

Discrete Particle Swarm Optimization
Algorithm and Its Application

离散粒子群优化 算法及其应用

郭文忠 陈国龙 著

清华大学出版社

Discrete Particle Swarm Optimization
Algorithm and Its Application

 **离散粒子群优化
算法及其应用**

郭文忠 陈国龙 著

清华大学出版社
北京

过去几十年来,如何运用优化技术解决工业工程中的各类问题已成为管理科学、运筹学、工业与系统工程等领域许多研究人员和工程实践人员的重要研究课题。鉴于实际工程问题的复杂性、约束性、非线性、多局部最小和建模困难等特点,寻找各种适合于工程实践需求的新型智能优化方法一直是许多学科的一个重要研究方向。粒子群优化(particle swarm optimization, PSO)算法是由 Eberhart 博士和 Kennedy 博士于 1995 年提出的一种新型群智能优化算法,与其他进化算法相比,其最吸引人的特征是易于实现和更强的全局优化能力。基于 PSO 算法是一种强有力且应用范围十分广泛的智能优化技术,它已被 IEEE Congress on Evolutionary Computation 国际年会作为一个独立的分支。PSO 算法自提出之后,引起了众多学者的极大关注,在短短几年内就成为一个研究热点并出现了大量的研究成果,大量实验结果也表明了 PSO 算法确实是一种有力的优化工具且具有强大的生命力。

近年来,作者及其科研团队一直致力于 PSO 算法的理论及应用研究,特别是离散 PSO (discrete PSO, DPSO)算法的构建及其应用,并在此基础上撰写了本书。本书内容是作者基于自身所主持和参与的科技部“973”计划项目、国家自然科学基金项目、国家“863”计划子课题以及教育部科技重点项目等的研究成果,吸纳了国内外许多具有代表性的研究成果,并融合了课题组近年来在国内外重要学术刊物和国际会议上发表的研究成果,力图体现国内外在这一领域的最新研究进展。本书可作为计算机科学、自动化科学、管理科学、控制科学等相关学科专业高年级本科生、研究生以及广大研究计算智能的科技工作者的参考用书。由于作者水平有限,书中难免有疏漏之处,对于本书的不足之处,恳请读者批评指正。

全书由 11 章构成,内容自成体系,各章节内容具体安排如下:第 1 章是绪论,主要介绍了基本 PSO 算法的原理机制及其发展现状,并着重介绍了 PSO 算法的三种常见离散化策略,阐述了 DPSO 算法的应用成果;第 2 章主要介绍了 PSO 算法在 TSP 优化问题中的应用;第 3 章刻画了一种基于表现型共享函数的多目标粒子群优化算法,继而在此基础上详细介绍其在多工作调度问题中的应用;第 4 章以多目标最小生成树问题为应用背景,给出了一种改进的计数算法,然后在上一章基于表现型共享函数的基础上,详细介绍了一种用于求解多目标最小生成树问题的新型 DPSO 算法的具体设计过程;第 5 章主要介绍了 PSO 算法在入侵检测数据特征选择中的应用;第 6 章主要阐述了 PSO 算法在入侵检测系统异常检测和误用检测中的具体应用;第 7 章主要介绍了 PSO 算法在网络安全态势感知中态势要素获取、理解以及预测等各步骤中的应用;第 8 章阐述了 PSO 算法在异构集群数据流分配问题

中的应用;第9章主要讨论了在无线传感器网络中基于PSO的拓扑生成及其容错拓扑控制机制,详细介绍了各种算法的具体设计过程;第10章介绍了基于PSO算法的无线传感器网络任务调度策略的设计;最后,第11章分别介绍了PSO算法在超大规模集成电路物理设计中的电路划分和布图/布局规划优化问题中的具体应用。其中,第1~3章和第5~11章由郭文忠完成,第4章由陈国龙完成。

最后,感谢清华大学出版社的大力支持。同时,对课题组内参与有关研究工作的陈羽中博士、程红举博士、陈振助教、林嘉雯助教、刘延华博士生、刘耿耿博士生以及涂雪珠、陈震亦、陈庆良、黄敏、洪玉玲、夏添、冯翔、翁谦、林宗明、陈仕涛、黄巧云、高洪垒、周柳鸿、尤秉煜、彭少君、陈锦珠、陈颖等硕士生表示衷心的感谢。最后,感谢科技部“973”计划项目(2006CB805904,2011CB808000)、国家“863”计划子课题(2008AA03A313-3)、国家自然科学基金项目(60673161,10871221,61103175,11141005)、福建省科技创新平台项目(2009J1007)、福建省自然科学基金重点项目(A0820002)和教育部科技重点项目(206073,212086)等对相关研究工作的资助。

作 者

2011年8月

于福州大学-福建省科学与工程计算重点实验室

第 1 章 绪论	1
1.1 引言	1
1.2 基本粒子群优化算法	2
1.2.1 粒子群优化算法的基本原理	2
1.2.2 基本粒子群优化算法模型	3
1.2.3 基本粒子群优化算法流程	4
1.2.4 参数分析与设置	4
1.3 粒子群优化算法的改进综述	5
1.3.1 基于惯性权值的改进	6
1.3.2 基于加速因子的改进	8
1.3.3 基于邻近群拓扑的改进	9
1.3.4 基于种群规模的改进	10
1.3.5 混合粒子群优化算法	11
1.4 粒子群优化算法的机理研究	11
1.5 粒子群优化算法的应用研究	12
1.6 离散粒子群优化算法	13
1.6.1 将速度作为位置变化的概率	13
1.6.2 直接将连续 PSO 用于离散问题的求解	13
1.6.3 重新定义 PSO 算法操作算子	14
1.7 DPSO 算法应用	15
1.8 DPSO 算法研究展望	16
参考文献	17
第 2 章 在 TSP 问题中的应用	22
2.1 引言	22
2.2 求解 TSP 问题的自适应粒子群优化算法	23
2.2.1 离散 PSO 算法	23
2.2.2 求解 TSP 问题的 PSO 算法设计	24

2.2.3	惯性权值在离散 PSO 算法中的作用	25
2.2.4	实验结果与分析	26
2.3	求解 TSP 问题的动态领域 PSO 算法	31
2.3.1	相关概念	31
2.3.2	TSP 问题的 PSO 操作	33
2.3.3	动态领域 PSO 算法的设计	33
2.3.4	实验结果及分析	34
2.4	求解 TSP 问题的 PSO-ACO 算法	35
2.4.1	模拟进化的蚁群算法	35
2.4.2	PSO-ACO 算法的设计思想及总体框架	40
2.4.3	实验结果与分析	40
	参考文献	41
第 3 章	在多 workflow 调度中的应用	43
3.1	引言	43
3.2	问题描述	44
3.2.1	多目标优化问题	44
3.2.2	求解多目标优化问题的基本方法	47
3.3	多目标 workflow 调度问题	49
3.4	基于表现型共享的多目标粒子群优化算法	50
3.4.1	基于表现型共享的适应度函数	50
3.4.2	算法的基本模型	51
3.4.3	算法步骤	52
3.4.4	算例测试与结果分析	52
3.5	求解多目标 workflow 调度问题的离散粒子群优化算法	60
3.5.1	算法基本模型	60
3.5.2	算法主要步骤	62
3.5.3	实验结果	62
	参考文献	71
第 4 章	在多目标最小生成树问题中的应用	73
4.1	引言	73
4.2	问题模型	74
4.2.1	MST 问题	74
4.2.2	mc-MST 问题	74
4.3	改进的计数算法	75
4.4	求解 mc-MST 问题的 NDPSO 算法	81
4.4.1	粒子的编码机制	81
4.4.2	粒子的适应度函数	82

4.4.3 粒子的更新公式	83
4.4.4 算法描述	83
4.4.5 收敛性分析	84
4.5 实验结果与分析	86
4.5.1 测试问题	86
4.5.2 结果与分析	87
参考文献	89
第 5 章 在入侵检测数据特征选择中的应用	90
5.1 引言	90
5.2 特征选择	91
5.3 基于 PSO 和相关性分析的特征选择算法	92
5.3.1 粒子编码模式	92
5.3.2 适应度函数	92
5.3.3 参数设置	93
5.3.4 算法描述	93
5.3.5 实验结果与分析	94
5.4 基于 PSO 和邻域约简模型的特征选择算法	96
5.4.1 邻域粗糙集	97
5.4.2 算法的具体设计	98
5.4.3 仿真实验	99
5.5 基于 PSO 和云模型的特征选择算法	101
5.5.1 云的概念	102
5.5.2 云的对象隶属度计算	103
5.5.3 算法的具体设计	103
5.5.4 实验结果与分析	105
参考文献	108
第 6 章 在入侵检测系统中的应用	110
6.1 引言	110
6.2 基于连续粒子群分类算法的误用检测	112
6.2.1 目前入侵检测产品存在的缺陷	112
6.2.2 分类算法	112
6.2.3 基于连续粒子群的分类算法	114
6.3 基于否定选择算法的异常检测	121
6.3.1 基于异常的入侵检测系统的缺陷	121
6.3.2 人工免疫与否定选择算法	121
6.3.3 修改的否定选择算法	124
6.4 混合的网络入侵检测引擎	133

6.4.1	引入混合方式的目的	133
6.4.2	混合方式	134
6.4.3	混合的入侵检测引擎的整体结构	135
6.4.4	仿真实验	137
	参考文献	138
第7章	在网络安全态势感知中的应用	141
7.1	引言	141
7.2	基于 PSO-FNN 的安全态势感知要素提取算法	142
7.2.1	相关算法	142
7.2.2	基于 PSO-FNN 的安全态势要素提取模型	144
7.2.3	基于 PSO-FNN 的安全态势要素提取方法	144
7.2.4	仿真实验与结果分析	147
7.3	基于 PSO-BPNN 的安全态势预测算法	151
7.3.1	基于 PSO-BPNN 的网络安全态势预测模型	151
7.3.2	基于 PSO-BPNN 网络安全态势预测方法	152
7.3.3	仿真实验	152
7.4	网络安全系统中的组态势感知研究	155
7.4.1	个体态势感知与组态势感知	155
7.4.2	基于 PSO 的聚类分析实验设计	156
7.4.3	算法流程	157
7.4.4	仿真实验	157
	参考文献	159
第8章	在异构集群数据流分配中的应用	162
8.1	引言	162
8.2	数据流分配算法	163
8.3	基于 PSO 的异构集群数据流自适应分配策略	165
8.3.1	问题建模	165
8.3.2	带动态反馈机制的数据流自适应分配模型	166
8.3.3	改进的粒子群优化算法	167
8.3.4	仿真实验结果与分析	170
8.4	动态联盟思想的引入	173
8.4.1	动态联盟思想	174
8.4.2	问题建模	174
8.4.3	算法描述	176
8.4.4	算法仿真与结果分析	179
	参考文献	182

第 9 章 在 WSN 拓扑控制中的应用	184
9.1 引言	184
9.2 基于度约束最小生成树的 WSN 分布式拓扑控制	185
9.2.1 网络模型与问题描述	186
9.2.2 求解 dc-MST 问题的 DPSO	187
9.2.3 分布式拓扑控制方案	189
9.2.4 仿真实验	190
9.3 基于二连通的 WSN 拓扑控制方案	192
9.3.1 网络模型及问题描述	193
9.3.2 求解 WSN 二连通拓扑结构的 DPSO 算法	195
9.3.3 仿真实验	196
9.4 基于 k -连通问题的 WSN 拓扑控制方案	197
9.4.1 相关工作	199
9.4.2 相关定义	200
9.4.3 集中式的 KTCPSO 算法描述	200
9.4.4 分布式 KLPSO 算法描述	202
9.4.5 算法的时间复杂度分析	203
9.4.6 仿真实验	203
参考文献	205
第 10 章 在 WSN 任务调度中的应用	208
10.1 引言	208
10.2 任务调度相关概念	210
10.3 WSN 任务分配动态联盟模型及其算法	211
10.3.1 问题描述	211
10.3.2 任务分配动态联盟模型的构建	212
10.3.3 求解动态联盟模型的 PSO 算法	213
10.3.4 实验结果与分析	216
10.4 带多 Agent 的 WSN 自适应任务调度策略	221
10.4.1 多 Agent 系统	221
10.4.2 基于多 Agent 的无线传感器网络体系结构及系统模型	221
10.4.3 基于多 Agent 的无线传感器网络自适应任务调度策略	225
10.4.4 仿真实验与结果分析	227
10.5 基于串行联盟的动态任务分配算法	228
10.5.1 串行联盟思想的引入	228
10.5.2 基于 DPSO 的联盟形成算法	230
10.5.3 基于串行联盟的任务分配体系结构	232
10.5.4 仿真实验	233
10.6 基于并行联盟的动态任务分配算法	238

10.6.1	引言	238
10.6.2	并行联盟概述	239
10.6.3	基于并行联盟的任务分配算法	240
10.6.4	基于并行联盟的任务分配体系结构	243
10.6.5	仿真实验	244
参考文献		249

第 11 章	在 VLSI 物理设计中的应用	252
11.1	引言	252
11.2	VLSI 设计概述	253
11.2.1	VLSI 设计流程	253
11.2.2	物理设计过程	254
11.3	单目标电路划分的离散 PSO 算法	255
11.3.1	相关工作	255
11.3.2	问题模型	255
11.3.3	算法描述	256
11.3.4	实验结果分析	260
11.4	单目标电路划分的混合 PSO 算法	263
11.4.1	算法的具体设计过程	263
11.4.2	实验结果与分析	266
11.5	多目标电路划分的离散 PSO 算法	269
11.5.1	相关工作	269
11.5.2	多目标划分问题模型	270
11.5.3	基于 DPSO 框架下的多目标划分算法	271
11.5.4	实验结果与分析	274
11.6	解决布图规划的 DPSO 算法	276
11.6.1	VLSI 布图模式与相关工作	276
11.6.2	问题描述	277
11.6.3	算法描述	278
11.6.4	实验结果与分析	281
11.7	解决布图规划的多目标 PSO 算法	282
11.7.1	采用整数序列编码的布图规划算法	282
11.7.2	采用序列对编码的布图规划算法	284
11.8	解决布图规划的协同多目标 PSO 算法	288
11.8.1	协同多目标算法概述	288
11.8.2	解决布图规划问题的协同多目标 PSO 算法	291
11.8.3	实验结果分析	293
参考文献		296

绪 论

1.1 引言

人工智能在经历了 20 世纪 80 年代整整 10 多年的繁荣之后,随着人类探索脚步的不断前进,其复杂性、非线性、系统性的问题越来越多地呈现在人们眼前,面对系统的复杂性,由于在方法论上始终没有改变和突破经典计算的思想,已经逐渐陷入了困境。与此同时,随着人们对生命本质的不断了解,生命科学却以前所未有的速度迅猛发展,使人工智能的研究开始摆脱经典逻辑计算的束缚,大胆探索起新的非经典计算模式。在这种背景下,社会性动物(如蚁群、蜂群、鸟群)的自组织行为吸引着越来越多的学者进入到这个领域,研究这些简单的个体如何通过协作呈现出如此复杂而奇妙的行为,同时通过计算机模拟来探索其中的可循规律,并用于指导和解决一些常规方法没有解决的传统问题以及实际应用中出现的新问题,这就产生了一种新型的智能计算技术,即所谓的“群智能(swarm intelligence, SI)”,或称为“群集智能”^[1,2]。

群智能算法与人工生命,特别是进化策略以及遗传算法(genetic algorithm, GA)有着特殊的联系,目前已引起了越来越多研究者的关注,并成为一个研究热点。在群智能中,群体指的是一组相互之间可以通过直接通信或间接通信从而能够达成合作以对分布式问题进行求解的主体。为此,在不存在集中控制和不提供全局模型的前提下,群智能为寻找复杂的分布式问题求解方案提供了研究基础和技术支持。粒子群优化(particle swarm optimization, PSO)算法和蚁群(ant colony optimization, ACO)算法是目前群智能研究领域的两种主要算法,前者是对鸟群觅食过程的模拟,而后者主要是对蚂蚁群落食物采集过程的模拟。

群智能算法仅涉及各种基本的数学运算,对计算机性能要求不高,是一种简单且易于实现的算法。群智能算法基于概率搜索的算法,只需目标函数的输出值,并不需要相关的梯度信息,这也是与多数基于梯度应用的优化算法不同的地方。虽然概率搜索算法通常采用较多评价函数,但与梯度方法及传统的演化算法相比,由于其所具有的分布式、自组织性、协作性、鲁棒性和实现简单等特点,使其在没有全局信息的情况下,为寻找复杂问题特别是典型的系统复杂性问题(如 NP 问题等)的求解方案提供了快速、可靠的基础,为人工智能、认知科学等领域的基础理论问题的研究开辟了新的途径,同时也为诸如实际优化、工业动态生产

控制、工程设计优化、数据挖掘、电子系统优化等实际工程问题提供了新的解决方案。因此,无论从理论研究还是应用研究的角度出发,群智能理论及其应用研究都是具有重要学术意义和现实价值的,它也越来越受到国际智能计算等相关研究领域学者的关注,逐渐成为一个新的重要的研究方向。

1.2 基本粒子群优化算法

1995年,Kennedy博士和Eberhart博士受自然界中鸟群运动模型的启发提出了一种新的基于种群的搜索算法——PSO算法^[3]。在PSO算法中,它把所求解问题空间中可能解的位置看成是鸟群运动模型中的栖息地,然后通过个体之间的信息交互,逐步提高在求解过程中发现较好解的可能性,并指引群体中所有的粒子朝着可能解的位置不断聚集。PSO算法相对于其他进化算法的最大优势在于实现简单和具有更强的全局优化能力。为此,算法一经提出,立刻引起了学术界的广泛关注,并在短短的几年时间里出现大量的研究成果,形成了一个研究热点。大量实验结果表明,PSO算法能够解决GA所能解决的各类优化问题,也显示出PSO算法确实具有强大的生命力^[4]。

1.2.1 粒子群优化算法的基本原理

PSO算法的思想来源于社会认知理论,体现了群智能的特征——简单智能的主体通过简单合作表现出复杂智能行为。根据社会认知理论,可以将PSO算法的求解概括为评价、比较和模仿三个过程^[2]:

(1)评价过程 对各种激励进行评估,根据正反馈或负反馈、吸引或排斥等激励特性对其进行排序。这种行为特征在自然界中普遍存在,有机生命只有通过对外部激励的评价,才能完成对环境的学习。PSO算法中对环境的自适应学习过程也包含了这种评价过程。

(2)比较过程 指粒子群中的个体对其他个体标准进行测量,并与自身条件相比较,从而确立自身的学习和修正动机。

(3)模仿过程 粒子群中的个体根据自身判断模拟其他个体行为的过程,这是自然界中普遍存在的一种非常有效的学习模式,通常只模拟那些优于自身的个体。

通过以上三个过程的有机结合即可构成能适应复杂环境变化、解决困难问题的简单社会个体,并以计算机程序的形式实现。

在PSO算法中,可以将种群中的每个个体看成是在寻优空间中的一个没有质量,也没有体积的粒子,个体在搜索空间中以一定的速度飞行,根据个体与群体的飞行经验的综合分析结果动态调整飞行速度,逐渐向问题空间的更好区域移动。可以用粒子的位置来表示被优化问题在搜索空间中的潜在解,粒子飞翔的方向和距离可以通过粒子的速度来加以控制,而每个粒子的优劣则可以通过一个适应度值(fitness value)函数来加以评价。PSO算法初始化为一群随机粒子,然后通过追随当前最优粒子进行不断迭代的搜索直至找到最优解或有效解。在每一次迭代过程中,算法主要是通过跟踪个体极值 $pbest$ 和全局极值 $gbest$ 来更新各个粒子。其中,个体极值是目前粒子本身所找到的最优解,而全局极值则是整个种群目前所找到的最优解。

文献[5]指出 PSO 算法与其他进化算法之间存在许多不同之处。首先,GA 等一些进化算法主要是通过交叉(crossover)和变异(mutation)等遗传操作来控制算法的运作,而 PSO 算法则是通过粒子的速度来决定粒子的搜索过程,从而 PSO 算法不但要维护位置信息,还要维护飞行矢量。其次,PSO 算法具有记忆能力这一重要特点,算法中的每个粒子可以保存各自搜索到的最好位置,所得到的新的全局最优位置虽然对历史记录毫无影响,但却会影响其飞行矢量,从而使搜索和记忆在某种程度上分离^[6]。最后,PSO 算法还拥有与 GA 等其他进化算法不同的信息共享机制。前者是通过 *gbest* 和 *pbest* 这两个极值来传递信息给其他粒子,这是一个通过追随当前最优粒子的单向信息流动过程,而后者则主要是通过染色体(chromosomes)来实现,从而是一个整个种群比较均匀地向最优区域进行移动的过程。

1.2.2 基本粒子群优化算法模型

PSO 算法最初应用于连续空间的优化,在连续空间中,算法的数学描述如下:

设粒子群的种群规模为 M ,决策空间 n 维,其中粒子 i 在时刻 t 的坐标位置可以表示为 $X_i^t = (x_{i1}^t, x_{i2}^t, \dots, x_{in}^t)$, $i=1, 2, \dots, M$, 粒子 i 的速度定义为每次迭代中粒子移动的距离,用 $V_i = (v_{i1}^t, v_{i2}^t, \dots, v_{in}^t)$ 表示,则粒子 i 在时刻 t 的第 j ($j=1, 2, \dots, n$) 维子空间中的飞行速度和位置根据下式进行调整:

$$v_{ij}^t = wv_{ij}^{t-1} + c_1 r_1 (p_{ij} - x_{ij}^{t-1}) + c_2 r_2 (g_j - x_{ij}^{t-1}) \quad (1.1)$$

$$v_{ij}^t = \begin{cases} v_{\max}, & v_{ij}^t > v_{\max} \\ -v_{\max}, & v_{ij}^t < -v_{\max} \end{cases} \quad (1.2)$$

$$x_{ij}^t = x_{ij}^{t-1} + v_{ij}^t \quad (1.3)$$

其中, w 为惯性权值(inertia weight), c_1 和 c_2 为加速因子(acceleration constants), r_1 和 r_2 是在 $[0, 1]$ 范围内的两个随机数,通常使用一个常量 $V_{\max}$, 即最大速度来限制粒子的速度,以改善搜索结果。 g_j 既可以表示整个群体中的历史最优位置记录,即全局极值 *gbest*, 也可以是局部粒子群的历史最优位置。此时 g_j 可改为 l_j , 即表示局部极值 *lbest*, 公式中是采用全局极值 *gbest* 还是局部极值 *lbest* 主要是由 PSO 算法的邻近拓扑结构来决定,具体分析可见 1.3.3 节,而 p_{ij} 则是当前粒子的历史最优位置记录,即个体极值 *pbest*。

粒子的运动过程由公式(1.1)~公式(1.3)共同作用,粒子在运动过程中的速度增量与其自身的历史飞行经验和群体飞行经验紧密相关,并受最大飞行速度的限制。以二维空间为例,图 1.1 描述了 PSO 算法中粒子根据公式(1.1)~公式(1.3)移动的原理。

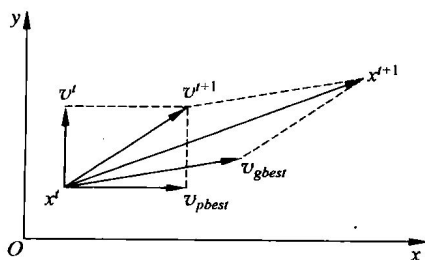


图 1.1 粒子运动原理

v^t —当前速度; v^{t+1} —改进后的速度; x^t —目前搜索到的点; x^{t+1} —改进后的搜索点;
 v_{pbest} —基于个体极值的速度; v_{gbest} —基于全局极值的速度

从公式(1.1)可以看出,粒子的速度更新模型主要由以下三个部分组成^[2]:①第一部分为粒子以前的速度对粒子飞行轨迹的影响,主要表示粒子对当前自身运动状态的信任,它主要是一个通过对粒子前一时刻的速度乘一个称为“惯性权值”的控制因子来实现惯性运动的过程;②第二部分为“认知(cognition)”部分,表示粒子自身经验对粒子运动轨迹的影响,也就是对粒子本身的思考,具体体现在粒子自身最优位置与当前位置之间的距离;③第三部分为“社会(social)”部分,表示群体经验对粒子运动轨迹的影响,也就是粒子之间的信息共享与相互合作,具体体现在群体最优位置与粒子当前位置之间的距离。后面这两个部分主要是通过两个称为“加速因子”的控制因子来进行调整。

1.2.3 基本粒子群优化算法流程

基本的粒子群优化算法流程可具体描述如下^[2]:

Step 1: 初始化所有粒子(群体规模为 M),在允许范围内随机设置粒子的初始位置和速度,每个粒子的局部最优解 $pbest$ 设为其初始位置, $pbest$ 中的最好值设为 $gbest$;

Step 2: 评价每个粒子的适应值,根据适应度函数计算每个粒子的目标函数;

Step 3: 更新每个粒子所经过的最好位置 $pbest$;

Step 4: 更新群体所经历过的最好位置 $gbest$;

Step 5: 根据公式(1.1)~公式(1.3)更新当前粒子的速度和位置;

Step 6: 检查终止条件(通常为达到最大迭代次数,或者取得足够好的适应值,或者最优解停滞不再变化)是否满足,如果满足则终止迭代,否则返回 Step 2。

基本 PSO 算法流程图如图 1.2 所示。

1.2.4 参数分析与设置

在 PSO 算法中存在几个显参数和隐参数,它们的值可以被调整,以产生算法搜索问题空间的方式变化。算法参数分析是 PSO 算法研究中的重要问题之一,许多研究人员对参数的选择及其对算法性能的影响进行了大量的分析和实验,为 PSO 算法的理论和应用研究奠定了坚实的基础。在基本 PSO 算法中,需要调节的参数主要有种群规模 M 、最大速度 $V_{\max}$ 、惯性权值 w 以及加速因子 c_1 和 c_2 等。

1. 种群规模

• 一般情况下,种群规模在 20~40 区间取值就能保证对解空间进行充分的搜索,而且对于大部分问题,种群规模取 10 就足以取得较好的结果,但对于一些特定类别或者比较难的问题,种群规模甚至有时需要在 100~200 区间取值。

2. 最大速度

最大速度 $V_{\max}$ 决定当前位置与最好位置之间区域的分辨率(或精度)。如果 $V_{\max}$ 太大,

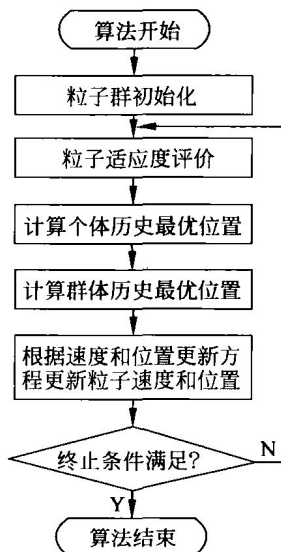


图 1.2 基本粒子群优化算法流程图

粒子可能会飞过好解;如果 $V_{\max}$ 太小,粒子不能在局部好区域之外进行足够的探索,导致容易陷入局部最优值。 $V_{\max}$ 通常设定为粒子的范围宽度,例如,粒子 (x_1, x_2, x_3) , x_1 属于 $[-10, 10]$,那么 $V_{\max}$ 的大小就是 10。

3. 惯性权值

在公式(1.1)中,如果没有后面两个部分,即 $c_1=c_2=0$,粒子将一直以当前的方向飞行,直到到达边界。由于它只能搜索有限区域,所以很难找到较好的解,而加上后面两部分之后,粒子则有扩展搜索空间的趋势。

性质 1.1^[7] 惯性权值的设置影响了粒子的全局搜索能力与局部搜索能力之间的平衡。

从速度更新模型的公式(1.1)可以看出,公式的第一部分是 PSO 算法中的一个关键部分,它主要提供了粒子在搜索空间中飞行的动力,并表示了粒子以前的速度对粒子飞行轨迹的影响,而惯性权值就是一个用来表征粒子以前经历过的速度对当前速度的影响程度的数字量。因此,惯性权值的设置影响了粒子的全局搜索能力与局部搜索能力之间的平衡。

性质 1.2^[7] 使用较大的惯性权值,算法具有较强的全局搜索能力。

从公式(1.1)和性质 1.1 可知,惯性权值是决定保留多少上一时刻的速度,从而取较大的值可以加强搜索以前所未能达到区域的能力,有利于增强算法的全局搜索能力并跳出局部极小点。而取较小的值则说明当前算法主要是停留在当前解的附近搜索,从而有利于提高算法的局部搜索能力,并加速算法的收敛。

因此,该参数对算法的性能影响很大,其大小控制着以前速度对当前速度的影响,体现了全局搜索与局部搜索的一个折衷^[8],对 PSO 算法的收敛性起着至关重要的作用。目前对惯性权值 w 的调整策略主要有线性变化、模糊自适应和随机变化等。其中应用最多的是线性递减策略,Shi 认为这样可以在开始优化时搜索较大的解空间,得到合适的粒子,然后在后期逐渐收缩到较好的区域进行更精细的搜索,以加快收敛速度^[9]。

4. 加速因子

加速因子 c_1 和 c_2 是一组调整粒子自身经验与群体经验影响粒子运动轨迹的重要参数。如果 c_1 值为 0,则粒子仅有群体经验作用于粒子的运动,这时它的收敛速度可能较快,但对一些复杂问题可能容易导致局部收敛;如果 c_2 值为 0,则仅有自身经验对粒子的运动起作用,群体中的粒子之间不具备信息交互的能力,那么,一个规模为 M 的群体就等同于运行了 M 次单个粒子,这就失去了群智能算法本身所具备的特性,从而难以得到最优解;如果 c_1 和 c_2 均为 0,则粒子不包含任何的经验信息并只能搜索有限区域,从而难以找到较好的解。

可以把加速因子 c_1 和 c_2 看成一个控制参数,即 $\varphi=c_1+c_2$ 。如果 $\varphi=0$,粒子的坐标值 X 只是简单的线性增加。如果 φ 非常小,则对粒子速度的控制很微弱,因此群体的运动轨迹随时间推移而变化得非常缓慢。当 φ 较大时,粒子的空间位置变化频率增大,粒子变化步长也随之增大。一般情况下,当 $\varphi=4.1$ 时,具有较好的收敛效果。

1.3 粒子群优化算法的改进综述

在现实生活中,不同领域的优化问题不但种类极其繁多,而且具备各自不同的特点。为此,PSO 算法自提出至今,引起了学术界的广泛关注并开展了大量的研究工作,这也有力地

推动了该算法的发展。目前对 PSO 算法的改进研究主要体现在算法的参数选择与设计、领域拓扑结构、群体组织与进化、融合进化计算等概念的混合算法等几个方面,然而正如文献[10]中的“没有免费的午餐”这一理论所认为的,这些改进算法都需要根据具体问题的领域知识等对算法参数进行相应的设置,从而在提高某种性能的同时也付出其他代价。

1.3.1 基于惯性权值的改进

从前面的分析可以知道,公式(1.1)的第一部分主要提供了粒子在搜索空间飞行的动力,其中的惯性权值用来表征粒子以前经历过的速度对当前速度的影响程度,于是,惯性权值的设置影响了粒子的局部搜索能力与全局搜索能力之间的平衡^[6,7]。

这里可以通过令 $c_1 = c_2 = 0$ 来简要解释惯性权值 w 的效果。假设初始速度非零,则当 w 大于 1.0 时,算法将使粒子的速度不断增大直至速度达到最大速度 $V_{\max}$ 为止;而当 w 小于 1.0 时,算法将使粒子的速度不断减小直至粒子最终速度为 0,停止粒子的运动过程。Shi 和 Eberhart 在文献[11]中研究了 w 在范围 $[0, 1.4]$ 内的影响,并声称选择 $[0.8, 1.2]$ 范围内的 w 值会导致收敛加速,而选择大于 1.2 的 w 值则会导致收敛失败。

基于惯性权值的设置对 PSO 算法的性能起着至关重要的作用,目前已有很多研究者针对惯性权值的改进研究开展了一系列工作,并对算法作了不同程度的改进^[7,8,11-15],同时也取得了一些研究成果。

1. 惯性权值线性递减

Shi 和 Eberhart 在文献[11]中所取得的研究成果,对于洞察惯性权值在 PSO 算法中的运作机理起到了一定的作用,在此研究的基础上,他们继而提出了一种基于惯性权值线性递减的 PSO 算法^[8,12]。该算法为了保证算法的全局搜索能力,在算法运行初期使用较大的惯性权值,而为了保证算法的局部搜索能力,则在算法运行后期使用较小的惯性权值,惯性权值的具体计算公式如下:

$$w = (w_1 - w_2) \times \frac{MaxIter - CurIter}{MaxIter} + w_2 \quad (1.4)$$

其中, w_1 和 w_2 是惯性权值的初始值和最终值, $CurIter$ 和 $MaxIter$ 是算法的当前代数和最大迭代次数。

为验证该方法的有效性,文献中使用了 4 个不同的基准函数进行模拟仿真实验,发现这种参数调整方法确实提高了算法的性能和收敛速度,并且在 $w_1 = 0.9$ 和 $w_2 = 0.4$ 时,算法的前 1500 次迭代效果更好。然而,该方法在解决多峰值函数问题时却容易陷入局部最优。

2. 模糊惯性权值

在文献[13,14]中,Shi 和 Eberhart 提出了一种利用模糊控制系统对惯性权值进行自适应调整的方法,其基本思想就是利用模糊控制系统将对一个问题的语言描述转化成为一个具体的模型,然后通过数字输入来描述变量该如何调整^[15]。

Shi 和 Eberhart 把当前惯性权值以及与函数值所对应的目前为止发现的最优解 $f(\hat{y})$ 作为模糊控制系统的两个输入,之后根据该系统所得到的输出结果来对惯性权值进行动态修正。由于不同范围的问题具有不同的函数值,Shi 和 Eberhart 采用了公式(1.5)的规划方法对函数值 $f(\hat{y})$ 进行规范化:

$$f_{\text{norm}}(\hat{y}) = \frac{f(\hat{y}) - f_{\min}}{f_{\max} - f_{\min}} \quad (1.5)$$

其中, $f_{\max}$ 和 $f_{\min}$ 值与具体问题有关, 须事先获知或能够预先估计值的大小。

为了对应输入变量所属的低、中、高三种模糊设置, Shi 和 Eberhart 使用了以下 3 个隶属度函数:

$$f_{\text{left_triangle}}(x) = \begin{cases} 1, & x < x_1 \\ \frac{x_2 - x}{x_2 - x_1}, & x_1 \leq x \leq x_2 \\ 0, & x > x_2 \end{cases} \quad (1.6)$$

$$f_{\text{triangle}}(x) = \begin{cases} 0, & x < x_1 \\ 2 \frac{x - x_1}{x_2 - x_1}, & x_1 \leq x \leq \frac{x_2 + x_1}{2} \\ 2 \frac{x_2 - x}{x_2 - x_1}, & \frac{x_2 + x_1}{2} \leq x \leq x_2 \\ 0, & x > x_2 \end{cases} \quad (1.7)$$

$$f_{\text{right_triangle}}(x) = \begin{cases} 1, & x < x_1 \\ \frac{x - x_1}{x_2 - x_1}, & x_1 \leq x \leq x_2 \\ 0, & x > x_2 \end{cases} \quad (1.8)$$

其中, x_1 和 x_2 为决定函数形状位置的重要参数。

为了说明所提算法的有效性, Shi 和 Eberhart 通过与线性递减惯性权值的 PSO 算法的实验结果进行比较, 发现在确定的参数设置情况下, 该算法对某些测试函数具有更好的效果。然而, 该算法虽然在单峰值函数的表现相当优异, 但对于多峰值函数, 却难以找到最佳的惯性权值, 并且模糊控制系统也难以区分算法的状态到底是陷入局部最小值还是靠近单峰值函数的最小值。此外, 这种方法中的 $f_{\max}$ 和 $f_{\min}$ 值事先也很难预知, 从而实现起来比较困难。

3. 随机惯性权值

现实中所存在的许多环境随时间不断改变的动态优化问题需要构建具有一定非线性搜索能力的算法, 为此, 文献[16]提出了一种基于随机惯性权值方法的 PSO 算法, 算法中的惯性权值按照下式进行随机调整:

$$w = 0.5 + \frac{\text{rand}()}{2} \quad (1.9)$$

其中, $\text{rand}()$ 为 0~1 之间的一个随机数。

为了说明这种参数调整策略的有效性, 文献[16]采用了一些著名的基准函数进行测试, 实验结果表明, 该策略在算法初期确实能够加速粒子的收敛, 且对大部分优化函数都能找到相当好的解。

4. 惯性权值自适应调整策略

上述的改进算法确实能够提高算法的性能, 并在单峰问题的测试中表现优异, 但在多峰函数的测试中却出现算法容易过早收敛的现象。另外, 上述所有的算法在同一代种群的速