于群体智能优化算法研究范畴

群体智能（Swarm Intelligence-SI）是指由许多简单个体所组成的群体所呈现出的涌现（emergence）行为所表现出的集体智能，是单个个体所不具备的强大能力。例如：生物群体系统有蚂蚁、鸟群、蜜蜂、鱼群、蜘蛛、萤火虫、细菌、等等。鸟群掠过天空、蚁群寻觅食物、鱼群在水中游荡、烟花在空中爆炸、蜘蛛的爬行等，这种群体的运动称为群体行为。尽管这些群体中的各个个体都是非常简单，但大量个体组成的群体所表现出的集体行为却是非常复杂的，呈现出智能的特色。

群体智能是进化计算的一个活跃的分支，隶属于计算智能的范畴。而近几十年来，计算智能的丰富研究成果，包括：人工神经网络[2-5]、模糊逻辑与系统[6]、进化计算[7, 8]、混沌计算[9]、模拟退火[10]、禁忌搜索[11]、以及各种混合策略等等，都是通过模拟或揭示某些自然现象或过程而得以实现。

优化问题是一个古老且永恒的研究问题，是解决众多问题的基础。大量学者和实际工作者致力于对其不懈的研究。不同于经典优化算法采用确定性规则的方式，群体智能优化算法利用一种概率转移方法，通过利用各种随机因素结合元启发性规则，采用群体中的多个个体同时对解空间进行并行搜索的方式，通过群体中个体的相互协作与竞争来实现对优化问题的最优解的有效搜索。它具有随机性、自适应性、鲁棒性、并行性等显著特点。在求解复杂优化问题时表现出了非常明显的优势，从

而引起人们对群体智能优化算法的高度重视，成为目前的研究热点 [12–15]。

群体智能优化算法可以分为两大类：基于生物群体的群体智能优化方法和基于非生物群体的群体智能优化方法。前者包括：蚁群优化[16]、粒子群优化[17]、鱼群搜索[18]、虚拟蜜蜂算法[19]、萤火虫算法 -I[20]、萤火虫算法 -II[21]、布谷鸟算法[22]、蝙蝠算法[23]、人工蜂群算法[24][25][26]、人工鱼群算法[27]、磷虾群算法[28]、细菌觅食优化算法[29] 等，而后者包括：烟花算法（FWA）[1]、水滴算法[30]、头脑风暴优化算法（BSO）[31]，磁铁优化算法[32]，等等。目前，群体智能算法研究主要包括三个方面：算法、求解问题类型、应用。其发展趋势包括：混合算法、求解大规模问题（面临维数灾难、大数据难题）、新型改进算法、理论分析等。

通常，群体智能优化算法都有一定的共同性，即是由组成的群体多个个体（社会昆虫、或粒子）的相互协同，具有交互传递信息（直接或间接地）和交互式地适应环境的能力，使得群体中个体对环境的适应性逐代变得越来越好，逐渐求得问题的全局最优解的足够好的近似解。

群体智能优化算法所具有的这种协同交互能力能够打破没有免费午餐 (NFL) 定理的魔咒 [33]，预示了存在并能够发展出具有性能更加优良的高效算法 [34]。这预示着对群体智能优化算法的深入研究将可能给我们带来难以预想的益处，从而，激励更多的研究人员从事群体智能的研究和在更多的实际领域里积极采用群体智能的最新研究成果，使得群体智能更好地为人类社会服务。

### 1.3 烟花算法的组成与研究内容

烟花算法的基本组成框架如图1.1所示。它主要由爆炸算子 (Explosive operator)、变异操作 (Mutation operation)、映射规则 (Mapping rule) 和选择策略 (Selection Strategy) 四大部分组成。其中，爆炸算子包括爆炸强度、爆炸幅度、位移变异等操作；变异操作主要包括高斯变异操作；映射规则包括有模运算规则，镜面反射规则和随机映射规则等操作；选择策略包括有基于距离的选择和随机选择等操作 [35]。关于各项算子的具体要求和实现将在下一章中详细介绍。

烟花算法的工作过程与群体智能优化算法相似，首先，随机选择  $N$  个烟花初始化群体，然后，让群体中的每个烟花经历爆炸操作和变异操作，并应用映射规则保证变异后的个体仍处于可行域内，最后，在保留最优个体（即精英策略）的前提下，应用选择策略从生成的所有个体（烟花和火花）中选择出余下的  $N-1$  个个体共同组成下一代的群体。这样周而复始，逐一迭代下去，通过这种交互传递信息（直接或间接地），使得群体对环境的适应性逐代变得越来越好，从而求得问题的全局最优解

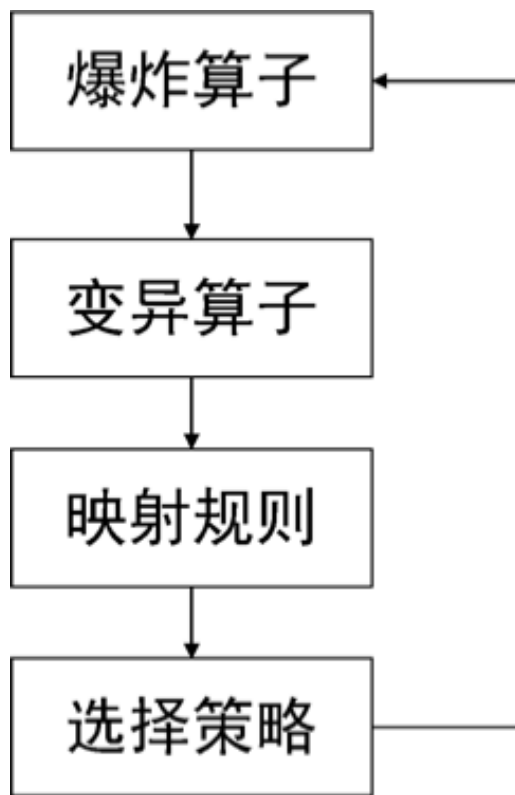


图 1.1 烟花算法（FWA）的基本组成框架图。

的足够好的近似解。

目前，对烟花算法的研究工作主要包括四个方面，即理论分析、算法研究、求解问题类型和应用研究。

### 1. 理论分析

研究烟花算法的求解机理，收敛性质，演化轨迹特点，各参数对算法的影响等理论问题，为开发新的算法和改进现有算法提供理论指导。

### 2. 算法研究方面

在烟花算法的算法研究方面，通过对烟花算法基本组成的各个成分进行深入分析和调整，不断改进烟花算法的性能（收敛性、解的精度、时间效率），从而提出了各种不同的改进烟花算法。同时，通过与其他方法的结合，取长补短，发展出高效的混合型方法。

### 3. 求解问题类型

由于烟花算法的通用性，它可以用于求解下列不同问题类型：

- 单目标优化问题 (single-objective optimization problem)
- 约束单目标优化问题 (constrained single-objective optimization problem)
- 多目标优化问题 (Multiple-objective optimization problem -MOO)
- 约束多目标优化问题 (constrained multiple-objective optimization problem)
- 许多目标优化问题 (Many-objective optimization problem - ManyOO)
- 组合优化问题(Combinatorial optimization problem-CO)
- 动态优化问题(Dynamic optimization problem - DOP)
- 其他优化问题 (Other optimization problem)

#### 4. 应用

由于烟花算法是一类群体智能优化方法，具有求解复杂问题的全局最优解的能力，同时对求解问题的目标要求很低，不要求问题的目标函数的梯度信息，因此具有广泛的适应性。故，烟花算法可以用到许多实际应用领域，求解人们可能遇到的各类问题。

## 1.4 优点与特色

烟花算法不仅继承了现有群体智能优化算法的许多优点，还具有明显的自身特色，归纳起来，烟花算法具有下面一些优点：

1. 爆发性 (explosive): 每次迭代开始，需要让烟花进行爆炸，在辐射范围内产生许多与该烟花本身不同的火花。之后，依据特定选择策略选择  $N$  个火花或烟花保存为下一代烟花群体，恢复烟花数目，并为下次爆炸过程做好准备。
2. 瞬时性 (instantaneity): 烟花算法中爆炸产生的火花，如果没有在选择策略中被选中成为下一代的烟花，这些火花或烟花本身都将在本次迭代中消亡，也就是说，一次特定的爆炸只存在于一次特定的迭代之中，具有瞬时存在性。
3. 简单性 (simplicity): 每个个体只能感知局部信息，个体的能力或遵循的规则非常简单，因此，算法的组成和实现都非常简单。
4. 局部覆盖性 (locality): 对于某一个烟花而言，它的爆炸范围是整个自变量取值范围的一个小部分，其爆炸出的火花是这个爆炸范围内的一些局部点，只是对

爆炸范围的区间点有一定程度的随机覆盖，但是不会涉及到爆炸范围之外的点，因此说这种爆炸具有一定的局部性。

5. 涌现性 (emergent properties): 使用简单交互规则，通过协同与竞争方式个体间相互作用，其群体总体表现出来的单个个体不具有的复杂行为，呈现出极高的智能特点。涌现 (emergence) 现象是以相互作用为中心的，它比单个行为的简单累加要复杂得多。
6. 分布并行性 (distributed parallelism): 群体中个体相对简单，没有一个直接的中心控制约束，各个个体进行局部相互作用，本质上是一个分布式方法，呈现出高度并行的特色，特别适合并行化。
7. 多样性 (diversity): 首先，烟花个体的多样性：我们通过一定的选择机制，使得选择保留下来的烟花具有不同的位置，以保证算法的多样性特征。其次，爆炸强度和爆炸幅度的多样性：在爆炸强度的作用下，根据各个烟花的优良度不同（适应度函数值大小不同），各个烟花产生不同个数的火花。在爆炸幅度的作用下，根据各个烟花不同的优良度，各个烟花产生的火花拥有不同的变异幅度。最后，爆炸算子中的多种变异共存：正如火花有多个隔层那样，我们设计出的爆炸幅度中存在有多种变异，目前有两种变异：一种是位移变异；另一种是高斯变异。其中，第一种位移变异是跟自变量的取值区间，以及粒子本身的优良度（决定了变异幅度的大小）相关的一种变异；第二种高斯变异只与烟花本身的位置有关。这两种变异是本质上不同的变异，保证了变异的多样性。
8. 可扩充性 (scalability): 由于个体相对独立，个体间的协作通常通过间接的方式实现信息交流，增加或减少部分个体，对系统的影响都不剧烈，从而保证系统具有很强的可扩展性。
9. 适应性 (adaptability): 由于只使用各个个体的适应性来对系统求解能力进行评估，以此，对所求解问题的要求非常低。甚至，不要求所求解问题具有显示的表达。

## 1.5 研究历史与现状

自从烟花算法的开创性论文“Fireworks algorithm for optimization” [1] 由作者于 2010 年在“The First International Conference on Swarm Intelligence (ICSI 2010) - 首届国际群体智能大会”上发表以后，业界对烟花算法的研究就逐步深入和铺开了。通过对

原始烟花算法的深入、细致的分析,针对原始烟花算法(FWA)所存在的不足,相继提出了大量的改进方法,并据此发展了各种改进算法,以及几种混合型方法,极大地提高了原始烟花算法的性能。进一步,研究了烟花算法在求解不同类型优化问题的能力,还有大量的研究人员进行了烟花算法的应用研究,给出了一些典型的成功应用案例。

具体现状简述如下:

### 1. 理论研究方面

在收敛性的理论分析工作方面, Liu et al. [36] 从理论上详细分析了烟花算法的收敛性, Liu 等人指出烟花算法是一个吸收马尔科夫过程,进而给出了收敛性定理并予以了严格的数学证明。此外,在本书中,我们首次研究了随机数对于烟花算法性能的影响,实验表明烟花算法对于随机数产生器的要求不高,不同的随机数生成方法对于算法性能的影响不明显。

### 2. 算法研究方面

烟花算法(FWA)研究的开创性论文是由 Tan and Zhu [1] 发表于首届国际群体智能大会(ICSIS 2010)。该文首次提出了受烟花爆炸启发的群体协同优化算法,即烟花算法(FWA)。该文详细介绍了FWA的组成、各个算子的要求和原则、及各个算子的具体实现方法。该文指出烟花算法包括爆炸算子、高斯变异算子、映射规则和选择策略,并给出了满足原则的具体实现方案。为了验证烟花算法的有效性,实验对比了两个典型群体智能算法——标准粒子群优化算法和克隆粒子群优化算法,在由11个测试函数组成的集合上,烟花算法具有明显的性能优势。Pei et al. [37] 研究了适应度函数估计对于烟花算法加速性能的影响。文中讨论了不同的适应度函数值的估计方法对于性能的影响,实验表明二次多项式模型和随机选择样本策略的性能最优,并且相对于烟花算法其性能优势具有显著性。Ding et al. [38] 提出了一种并行烟花算法GPU-FWA,它是基于图形处理单元(GPU)的FWA的高效并行实现方案,可以全面加速FWA的运行速度,在当前流行的GPU硬件和CUDA平台下,现实了近200倍的加速性能。相对于FWA, GPU-FWA做了一些算子上的改动,主要的目的是减少烟花之间的交互同时使得性能损失在一个可以接受的范围内。在[38]中,烟花之间每隔一定代数才会计算计算爆炸半径和爆炸幅度。这极大地降低了烟花之间的交互,提高加速比。在[39]中, Zheng S. 等对于烟花算法的算子进行了细致的分析,针对FWA存在的缺陷进行了改进,并最终提出了增强烟花算法。改进的工作包括基本烟花算法中的爆炸算子、高斯变异算子、选择算子以及映射规则等4个方面。在[40]和[41]中, Zheng S. 等和 Li J. 等细致地研究了烟花算法中

爆炸幅度的自适应策略，并分别提出了动态搜索烟花算法和自适应烟花算法。此外，有部分学者研究了烟花算法和其他算法的混合算法。在 [42] 和 [43] 中，Zheng Y. 等和 Yu C. 等分别尝试将烟花算法和差分演化算法进行混合。在 [42] 中，混合算法 FWA-DE 相对于烟花算法和差分演化算法在测试函数集合上面具有更好的性能。在 [44] 中，Gao 等将烟花算法和文化算法进行混合，并应用到滤波器设计的优化问题中。Gao 等对比了量子粒子群优化算法 [45] 和自适应量子粒子群优化算法 [46]，实验结果表明文化烟花算法具有更好的性能。此外，Zhang 等提出了生物地理学优化 - 烟花混合算法 (BBO-FWA)，BBO-FWA 算法的性能要远远好于所基于的 BBO 和 FWA 两种算法 [47]。在 [48] 中，Nantiwat Pholdee 和 Sujin Bureerat 系统比较研究了 24 种元启发 (meta-heuristic) 算法的优化性能，他们主要是针对具有动态约束的桁架 (truss) 质量最小化问题，在不同问题规模的多种情况下仔细地比较了这些算法的性能，给出了客观的排名，其中烟花算法处于中上游并被证明是一种有效的算法 [48]。

### 3. 求解不同类型优化问题方面

连续优化问题，又分单目标问题和多目标问题。前述的大量算法都是针对单目标优化问题进行的，已经产生了大量的高效算法。对于多目标优化问题，目前研究的还不多。Zheng Y. 等是最先将烟花算法应用于多目标问题提出了多目标烟花算法 (MOFOA)，并应用到施肥问题求解中 [49]。相对于其他著名的多目标群体智能算法和多目标进化计算方法，MOFOA 表现出了非常优异的性能，详情参见第 11 章。目前，我们实验室也在进行着这方面的研究工作，研究成果将另文报告。此外，在本书中，我们首次尝试使用离散烟花算法用于求解旅行售货商问题，提出了求解旅行商问题的离散烟花算法 (TSP-FWA)，并将其应用到标准 TSP 标准数据集上，取得了很不错的实验效果，详情参见第 12 章。

### 4. 应用方面

目前，烟花算法及其改进算法被应用到了许多实际优化问题求解中。应用领域主要包括有：方程组求解 [50]，非负矩阵分解 (NMF) 计算 [51]，垃圾邮件检测算法中参数优化 [52]，方向性特征距离度量 [53]，数字滤波器 FIR 和 IIR 的设计 [44]、油料作物的施肥问题 [49]、群体机器人多目标搜索 [54]、电力系统重构问题 [55]，等等。同时，在本书中，我们也尝试将烟花算法扩展应用到文本聚类问题优化中，我们实验室的研究生也正在研究如何将烟花算法有效的应用于大数据的文本聚类问题。

此外，关于烟花算法研究历史的简要综述还可参考文献 [56] 和 [57]。

## 1.6 未来发展方向

到目前为止，烟花算法的研究还是很初步的，还处在逐渐发展的过程中。新的思想和算法还在不断涌现，在许多方面，目前的研究还很肤浅，在有些方面，还存在空白，急需广大有兴趣的研究人员对其进行广泛探讨和深入研究。

烟花算法的未来发展归纳起来，可以有下列几个方面：

1. 算法的理论基础和分析（稳定性、收敛性、收敛特性、参数灵敏度分析）
2. 各种改进方法的深入研究（各种因素对烟花算法性能的影响，如何有效控制和调整。重点是：子烟花间的协同机制建立和研究，发展出协作型烟花算法）
3. 混合方法的研究
4. 大数据问题的求解（如何处理巨大数据？如何处理大量目标的协同优化？）
5. 动态优化问题的求解（优化目标值随时间变化的情况，如：群体机器人的动态目标追踪搜索问题）
6. 发展更为广泛的应用。

## 1.7 未来五年内需要研究的 25 个问题

根据我们对烟花算法的长期研究，经过归纳整理，我们可以提出未来五年内有关烟花算法需要研究的 25 个问题为：

1. 研究烟花算法求解轨迹特点。
2. 研究烟花算法的收敛性、稳定性、全局性能。
3. 研究建立烟花算法求解的概率模型，分析其求解性能。
4. 研究烟花算法的求解效率。
5. 研究烟花算法各种参数的作用和如何合理设定。
6. 研究烟花算法爆炸算子的作用。
7. 研究更少设定参数的简化烟花算法。
8. 研究不同概率分布类型的随机数对烟花算法的影响

9. 研究莱维 (Lèvy) 烟花算法。
10. 研究烟花算法的协作和竞争机制, 发展协作型 (Cooperative) 烟花算法。
11. 研究现有烟花算法的改进方法。
12. 研究混合型烟花算法, 即, 其他计算方法与烟花算法结合的高效方法。
13. 研究多目标 (Multi-objective) 烟花算法, 以及求解大量目标 (Many-objective) 的优化烟花算法。
14. 研究烟花算法求解动态优化问题。
15. 研究求解约束优化问题的烟花算法。
16. 研究求解组和优化问题的离散烟花算法。
17. 研究处理大数据的烟花算法。
18. 研究高维空间搜索的烟花算法。
19. 研究烟花算法的并行化实现方法。
20. 研究基于 GPU 的并行烟花算法。
21. 研究烟花算法在智能电网中的应用问题。
22. 研究烟花算法在数据挖掘与知识发现中的应用问题。
23. 研究烟花算法在群体机器人搜索控制中的应用问题。
24. 研究烟花算法在语音、图像处理与分析中的应用问题。
25. 研究烟花算法在互联网搜索技术中的应用问题。

## 1.8 小结

本章对烟花算法进行了简要概述。具体地, 首先, 介绍了烟花算法的研究动机和起源, 其次梳理了烟花算法与群体智能优化算法的关系, 并给出了相关术语解释, 然后, 介绍了烟花算法的组成、研究内容、特点, 接着, 详细介绍了烟花算法的研究历史和研究现状, 综述了到目前为止的烟花算法研究文献, 之后指明了烟花算法

的未来发展方向，最后，给出了未来五年内烟花算法研究中需要研究的 25 个问题供相关研究人员参考。

接下来的各个章节将逐一对烟花算法的研究成果进行全面详细地介绍。



## 第二章 烟花算法

烟花算法 (Fireworks Algorithm, 缩写为 FWA) 是作者受到烟花在夜空中爆炸的启发而提出的一种群体智能算法 [1]。本章介绍烟花算法的基本概念、组成和操作等, 目的是对烟花算法进行系统地阐述。主要内容包括烟花算法的组成、算法实现、算法特征分析、各种因子对算法性能的影响、实验结果及其分析、以及与其他算法的比较等。

### 2.1 引言

在中国的传统节日尤其是春节, 燃放烟花 (Fireworks) 或爆竹 (Crackers) 是中国人最重要的节日活动之一, 具有上千年的历史传统。通常, 在这一天, 成千上万的烟花或爆竹在夜空中爆炸并产生出五彩缤纷的美丽图案。不同价格和规格的烟花在黑夜中爆炸产生不同的效果。一般价格高昂的烟花产生的火花数量比较多, 爆炸产生的火花分布的范围也较集中; 价格低廉的烟花产生的火花数量比较少, 爆炸产生的火花分布的范围也较分散。

烟花算法通过模拟燃放的烟花在空中爆炸的这种行为建立相应的数学模型, 通过引入随机因素和选择策略而形成一种并行爆炸式搜索方式, 进而发展成为能够求解复杂问题最优解的全局概率搜索方法。

不失一般性, 通常我们可以将求解的优化问题转化为求解如下最小化优化问题:

$$\min f(X) \text{ s.t. } g_i(X) \leq 0. (i = 1, \dots, m), \quad (2.1)$$

其中,  $f(X)$  为目标函数,  $g_i(X)$  代表约束函数,  $X$  为  $n$  维优化变量。

**烟花算法的基本原则:** 若烟花对应的适应度函数值越小, 则该烟花爆炸所产生的火花数量越多且爆炸幅度越小, 相反, 若烟花对应的适应度函数值越大, 则该烟花爆炸产生的火花数量越少且爆炸幅度越大。

一般地, 烟花算法由爆炸算子、变异算子、映射规则和选择策略四部分组成。其中, 爆炸算子包括爆炸强度、爆炸幅度、位移变异等操作; 变异操作主要包括高斯

变异操作；映射规则包括有模运算规则，镜面反射规则和随机映射规则等操作；选择策略包括有基于距离的选择和随机选择等操作。

烟花算法中的核心是爆炸算子，其作用是在烟花的周围产生一批新的火花。产生火花的个数以及爆炸的幅度都由爆炸算子来确定。因此，我们希望能根据适应度函数值来自适应地控制爆炸算子产生火花的数量和爆炸幅度大小。另外，通过变异算子产生的火花服从高斯分布。两种算子的作用下，如果产生的火花不在可行域范围内，需要运用映射规则将新产生的火花映射至可行域范围内，再利用选择策略选择新的火花作为下一代烟花。

烟花算法的框架图如图2.1所示。

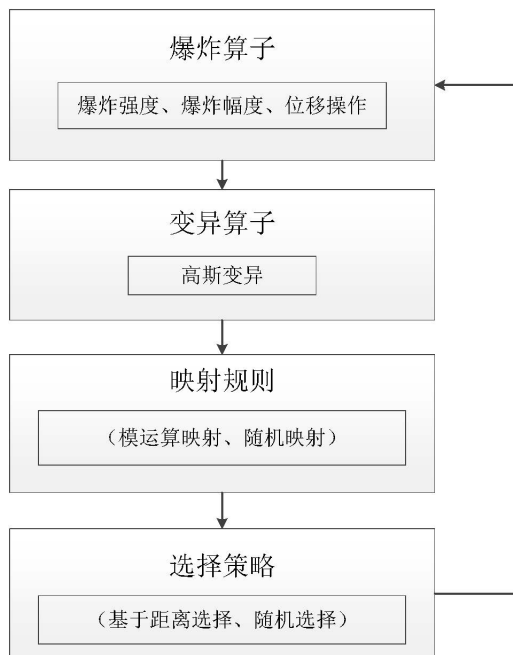


图 2.1 烟花算法的框架图。

下面详细介绍烟花算法。

## 2.2 烟花算法的组成

### 2.2.1 爆炸算子

烟花算法的初始化是随机生成  $N$  个烟花的过程。接着，需要对生成的这  $N$  个烟花应用爆炸算子，以便产生新的火花。爆炸算子是烟花算法的关键核心并起关键性的作用，包含爆炸强度、爆炸幅度和位移操作。

#### 1) 爆炸强度

爆炸强度是烟花算法中爆炸算子的核心，它模拟的是现实生活中烟花爆炸的方式。任何一个烟花爆炸时，这个烟花周围都会产生一批火花。烟花算法首先需要确定每个烟花爆炸产生火花的个数，以及在什么幅度内产生这些火花。

通过观察一些典型优化函数的曲线图，可以直观地看出，最优点附近的优值点也相应较多较密。因此，通过爆炸强度让适应度函数值好的烟花，产生的火花个数较多。这样可以避免寻优时火花总是在最优值附近摆动，而无法精准地找到最优值。对于适应度函数值较差的烟花，由于产生适应度函数值好的火花的几率较小，为避免做过多的不必要的计算，通过爆炸强度使其产生火花数较少。这种适应度函数值差的烟花的作用是对其余空间做适度的探索，避免早熟。根据各个烟花适应度值的大小，确定每一个烟花产生火花的数量，让适应度值好的烟花产生更多的火花，适应度值差的烟花产生更少的火花，如图2.2所示。

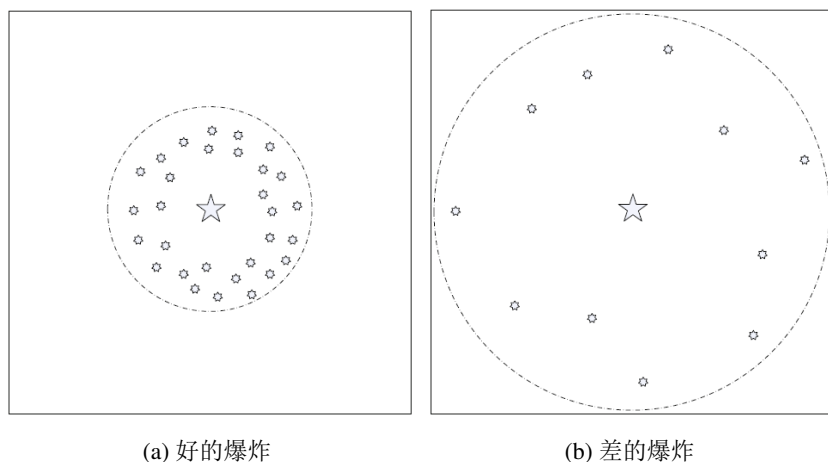


图 2.2 烟花产生爆炸火花示意图。

从图2.2中可以看出，适应度值高的烟花产生了更多的火花，而适应度值差的烟

花产生更少的火花。在确定了爆炸的火花数目之后，接着需要计算烟花爆炸的幅度。

## 2) 爆炸幅度

通过观察一些函数的曲线图，可以直观地看出，通常函数的最优值、局部极值附近的点的函数值通常也较优。因此在烟花算法中，通过控制爆炸幅度，让适应度函数值好的烟花爆炸幅度减小，这样才能更有效的收敛到各个极值，直至最终找到最优值。相反，适应度函数值较差的点，往往离最优值都较远，只有让这些适应度函数值差的烟花产生大幅度的变异，才能使其有效地到达最优值附近。这就是控制烟花爆炸幅度的基本思想。

## 3) 位移操作

在计算出爆炸幅度之后，需要确定烟花在爆炸幅度范围内的位移。这里用到的是随机位移的方法，对烟花进行位移变异。这样，每个烟花都有自己特定的火花数目和爆炸幅度。在某个爆炸幅度内，能随机产生一个位移，生成新的火花，保证了种群多样性。通过爆炸算子，每个烟花都能生成一批新的火花，为寻找全局最优解提供了保障。

### 2.2.2 变异算子

#### 1) 高斯变异

为进一步提高种群的多样性，在烟花算法中引入高斯变异。高斯变异火花产生的过程如下：首先在烟花种群中随机选择一个烟花，对于选择得到的烟花随机选择一定数量的维度进行高斯变异。

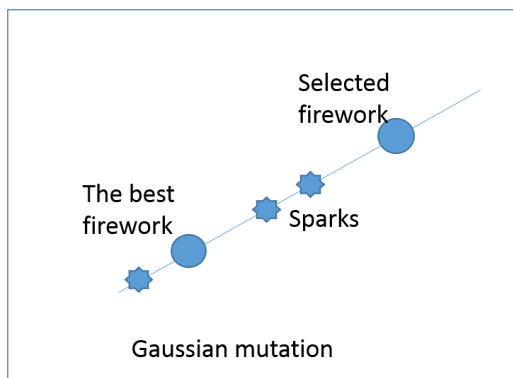


图 2.3 高斯变异的示意图。

高斯变异在选中的烟花和最好的烟花之间进行变异，产生新的火花。高斯变异可能产生超出可行“解空间”范围的火花。当火花在某一维度上超出边界，将通过映射规则映射到一个新的位置。

### 2.2.3 映射规则

如果某一个烟花靠近可行域的边界，而爆炸幅度范围又覆盖到边界以外的区域，那么将可能在可行域范围外产生火花。这种火花是无用的，因此需要通过一种规则将其拉回到可行域范围内。这里采用映射规则来应对这种情况。映射规则确保所有个体留在可行的空间。如果有在边界附近产生一些越界的火花，它们将被映射到可行域的范围。

### 2.2.4 选择策略

运用爆炸算子和变异算子并保证产生的火花在可行域范围之后，需要从产生的火花中选择出一部分作为下一代的烟花。烟花算法用到的是基于距离的选择策略。为了选择进入下一代的个体，选择策略采用的方式是每次都留下最优个体，再选择其他的  $(N - 1)$  个个体。为保证种群的多样性，采用  $(N - 1)$  个个体中和其他个体距离更远的个体有更多的机会被选中的选择策略。

## 2.3 烟花算法的实现

烟花算法开始迭代，依次利用爆炸算子、变异算子、映射规则 and 选择策略，直到达到终止条件，即满足问题的精度要求或者达到最大函数评估次数。

烟花算法的实现包括如下的几个步骤：

- 1) 在特定的解空间中随机产生一些烟花，每一个烟花代表解空间的一个解。
- 2) 根据适应度函数计算每一个烟花的适应度值，并根据适应度值产生火花。火花的个数是基于免疫学中的免疫浓度的思想来计算的，即适应度值越好的烟花产生火花的数目越多。
- 3) 根据现实中的烟花属性并结合搜索问题的实际情况，在烟花的辐射空间内产生火花。（某个烟花的爆炸幅度的大小由该烟花在函数上的适应度值决定，适应度值越大，爆炸幅度越小，反之亦然）。每一个火花代表解空间中的一个解。为了保证种群的多样性，需要对烟花进行适当变异，如高斯变异。

4) 计算种群的最优解, 判定是否满足要求, 如果满足则停止搜索, 没有满足则继续迭代。迭代的初始值为此次循环得到的最好的解和选择的其他的解。

### 2.3.1 爆炸算子

#### 1) 爆炸强度

在烟花算法中, 产生火花个数的公式如下:

$$S_i = m * \frac{Y_{\max} - f(x_i) + \varepsilon}{\sum_{i=1}^N (Y_{\max} - f(x_i)) + \varepsilon}, \quad (2.2)$$

其中,  $S_i$  表示第  $i$  个烟花产生的火花个数, 参数  $i$  的取值范围从 1 到  $N$ 。  $m$  是一个常数, 用来限制产生的火花总数。  $Y_{\max}$  是当前种群中适应度值最差的个体的适应度值。  $f(x_i)$  表示个体  $x_i$  的适应度值。 最后一个参数  $\varepsilon$  取一个极小的常数, 以避免出现分母为零的情况。

为了限制烟花爆炸产生火花的数量太多或过少, 我们为每个烟花设定了如下的产生火花数量的限制公式:

$$\hat{s}_i = \begin{cases} \text{round}(a \cdot m), & \text{if } s_i < am \\ \text{round}(b \cdot m), & \text{if } s_i > bm, a < b < 1 \\ \text{round}(a \cdot m), & \text{otherwise} \end{cases}, \quad (2.3)$$

其中,  $\hat{s}_i$  是第  $i$  烟花可以产生的火花数量,  $\text{round}()$  是取整函数,  $a$  和  $b$  是给定常数,。

#### 2) 爆炸幅度

烟花爆炸幅度范围的计算公式如下:

$$A_i = \hat{A} * \frac{f(x_i) - Y_{\min} + \varepsilon}{\sum_{i=1}^N (f(x_i) - Y_{\min}) + \varepsilon}, \quad (2.4)$$

其中,  $A_i$  表示第  $i$  个烟花的爆炸幅度范围, 即爆炸的火花将在这个范围内随机产生位移, 但不能超出这个范围。  $\hat{A}$  是一个常数, 表示最大的爆炸幅度。 参数  $Y_{\min}$  是当前种群中适应度值最好的个体的适应度值。  $f(x_i)$  和参数  $\varepsilon$  的意义与公式 (2.2) 相同。

### 3) 位移操作

位移操作是对烟花的每一维进行位移，其公式如下：

$$\Delta x_i^k = x_i^k + rand(0, A_i), \quad (2.5)$$

其中， $rand(0, A_i)$  表示在幅度  $A_i$  内生成的均匀随机数。

算法1给出了烟花算法产生火花的伪代码。

---

#### 算法 1 产生火花

---

```

1: 初始化烟花，并计算出每个烟花的适应度值  $f(x_i)$ 
2: 计算每个烟花生成的火花个数  $S_i$ 
3: 计算每个烟花生成火花的爆炸幅度  $A_i$ 
4:  $z = rand(1, dimension)$  // 随机选择  $z$  个维度
5: for  $k = 1 \rightarrow dimension$  do
6:   if  $k \in z$  then
7:      $x_i^k = x_i^k + rand(0, A_i)$ 
8:   end if
9: end for

```

---

## 2.3.2 变异算子

### 1) 高斯变异

用  $x_i^k$  表示第  $i$  个个体在第  $k$  维上的位置，此时高斯变异的计算方式如下：

$$x_i^k = x_i^k * g, \quad (2.6)$$

其中， $g$  是服从如下均值为 1，方差为 1 的高斯分布的随机数。

$$g = \mathcal{N}(1, 1). \quad (2.7)$$

下面算法2给出了烟花算法中高斯变异的伪代码。

### 2.3.3 映射规则

采用模运算的映射规则，其公式如下：

---

**算法 2** 高斯变异
 

---

```

1: 初始化烟花，并计算出每个烟花的适应度值  $f(x_i)$ 
2: 计算高斯变异的系数  $g = \mathcal{N}(1, 1)$ 
3:  $z = \text{rand}(1, \text{dimension})$  // 随机选择  $z$  个维度
4: for  $k = 1 \rightarrow \text{dimension}$  do
5:   if  $k \in z$  then
6:      $x_i^k = x_i^k * g$ 
7:   end if
8: end for
  
```

---

$$x_i^k = x_{min}^k + |x_i^k| \% (x_{max}^k - x_{min}^k), \quad (2.8)$$

其中， $x_i^k$  表示超出边界的第  $i$  个个体在第  $k$  维上的位置， $x_{max}^k$  和  $x_{min}^k$  分别表示第  $k$  维上的边界上下界。百分号代表模运算。

### 2.3.4 选择策略

在烟花算法中，采用欧式距离用来度量任意两个个体之间的距离。 $d(x_i, x_j)$  表示任意两个个体  $x_i$  和  $x_j$  之间的欧式距离。

$$R(x_i) = \sum_{j=1}^K d(x_i - x_j) = \sum_{j=1}^K \|x_i - x_j\|. \quad (2.9)$$

用  $R(x_i)$  表示个体  $x_i$  与其他个体的距离之和， $j \in K$  是指第  $j$  个位置属于集合  $K$ 。其中，集合  $K$  是爆炸算子和高斯变异产生的火花的位置集合。个体选择采用轮盘赌的方式，每个个体被选择的概率用  $p(x_i)$  表示。

$$p(x_i) = \frac{R(x_i)}{\sum_{j \in K} R(x_j)}. \quad (2.10)$$

由 (2.10) 可看出，离其他个体距离更远的个体具有更多的机会成为下一代个体。这种选择方式保证了烟花算法的种群多样性。

烟花算法的流程图如图2.4所示。

从流程图中可以看出，烟花算法的流程和其他群体智能算法相似，需要通过循环迭代产生下一代个体。

算法3是烟花算法的伪代码。

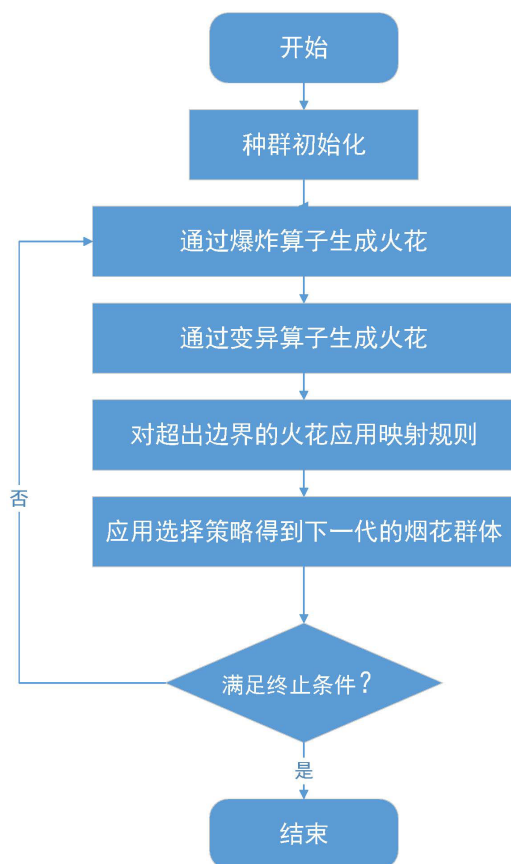


图 2.4 烟花算法的流程图。

## 2.4 基本烟花算法特点分析

基本烟花算法具有如下特点：随机性、局部性、爆发性、隐并行性、多样性和瞬时性。下面具体说明烟花算法的这些特点。

### 2.4.1 爆发性

在烟花算法的一次迭代开始后，烟花在辐射范围内爆炸，会产生其他的火花。等本次迭代结束后，烟花算法选择  $N$  个火花作为下一代的烟花，恢复了烟花的数目，并为下次迭代的爆发做好准备。每一次迭代，烟花都会爆发，说明烟花算法具有爆发性。

---

**算法 3** 烟花算法的伪代码
 

---

```

1: 随机选择  $n$  个烟花的位置
2: while 当前函数评估次数 < 最大函数评估次数 do
3:   对于  $n$  个烟花
4:     for 所有烟花  $x_i$  do
5:       计算每个烟花产生的火花个数  $S_i$ 
6:       计算每个烟花产生火花的幅度  $A_i$ 
7:     end for
8:     随机产生火花
      //  $\hat{m}$  是烟花高斯变异产生的火花数
9:     for  $k = 1 \rightarrow \hat{m}$  do
10:      随机选择一个烟花  $x_i$  并产生一个火花
11:    end for
12:    依据映射规则对火花进行映射
13:    依据选择策略选择最好的烟花以及其他的烟花
14: end while
  
```

---

### 2.4.2 瞬时性

当一次迭代计算开始后，各个烟花依据适应度值的不同，产生不同的火花个数和爆炸幅度。接着，烟花算法将在爆炸算子和变异算子的作用下产生火花。最后，首先选出最优个体，再依据距离选择其他的  $(N - 1)$  个个体。这些选择出来的  $N$  个个体将作为下一代爆炸的烟花，其余的火花不再保留。不保留的火花或烟花将消亡，说明烟花算法具有瞬时性。

### 2.4.3 简单性

和群体智能算法一样，每个烟花只需要感知自身周围的信息，遵循简单的规则，完成自身的使命。总体看来，烟花算法本身并不复杂，由简单个体组成，说明烟花算法具有简单性。

### 2.4.4 局部覆盖性

在烟花算法中，所有的烟花都会在相应的爆炸幅度内产生火花。除非超出可行域，产生的火花都局限在一定的范围内。烟花算法的局部性特点体现了烟花算法强

大的局部搜索能力，可以用在算法运算的后期更加精细的搜索最优解。因此，烟花算法具有局部性。

### 2.4.5 涌现性

烟花之间通过竞争与协作，群体之间表现出简单个体不具有的高度智能性。烟花之间相互作用，比单个个体的行为要复杂得多，因此烟花算法具有涌现性。

### 2.4.6 分布并行性

在烟花算法的每次迭代过程中，各个烟花在不同坐标范围内依次爆炸，即对不同的坐标区间进行一次搜索，在最后会将所有的火花和烟花综合起来，进行下一代烟花的选择。在一次迭代中，算法实质上是并行搜索，表现出烟花算法的分布并行性。

### 2.4.7 多样性

种群多样性是影响群体优化算法性能的关键。群体多样性的保持，可以保证算法跳出局部极值点，从而可以收敛到全局最优点，这正是群体优化算法与一般优化算法的显著区别。群体多样性越大，算法中的个体分布越广，找到最优值的可能越大，同时还不会明显影响算法的收敛能力。因此，种群多样性是烟花算法的一个重要组成部分。烟花算法的多样性主要体现在下面三个方面。

#### 1) 火花个数和爆炸幅度的多样性

在爆炸算子的作用下，依据各个烟花的适应度值，其产生不同个数的火花和不同的爆炸幅度。适应度值高的烟花产生更多的火花，爆炸幅度相对较小，而适应度值低的烟花产生更少的火花，爆炸幅度相对较大。因此，保证了火花个数和爆炸幅度的多样性。

#### 2) 位移操作和高斯变异的多样性

烟花算法有两种算子，第一种是爆炸算子，第二种是变异算子。在爆炸算子的位移操作中，对计算出来的幅度范围，随机产生一个位移，将在选中的烟花加上这个随机位移。在变异算子的作用下，选中的烟花需要乘以一个满足高斯分布的随机数。爆炸算子与烟花的适应度值有关，变异算子与烟花本身的坐标值有关。两种算子是本质上不同的，都保证了爆炸的多样性。

### 3) 烟花的多样性

通过一定的选择机制，保留下来的烟花坐标值各不相同，从而保证了烟花算法的多样性特征。另外，在选择策略中，距离其他火花距离更大的火花更容易被选中，也体现出烟花算法中烟花的多样性特征。

#### 2.4.8 可扩充性

烟花算法中烟花和火花数量不确定，可以依据问题的复杂度来确定。烟花和火花的数目可多可少，增加和减少个体都能有效地求解问题，因此烟花算法具有可扩充性。

#### 2.4.9 适应性

烟花算法求解问题时，不要求问题具有显示表达，只要计算适应度值就能求解问题。同时，烟花算法对问题的要求低，也能求解显示表达的问题。因此烟花算法具有适应性。

## 2.5 影响算法性能的因素分析

### 2.5.1 爆炸算子

在爆炸算子的作用下，烟花在其周围区域进行搜索。烟花的适应度值越好，其爆炸幅度越小，产生的火花数越多；烟花的适应度值越差，其爆炸幅度越大，产生的火花数越少。在这种情况下，具有较好的适应度值的烟花在更小的区域内更仔细地搜索，而较差的火花则搜索更广泛的区域进行搜索。

爆炸算子中有两个参数需要设定，第一个参数是用来限制产生火花的总数  $m$ ，第二个参数是烟花的个数  $N$ 。

这里选用 Generalized Rosenbrock 来说明爆炸算子对烟花算法性能的影响。

图2.5给出了保持其他参数不变的情况下，产生火花的总数  $m$  对算法在 Generalized Rosenbrock 函数上性能的影响。

从图2.5中看出，初步实验说明，在 Generalized Rosenbrock 函数上其他参数保持不变的条件下，烟花算法的火花总数在 10 和 50 之间可以得到相对好的结果。

在 Generalized Rosenbrock 函数上进行实验。图2.6说明烟花的个数对算法性能的影响。

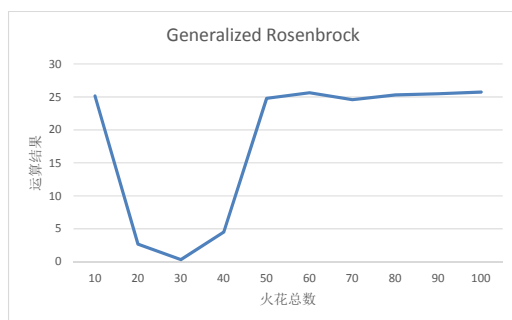


图 2.5 在 Generalized Rosenbrock 函数上，不同烟花总数对算法性能的影响。

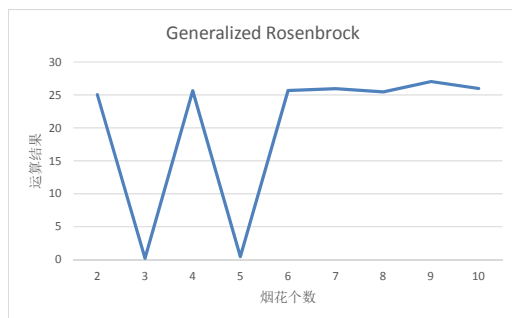


图 2.6 在 Generalized Rosenbrock 函数上，不同烟花个数对算法性能的影响。

从图2.6中看出，在其他参数保持不变的条件下，烟花个数设置为 3 或着 5，烟花算法可以在 Generalized Rosenbrock 函数上得到相对好的结果。

显而易见，针对不同的优化问题，选取不同的烟花算法爆炸算子包含的两个参数，对算法性能有一定的影响。在保持其他参数不变的情况下，火花总数在 20 到 40 之间，烟花算法在 Generalized Rosenbrock 函数上的性能较好。

## 2.5.2 高斯变异

高斯变异在烟花算法中可以增加算法的多样性。烟花高斯变异所产生的火花，不局限于在烟花爆炸的周围地区。因此，烟花算法的多样性提高了。同时，因为高斯变异 ( $x_i^k = x_i^k * g$ ) 在当前位置和原点之间产生火花，所以在某些最优值在原点的函数上表现出很好的性能。例如，Sphere 函数的最优值在原点，因此烟花算法的高斯变异可以很容易地找到 Sphere 函数的最优值的位置。

表2.1是烟花算法在两个函数上的运算结果。函数的维数是 30，烟花算法运行

300,000 次。

表 2.1 烟花算法高斯变异和无变异的结果

| 有无高斯变异 | Sphere   | Generalized Rosenbrock |
|--------|----------|------------------------|
| 有      | 0        | 25.209447              |
| 无      | 1.095037 | 706.936069             |

从表2.1可以看出，烟花算法的高斯变异在两个函数上都提高了运算结果。主要原因在于，高斯变异提高了烟花算法的多样性，增强了烟花算法的全局搜索能力，避免烟花算法陷入局部最优解。

### 2.5.3 映射规则

映射规则保证所有产生的火花都在可行域范围内。如果一个烟花靠近边界，且其爆炸幅度很大，那么其产生的火花可能越界。因此，为防止不必要的计算，需要把超出范围的火花拉回可行域。但是，烟花算法的映射规则也有一定的缺点。例如，映射规则极易把火花拉到原点附近，从而产生靠近原点的火花。

烟花算法中映射规则采用一种模运算方式，这种运算保证超出边界的火花被拉回到可行域范围内。图2.7给出了映射规则对超出边界个体的处理方式。

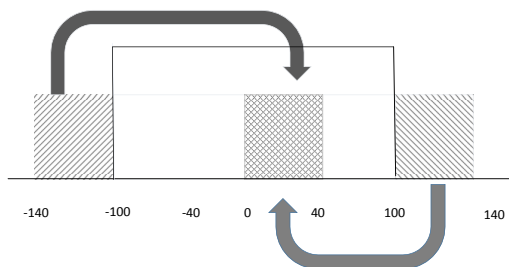


图 2.7 映射规则对火花位置的操作。

如果问题的可行域范围在 -100 到 100 之间，爆炸幅度最大为 40，那么超出边界的点会落在 -140 到 -100 和 100 到 140 区间内。依据映射规则，这两个区间内的点都落在 0 到 40 范围。

### 2.5.4 选择策略

烟花算法选择策略的目的是选择下一代的个体。一般而言，最优的个体总是保留给下一代，而其余的  $(N - 1)$  个个体是按照距离的远近随机选择的。通常，距离其他火花更远的火花更容易被选中。这样，烟花算法的多样性也得到了保障。

## 2.6 烟花算法与遗传算法和粒子群优化算法的比较

### 2.6.1 烟花算法与遗传算法的思想对比

在烟花算法中，对下一代的选择是直接地引入了免疫浓度思想，没有专门设计为该算法的独立算子。这种运算与遗传算法的选择算子名称上有相似性，但却有着本质上的不同：基于免疫浓度的思想，在选择时，与火花（抗体）相似的火花（抗体）越多，火花（抗体）被选中的概率就越小。反之，与火花（抗体）相似的火花越少，火花（抗体）被选中的概率就越大。这使得低适应度值的火花（抗体）也可获得继续演化的机会。因此基于火花（抗体）浓度的概率选择公式在理论上保证了火花（抗体）的多样性。而遗传算法的选择算子按各染色体适应度大小比例来决定其被选择数目的多少，依据的是轮盘赌的方法，并不能保证选择的子个体具有多样性。

遗传算法最初于上世纪 60 年代，由美国密歇根大学的 Holland 教授提出。当时，Holland 认识到了生物的遗传和自然进化现象与人工自适应系统的相似关系，运用生物遗传和进化的思想来研究自然和人工自适应系统的生成以及它们与环境的关系，提出在研究和设计人工自适应系统时，可以借鉴生物遗传的机制，以群体的方法进行自适应搜索，并且充分认识到了交叉、变异等运算策略在自适应系统中的重要性。

两种算法都具有大致相似的过程：

- 1) 初始种群随机初始化；
- 2) 对步骤 1 中产生的每一个个体计算其适应度值；
- 3) 根据适应度值对初始的群体进行必要的操作，遗传算法中进行遗传运算（选择、交叉、变异），烟花算法中执行两个主要的算子（爆炸算子和变异算子）；
- 4) 依据个体的适应度值选择下次迭代的种群；
- 5) 如果终止条件满足，则停止，否则转步骤 2。

从以上步骤，我们可以看到烟花算法和遗传算法有很多共同之处。两者都随机初始化种群，而且都使用适应值来评价个体的优良度，而且都是根据适应值进行一定的随机搜索。两个算法都不是能保证一定找到最优解。

但是，烟花算法中没有交叉操作，而且烟花算法中的变异只是借鉴了遗传算法中的一些思想，有着本质上不同的实现。

与遗传算法比较，烟花算法的信息共享机制是很不同的。在遗传算法中，染色体互相共享信息，所以整个种群的移动是比较均匀的向最优区域移动。在烟花算法中，采用的是一种分布式信息共享机制，根据分布在不同区域烟花的适应值决定爆炸产生的火花个数和爆炸幅度。而且，基于免疫浓度思想的选择使得烟花总能分布在不同的区域，而不会产生聚集。与遗传算法比较，烟花算法有更多的机制来避免早熟。

## 2.6.2 烟花算法与粒子群优化算法的思想对比

这里首先介绍两种粒子群算法。它们是克隆粒子群算法 (Clonal particle swarm optimization, CPSO)[58] 和标准粒子群算法 (Standard particle swarm optimization, SPSO)[17]。

在生物免疫系统中，当抗原侵入生物机体时，其免疫系统在机体内选择出能识别和消灭相应抗原的抗体，这一过程主要借助克隆（无性繁殖）使机体内的抗体激活、分化和增殖，以增加其数量，进一步进行免疫应答并最终清除抗原 (刘若辰 et al. [59])。基于对克隆在免疫响应中的重要性的认识，Tan and Xiao [58] 提出适用于 PSO 的克隆算子，以此对标准 PSO 算法做出改进。

三种算法都具有大致相似的过程：

- 1) 种群随机初始化，烟花算法是对烟花随机初始化，两种 PSO 算法是对所有微粒的初始化；
- 2) 对步骤 1 中产生的每一个个体计算其适应度值；
- 3) 根据步骤 1 中的适应度值对初始的群体进行必要的操作，PSO 算法中更新 pbest、gbest，以及各粒子的位置与速度，烟花算法中执行两个主要的算子（爆炸算子和变异算子）；
- 4) 依据个体的适应度值选择下次迭代的种群；
- 5) 如果终止条件满足，则停止，否则转步骤 2。

从以上步骤，我们可以看到烟花算法和两种 PSO 算法有许多共同之处。他们都采用随机初始化种群，都使用适应值来评价系统，而且都是根据适应值进行一定的随机搜索。两个算法都不是能保证一定找到最优解。

但是，SPSO 算法中没有变异操作，CPSO 算法中加入了高斯变异，而烟花算法则包含位移操作和高斯变异。进一步，烟花算法中的高斯变异对某一次变异中选出的不同维，在每个维度上的位移是相同的，保证了一些维度之间的可能联系。而 CPSO 中各维变异是不相同的。另外，烟花算法中的高斯变异每代都要进行，而 CPSO 中的高斯变异每隔一定的迭代次数才运行一次。

与两种 PSO 算法比较,烟花算法的信息共享机制是很不同的。在 PSO 中,只有 gbest 给出信息给其他的粒子,这是单项的信息流动,整个搜索更新过程是跟随当前最优解的过程。在烟花算法中,采用的是一种分布式信息共享机制,根据分布在不同区域烟花的适应值决定爆炸强度大小和辐射幅度。同时,也需要在整个迭代进程中维护一个最优烟花,采用的是精英策略。

此外,烟花算法中采取了与 CPSO 相同的免疫浓度思想来保持多样性。这个思想在 SPSO 中是不存在的。

## 2.7 实验结果及分析

### 2.7.1 测试函数

为了证明烟花算法在函数优化问题上的可应用性和性能,烟花算法与 SPSO[17] 和 CPSO[58] 进行了实验对比。实验中使用了六个测试函数,见附录 A。函数性质如表 2.2 所示。

表 2.2 实验中函数设置

| 函数名称              | 维度 | 最优点                | 最优值 | 算法初始化区间  | 算法搜索区间        |
|-------------------|----|--------------------|-----|----------|---------------|
| Sphere            | 30 | $[0, 0, \dots, 0]$ | 0   | (30, 50) | $(-100, 100)$ |
| Rosenbrock        | 30 | $[1, 1, \dots, 1]$ | 0   | (30, 50) | $(-100, 100)$ |
| Griewank          | 30 | $[0, 0, \dots, 0]$ | 0   | (30, 50) | $(-100, 100)$ |
| Rastrigin         | 30 | $[0, 0, \dots, 0]$ | 0   | (30, 50) | $(-100, 100)$ |
| Rotated Griewank  | 30 | $[0, 0, \dots, 0]$ | 0   | (15, 30) | $(-100, 100)$ |
| Rotated Rastrigin | 30 | $[0, 0, \dots, 0]$ | 0   | (15, 30) | $(-100, 100)$ |

### 2.7.2 参数设定

经过大量实验,下面这种参数组合的性能较好。但是,这些参数是在初步的较少次数实验下确定的经验值,还缺乏坚实的理论支撑。所以,很多的参数仍然需要

通过进一步实验，或通过相关的理论分析进行适当调节。

- 1) 种群大小设置为 5，烟花高斯爆炸的个数为 5。
- 2) 火花总数  $m$  设置为 50，参数  $a$  设置为 0.8，参数  $b$  设置成 0.04。
- 3) 爆炸幅度之和  $\hat{A}$  设为 40，爆炸幅度无下限。
- 4) 实验中函数的维数都是 30 维，运行 20 次，函数评估次数为 400,000 次。

### 2.7.3 实验结果

目标函数计算 400,000 次时，各个算法的 20 次实验结果数据如表 2.3 所示（精确到  $10^{-6}$ ）。收敛曲线图如图 2.8 所示。

表 2.3 FWA 对比 CPSO 和 SPSO 的实验结果（精确到  $10^{-6}$ ）

|                      | FWA      |          | CPSO     |          | SPSO     |          |
|----------------------|----------|----------|----------|----------|----------|----------|
|                      | 均值       | 标准差      | 均值       | 标准差      | 均值       | 标准差      |
| Sphere               | 0        | 0        | 0        | 0        | 367.1166 | 186.7949 |
| Rosenbrock           | 12.16293 | 12.82113 | 66.58722 | 204.2907 | 5692076  | 4087432  |
| Griewank             | 0        | 0        | 0.003693 | 0.011792 | 1.088648 | 0.042218 |
| Rastrigin            | 0        | 0        | 6.769299 | 7.701368 | 676.1549 | 197.9695 |
| Rotated<br>Griewank  | 0        | 0        | 0.043401 | 0.042286 | 0.920613 | 0.088088 |
| Rotated<br>Rastrigin | 0        | 0        | 23.92579 | 13.6093  | 339.2073 | 62.38145 |

### 2.7.4 分析

从上述实验中可以明显地看出，在对上面六个标准测试函数的优化中，烟花算法不仅在收敛速度方面，而且在得到优化解的精度上都明显优于 SPSO 和 CPSO。这说明该算法具有良好的收敛性能和结果精度，能够成功应用于函数优化问题。在与 SPSO、CPSO 的函数优化对比实验中，烟花算法也取得了明显优势，这表明了烟花算法是非常成功的，展示了其良好的前景。另外，由于很多实际的工程、科研问题

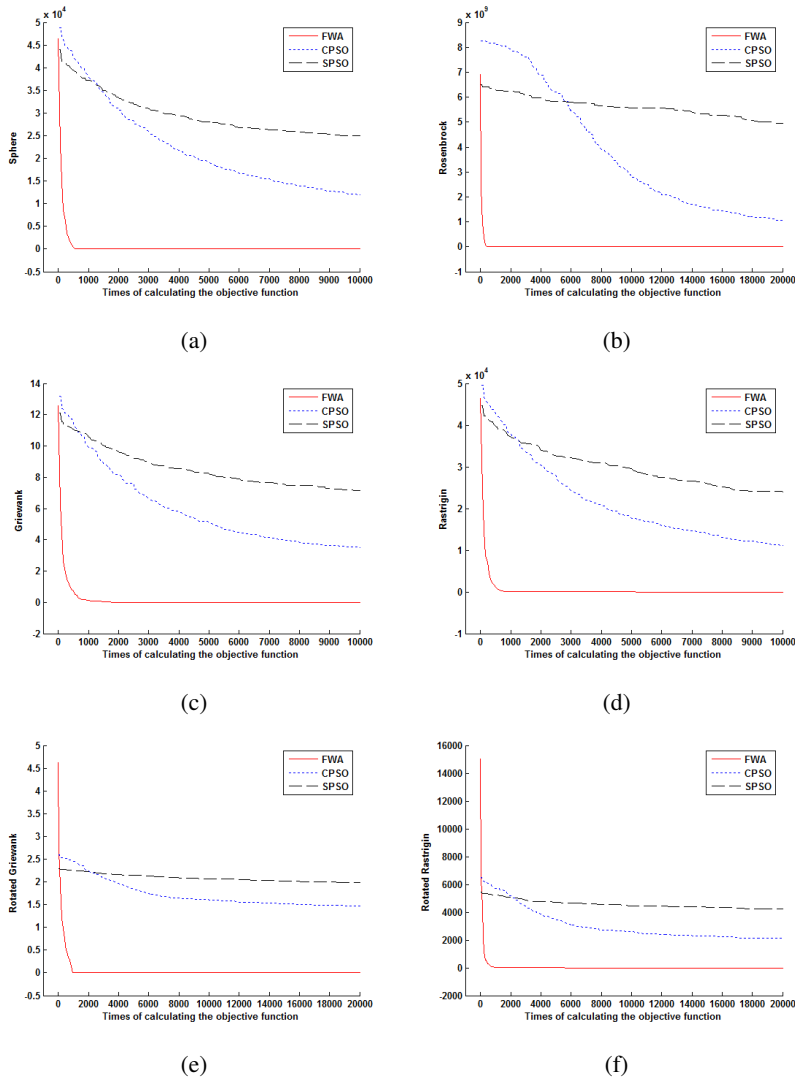


图 2.8 FWA、CPSO 和 SPSO 在六个函数上的收敛曲线

可以直接或间接地转化为函数优化问题，因此，这为推广烟花算法的实际应用提供了必备的基础和有利的条件。

然而，对烟花算法的研究还处于初步阶段，现有的烟花算法还存在一些问题，仍有许多改进和完善的工作需要做：

(1) 烟花算法的两个主要算子虽已初步实现，但还不够完善，如：高斯变异虽取得了很好的效果，但不能保证对所有的问题都能有效地处理。需要不断的加入新的

机制，对算法进行改进、完善和扩充。

(2) 算法的参数还都是在简单的测试条件中的初步设置，测试实验的强度不够，更加没有理论的分析保证，下一步需要找到参数合理以及最优设置。

(3) 算法的函数优化实验是在有限的几个标准函数上进行的，不可避免地会有一些片面性，需要在其他更多的优化函数或一些合成的复杂函数上进行完整全面的测试。

(4) 在应用领域，应将烟花算法推广到函数优化以外的众多方面，如：神经网络训练、模糊系统控制、模糊规则学习等，也可将其应用到离散问题的求解中。

## 2.8 小结

本章详细介绍了烟花算法，包括爆炸算子、变异算子、映射规则 and 选择策略。本章给出了烟花算法的步骤、流程图和伪代码，分析了烟花算法的特点和各个因子对算法性能的影响。此外，还与遗传算法、粒子群算法和烟花算法进行了比较。实验结果显示，与粒子群算法相比，烟花算法表现出了更好的性能。烟花算法提供了一种爆炸式的随机搜索方法求解复杂问题。因此，无论是理论、算法，还是应用研究，烟花算法都是值得深入研究的，并且可以带来巨大的经济利益。

## 参考文献

- [1] Ying Tan and Yuanchun Zhu. “Fireworks algorithm for optimization”. In: *Advances in Swarm Intelligence*. Springer, 2010, pp. 355–364.
- [2] Martin T Hagan, Howard B Demuth, Mark H Beale, et al. *Neural network design*. Pws Pub. Boston, 1996.
- [3] Y. Tan and C. Deng. “Solving for a Quadratic Programming with a Quadratic Constraint Based on a Neural Network Frame”. In: *Neurocomputing* 30 (2000), pp. 117–128.
- [4] et.al. Y. Tan. “Neural Network Design Approach of Cosine-modulated FIR Filter Bank and Compactly Supported Wavelets with Almost PR Property”. In: *Signal Processing* 69.1 (1998), pp. 29–48.
- [5] Y. Tan and Z.K. Liu. “On Matrix Eigendecomposition by Neural Networks”. In: *Neural Network World, International Journal on Neural and Mass-Parallel Computing and Information Systems* 8.3 (1998), pp. 337–352.
- [6] George J Klir and Bo Yuan. *Fuzzy sets and fuzzy logic*. Vol. 4. Prentice Hall New Jersey, 1995.
- [7] Agoston E Eiben and James E Smith. *Introduction to evolutionary computing*. springer, 2003.
- [8] Y. Tan and J. Wang. “Nonlinear Blind Separation Using Higher-Order Statistics and A Genetic Algorithm”. In: *IEEE Transaction on Evolutionary Computation* 5.6 (2001), pp. 600–612.
- [9] Heinz-Otto Peitgen, Hartmut Jürgens, and Dietmar Saupe. *Chaos and fractals: new frontiers of science*. Springer, 2004.
- [10] Peter JM Van Laarhoven and Emile HL Aarts. *Simulated annealing*. Springer, 1987.
- [11] Fred Glover and Manuel Laguna. *Tabu search*. Springer, 1999.

- [12] Y. Tan. “Particle Swarm Optimizer Algorithms Inspired by Immunity-Clonal Mechanism and Their Application to Spam Detection”. In: *International Journal of Swarm Intelligence Research* 1.1 (2010), pp. 64–86.
- [13] Junqi Zhang, Y. Tan, Lina Ni, Chen Xie, and Zheng Tang. “AMT-PSO: An Adaptive Magnification Transformation Based Particle Swarm Optimizer”. In: *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences* E94-D.4 (2011), pp. 786–797.
- [14] Junqi Zhang, Ying Tan, Ni Lina, Chen Xie, and Zheng Tang. “Hybrid Uniform Distribution of Particle Swarm Optimizer”. In: *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences* E93-A.10 (2010), pp. 1782–1791.
- [15] 刘坤, 谭莹, and 何新贵. “基于粒子群优化的过程神经网络学习算法”. In: 北京大学学报 (自然科学版) 47.2 (2011), pp. 238–244.
- [16] M. Dorigo, M. Birattari, and T. Stutzle. “Ant colony optimization”. In: *Computational Intelligence Magazine, IEEE* 1.4 (2006), pp. 28–39.
- [17] James Kennedy, Russell Eberhart, et al. “Particle swarm optimization”. In: *Proceedings of IEEE international conference on neural networks*. Vol. 4. 2. Perth, Australia. 1995, pp. 1942–1948.
- [18] Carmelo JA Bastos Filho, Fernando B de Lima Neto, Anthony JCC Lins, Antônio IS Nascimento, and Mar’lia P Lima. “Fish school search”. In: *Nature-Inspired Algorithms for Optimisation*. Springer, 2009, pp. 261–277.
- [19] Xin-She Yang. “Engineering optimizations via nature-inspired virtual bee algorithms”. In: *Artificial Intelligence and Knowledge Engineering Applications: A Bioinspired Approach*. Springer, 2005, pp. 317–323.
- [20] Szymon ukasik and Sawomir ak. “Firefly algorithm for continuous constrained optimization tasks”. In: *Computational Collective Intelligence. Semantic Web, Social Networks and Multiagent Systems*. Springer, 2009, pp. 97–106.
- [21] Xin-She Yang. “Firefly algorithms for multimodal optimization”. In: *Stochastic algorithms: foundations and applications*. Springer, 2009, pp. 169–178.

- 
- [22] Xin-She Yang and Suash Deb. “Cuckoo search via Lévy flights”. In: *Nature & Biologically Inspired Computing, 2009. NaBIC 2009. World Congress on*. IEEE. 2009, pp. 210–214.
- [23] Xin-She Yang. “A new metaheuristic bat-inspired algorithm”. In: *Nature inspired cooperative strategies for optimization (NISCO 2010)*. Springer, 2010, pp. 65–74.
- [24] Dervis Karaboga. *An idea based on honey bee swarm for numerical optimization*. Tech. rep. Technical report-tr06, Erciyes university, engineering faculty, computer engineering department, 2005.
- [25] Dervis Karaboga and Bahriye Basturk. “A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm”. In: *Journal of global optimization* 39.3 (2007), pp. 459–471.
- [26] Dervis Karaboga and Bahriye Basturk. “On the performance of artificial bee colony (ABC) algorithm”. In: *Applied soft computing* 8.1 (2008), pp. 687–697.
- [27] Carmelo JA Bastos Filho, Fernando B de Lima Neto, Anthony JCC Lins, Antônio IS Nascimento, and Mar´lia P Lima. “Fish school search”. In: *Nature-Inspired Algorithms for Optimisation*. Springer, 2009, pp. 261–277.
- [28] Amir Hossein Gandomi and Amir Hossein Alavi. “Krill herd: a new bio-inspired optimization algorithm”. In: *Communications in Nonlinear Science and Numerical Simulation* 17.12 (2012), pp. 4831–4845.
- [29] Christine R Blomeke, Stephen J Elliott, and Thomas M Walter. “Bacterial survivability and transferability on biometric devices”. In: *Security Technology, 2007 41st Annual IEEE International Carnahan Conference on*. IEEE. 2007, pp. 80–84.
- [30] Hamed Shah-Hosseini. “The intelligent water drops algorithm: a nature-inspired swarm-based optimization algorithm”. In: *International Journal of Bio-Inspired Computation* 1.1 (2009), pp. 71–79.
- [31] Yuhui Shi. “Brain storm optimization algorithm”. In: *Advances in Swarm Intelligence*. Springer, 2011, pp. 303–309.
- [32] NMH Tayarani and MR Akbarzadeh-T. “Magnetic optimization algorithms a new synthesis”. In: *Evolutionary Computation, 2008. CEC 2008.(IEEE World Congress on Computational Intelligence)*. IEEE Congress on. IEEE. 2008, pp. 2659–2664.

- [33] David H. Wolpert and William G. Macready. “No Free Lunch Theorems for Optimization”. In: *IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION* 1.1 (1997), pp. 67–322.
- [34] David H. Wolpert and William G. Macready. “Coevolutionary Free Lunches”. In: *IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION* 9.6 (2005), pp. 721–735.
- [35] Y. Tan, C. Yu, S.Q. Zheng, and K. Ding. “Introduction to Fireworks Algorithms”. In: *International Journal of Swarm Intelligence Research (IJSIR)* 4.4 (2013), pp. 39–71.
- [36] Jianhua Liu, Shaoqiu Zheng, and Ying Tan. “Analysis on Global Convergence and Time Complexity of Fireworks Algorithm”. In: (2014), pp. 3207–3213.
- [37] Yan Pei, Shaoqiu Zheng, Ying Tan, and Takagi Hideyuki. “An Empirical Study on Influence of Approximation Approaches on Enhancing Fireworks Algorithm”. In: *Proceedings of the 2012 IEEE Congress on System, Man and Cybernetics*. IEEE. 2012, pp. 1322–1327.
- [38] Ke Ding, Shaoqiu Zheng, and Ying Tan. “A GPU-based Parallel Fireworks Algorithm for Optimization”. In: *Proceeding of the Fifteenth Annual Conference on Genetic and Evolutionary Computation Conference*. GECCO 2013. Amsterdam, The Netherlands: ACM, 2013, pp. 9–16. ISBN: 978-1-4503-1963-8.
- [39] Shaoqiu Zheng, Andreas Janecek, and Ying Tan. “Enhanced Fireworks Algorithm”. In: *Evolutionary Computation (CEC), 2013 IEEE Congress on*. IEEE. 2013, pp. 2069–2077.
- [40] Junzhi Li Shaoqiu Zheng Andreas Janecek and Ying Tan. “Dynamic Search in Fireworks Algorithm”. In: (2014), pp. 3222–3229.
- [41] Shaoqiu Zheng Junzhi Li and Ying Tan. “Adaptive Fireworks Algorithm”. In: (2014), pp. 3214–3221.
- [42] Y.J. Zheng, X.L. Xu, and H.F. Ling. “A Hybrid Fireworks Optimization Method with Differential Evolution”. In: *Neurocomputing* (2012).
- [43] Shaoqiu Zheng Chao Yu Lingchen Kelley and Ying Tan. “Fireworks Algorithm with Differential Mutation for Solving the CEC 2014 Competition Problems”. In: (2014), pp. 3238–3245.

- 
- [44] Hongyuan Gao and Ming Diao. “Cultural firework algorithm and its application for digital filters design”. In: *International Journal of Modelling, Identification and Control* 14.4 (2011), pp. 324–331.
- [45] Wei Fang, Jun Sun, Wenbo Xu, and Jing Liu. “FIR digital filters design based on quantum-behaved particle swarm optimization”. In: *Innovative Computing, Information and Control, 2006. ICICIC’06. First International Conference on*. Vol. 1. IEEE. 2006, pp. 615–619.
- [46] Wei Fang, Jun Sun, and Wen-bo Xu. “FIR filter design based on adaptive quantum-behaved particle swarm optimization algorithm”. In: *Systems Engineering and Electronics* 30.7 (2008), pp. 1378–81.
- [47] Minxia Zhang Bei Zhang and Yujun Zheng. “A hybrid biogeography-based optimization and fireworks algorithm”. In: (2014), pp. 1–7.
- [48] Sujin Bureerat Nantiwat Pholdee. “Comparative performance of meta-heuristic algorithms for mass minimisation of trusses with dynamic constraints”. In: *Advances in Engineering Software* 75 (2014), pp. 1–13. DOI: 10.1016/j.advengsoft.2014.04.005.
- [49] Yu-Jun Zheng, Qin Song, and Sheng-Yong Chen. “Multiobjective fireworks optimization for variable-rate fertilization in oil crop production”. In: *Applied Soft Computing* 13.11 (2013), pp. 4253–4263.
- [50] 张家琴. “求解 0/1 背包问题的烟花算法研究”. In: *武汉工程职业技术学院学报* 23.3 (2011), pp. 64–66.
- [51] Andreas Janecek and Ying Tan. “Swarm intelligence for non-negative matrix factorization”. In: *International Journal of Swarm Intelligence Research (IJSIR)* 2.4 (2011), pp. 12–34.
- [52] Wenrui He, Guyue Mi, and Ying Tan. “Parameter Optimization of Local-Concentration Model for Spam Detection by Using Fireworks Algorithm”. In: *Advances in Swarm Intelligence*. Springer, 2013, pp. 439–450.
- [53] Zheng Shaoqiu and Ying Tan. “A Unified Distance Measure Scheme for Orientation Coding in Identification.” In: *Information Science and Technology, 2013 IEEE Congress on*. IEEE. 2013, pp. 979–985.

- 
- [54] Zhongyang Zheng and Ying Tan. “Group Explosion Strategy for Searching Multiple Targets using Swarm Robotic”. In: *IEEE Congress on Evolutionary Computation*. 2013, pp. 821–828.
- [55] A. Mohamed Imran and M. Kowsalya. “A new power system reconfiguration scheme for power loss minimization and voltage profile enhancement using Fireworks Algorithm”. In: *Electrical Power and Energy Systems* 62 (2014), pp. 312–322. DOI: 10.1016/j.ijepes.2014.04.034.
- [56] Ying Tan, Chao Yu, Shaoqiu Zheng, and Ding Ke. “Introduction to Fireworks Algorithm”. In: *International Journal of Swarm Intelligence Research*. Vol. 4. 4. IGI Global, 2013, pp. 39–70.
- [57] 谭营 and 郑少秋. “烟花算法研究进展”. In: *智能系统学报*. Vol. 9. 10. 2014, pp. 1–17.
- [58] Y. Tan and Z.M. Xiao. “Clonal particle swarm optimization and its applications”. In: *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on*. 2007, pp. 2303–2309.
- [59] 刘若辰, 杜海峰, and 焦李成. “免疫多克隆策略”. In: *计算机研究与发展* 41.4 (2004), pp. 571–576.
- [60] Jianhua Liu, Shaoqiu Zheng, and Ying Tan. “The Improvement on Controlling Exploration and Exploitation of Firework Algorithm”. In: *Advances in Swarm Intelligence*. Vol. 7928. Springer, 2013, pp. 11–23.
- [61] 黄翰, 郝志峰, and 秦勇. “进化规划算法的时间复杂度分析”. In: *计算机研究与发展* 45.11 (2008), pp. 1850–1857.
- [62] 黄翰, 郝志峰, 吴春国, and 秦勇. “蚁群算法的收敛速度分析”. In: *计算机学报* 30.8 (2007), pp. 1344–1353.
- [63] Donald E Knuth. “The Art of Computer Programming, volume 2: Seminumerical Algorithms”. In: *Amsterdam, London* (1969).
- [64] Pierre L’Ecuyer. “Random Number Generation”. English. In: *Handbook of Computational Statistics*. Ed. by James E. Gentle, Wolfgang Karl H. “ardle, and Yuichi Mori. Springer Handbooks of Computational Statistics. Springer Berlin Heidelberg, 2012, pp. 35–71. ISBN: 978-3-642-21550-6. DOI: 10.1007/978-3-642-

- 21551-3\_3. URL: [http://dx.doi.org/10.1007/978-3-642-21551-3\\_3](http://dx.doi.org/10.1007/978-3-642-21551-3_3).
- [65] Pierre L'Ecuyer and Richard Simard. "TestU01: A C library for empirical testing of random number generators". In: *ACM Trans. Math. Softw.* 33.4 (Aug. 2007). ISSN: 0098-3500. DOI: 10.1145/1268776.1268777. URL: <http://doi.acm.org/10.1145/1268776.1268777>.
- [66] Maurice Clerc. "List-Based Optimisers: Experiments and Open Questions". In: *International Journal of Swarm Intelligence Research (IJSIR)* 4.4 (2013), pp. 23–38.
- [67] C. J A Bastos-Filho, J.D. Andrade, M.R.S. Pita, and A.D. Ramos. "Impact of the quality of random numbers generators on the performance of particle swarm optimization". In: *Systems, Man and Cybernetics, 2009. SMC 2009. IEEE International Conference on*. 2009, pp. 4988–4993. DOI: 10.1109/ICSMC.2009.5346366.
- [68] C. J A Bastos-Filho, M.A.C. Oliveira, D. N O Nascimento, and A. D. Ramos. "Impact of the Random Number generator quality on particle swarm optimization algorithm running on graphic processor units". In: *Hybrid Intelligent Systems (HIS), 2010 10th International Conference on*. 2010, pp. 85–90. DOI: 10.1109/HIS.2010.5601073.
- [69] George Marsaglia. "Xorshift RNGs". In: *Journal of Statistical Software* 8.14 (2003), pp. 1–6.
- [70] NVIDIA Corp. *CURAND Library Programming Guide v5.5*. July 2013.
- [71] Intel Corp. *The Math Kernel Library*.
- [72] Ke Ding and Ying Tan. "cuROB: A GPU-based Test Suit for Real-Parameter Optimization". In: *Advances in Swarm Intelligence*. Ed. by C. A. C. Coello Y. Tan Y.H. Shi. Vol. 8794. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2014, pp. 66–78.
- [73] Lee Howes and David Thomas. "GPU gems 3". In: ed. by Hubert Nguyen. Addison-Wesley Professional, 2007. Chap. Efficient random number generation and application using CUDA, pp. 805–830.
- [74] John A Rice. *Mathematical Statistics and Data Analysis*. Belmont, CA USA: Thomson Higher Education, 2007.

- 
- [75] Pierre L'Ecuyer. "Tables of Linear Congruential Generators of Different Sizes and Good Lattice Structure". In: *Math. Comput.* 68.225 (Jan. 1999), pp. 249–260. ISSN: 0025-5718. DOI: 10.1090/S0025-5718-99-00996-5. URL: <http://dx.doi.org/10.1090/S0025-5718-99-00996-5>.
- [76] The Numerical Algorithms Group Ltd. *NAG Library Manual, Mark 23*. 2011.
- [77] Pierre L'Ecuyer. "Good parameters and implementations for combined multiple recursive random number generators". In: *Operations Research* 47.1 (1999), pp. 159–164.
- [78] Makoto Matsumoto and Takuji Nishimura. "Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator". In: *ACM Transactions on Modeling and Computer Simulation (TOMACS)* 8.1 (1998), pp. 3–30.
- [79] Mutsuo Saito and Makoto Matsumoto. "SIMD-Oriented Fast Mersenne Twister: a 128-bit Pseudorandom Number Generator". In: *Monte Carlo and Quasi-Monte Carlo Methods 2006*. Ed. by Alexander Keller, Stefan Heinrich, and Harald Niederreiter. Springer Berlin Heidelberg, 2008, pp. 607–622. ISBN: 978-3-540-74495-5. DOI: 10.1007/978-3-540-74496-2\_36. URL: [http://dx.doi.org/10.1007/978-3-540-74496-2\\_36](http://dx.doi.org/10.1007/978-3-540-74496-2_36).
- [80] Mutsuo Saito and Makoto Matsumoto. "Variants of Mersenne Twister Suitable for Graphic Processors". In: *ACM Trans. Math. Softw.* 39.2 (Feb. 2013), 12:1–12:20. ISSN: 0098-3500. DOI: 10.1145/2427023.2427029. URL: <http://doi.acm.org/10.1145/2427023.2427029>.
- [81] AMD Inc. *Core Math Library (ACML)*.
- [82] *The Boost Random Number Library*.
- [83] W. B. Langdon. "A Fast High Quality Pseudo Random Number Generator for nVidia CUDA". In: *Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference: Late Breaking Papers*. GECCO '09. Montreal, Quebec, Canada: ACM, 2009, pp. 2511–2514. ISBN: 978-1-60558-505-5. DOI: 10.1145/1570256.1570353. URL: <http://doi.acm.org/10.1145/1570256.1570353>.

- 
- [84] Scott Kirkpatrick and Erich P Stoll. “A very fast shift-register sequence random number generator”. In: *Journal of Computational Physics* 40.2 (1981), pp. 517–526. ISSN: 0021-9991. DOI: [http://dx.doi.org/10.1016/0021-9991\(81\)90227-8](http://dx.doi.org/10.1016/0021-9991(81)90227-8). URL: <http://www.sciencedirect.com/science/article/pii/0021999181902278>.
- [85] Makoto Matsumoto and Takuji Nishimura. “Dynamic Creation of Pseudorandom Number Generators”. In: *Monte Carlo and Quasi-Monte Carlo Methods 1998*. Ed. by Harald Niederreiter and J. Spanier. Springer Berlin Heidelberg, 2000, pp. 56–69.
- [86] Yan Pei and Hideyuki Takagi. “A survey on accelerating evolutionary computation approaches”. In: *Soft Computing and Pattern Recognition (SoCPaR), 2011 International Conference of*. IEEE. 2011, pp. 201–206.
- [87] Takeo Ingu and Hideyuki Takagi. “Accelerating a GA convergence by fitting a single-peak function”. In: *Fuzzy Systems Conference Proceedings, 1999. FUZZ-IEEE'99. 1999 IEEE International*. Vol. 3. IEEE. 1999, pp. 1415–1420.
- [88] Yan Pei and Hideyuki Takagi. “Accelerating evolutionary computation with elite obtained in projected one-dimensional spaces”. In: *Genetic and Evolutionary Computing (ICGEC), 2011 Fifth International Conference on*. IEEE. 2011, pp. 89–92.
- [89] Yan Pei and Hideyuki Takagi. “Comparative evaluations of evolutionary computation with elite obtained in reduced dimensional spaces”. In: *Intelligent Networking and Collaborative Systems (INCoS), 2011 Third International Conference on*. IEEE. 2011, pp. 35–40.
- [90] Ponnuthurai N Suganthan, Nikolaus Hansen, Jing J Liang, Kalyanmoy Deb, Y-Po Chen, Anne Auger, et al. “Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization”. In: *KanGAL Report 2005005* (2005).
- [91] A. Jati, G. Singh, P. Rakshit, A. Konar, E. Kim, and A.K. Nagar. “A hybridisation of Improved Harmony Search and Bacterial Foraging for multi-robot motion planning”. In: *Evolutionary Computation (CEC), 2012 IEEE Congress on*. IEEE. 2012, pp. 1–8.

- 
- [92] Andreas Janecek and Ying Tan. “Iterative improvement of the multiplicative update nmf algorithm using nature-inspired optimization”. In: *Natural Computation (ICNC), 2011 Seventh International Conference on*. Vol. 3. IEEE. 2011, pp. 1668–1672.
- [93] Yuhui Shi and Russell Eberhart. “A modified particle swarm optimizer”. In: *Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on*. IEEE. 1998, pp. 69–73.
- [94] D. Bratton and J. Kennedy. “Defining a standard for particle swarm optimization”. In: *Swarm Intelligence Symposium, 2007. SIS 2007. IEEE*. IEEE. 2007, pp. 120–127.
- [95] M. Clerc and J. Kennedy. “The particle swarm - explosion, stability, and convergence in a multidimensional complex space”. In: *Trans. Evol. Comp* 6.1 (2002), pp. 58–73.
- [96] JJ Liang, BY Qu, PN Suganthan, and Alfredo G Hernández-D’az. *Problem Definitions and Evaluation Criteria for the CEC 2013 Special Session on Real-Parameter Optimization*. 2013.
- [97] M. Zambrano-Bigiarini, M. Clerc, and R. Rojas. “Standard Particle Swarm Optimisation 2011 at CEC-2013: A baseline for future PSO improvements”. In: *Evolutionary Computation (CEC), 2013 IEEE Congress on*. 2013, pp. 2337–2344. DOI: 10.1109/CEC.2013.6557848.
- [98] Frank Wilcoxon. “Individual comparisons by ranking methods”. In: *Biometrics bulletin* 1.6 (1945), pp. 80–83.
- [99] D. Bratton and J. Kennedy. “Defining a standard for particle swarm optimization”. In: *Swarm Intelligence Symposium, 2007. SIS 2007. IEEE*. IEEE. 2007, pp. 120–127.
- [100] M Clerc. “Standard Particle Swarm Optimization, from 2006 to 2011”. In: *Particle Swarm Central* (2011).
- [101] Rainer Storn and Kenneth Price. “Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces”. In: *Journal of global optimization* 11.4 (1997), pp. 341–359.

- 
- [102] JJ Liang, BY Qu, and PN Suganthan. *Problem Definitions and Evaluation Criteria for the CEC 2014 Special Session and Competition on Single Objective Real-Parameter Numerical Optimization*. Tech. rep. Technical Report 201311, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University, Singapore, 2013.
- [103] Dan Simon. “Biogeography-based optimization”. In: *Evolutionary Computation, IEEE Transactions on* 12.6 (2008), pp. 702–713.
- [104] Aniruddha Bhattacharya and Pranab Kumar Chattopadhyay. “Biogeography-based optimization for different economic load dispatch problems”. In: *Power Systems, IEEE Transactions on* 25.2 (2010), pp. 1064–1077.
- [105] Urvinder Singh, Harish Kumar, and Tara Singh Kamal. “Design of Yagi-Uda antenna using biogeography based optimization”. In: *Antennas and Propagation, IEEE Transactions on* 58.10 (2010), pp. 3375–3379.
- [106] Ilhem Boussaïd, Amitava Chatterjee, Patrick Siarry, and Mohamed Ahmed-Nacer. “Biogeography-based optimization for constrained optimization problems”. In: *Computers & Operations Research* 39.12 (2012), pp. 3293–3304.
- [107] Yu-Jun Zheng, Hai-Feng Ling, Hai-He Shi, Hai-Song Chen, and Sheng-Yong Chen. “Emergency railway wagon scheduling by hybrid biogeography-based optimization”. In: *Computers & Operations Research* 43 (2014), pp. 1–8.
- [108] Nidamarthi Srinivas and Kalyanmoy Deb. “Multiobjective optimization using non-dominated sorting in genetic algorithms”. In: *Evolutionary computation* 2.3 (1994), pp. 221–248.
- [109] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. “A fast and elitist multiobjective genetic algorithm: NSGA-II”. In: *Evolutionary Computation, IEEE Transactions on* 6.2 (2002), pp. 182–197.
- [110] Eckart Zitzler and Lothar Thiele. “Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach”. In: *Evolutionary Computation, IEEE Transactions on* 3.4 (1999), pp. 257–271.
- [111] Carlos A Coello Coello, David A Van Veldhuizen, and Gary B Lamont. *Evolutionary algorithms for solving multi-objective problems*. Vol. 242. Springer, 2002.

- 
- [112] Joshua D Knowles and David W Corne. “M-PAES: A memetic algorithm for multiobjective optimization”. In: *Evolutionary Computation, 2000. Proceedings of the 2000 Congress on*. Vol. 1. IEEE. 2000, pp. 325–332.
- [113] Hussein A Abbass, Ruhul Sarker, and Charles Newton. “PDE: a Pareto-frontier differential evolution approach for multi-objective optimization problems”. In: *Evolutionary Computation, 2001. Proceedings of the 2001 Congress on*. Vol. 2. IEEE. 2001, pp. 971–978.
- [114] Wenyin Gong and Zhihua Cai. “A multiobjective differential evolution algorithm for constrained optimization”. In: *Evolutionary Computation, 2008. CEC 2008.(IEEE World Congress on Computational Intelligence). IEEE Congress on*. IEEE. 2008, pp. 181–188.
- [115] Xiaodong Li. “A non-dominated sorting particle swarm optimizer for multiobjective optimization”. In: *Genetic and Evolutionary Computation—GECCO 2003*. Springer. 2003, pp. 37–48.
- [116] B.W. Silverman. “Density Estimation for Statistics and Data Analysis”. In: Chapman and Hall, London. 1986.
- [117] Qing-yuan GUO, Zhi-yu LI, and Xue-wen TU. “Plant nutritional aspects and effects of fertilizer application in rapeseed in red-yellow soil of south China . Fertilizer application of double-low rapeseed cultivar, Zhongshuang No. 7 in red paddy soil [J]”. In: *Chinese Journal of Oil Crop Scieives* 1 (2001), p. 011.
- [118] Liu Yankun<sup>1</sup> Wu Huishan<sup>2</sup> Huang Jinlan<sup>1</sup> Zhuang and Zengfu<sup>3</sup> Lin Jia<sup>1</sup> Liu Yongye. “Preliminary Report on Fertilization Trial of Canarium album”. In: *Guangdong Forestry Science and Technology* 5 (2007), p. 004.
- [119] George Bernard Dantzig, Delbert R Fulkerson, and Selmer M Johnson. “On a linear-programming, combinatorial approach to the traveling-salesman problem”. In: *Operations Research* 7.1 (1959), pp. 58–66.
- [120] M Grotschel and L Lovász. “Combinatorial optimization”. In: *Handbook of combinatorics* 2 (1995), pp. 1541–1597.
- [121] Robert G Bland and David F Shallcross. “Large travelling salesman problems arising from experiments in X-ray crystallography: a preliminary report on computation”. In: *Operations Research Letters* 8.3 (1989), pp. 125–128.

- 
- [122] R Garey Michael and David S Johnson. “Computers and Intractability: A guide to the theory of NP-completeness”. In: *WH Freeman & Co., San Francisco* (1979).
- [123] Harlan Crowder and Manfred W Padberg. “Solving large-scale symmetric travelling salesman problems to optimality”. In: *Management Science* 26.5 (1980), pp. 495–509.
- [124] Manfred Padberg and Giovanni Rinaldi. “Optimization of a 532-city symmetric traveling salesman problem by branch and cut”. In: *Operations Research Letters* 6.1 (1987), pp. 1–7.
- [125] David S Johnson and Lyle A McGeoch. “The traveling salesman problem: A case study in local optimization”. In: *Local search in combinatorial optimization* 1 (1997), pp. 215–310.
- [126] John J Hopfield and David W Tank. “ “Neural” computation of decisions in optimization problems”. In: *Biological cybernetics* 52.3 (1985), pp. 141–152.
- [127] Heinz Mühlenbein, Martina Gorges-Schleuter, and Ottmar Krämer. “Evolution algorithms in combinatorial optimization”. In: *Parallel Computing* 7.1 (1988), pp. 65–85.
- [128] Heinrich Braun. “On solving travelling salesman problems by genetic algorithms”. In: *Parallel problem solving from nature*. Springer, 1991, pp. 129–133.
- [129] Maurice Clerc. “Discrete particle swarm optimization, illustrated by the traveling salesman problem”. In: *New optimization techniques in engineering*. Springer, 2004, pp. 219–239.
- [130] Mehmet Fatih Tasgetiren, Ponnuthurai N Suganthan, and Quan-Qe Pan. “A discrete particle swarm optimization algorithm for the generalized traveling salesman problem”. In: *Proceedings of the 9th annual conference on Genetic and evolutionary computation*. ACM. 2007, pp. 158–167.
- [131] Marco Dorigo and Luca Maria Gambardella. “Ant colony system: a cooperative learning approach to the traveling salesman problem”. In: *Evolutionary Computation, IEEE Transactions on* 1.1 (1997), pp. 53–66.

- 
- [132] Bernd Freisleben and Peter Merz. “A genetic local search algorithm for solving symmetric and asymmetric traveling salesman problems”. In: *Evolutionary Computation, 1996., Proceedings of IEEE International Conference on*. IEEE. 1996, pp. 616–621.
- [133] Bernd Freisleben and Peter Merz. “New genetic local search operators for the traveling salesman problem”. In: *Parallel Problem Solving from Nature—PPSN IV*. Springer, 1996, pp. 890–899.
- [134] Holger H. Hoos<sup>2</sup> Thomas Stutzle. “Max min ant system”. In: *Future generation computer systems, Elsevier*. 2000.
- [135] S. Bureerat. “Hybrid population-based incremental learning using real codes”. In: *Learning and Intelligent Optimization* (2011), pp. 379–391.
- [136] Gerhard Reinelt. “TSPLIB—A traveling salesman problem library”. In: *ORSA journal on computing* 3.4 (1991), pp. 376–384.
- [137] Luping Fang, Pan Chen, and Shihua Liu. “Particle swarm optimization with simulated annealing for TSP”. In: *Proceedings of the 6th WSEAS Int. Conf. on Artificial Intelligence, Knowledge Engineering and Data Bases, Corfu Island, Greece, February*. 2007, pp. 16–19.
- [138] J.D. Owens, M. Houston, D. Luebke, S. Green, J.E. Stone, and J.C. Phillips. “GPU Computing”. In: *Proceedings of the IEEE* 96.5 (2008), pp. 879–899.
- [139] Stanislav Stankovi and Jaakko Astola. “GPU Computing with Applications in Digital Logic”. In: ed. by Jaakko Astola, Mititaka Kameyama, Martin Lukac, and Radomir S. Stankovi. Tampere International Center for Signal Processing. 2012. Chap. An Overview of Miscellaneous Applications of GPU Computing, pp. 191–215.
- [140] Rob Farber. 高性能 CUDA 应用设计与开发. 机械工业出版社, 2013.
- [141] You Zhou and Ying Tan. “GPU-based Parallel Particle Swarm Optimization”. In: *Evolutionary Computation, 2009. CEC '09. IEEE Congress on*. May 2009, pp. 1493–1500.
- [142] You Zhou and Ying Tan. “Particle swarm optimization with triggered mutation and its implementation based on GPU”. In: *Proceedings of the 12th annual conference on Genetic and evolutionary computation*. GECCO '10. Portland, Oregon, USA: ACM, 2010, pp. 1–8. ISBN: 978-1-4503-0072-8.

- [143] You Zhou and Ying Tan. “GPU-based parallel multi-objective particle swarm optimization”. In: *International Journal of Artificial Intelligence* 7.A11 (Autumn (October) 2011), pp. 125–141.
- [144] Boguslaw Rymut and Bogdan Kwolek. “GPU-Supported Object Tracking Using Adaptive Appearance Models and Particle Swarm Optimization”. In: *Computer Vision and Graphics*. Ed. by Leonard Bolc, Ryszard Tadeusiewicz, LeszekJ. Chmielewski, and Konrad Wojciechowski. Vol. 6375. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2010, pp. 227–234. ISBN: 978-3-642-15906-0. DOI: 10.1007/978-3-642-15907-7\_28. URL: [http://dx.doi.org/10.1007/978-3-642-15907-7\\_28](http://dx.doi.org/10.1007/978-3-642-15907-7_28).
- [145] Luca Mussi, Spela Ivekovic, and Stefano Cagnoni. “Markerless Articulated Human Body Tracking from Multi-view Video with GPU-PSO”. In: *Evolvable Systems: From Biology to Hardware*. Ed. by Gianluca Tempesti, AndyM. Tyrrell, and JulianF. Miller. Vol. 6274. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2010, pp. 97–108. ISBN: 978-3-642-15322-8. DOI: 10.1007/978-3-642-15323-5\_9. URL: [http://dx.doi.org/10.1007/978-3-642-15323-5\\_9](http://dx.doi.org/10.1007/978-3-642-15323-5_9).
- [146] MarcoS. Nobile, Daniela Besozzi, Paolo Cazzaniga, Giancarlo Mauri, and Dario Pescini. “A GPU-Based Multi-swarm PSO Method for Parameter Estimation in Stochastic Biological Systems Exploiting Discrete-Time Target Series”. In: *Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics*. Vol. 7246. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2012, pp. 74–85. ISBN: 978-3-642-29065-7. DOI: 10.1007/978-3-642-29066-4\_7. URL: [http://dx.doi.org/10.1007/978-3-642-29066-4\\_7](http://dx.doi.org/10.1007/978-3-642-29066-4_7).
- [147] V.V. Zhirnov, R.K. Cavin, J.A. Hutchby, and G.I. Bourianoff. “Limits to binary logic switch scaling - a gedanken model”. In: *Proceedings of the IEEE* 91.11 (2003), pp. 1934–1939. ISSN: 0018-9219. DOI: 10.1109/JPROC.2003.818324.
- [148] Herb Sutter. “The free lunch is over: A fundamental turn toward concurrency in software”. In: *Dr. Dobbs’ s Journal* 30.3 (2005), pp. 202–210.
- [149] P.E. Ross. “Why CPU Frequency Stalled”. In: *Spectrum, IEEE* 45.4 (Apr. 2008), p. 72. ISSN: 0018-9235.

- 
- [150] Shekhar Borkar. “Getting Gigascale Chips: Challenges and Opportunities in Continuing Moore’s Law”. In: *Queue* 1.7 (Oct. 2003), pp. 26–33. ISSN: 1542-7730. DOI: 10.1145/957717.957757. URL: <http://doi.acm.org/10.1145/957717.957757>.
- [151] NVIDIA Corp. *CUDA C Programming Guide*. July 2013.
- [152] Yaser Jararweh, Moath Jarrah, Abdelkader Bousselham, and Salim Hariri. “Gpu-based personal supercomputing”. In: *Applied Electrical Engineering and Computing Technologies (AEECT), 2013 IEEE Jordan Conference on*. 2013, pp. 1–5. DOI: 10.1109/AEECT.2013.6716444.
- [153] NVIDIA Corp. *NVIDIA’s Next Generation CUDA™ Compute Architecture: Fermi™*. 2009.
- [154] Mark Segal and Kurt Akeley. *The OpenGL® Graphics System: A Specification (Version 4.4)*. The Khronos Group Inc. July 2013.
- [155] Ian Buck, Tim Foley, Daniel Horn, Jeremy Sugerman, Kayvon Fatahalian, Mike Houston, et al. “Brook for GPUs: stream computing on graphics hardware”. In: *ACM Trans. Graph.* 23.3 (Aug. 2004), pp. 777–786. ISSN: 0730-0301. DOI: 10.1145/1015706.1015800. URL: <http://doi.acm.org/10.1145/1015706.1015800>.
- [156] Michael McCool, Stefanus Du Toit, Tiberiu Popa, Bryan Chan, and Kevin Moule. “Shader algebra”. In: *ACM Trans. Graph.* 23.3 (Aug. 2004), pp. 787–795. ISSN: 0730-0301. DOI: 10.1145/1015706.1015801. URL: <http://doi.acm.org/10.1145/1015706.1015801>.
- [157] OpenACC Standards Group. *The OpenACC® Application Programming Interface (version 2.0)*. June 2013.
- [158] Microsoft Corp. *C++ AMP (C++ Accelerated Massive Parallelism)*. <http://msdn.microsoft.com/en-us/library/hh265137.aspx>.
- [159] NVIDIA Corp. *CUDA C Best Practices Guide*. July 2013.
- [160] Benedict Gaster, Lee Howes, David R. Kaeli, Perhaad Mistry, and Dana Schaa. *OpenCL 异构计算*. 清华大学出版社, 2012.
- [161] David B. Kirk and Wen-mei W. Hwu. *Programming Massively Parallel Processors: A Hands-on Approach*. Beijing, China: Tsinghua University Press, 2010.

- 
- [162] Aaftab Munshi, Benedict Gaster, Timothy G. Mattson, James Fung, and Dan Ginsburg. *OpenCL Programming Guide*. Addison-Wesley Professional, 2011. ISBN: 0321749642.
- [163] AMD 上海研发中心. 跨平台的多核与从核编程讲义——*OpenCL* 的方式. 2010.
- [164] Arian Maghazeh, Unmesh D. Bordoloi, Petru Eles, and Zebo Peng. *General Purpose Computing on Low-Power Embedded GPUs : Has It Come of Age?* Tech. rep. Linköping University, The Institute of Technology, 2013, p. 10.
- [165] Daniel Hallmans, Kristian Sandstrom, Markus Lindgren, and Thomas Nolte. “GPGPU for industrial control systems”. In: *Emerging Technologies Factory Automation (ETFA), 2013 IEEE 18th Conference on*. 2013, pp. 1–4. DOI: 10.1109/ETFA.2013.6648166.
- [166] You Zhou and Ying Tan. “GPU-based parallel multi-objective particle swarm optimization”. In: *International Journal of Artificial Intelligence* 7.A11 (Autumn (October) 2011), pp. 125–141.
- [167] Vincent Roberge and Mohammed Tarbouchi. “Parallel Particle Swarm Optimization on Graphical Processing Unit for Pose Estimation”. In: *WSEAS TRANSACTIONS on COMPUTERS* 11.6 (June 2012), pp. 170–179.
- [168] M. Molga and C. Smutnicki. “Test functions for optimization needs”. In: (2005). URL: <http://www.zsd.ict.pwr.wroc.pl/files/docs/functions.pdf>.
- [169] Daniel D Lee and H Sebastian Seung. “Learning the parts of objects by non-negative matrix factorization”. In: *Nature* 401.6755 (1999), pp. 788–791.
- [170] Chih-Jen Lin. “Projected gradient methods for nonnegative matrix factorization”. In: *Neural computation* 19.10 (2007), pp. 2756–2779.
- [171] Hyunsoo Kim and Haesun Park. “Nonnegative matrix factorization based on alternating nonnegativity constrained least squares and active set method”. In: *SIAM Journal on Matrix Analysis and Applications* 30.2 (2008), pp. 713–730.
- [172] Mikkel N Schmidt and Hans Laurberg. “Nonnegative matrix factorization with Gaussian process priors”. In: *Computational intelligence and neuroscience* 2008 (2008), p. 3.

- 
- [173] Stefan Wild, James Curry, and Anne Dougherty. “Improving non-negative matrix factorizations through structured initialization”. In: *Pattern Recognition* 37.11 (2004), pp. 2217–2232.
- [174] Yun Xue, Chong Sze Tong, Ying Chen, and Wen-Sheng Chen. “Clustering-based initialization for non-negative matrix factorization”. In: *Applied Mathematics and Computation* 205.2 (2008), pp. 525–536.
- [175] Christos Boutsidis and Efstratios Gallopoulos. “SVD based initialization: A head start for nonnegative matrix factorization”. In: *Pattern Recognition* 41.4 (2008), pp. 1350–1362.
- [176] Kurt Stadthanner, Dominik Lutter, Fabian J Theis, Elmar Wolfgang Lang, Ana Maria Tomé, Petia Georgieva, et al. “Sparse nonnegative matrix factorization with genetic algorithms for microarray analysis”. In: *Neural Networks, 2007. IJCNN 2007. International Joint Conference on*. IEEE. 2007, pp. 294–299.
- [177] Andreas Janecek and Ying Tan. “Using population based algorithms for initializing nonnegative matrix factorization”. In: *Advances in Swarm Intelligence*. Springer, 2011, pp. 307–316.
- [178] Michael W Berry, Zlatko Drmac, and Elizabeth R Jessup. “Matrices, vector spaces, and information retrieval”. In: *SIAM review* 41.2 (1999), pp. 335–362.
- [179] MEH Pedersen. “SwarmOps: Black-Box Optimization in ANSI C”. In: *Hvass Lab., Southampton, UK* (2008).
- [180] Randy L Haupt and Sue Ellen Haupt. *Practical genetic algorithms*. John Wiley & Sons, 2004.
- [181] Andreas Janecek, Wilfried N Gansterer, Michael Demel, and Gerhard Ecker. “On the Relationship Between Feature Selection and Classification Accuracy.” In: *Journal of Machine Learning Research-Proceedings Track 4* (2008), pp. 90–105.
- [182] Kjellerstrand. “hakanks hemsida”. In: [http : //www.hakank.org/weka/](http://www.hakank.org/weka/) (2011).
- [183] Ian H Witten and Eibe Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2005.
- [184] Gareth Jones, Alexander Robertson, Chawchat Santimetvirul, and Peter Willett. “Non-hierarchic document clustering using a genetic algorithm”. In: *Information Research* 1.1 (1995), pp. 1–1.

- 
- [185] Vijay V Raghavan and Kim Birchard. “A clustering strategy based on a formalism of the reproductive process in natural systems”. In: *ACM SIGIR Forum*. Vol. 14. 2. ACM. 1979, pp. 10–22.
- [186] Xiaohui Cui, Thomas E Potok, and Paul Palathingal. “Document clustering using particle swarm optimization”. In: *Swarm Intelligence Symposium, 2005. SIS 2005. Proceedings 2005 IEEE*. IEEE. 2005, pp. 185–191.
- [187] S.C. Gu, Y. Tan, and X. He. “Recentness biased Learning for Time Series Forecasting”. In: *Information Sciences 237* (2013), pp. 29–38.
- [188] B.S. Everitt, S. Landau, M. Leese, and D. Stahl. *Cluster Analysis*. Wiley, 2011.
- [189] Yiming Yang and Jan O Pedersen. “A comparative study on feature selection in text categorization”. In: (1997).
- [190] Thorsten Joachims. *A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization*. Tech. rep. DTIC Document, 1996.
- [191] “Principal Component Analysis”. In: John Wiley & Sons, Ltd, 2005.
- [192] J.Wang and Y. Tan. “Efficient Euclidean Distance Transform Algorithm of Binary Images in Arbitrary Dimensions”. In: *Pattern Recognition 46.1* (2013), pp. 230–242.
- [193] X. Huang, Y. Tan, and X.G. He. “An Intelligent Multi-feature Statistical Approach for Discrimination of Driving Conditions of Hybrid Electric Vehicle”. In: *IEEE Transactions on Intelligent Transportation Systems 12.2* (2011), pp. 453–456.
- [194] 谭营 and 朱元春. “反垃圾电子邮件研究进展”. In: *智能系统学报 5.3* (2010), pp. 189–201.
- [195] Enrico Blanzieri and Anton Bryl. “A survey of learning-based techniques of email spam filtering”. In: *Artificial Intelligence Review 29.1* (2008), pp. 63–92.
- [196] Evangelos Moustakas, Chandrasekaran Ranganathan, and Penny Duquenoy. “Combating Spam through Legislation: A Comparative Analysis of US and European Approaches.” In: *CEAS*. 2005.
- [197] Shlomo Hershkop. “Behavior-based email analysis with application to spam detection”. PhD thesis. Columbia University, 2006.
- [198] Enrique Puertas Sanz, José Mar’a Gómez Hidalgo, and José Carlos Cortizo Pérez. “Email spam filtering”. In: *Advances in Computers 74* (2008), pp. 45–114.

- 
- [199] Zhenhai Duan, Yingfei Dong, and Kartik Gopalan. “DMTP: Controlling spam through message delivery differentiation”. In: *Computer Networks* 51.10 (2007), pp. 2616–2630.
- [200] John Ioannidis. “Fighting Spam by Encapsulating Policy in Email Addresses.” In: *NDSS*. 2003.
- [201] James Carpinter and Ray Hunt. “Tightening the net: A review of current and next generation spam filtering tools”. In: *Computers & security* 25.8 (2006), pp. 566–578.
- [202] Gordon V Cormack. “Email spam filtering: A systematic review”. In: *Foundations and Trends in Information Retrieval* 1.4 (2007), pp. 335–455.
- [203] Sotiris B Kotsiantis. “Supervised machine learning: a review of classification techniques.” In: *Informatica (03505596)* 31.3 (2007).
- [204] Karl-Michael Schneider. “A comparison of event models for Naive Bayes anti-spam e-mail filtering”. In: *Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics-Volume 1*. Association for Computational Linguistics. 2003, pp. 307–314.
- [205] Kevin R Gee. “Using latent semantic indexing to filter spam”. In: *Proceedings of the 2003 ACM symposium on Applied computing*. ACM. 2003, pp. 460–464.
- [206] Ying Tan, Chao Deng, and Guangchen Ruan. “Concentration based feature construction approach for spam detection”. In: *Neural Networks, 2009. IJCNN 2009. International Joint Conference on*. IEEE. 2009, pp. 3088–3093.
- [207] Guangchen Ruan and Ying Tan. “A three-layer back-propagation neural network for spam detection using artificial immune concentration”. In: *Soft Computing* 14.2 (2010), pp. 139–150.
- [208] Mark Dredze, Reuven Gevartyahu, and Ari Elias-Bachrach. “Learning Fast Classifiers for Image Spam.” In: *CEAS*. 2007.
- [209] Giorgio Fumera, Ignazio Pillai, and Fabio Roli. “Spam filtering based on the analysis of text information embedded into images”. In: *The Journal of Machine Learning Research* 7 (2006), pp. 2699–2720.

- 
- [210] Chi-Yuan Yeh, Chih-Hung Wu, and Shing-Hwang Doong. “Effective spam classification based on meta-heuristics”. In: *Systems, Man and Cybernetics, 2005 IEEE International Conference on*. Vol. 4. IEEE. 2005, pp. 3872–3877.
- [211] Anirudh Vadakkedath Ramachandran. “Mitigating spam using network-level features”. In: (2011).
- [212] Wenrui He, Guyue Mi, and Ying Tan. “Parameter Optimization of Local-Concentration Model for Spam Detection by Using Fireworks Algorithm”. In: *Advances in Swarm Intelligence*. Springer, 2013, pp. 439–450.
- [213] G.C. Ruan and Y. Tan. “A Three-layer Back-propagation Neural Network for Spam Detection Using Artificial Immune Concentration”. In: *Softcomputing* 14 (2010), pp. 139–150.
- [214] Y. Tan and G.C. Ruan. “Uninterrupted Approaches for Spam Detection Based on SVM and AIS”. In: *International Journal of Computational Intelligence and Pattern Recognition (IJCIPR)* 1.1 (2014), pp. 1–26.
- [215] Yuanchun Zhu and Ying Tan. “A local-concentration-based feature extraction approach for spam filtering”. In: *Information Forensics and Security, IEEE Transactions on* 6.2 (2011), pp. 486–497.
- [216] Harris Drucker, S Wu, and Vladimir N Vapnik. “Support vector machines for spam categorization”. In: *Neural Networks, IEEE Transactions on* 10.5 (1999), pp. 1048–1054.
- [217] S.C. Gu, Y. Tan, and X.G. He. “Discriminant Analysis via Support Vectors”. In: *Neurocomputing* 73.10-12 (2010), pp. 1669–1675.
- [218] Y. Tan and J. Wang. “A Support Vector Network with Hybrid Kernel and Minimal Vapnik-Chervonenkis Dimension”. In: *IEEE Trans. On Knowledge and Data Engineering* 26.2 (2004), pp. 385–395.
- [219] I. Androutsopoulos, G. Paliouras, and E. Michelakis. *Learning to filter unsolicited commercial e-mail*. Tech. rep. “DEMOKRITOS”, National Center for Scientific Research, 2004, pp. 1–52.
- [220] 谭营 and 王军. “手指静脉身份识别技术最新进展”. In: *智能系统学报* 6.6 (2011), pp. 471–482.

- 
- [221] 顾岁成, 谭营, and 何新贵. “Laplace 平滑变换及其在人脸识别中的应用”. In: 中国科学: 信息科学 41.3 (2011), pp. 257–268.
- [222] S.C. Gu, Y. Tan, and X.G. He. “Laplacian Smoothing Transform for Face Recognition”. In: *Science China (Information Science)* 53.12 (2010), pp. 24150–2428.
- [223] A.W.K. Kong and D. Zhang. “Competitive coding scheme for palmprint verification”. In: *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*. Vol. 1. IEEE. 2004, pp. 520–523.
- [224] X. Wu, K. Wang, and D. Zhang. “Palmprint authentication based on orientation code matching”. In: *Audio-and Video-Based Biometric Person Authentication*. Springer. 2005, pp. 83–132.
- [225] A.W.K. Kong. “Palmprint identification based on generalization of iriscodes”. PhD thesis. University of Waterloo, 2007.
- [226] Z. Guo, W. Zuo, L. Zhang, and D. Zhang. “A unified distance measurement for orientation coding in palmprint verification”. In: *Neurocomputing* 73.4 (2010), pp. 944–950.
- [227] R. Storn and K. Price. “Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces”. In: *Journal of global optimization* 11.4 (1997), pp. 341–359.
- [228] Y. Tan. “Swarm Robotics: Collective Behavior Inspired by Nature”. In: *Journal of Comput Sci Syst Biol (JCSB)* ().
- [229] Claire Detrain and Jean-Louis Deneubourg. “Self-organized structures in a superorganism: do ants “behave” like molecules?”. In: *Physics of Life Reviews* 3.3 (2006), pp. 162–187.
- [230] Scott Camazine, Jean-Louis Deneubourg, Nigel R. Franks, James Sneyd, Guy Theraula, and Eric Bonabeau. *Self-organization in biological systems*. Princeton University Press, 2003.
- [231] Y. Tan and Z.Y. Zheng. “Research Advance in Swarm Robotics”. In: *Defence Technology* 9.1 (2013), pp. 31–62.
- [232] Kurt Derr and Milos Manic. “Multi-robot, multi-target particle swarm optimization search in noisy wireless environments”. In: *Human System Interactions, 2009. HSI’09. 2nd Conference on*. IEEE. 2009, pp. 81–86.

- 
- [233] Ali Marjovi, Joao Nunes, Pedro Sousa, Ricardo Faria, and Lino Marques. “An olfactory-based robot swarm navigation method”. In: *Robotics and Automation (ICRA), 2010 IEEE International Conference on*. IEEE. 2010, pp. 4958–4963.
- [234] Goncalo Cabrita and Lino Marques. “Divergence-based odor source declaration”. In: *Control Conference (ASCC), 2013 9th Asian*. IEEE. 2013, pp. 1–6.
- [235] H.E. Espitia and J.I. Sofrony. “Path planning of mobile robots using potential fields and swarms of Brownian particles”. In: *Evolutionary Computation (CEC), 2011 IEEE Congress on*. IEEE. 2011, pp. 123–129.
- [236] Qian Zhu, Alei Liang, and Haibing Guan. “A PSO-inspired multi-robot search algorithm independent of global information”. In: *Swarm Intelligence (SIS), 2011 IEEE Symposium on*. IEEE. 2011, pp. 1–7.
- [237] Jim Pugh and Alcherio Martinoli. “Multi-robot learning with particle swarm optimization”. In: *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*. ACM. 2006, pp. 441–448.
- [238] Tamanna Arora and Melanie E. Moses. “Ant Colony Optimization for power efficient routing in manhattan and non-manhattan VLSI architectures”. In: *Swarm Intelligence Symposium, 2009. SIS’09. IEEE*. IEEE. 2009, pp. 137–144.
- [239] C. Blum. “Ant colony optimization: Introduction and recent trends”. In: *Physics of Life reviews* 2.4 (2005), pp. 353–373.
- [240] Jigong Li, Jing Yang, Shigang Cui, and Lihui Geng. “Speed limitation of a mobile robot and methodology of tracing odor plume in airflow environments”. In: *Procedia Engineering* 15 (2011), pp. 1041–1045.
- [241] Micael S. Couceiro, Rui P. Rocha, and Nuno M.F. Ferreira. “A novel multi-robot exploration approach based on Particle Swarm Optimization algorithms”. In: *Safety, Security, and Rescue Robotics (SSRR), 2011 IEEE International Symposium on*. IEEE. 2011, pp. 327–332.
- [242] Yuli Zhang, Xiaoping Ma, and Yanzi Miao. “Localization of multiple odor sources using modified glowworm swarm optimization with collective robots”. In: *Control Conference (CCC), 2011 30th Chinese*. IEEE. 2011, pp. 1899–1904.

- 
- [243] Áron Kisdi and Adrian R.L. Tatnall. “Future robotic exploration using honeybee search strategy: Example search for caves on Mars”. In: *Acta Astronautica* 68.11 (2011), pp. 1790–1799.
- [244] Adam T. Hayes, Alcherio Martinoli, and Rodney M. Goodman. “Distributed odor source localization”. In: *Sensors Journal, IEEE* 2.3 (2002), pp. 260–271.
- [245] Wei Li, Jay A. Farrell, Shuo Pang, and Richard M. Arrieta. “Moth-inspired chemical plume tracing on an autonomous underwater vehicle”. In: *Robotics, IEEE Transactions on* 22.2 (2006), pp. 292–307.
- [246] R. Andrew Russell. “A ground-penetrating robot for underground chemical source location”. In: *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*. IEEE. 2005, pp. 175–180.
- [247] Wei Li, Jay A. Farrell, and Ring T. Card. “Tracking of fluid-advected odor plumes: strategies inspired by insect orientation to pheromone”. In: *Adaptive Behavior* 9.3-4 (2001), pp. 143–170.
- [248] Adam T. Hayes, Alcherio Martinoli, and Rodney M. Goodman. “Swarm robotic odor localization: Off-line optimization and validation with real robots”. In: *Robotica* 21.4 (2003), pp. 427–441.
- [249] Gervasio Varela, Pilar Caamamo, Felix Orjales, Álvaro Deibe, Fernando Lopez-Pena, and Richard J. Duro. “Swarm intelligence based approach for real time UAV team coordination in search operations”. In: *Nature and Biologically Inspired Computing (NaBIC), 2011 Third World Congress on*. IEEE. 2011, pp. 365–370.
- [250] Zhongyang Zheng and Ying Tan. “An Indexed K-D Tree for Neighborhood Generation in Swarm Robotics Simulation”. In: *Advances in Swarm Intelligence*. Vol. 7929. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2013, pp. 53–62.
- [251] Q. Tang and P. Eberhard. “A PSO-based algorithm designed for a swarm of mobile robots”. In: *Structural and Multidisciplinary Optimization* 44.4 (2011), pp. 483–498.
- [252] Songdong Xue, Yunlong Zan, Jianchao Zeng, Zhibin Xue, and Jing Du. “Group Decision Making Aided PSO-type Swarm Robotic Search”. In: *Computer, Consumer and Control (IS3C), 2012 International Symposium on*. IEEE. 2012, pp. 785–788.

- 
- [253] T. Schmickl and K. Crailsheim. “Trophallaxis within a robotic swarm: bio-inspired communication among robots in a swarm”. In: *Autonomous Robots* 25.1 (2008), pp. 171–188.
- [254] Ji-Gong Li, Qing-Hao Meng, Yang Wang, and Ming Zeng. “Odor source localization using a mobile robot in outdoor airflow environments with a particle filter algorithm”. In: *Autonomous Robots* 30.3 (2011), pp. 281–292.
- [255] James McLurkin, Andrew J Lynch, Scott Rixner, Thomas W Barr, Alvin Chou, Kathleen Foster, et al. “A low-cost multi-robot system for research, teaching, and outreach”. In: *Distributed Autonomous Robotic Systems*. Springer Berlin Heidelberg, 2013, pp. 597–609.
- [256] A. Gasparri and M. Prosperi. “A bacterial colony growth algorithm for mobile robot localization”. In: *Autonomous Robots* 24.4 (2008), pp. 349–364.
- [257] 安艺敬一 and P.G. 理查兹. 定量地震学：理论与方法. Vol. 1. 地震出版社, 1986.



## 第六部分

### 附录



# 附录 E 图表列表

## 插图

|     |                                                                                                  |    |
|-----|--------------------------------------------------------------------------------------------------|----|
| 1.1 | 烟花算法（FWA）的基本组成框架图。 . . . . .                                                                     | 10 |
| 2.1 | 烟花算法的框架图。 . . . . .                                                                              | 20 |
| 2.2 | 烟花产生爆炸火花示意图。 . . . . .                                                                           | 21 |
| 2.3 | 高斯变异的示意图。 . . . . .                                                                              | 22 |
| 2.4 | 烟花算法的流程图。 . . . . .                                                                              | 27 |
| 2.5 | 在 Generalized Rosenbrock 函数上，不同烟花总数对算法性能的影响。                                                     | 31 |
| 2.6 | 在 Generalized Rosenbrock 函数上，不同烟花个数对算法性能的影响。                                                     | 31 |
| 2.7 | 映射规则对火花位置的操作。 . . . . .                                                                          | 32 |
| 2.8 | FWA、CPSO 和 SPSO 在六个函数上的收敛曲线 . . . . .                                                            | 37 |
| 5.1 | 单峰搜索空间估计示意图 [87] . . . . .                                                                       | 66 |
| 5.2 | 在 1 维空间进行估计的示意图 . . . . .                                                                        | 67 |
| 5.3 | 样本点在解空间各个维度上投影及估计 . . . . .                                                                      | 69 |
| 5.4 | 各算法在测试集合上 $t$ 检验有效函数个数 . . . . .                                                                 | 72 |
| 5.5 | 加速型烟花算法（AcFWA）与基本烟花算法（FWA）在测试函数（ $f_1$ 、 $f_3$ 、 $f_5$ 、 $f_6$ 、 $f_8$ 、 $f_9$ ）上性能对比。 . . . . . | 76 |
| 6.1 | 不同参数的转移函数图形 . . . . .                                                                            | 78 |
| 6.2 | 烟花爆炸个数与迭代次数的曲线 . . . . .                                                                         | 82 |
| 6.3 | 烟花爆炸幅度与迭代次数的曲线 . . . . .                                                                         | 83 |

|      |                                                                                  |     |
|------|----------------------------------------------------------------------------------|-----|
| 6.4  | 对比四种算法的收敛曲线 . . . . .                                                            | 84  |
| 6.5  | 对比四种算法的收敛曲线 . . . . .                                                            | 85  |
| 7.1  | 爆炸半径检测策略 . . . . .                                                               | 89  |
| 7.2  | 爆炸火花产生方式对比 . . . . .                                                             | 90  |
| 7.3  | 使用原始烟花算法中的高斯变异算子产生的高斯火花分布图。 . . . .                                              | 92  |
| 7.4  | 基本烟花算法和增强烟花算法中高斯变异算子对比 . . . . .                                                 | 93  |
| 7.5  | 时间消耗对比。 . . . .                                                                  | 97  |
| 8.1  | CF 的爆炸半径的放大与缩小的示意图 . . . . .                                                     | 106 |
| 8.2  | CF 的爆炸半径的缩小和放大 (Sphere 函数上, [96], 参看第 8.5 节) .                                   | 109 |
| 8.3  | 验证高斯变异算子有效性 (在测试集合上符号秩检验有效的函数个数)                                                 | 111 |
| 8.4  | 运行时间比较 (dynFWA 运行时间为 1) . . . . .                                                | 115 |
| 9.1  | 在 Sphere 函数上的自适应半径 . . . . .                                                     | 119 |
| 9.2  | 计算自适应半径的一个例子 . . . . .                                                           | 120 |
| 9.3  | 爆炸的两种情况左: $\hat{s}$ 和 $s^*$ 处于烟花的不同侧右: $\hat{s}$ 和 $s^*$ 处于烟花的<br>同一侧 . . . . .  | 122 |
| 9.4  | 局部搜索 . . . . .                                                                   | 123 |
| 9.5  | 在第二阶段半径增长比例的直方图 . . . . .                                                        | 123 |
| 9.6  | 微调搜索 . . . . .                                                                   | 124 |
| 9.7  | 各个算法在 28 个函数上的平均误差 . . . . .                                                     | 126 |
| 9.8  | 三种算法在 28 个函数上消耗的时间 . . . . .                                                     | 127 |
| 10.1 | 高斯变异和差分变异对比图。 . . . .                                                            | 130 |
| 10.2 | FWA-DM 算法示意图。 . . . .                                                            | 131 |
| 10.3 | EFWA, FWA-DM 和 SPSO2011 对比结果 . . . . .                                           | 134 |
| 10.4 | 在函数 1 到 15 上 EFWA, FWA-DM 和 SPSO2011 的收敛曲线 . . . . .                             | 136 |
| 10.5 | 在函数 16 到 30 上 EFWA, FWA-DM 和 SPSO2011 的收敛曲线 . . . . .                            | 137 |
| 11.1 | 帕累托前端示意图。 . . . .                                                                | 148 |
| 11.2 | 在 BN 施肥问题是 CPU 时间消耗对比; (a) $300 \leq m \leq 800$ ; (b)<br>$m \leq 200$ . . . . . | 156 |
| 11.3 | 在 CA 施肥问题是 CPU 时间消耗对比; (a) $300 \leq m \leq 800$ ; (b)<br>$m \leq 200$ . . . . . | 156 |

|                                                       |     |
|-------------------------------------------------------|-----|
| 11.4 对比在六个 BN 施肥优化问题上 HV 结果, HV 取 30 次平均。 . . . .     | 157 |
| 11.5 对比在六个 BN 施肥优化问题上 HV 结果, HV 取 30 次平均。 . . . .     | 158 |
| 11.6 在 CO 施肥优化问题上 MOFWA 和 MORS 优化算法求解得到的解的分布。 . . . . | 161 |
| 12.1 用于求解 TSP 的离散烟花算法流程图 . . . . .                    | 171 |
| 12.2 2-Opt 局部搜索操作: 左侧为原始回路, 右侧为结果 . . . . .           | 172 |
| 12.3 3-Opt 局部搜索操作: 左侧为原始回路, 右侧两个为结果 . . . . .         | 173 |
| 12.4 $\theta, L_m/L_o$ 对 $p_a$ 的影响 . . . . .          | 176 |
| 12.5 2h-Opt 局部搜索操作: 左侧为原始回路, 右侧为结果 . . . . .          | 177 |
| 12.6 烟花数量对算法性能的影响 . . . . .                           | 181 |
| 12.7 最大火花数量对算法性能的影响 . . . . .                         | 182 |
| 12.8 d198 数据集, DFWA-na 和 DFWA-ri 的迭代对比曲线 . . . . .    | 184 |
| 12.9 d198 数据集的一个优化结果, 误差为 0.05% . . . . .             | 185 |
| 13.1 CPU 时钟频率变化趋势 . . . . .                           | 189 |
| 13.2 GPU 的晶体管更多的用于计算单元 [151] . . . . .                | 190 |
| 13.3 GPU 和 CPU 浮点计算能力对比 [151] . . . . .               | 191 |
| 13.4 GPU 图形流水线 . . . . .                              | 193 |
| 13.5 统一渲染架构结构图 (NVIDIA Tesla) . . . . .               | 196 |
| 13.6 传统渲染架构与统一渲染架构比较 . . . . .                        | 197 |
| 13.7 线程的层级结构 . . . . .                                | 202 |
| 13.8 内存的层级结构 . . . . .                                | 203 |
| 13.9 吸引-排斥变异 . . . . .                                | 209 |
| 13.10 $A$ 取不同值时 $E[x]$ 的变化趋势 . . . . .                | 209 |
| 13.11 不同均匀分布下的仿真结果 . . . . .                          | 210 |
| 13.12 GPU-FWA 算法流程图 . . . . .                         | 211 |
| 13.13 数组的结构体 . . . . .                                | 212 |
| 13.14 结构体的数组 . . . . .                                | 212 |
| 13.15 GPU-FWA 相对于 FWA 的加速比 . . . . .                  | 215 |
| 13.16 GPU-FWA 相对于 PSO 的加速比 . . . . .                  | 216 |
| 14.1 NMF 方法示意图。 . . . .                               | 224 |
| 14.2 NMF 计算过程中对于 $W$ 矩阵的行 $l$ 的优化过程。 . . . .          | 226 |

|      |                                                                               |     |
|------|-------------------------------------------------------------------------------|-----|
| 14.3 | 左边：在初始化矩阵 $W$ 矩阵的每一行后，平均估计误差；右边：在初始化 $H$ 矩阵的每一列后，平均估计误差。NMF 的秩为 5。 . . . . .  | 231 |
| 14.4 | 左边：在初始化矩阵 $W$ 矩阵的每一行后，平均估计误差；右边：在初始化 $H$ 矩阵的每一列后，平均估计误差。NMF 的秩为 10。 . . . . . | 231 |
| 14.5 | 只对矩阵 $W$ 的每一行进行更新的精度 . . . . .                                                | 232 |
| 14.6 | 在不同 $k$ 值上，只对矩阵 $W$ 的每一行进行更新取得相同的性能的运行时间对比（独立运行 30 次结果） . . . . .             | 233 |
| 15.1 | 烟花算法在 20-newsgroup 上的聚类效果 . . . . .                                           | 244 |
| 15.2 | 垃圾电子邮件智能检测算法模型 . . . . .                                                      | 246 |
| 15.3 | 利用烟花算法优化垃圾电子邮件智能检测算法参数的框架 . . . . .                                           | 248 |
| 15.4 | 采用策略 1 条件下优化前后性能对比 . . . . .                                                  | 249 |
| 15.5 | 采用策略 2 条件下优化前后性能对比 . . . . .                                                  | 250 |
| 16.1 | 多目标搜索问题在模拟程序中的截图 . . . . .                                                    | 261 |
| 16.2 | 在群体机器人中引入烟花爆炸策略 . . . . .                                                     | 263 |
| 16.3 | 分组爆炸策略流程图 . . . . .                                                           | 265 |
| 16.4 | 可拓展性实验结果，图中的数值表示算法在迭代次数和移动距离上性能的比值。 . . . . .                                 | 271 |
| 17.1 | 地层结构示意图 . . . . .                                                             | 274 |
| 17.2 | 地学反演求解流程图 . . . . .                                                           | 275 |
| 17.3 | 地震传播的射线理论 . . . . .                                                           | 278 |
| 17.4 | 优化收敛曲线 . . . . .                                                              | 279 |
| 17.5 | 最优结果对比 . . . . .                                                              | 280 |

## 表格

|     |                                                    |    |
|-----|----------------------------------------------------|----|
| 2.1 | 烟花算法高斯变异和无变异的结果 . . . . .                          | 32 |
| 2.2 | 实验中函数设置 . . . . .                                  | 35 |
| 2.3 | FWA 对比 CPSO 和 SPSO 的实验结果（精确到 $10^{-6}$ ） . . . . . | 36 |

|      |                                                                      |     |
|------|----------------------------------------------------------------------|-----|
| 4.1  | 实验中用到的随机数生成器 . . . . .                                               | 54  |
| 4.2  | 各个函数求解质量 (均值/方差) . . . . .                                           | 56  |
| 4.3  | 各个 RNG 在不同测试函数上的 Lose Rank 值 . . . . .                               | 57  |
| 4.4  | 各个 RNG 在不同测试函数上的 Win Rank 值 . . . . .                                | 58  |
| 4.5  | RNG 效率对比 (个/纳秒) . . . . .                                            | 59  |
| 5.1  | 各算法在测试集合上面的计算结果 (实验结果格式: 均值 (方差), 这里 $N$ 指基本烟花算法) . . . . .          | 75  |
| 7.1  | 几种比较算法描述 . . . . .                                                   | 96  |
| 7.2  | 基本测试函数的位置偏移描述 . . . . .                                              | 96  |
| 7.3  | FWA、SPSO、eFWA-x、EFWA 算法在测试函数上的实验结果 (SI 为偏移索引) . . . . .              | 99  |
| 7.4  | EFWA 和 FWA $t$ -test 实验结果 (加粗表明优化结果显著) . . . . .                     | 100 |
| 8.1  | SPSO2011, EFWA 和 dynFWA 在测试函数上的平均适应度值和平均排名. . . . .                  | 113 |
| 8.2  | dynFWA 对 EFWA 的 Wilcoxon 符号秩检验结果 (粗体值表示改进显著)。 . . . .                | 114 |
| 9.1  | 28 个函数上平均误差的平均排名 . . . . .                                           | 126 |
| 9.2  | T-test results on AFWA vs. EFWA . . . . .                            | 127 |
| 10.1 | 测试函数 . . . . .                                                       | 133 |
| 10.2 | FWA-DM 相对于 EFWA 的 $t$ -检验结果 . . . . .                                | 135 |
| 11.1 | 算法参数设置 . . . . .                                                     | 154 |
| 11.2 | 多目标进化计算方法在 BN 问题上的性能比较。 . . . .                                      | 163 |
| 11.3 | 多目标进化计算方法在 CA 问题上的性能比较。 . . . .                                      | 164 |
| 11.4 | 在 CO 施肥优化问题上 MOFWA 和 MORS 优化算法求解得到的解 . . .                           | 165 |
| 12.1 | oliver30 数据集上的实验结果, 除 DFWA 以外的其他数据引用自文献 [137], 最优路径长度为 420 . . . . . | 183 |
| 12.2 | att48 数据集上的实验结果, 除 DFWA 以外的其他数据引用自文献 [137], 最优路径长度为 33522 . . . . .  | 183 |

|                                                                          |     |
|--------------------------------------------------------------------------|-----|
| 12.3 d198 数据集上的实验结果，除 DFWA 以外的其他数据引用自 Thomas Stutzle [134]，最优路径长度为 15780 | 184 |
| 12.4 大规模测试集，ACS 测试数据引自 [131]                                             | 185 |
| 13.1 实验中用到的基准测试函数                                                        | 214 |
| 13.2 结果精度比较                                                              | 214 |
| 13.3 T 检验 p 值                                                            | 215 |
| 13.4 运行时间和加速比 (Rosenbrock 函数)                                            | 216 |
| 14.1 在数据集 DS-SPAM1 上的静态分类方法实验结果                                          | 234 |
| 14.2 在数据集 DS-SPAM2 上静态方法分类结果                                             | 235 |
| 14.3 在数据集 DS-SPAM1 上动态分类方法实验结果                                           | 236 |
| 15.1 人工数据集实验结果                                                           | 253 |
| 15.2 掌纹数据集实验结果                                                           | 254 |
| 15.3 手指静脉图像数据集实验结果                                                       | 254 |
| 15.4 部分较好的参数列表 ( $K_1 = 1$ )                                             | 256 |
| 16.1 算法的性能验证结果                                                           | 270 |
| 17.1 优化结果对比                                                              | 280 |
| A.1 第 2 章用到的测试函数                                                         | 319 |
| A.2 第 7 章用到的测试函数                                                         | 319 |
| A.3 CEC2005 的 25 个测试函数 [90]                                              | 320 |
| A.4 CEC2013 的 28 个测试函数 [96]                                              | 321 |
| A.5 CEC2014 的 30 个测试函数 [102]                                             | 322 |
| C.1 术语列表                                                                 | 327 |
| D.1 符号列表                                                                 | 330 |