

大连理工大学

---

硕士学位论文

---

多目标优化的粒子群算法及其应用研究

---

姓名：陈绍新

---

申请学位级别：硕士

---

专业：软件工程

---

指导教师：周宽久

---

20071118

## 摘 要

粒子群算法(PSO)是近几年发展起来的解决多目标优化问题的群智能算法。该算法基于这样的假设:群体中的各个粒子能够从其过去的经历和其它粒子的经历得到有效的信息。实验发现,对于大多数优化问题,其有更快的收敛速度,需要设置的参数较少等特点,但它们在解集分布性、收敛性方面仍存在不足,涉及到的实际应用较少。本文对多目标粒子群算法进行了系统研究,在前人研究的基础上从三个方面改进了多目标粒子群算法,并将改进的多目标粒子群算法应用到营养配餐计算模型的求解过程之中,本文的工作主要有以下几个方面:

1. 总结了多目标优化的传统解决方法和基于进化算法计算的解决算法,并重点介绍了粒子群算法及其在多目标优化领域的研究现状。

2. 设计了一种新的多目标粒子群求解方法,主要从三个方面对算法进行了改进和优化:(1)在粒子全局最优值选取过程中,先后采用了拥挤机制和禁忌算法,从而使全局最优值的选取更为合理,避免了算法过早陷入局部最优,保持了解的分布性;(2)在处理约束条件方面,传统的处理方法大多只考虑粒子所在区域,致使边界值处理精度不高,本文引入半可行域的概念,进而改进了适应度函数的求解方式,克服了传统的约束处理存在的问题;(3)在时间复杂度方面,传统的构造非支配集的方法时间复杂度较高,本文采用了随机选择策略交换分组的非支配集构造方法,提高了构造非支配集的效率,减少了算法的时间复杂度。

3. 针对传统的营养配餐模型多为经济目标函数,主要考虑食物价格因素的特点,提出了新的营养配餐模型,能在满足人体各种营养需求的基础上,满足各类人群的需求,具有种类多样化、灵活化的特点。

最后,将提出的新的多目标粒子群算法应用到新的营养配餐模型中,针对营养配餐的特点设计并实现了基于 Web 的新的营养配餐决策支持系统。通过与传统营养配餐模型和求解方式的实验结果进行比对,验证了本文提出的新的多目标粒子群算法的可行性和有效性,拓展了新的多目标粒子群算法的应用领域。

**关键词:** 多目标优化; 多目标粒子群算法; 营养配餐计算模型

## Research on Multi-Objective Optimization Particle Swarm Algorithm and its Application

### Abstract

Particle Swarm Optimization algorithm is developed in the past few years to resolve the issue of multi-objective optimization algorithm intelligent group. The algorithm is based on this assumption: the various groups of particles can get effective message from past experience and the experience of other particles. The experiments was found that for the majority of optimization problems this algorithm has many features such as a faster convergence rate, the less the need for parameters and so on. But it remains inadequate in the distribution and convergence of its solution set. Its applications are less. From some current research, this paper designed a hybrid improved Particle Swarm Algorithm from three phrases, and applied to the Nutrition Catering Computing Model. The work in the dissertation is shown as follows:

1. Summing up the multi-objective optimization solutions of the traditional methods and evolutionary algorithm based on the solution algorithm, and focuses on the particle swarm algorithm and its multi-objective optimization in the field of the status quo.

2. This paper designed a new multi-objective PSO to improve the algorithm from three phrases: (1) It joined tabu algorithm and Crowded mechanism in selecting of the optimal value of the best, so that it avoided getting into the optimizing locality earlier and improved the solution precision. (2) A solution of an advanced fitness function, which introduced the conception of feasible zone to treat with the restrained conditions. (3) It put forward the non-dominated set: a strategy about the random choice in label switching. It improved the structural efficiency of the non-dominated set and reduced the algorithm's time complexity.

3. It designed a new nutrition Catering model which can meet all nutritional needs of the human body and the needs of the various type of the crowd. This model is species diversity and flexible.

In the end, the new nutrition Catering model is designed and realized with the new multi-objective PSO according to the characteristics. By comparison with the traditional way to solve the model and experimental results, it proved feasibility and effectiveness, expanding a new multi-target particle swarm algorithm applications.

**Key Words:** Multi-objective optimization; multi-objective PSO; nutrition Catering model

## 独创性说明

作者郑重声明：本硕士学位论文是我个人在导师指导下进行的研究工作及取得研究成果。尽我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写的研究成果，也不包含为获得大连理工大学或者其他单位的学位或证书所使用过的材料。与我一同工作的同志对本研究所做的贡献均已在论文中做了明确的说明并表示了谢意。

作者签名： 陈锡新 日期： 2007.12.15

## 大连理工大学学位论文版权使用授权书

本学位论文作者及指导教师完全了解“大连理工大学硕士、博士学位论文版权使用规定”，同意大连理工大学保留并向国家有关部门或机构送交学位论文的复印件和电子版，允许论文被查阅和借阅。本人授权大连理工大学可以将本学位论文的全部或部分内容编入有关数据库进行检索，也可采用影印、缩印或扫描等复制手段保存和汇编学位论文。

作者签名： 陈绍新

导师签名： 周竟久

2007年12月15日

# 1 绪 论

## 1.1 研究背景和课题意义

优化是个古老而且困难的课题，它所研究的问题是在众多方案中寻找最优方案，优化处理的是具有多个变量且通常需要服从等式和(或)不等式约束的最小化或最大化函数问题<sup>[1]</sup>。随着高速数字计算机日益广泛应用，使优化问题的研究不仅成为一种迫切需要，而且有了求解的有力工具。因此，优化理论和算法迅速发展起来，形成一门新的学科。

多目标优化是优化问题的主要研究领域之一。自 20 世纪 60 年代以来，多目标优化问题(MOP)吸引了越来越多不同背景研究人员的注意，这是因为多目标优化问题在现实生活中具有非常普遍和重要的地位。在同样条件下，经过优化技术的处理，对系统效率的提高、能耗的降低、资源的合理利用及经济效益的提高等均有显著的效果。如工厂的最低成本和最大效益问题就是一个典型的多目标优化问题，此外还有社会发展与国民经济的中长远发展计划的优化与决策问题等。一般来说，科学与工程实践中的优化问题大都是多目标优化与决策问题。而这些实际问题非常复杂、困难，要解决这类问题需要投入更多的精力。因此，解决多目标优化问题是一个非常具有科研价值和实际意义的课题。

随着人类生存空间的扩大以及认识世界和改造世界范围的拓宽，常规方法如评价函数法、分层序列法等已经无法处理人们所面对的复杂问题，因此高效的优化算法成为科学工作者的研究目标之一。

自然界生物体通过自身的演化来适应周围的环境从而不断地向前发展，进化算法就是基于这种思想逐步发展起来的一类随机搜索技术，是对生物进化过程进行的一种数学仿真，它们是模拟由个体组成的群体的集体学习过程。进化算法的出现为那些难以找到传统数学模型的难题指出了一条新的出路，这对于多目标优化这个领域同样如此，因为进化算法具有求解多目标优化问题的优点，受到了相当大的关注，这就导致了一类新的研究和应用，称作多目标进化优化。

进化算法已被用于解决复杂的单目标和多目标优化问题，本文研究的粒子群优化算法(particle swarm optimization)是在 1995 年由美国社会心理学家 Kennedy 和电气工程师 Eberhart 共同提出的，源于对鸟群觅食过程中的迁徙和聚集的模拟。它收敛速度快、易于实现且仅有少量参数需要调整，因而一经提出就成为进化计算领域的一个新的研究热点，目前已经被广泛应用于目标函数优化、动态环境优化、神经网络训练等许多领域。

其中最具应用前景的领域包括多目标问题的优化、系统设计、分类、模式识别、决策和模拟等。

多目标粒子群优化算法是一种新兴的多目标优化算法,和传统的多目标优化方法及其它进化优化算法比较起来其具有以下优点:

- (1) 在编码方式上 PSO 算法比较简单,可以直接根据被优化问题进行实数编码;
- (2) 对种群的初始化不像其它进化算法那样敏感,可达到较快的收敛速度;
- (3) 算法具有一般性的特点,适用于绝大多数的多目标优化问题;
- (4) 在优化过程中,每个粒子通过自身经验与群体经验进行更新,具有学习和记忆的功能;
- (5) 和遗传算法相似,搜索过程也是从一组解迭代到另一组解,采用同时处理群体中多个个体的方法,具有本质的并行性。

## 1.2 国内外的研究情况

多目标最优化(MOP)也称为多标准优化,多绩效或向量优化问题。正是由于多目标问题的广泛存在性和求解的困难,该问题一直富有吸引力和挑战性。它的最早出现,应追溯到 1772 年的 Franklin 提出的多目标矛盾如何协调的问题,国际上一般认为多目标最优问题最早由法国经济学家 V.Pareto 在 1896 年提出<sup>[2]</sup>。当时他从政治经济学的角度,把很多不好的目标归纳成多目标最优化问题。此后这方面的工作方兴未艾<sup>[3]</sup>: 1944 年, Von.Neumann 和 J.morgenstern 从对策论的角度,提出多个决策者而且彼此又相互矛盾的多目标决策问题; 1951 年, T.C.Koopmans 从生产与分配的活动分析中提出了多目标最优化问题,并且第一次提出了 Pareto 最优解的概念; 同年, H.W.T.Tucker 从数学规划的角度,给出了向量极值问题的 Pareto 最优解的概念,并且研究了这种解的充分与必要条件; 1953 年, Arron 等人对凸集提出了有效点的概念,从此多目标规划逐渐受到了人们的关注; 1963 年, L.A.Zadeh 又从控制论的角度提出多目标控制问题; 这期间 Charnes、Karlin、Klinger、Polak、Keeney、Geoffrion 等人先后都做出了较有影响的工作; 1968 年, Z.Johnsen 系统的提出了关于多目标决策问题的研究报告,这是多目标最优化这门学科开始大发展的一个转折点。到目前为止,已有超过 30 个数值计算技术用于解决该类问题。

多目标最优化问题从 Pareto 正式提出到 Johnsen 的系统总结,先后经历了六七十年时间。但是,传统的数学规划与模拟退火算法是以单点搜索为特征的串行算法,不可利用 Pareto 最优概念对解进行评估。研究多目标优化问题一直缺乏一种高效实用的求解

方法。好在九十年代开始流行的进化计算为求解多目标优化问题提供了有力的工具。进化算法搜索解的一个最大特点是它是一个群体搜索算法,很自然地就解决了上述传统的多目标算法的并行化困难的问题,有效的利用 Pareto 最优解也变得非常合理。并且进化算法对 Pareto 解集的诸如形状、连续性等性质并不敏感,而这两点却在实际工作中经常遇到,且用传统的数学方法很难处理。1967 年 Rosenberg 在其博士论文中曾提出可用基于遗传的搜索算法来求解多目标的优化问题,但直到 1985 年才出现基于向量评估的遗传算法<sup>[4]</sup>(VEGA),这是第一个多目标进化算法,真正引起演化界重视的是 1990 年后,相继提出了不同的多目标进化算法, Fonseca 和 Fleming 提出多目标遗传算法(MOGA), Srinivas 和 Deb 提出非劣分层遗传算法(NSGA), Horn 和 Nafpliotis 也提出了小组决胜遗传算法(NPGA),这些算法已经成为多目标进化算法研究的基石。90 年代后期,在学者总结了前人的经验教训之后,又提出了 PAES<sup>[5]</sup>、SPEA<sup>[6]</sup>、NSGA2<sup>[7]</sup>、SPEA2<sup>[8]</sup>等算法。可以发现,进化多目标算法最开始采用各种技术将多目标问题转化为单目标问题,然后逐步采用了 Pareto 解的概念,从而使我们处理问题的方法更符合客观事物。在采用了 Pareto 解集概念后,算法的求解精度有了极大的提高。然而,由于进化算法本身属于随机搜索算法,收敛速度不能保证,且进化多目标优化算法仍需将多个同时优化的目标转化为相应的评价适应值,运算量仍然很大<sup>[9]</sup>。

越来越多的学者试图将其它优化算法应用于多目标优化领域,粒子群算法就是其中一种<sup>[10]</sup>。它属于群聚智能算法,其消息传递机制有别于进化算法。学者们通过设计多目标粒子群算法来解决多目标进化算法的收敛速度慢、易于陷入局部最优解的问题。当前,多目标粒子群算法的研究处于起步阶段,研究的主要工作是模仿多目标进化算法的策略设计新的多目标粒子群算法。较有影响的工作如 K.E.Parsopoulos, M.N.Vrahatis<sup>[11]</sup>的 MOPSO 采用加权聚合法; Xiaodong Li 的 MOPSO 采用 NSGA2 算法的相应策略<sup>[12]</sup>; Gregorio Toscano-Pulido 和 Carlos A.Coello Coello 的采用簇技术来保证算法的有效性<sup>[13]</sup>; Fieldsend 的采用了新的数据结构来存储每代的精英<sup>[14]</sup>。另外, Thomas Bartz-Beielstein 等人对多目标粒子群算法优化的归档机制进行了实验分析<sup>[15]</sup>, Sanaz Mostaghim 等对多目标粒子群优化进行了一些总结<sup>[16]</sup>。

## 1.3 研究内容与框架

### 1.3.1 研究内容

在吸取已有研究成果的基础上,本文设计和讨论了一种改进的多目标粒子群算法,通过设计选择算子来处理约束条件,通过利用拥挤机制和禁忌算法来进行粒子全局最优值的选取,通过采用随机选择策略交换分组的非支配集构造方法来构造非支配集。最后



把改进后的多目标粒子群算法应用到新的营养配餐的计算模型的求解过程中。因此本文的研究工作主要有以下几点：

#### (1) 约束处理

大多数多目标优化问题均含有约束条件，大部分约束优化问题的解是在约束边界取得，因此求解约束优化问题的难点在于对约束条件的处理。本文引入半可行域的概念，改进了适应度函数的求解方式，用以处理约束条件。从而使约束条件的处理更为合理，满足实际的求解需要。

#### (2) 全局最优值的选取

全局最优值的选取在多目标粒子群算法中非常重要，直接关系到粒子的更新更直接影响到算法的收敛速度和解集的分布性，恰当的选取可以避免算法过早的陷入局部最优。本文采用外部集保存当前找到的最优解集，从外部集中随机选择粒子的全局最优值，前期采用拥挤机制用于粒子的全局最优值的选取，在外部集中粒子的分布日趋均匀的情况下采用禁忌算法用于粒子的全局最优值的选取。从而使全局最优值的选取更为合理，保持了解的分布性。

#### (3) 非支配集的构造

在算法的每次迭代过程中均须构造非支配集，因此非支配集的构造效率直接影响到整个算法的时间复杂度，传统的构造非支配集的方法时间复杂度较高，本文采用了随机选择策略交换分组的非支配集构造方法，在选取粒子进行分组的过程中，尽量选择上次分组时选取的比较粒子的支配粒子做为新的比较粒子，提高了构造非支配集的效率，减少了算法的时间复杂度。

#### (4) 多目标粒子群算法的应用

在实际应用中，多目标粒子群算法的应用范围较小。本文提出了新的营养配餐计算模型，将提出的新的多目标粒子群算法应用到新的营养配餐模型中，针对营养配餐的特点设计并实现了基于 Web 的新的营养配餐模型。通过与传统营养配餐模型和求解方式的实验结果进行比对，验证了本文提出的新的多目标粒子群算法的可行性和有效性，拓展了新的多目标粒子群算法的应用领域。

### 1.3.2 论文框架

本文后续章节组织如下：

第二章 多目标优化的粒子群算法：介绍什么是多目标优化问题，相关的基本概念和术语，传统的解决多目标优化问题的方法，现代的多目标进化算法的概况，并对两者

进行了比较。详细介绍了粒子群算法,并对粒子群算法和其它进化算法进行了分析比较,最后分析了将粒子群算法应用到多目标优化中的注意事项。

第三章 改进的多目标粒子群优化算法: 此章为全文的重点,给出了改进的多目标粒子群算法的设计原理,算法流程及具体的实现方案。给出了评价多目标优化算法性能的度量方法,通过一些标准测试函数的实验数据对比,讨论了改进算法的优缺点。

第四章 营养配餐计算模型: 设计了新的营养配餐计算模型,将多目标粒子群算法应用到营养配餐决策支持系统的营养计算模型的求解过程中,取得了良好的实验效果。

第五章 营养配餐决策支持系统: 设计并实现了营养配餐决策支持系统,重点介绍了营养配餐计算模型的设计和实现。

结论: 总结了论文的主要工作,对将来的研究工作进行展望,提出了进一步的研究方向。

## 2 多目标优化的粒子群算法

### 2.1 多目标优化

最优化处理的是在多个可能的选择中找寻对于某些目标来说是最优解的问题，这里的目标数目可能是只有一个或者多个，如果仅考虑一个目标，就称为单目标优化问题，这类问题已经得到了深入、广泛的研究，另外，如果存在的目标超过一个并需要同时处理，就成为多目标优化问题。

#### 2.1.1 多目标优化的基本概念

单目标优化问题通常可表述为下面的形式：

$$\begin{aligned} \max z &= f(x) \\ \text{s.t. } g_i(x) &\leq 0, \quad i=1,2,\dots,m \end{aligned} \quad (2.1)$$

其中  $x \in R^n$  是带有  $n$  个决策变量的向量， $f(x)$  是目标函数， $g_i(x)$  是  $m$  个不等式约束函数，由它们形成了可行解区域。通常在决策空间中用  $S$  来表示可行区域，如下表示：

$$S = \{x \in R^n \mid g_i(x) \leq 0, i=1,2,\dots,m, x \geq 0\} \quad (2.2)$$

不失一般性，多目标优化问题可以表述为下面的形式：

给定决策向量  $x = [x_1, x_2, \dots, x_n]^T$ ，它满足下列约束：

$$\begin{aligned} g_i(x) &\leq 0, \quad i=1,2,\dots,k \\ h_i(x) &= 0, \quad i=1,2,\dots,l \end{aligned}$$

设有  $m$  个优化目标，且这  $m$  个优化目标可能是相互冲突的，优化目标可表示为：

$$f(x) = [f_1(x), f_2(x), \dots, f_m(x)]^T \quad (2.3)$$

寻求  $x^* = [x_1^*, x_2^*, \dots, x_m^*]^T$ ，使  $f(x^*)$  在满足约束和的同时达到最优。

从 2.1.1 的定义中我们可以看出，多目标优化问题与单目标优化还是有很大差别的，在只有一个目标的时候，人们寻找的是一个最好的解，这个解比其它所有的解都要好。而当我们求解多目标优化问题时，由于有多个目标且存在目标之间的无法比较和冲突现象，不能期望它们的极小点重复在一起，即不能同时达到最优解；甚至有时还会产生对立的情况，即对一个目标函数是最优点，对另一个目标函数却是差点。要使所有的目标函数同时达到最大(或最小)是不可能的。多目标算法的核心就是协调各目标函数之间的关系，找出使各目标函数能尽量达到比较大(或比较小)的最优解集，一个解可能在其中某个上是最好的，但在其他目标上是最差的，不一定有在所有目标上都是最优的解。因此，在有多个目标时，通常存在一堆无法简单进行相互比较的解。这种解通常称作非

支配解(non-dominated solutions 或 non-inferior solutions)或 Pareto 最优解(Pareto optimal solutions), 这里所说的 Pareto 最优解<sup>[17]</sup>, 是由 Vilfredo Pareto 在 1896 年提出的, 定义如下:

如果  $x^* \in R^n$ , 不存在  $x \in R^n$  和任何的  $i=1,2,\dots,m$ , 使得  $f_i(x) \leq f_i(x^*)$ , 并且至少在一点上  $f_i(x) < f_i(x^*)$ , 把  $x^*$  叫做 Pareto 最优解。

也就是说, Pareto 边界上的可行解不能在改善一个目标的同时而不损害其他的至少一个目标。通常情况下, 最优解不只一个, 而是一个最优解集。多目标算法的工作就是构造非支配集, 并使非支配集不断逼近 Pareto 最优解集, 最终达到最优。在多目标进化群体中, 或对一个给定的多目标可行解集, 个体之间的关系不同于单目标优化下的大小关系。对于个体  $x$  与个体  $y$ , 或者  $x$  支配  $y$ , 或者  $y$  支配  $x$ , 或者  $x$  与  $y$  之间相互不被支配。为了描述方便起见, 下面给出有关定义。

定义 1 设  $P$  为一个集合, 其大小为  $n$ ,  $P$  中每个个体均有  $r$  个属性,  $f_i()$  是每个属性的评价函数( $i=1,2,\dots,m$ ),  $P$  中个体之间的关系定义为:

(1)支配关系:  $\forall x,y \in P$ , 若  $f_i(x) \leq f_i(y), (i=1,2,\dots,m)$ ; 且  $\exists k \in \{1,2,\dots,m\}$ , 使  $f_k(x) < f_k(y)$ , 则称  $x$  支配  $y$ , 表示为  $x \succ y$ 。此时称  $x$  为非支配的(non-dominated),  $y$  为被支配的(dominated), 其中 “ $\succ$ ” 是支配关系。

(2)不相关:  $\forall x,y \in P$ , 若  $x$  和  $y$  之间不存在支配关系, 则称  $x$  和  $y$  不相关或无关。

定义 2 对于给定的个体  $x \in P$ , 若不  $\exists y \in P$ , 使  $y \succ x$ , 则  $x$  称之为集合  $P$  的非支配个体。有所有  $P$  的非支配个体组成的集合, 称之为  $P$  的非支配集。

定义 3 设  $Nds$  是  $P$  的非支配集,  $\forall x \in P$ , 若  $x$  是  $P$  的非支配个体, 必有  $x \in Nds$ , 则称  $Nds$  为  $P$  的最大非支配集。

### 2.1.2 多目标优化算法分类

对于多目标优化问题, 从数学的角度讲, Pareto 最优解中的所有解都是可接受的解, 不同解之间并无任何差异, 然而人们往往需要一个或多个最终解来指导实际工作。从 Pareto 最优解集中选择某一个或多个解的过程称为决策(DM)。指导决策的人应该对实际的问题有深刻的认识, 再加上自己的喜好偏爱或经验知识, 选择其认为最合适的解。

获得 Pareto 解集的过程被称为解过程(solution process)或优化过程, 它往往和决策过程相互合作, 从而寻找到人们需要的最终解。

根据决策者在优化过程中的参与方式, 依赖于优化和决策这两个过程是如何组合的, 多目标优化方法可以被分为四类<sup>[18]</sup>:

在优化前决策(Priori methods): 多目标优化的多个目标被合并成一个单目标优化问题, 它隐式的包含决策者的喜好信息。

在优化后决策(Posteriori methods): 在没有任何喜好信息的情况下进行优化。优化过程的结果是候选解的集合(理想的是 Pareto 最优解集), 然后由决策者最终做出选择。

在优化的过程中进行决策(Interactive methods): 在优化的过程中, 决策者给出一些偏好信息。在优化过程的每一步, 求得一些折中方案, 在此基础上, 决策者给出更深层的偏好信息以导向更深层的优化。

没有明显的偏好信息(No-preference methods): 不明显属于上述类型的方法, 不需要任何决策信息的方法。

或者更简单的分为两类:

(1) 在优化过程中考虑决策者的偏好, 最后得到一个基于偏好的最终解的方法 (Preference-based methods);

(2) 在优化过程中不考虑决策者的偏好, 直接得到一个非支配解集的产生式方法 (Generating methods)。

基于偏好的方法试图得到一个最终解, 产生式方法试图得到一个非支配解集。如果能通过一些准则来判定目标间的相对重要性, 就可以利用这些准则来获得最终解。如果没有任何准则信息, 那么只能用产生式获得一个非支配解集并对其进行检验, 得到满意的结果。

### 2.1.3 基于偏好的方法

在应用进化算法之前, 大多数传统的多目标优化算法都是基于偏好的方法, 它们根据决策者提供的偏好信息, 通过一定的方法把多目标优化问题转化为一个或一系列单目标优化问题, 利用数学规划来求解获得最终解。基于偏好的多目标优化方法包括评价函数法、分层序列法、目标规划法等几类方法。

#### (1) 评价函数法

这类求解方法本着将多目标转化为单目标的思想, 将比较难统一的多个目标转化为一个利用现有单目标优化方法比较容易解决的单目标问题。转化成单目标的过程就是一个利用决策者提供的偏好信息构造实函数的过程, 使得寻找使决策者满意解的过程等价于解决以该实函数为新目标的单目标问题。评价函数就是用一个评价函数来集中反映各个不同目标的重要性等因素, 并极小化此评价函数, 得到问题的最优解。常用的评价函数法有: 主要目标法、线性加权法、极大极小法、理想点法等。

##### a.主要目标法

根据问题的实际意义, 确定一个主要目标而把其余目标在一定的允许界限内作为约束。

不妨设  $f_1(x)$  为主要目标, 其余目标给定一组界限, 例如  $f_i(x) \leq f_i^*$ , 这样可将原问题转化为:

$$\begin{cases} z = \min(f_1(x)) \\ g_j(x) \geq 0 \quad j = 1, \dots, n \\ f_i(x) \leq f_i^* \quad i = 2, \dots, m \end{cases} \quad (2.4)$$

### b. 线性加权法

线性加权法是最简单、最基本也是应用最广泛的多目标优化算法。其核心思想是根据各个目标  $f_i(x)$  在决策者心中的重要程度, 分别赋予一个非负的权系数

$w_i \geq 0 (i = 1, 2, \dots, m)$ ,  $\sum_{i=1}^m w_i = 1$ , 然后把这些带权的目标加起来构造评价函数

$$z = \sum_{i=1}^m w_i f_i(x) \quad (2.5)$$

通过这个评价函数, 可以把多目标优化问题转化成单目标优化问题:

$$\min z = \min \sum_{i=1}^m w_i f_i(x) \quad (2.6)$$

求解这个单目标问题可以得到最终解  $x^*$ 。由于限制了  $w_i \geq 0$ , 可以证明  $x^*$  是原目标问题的一个非支配解。

在不同的权值组合下求解这个单目标优化问题就可以求得一个解集。如果所求问题具有凸性, 那么理论上采用该方法反复迭代就能得到一个完全的非支配解集。然而, 如果该问题不满足凸性, 将不能保证最终得到一个满意的 Pareto 解集。

该算法有几个明显的缺点: 一方面, 权重参数的很小的改变能够引起所求目标向量的较显著的变化; 另一方面, 不同权重参数的显著变化可能得到相似的解向量。因此, 权重集合的均匀分布一般不会产生一个均匀分布的 Pareto 解集。

### c. 极大极小法

极大极小法是在对各个目标来说最不利的情况下找出最有力的解。对多目标最小化问题, 选取各个目标函数  $f_i(x)$  中最大的值构造评价函数, 即取  $z = \max_{1 \leq i \leq m} \{f_i(x)\}$ , 原问题就可以归结为数值极小化问题:

$$\min z = \min \max_{1 \leq i \leq m} \{f_i(x)\} \quad (2.7)$$

有时为了在评价函数中反映各个目标的重要程度。也可以赋给目标相应的权值，此时的评价函数为  $z = \max_{1 \leq i \leq m} \{w_i f_i(x)\}$ ，相应的单目标问题就是：

$$\min z = \min \max_{1 \leq i \leq m} \{w_i f_i(x)\} \quad (2.8)$$

实际应用极大极小法时，通常引入变量  $\lambda = \max_{1 \leq i \leq m} \{w_i f_i(x)\}$  把多目标优化问题转化为以下单目标优化问题：

$$\begin{aligned} & \min \lambda \\ & \text{s.t.} \begin{cases} w_i f_i(x) \leq \lambda, & i = 1, 2, \dots, m \\ x \in P \end{cases} \end{aligned} \quad (2.9)$$

d.理想点法

有  $m$  个目标  $f_1(x), \dots, f_m(x)$ ，对于每个目标分别有其最优值：

$$f_i^* = \min f_i(x) = f_i(x^i) \quad i = 1, 2, \dots, m \quad (2.10)$$

如果所有的  $x^i$  都相同，记为  $x^*$ ，则该解是绝对最优解。可惜一般该解不存在，因此对向量函数  $F(x) = (f_1(x), f_2(x), \dots, f_m(x))^T$ ，向量  $(f_1^*, f_2^*, \dots, f_m^*)$  只是理想点。

理想点法的基本思想是定义一个模，在这个模的意义下，找一个与理想点尽量接近的点。一般去欧式空间的距离为模。

令评价函数为： $\sqrt{\sum_{i=1}^m (f_i - f_i^*)^2}$ ，然后求  $z = \min \left( \sqrt{\sum_{i=1}^m (f_i - f_i^*)^2} \right)$  得最优解。

(2) 分层序列法

分层序列法的特点是各个目标函数不是同等的被优化，而是按照不同的优先层次进行优化。先求出第一层目标的最优解，在达到此目标的条件下求第二个目标的最优解，以此类推直到最后一个求解结束得到最优解。

一般，对于  $m(m \geq 2)$  个目标函数

$$f_{i_1}^1(x), \dots, f_{i_1}^1(x); f_{i_2}^2(x), \dots, f_{i_2}^2(x); \dots; f_{i_l}^l(x), \dots, f_{i_l}^l(x), (l_1 + l_2 + \dots + l_l = m) \quad (2.11)$$

按其重要程度分成  $L$  个优先层次：

$$\begin{cases} \text{第1优化层次} - f_{i_1}^1(x), \dots, f_{i_{l_1}}^1(x) \\ \text{第2优化层次} - f_{i_1}^2(x), \dots, f_{i_{l_2}}^2(x) \\ \dots \dots \dots \\ \text{第L优化层次} - f_{i_1}^L(x), \dots, f_{i_{l_L}}^L(x) \end{cases}$$

则它们的分层多目标优化问题可以记作

$$L - \min [P_1(f_1^1(x), \dots, f_{l_1}^1(x)), P_2(f_1^2(x), \dots, f_{l_2}^2(x)), \dots, P_l(f_1^l(x), \dots, f_{l_l}^l(x))] \quad (2.12)$$

其中,  $P_s (s=1, 2, \dots, L)$  是优先层次的记号, 表示后面括号中的目标函数属于第  $S$  优先层次, 并且各  $P_s$  之间有关系  $P_s \gg P_{s+1}, s=1, \dots, l$ , 它表示第  $S$  优先层次“优先于”第  $S+1$  优先层次。

常用的分层序列法有完全分层法、分层评价法和重点目标法等。其中完全分层法的每个优先层次中只有一个目标。采用此方法一般都能取得比较满意的最优解, 但是该方法存在一个缺点, 当前面的问题的最优解是唯一的时候, 后面的求解就失去意义了。

### (3) 目标规划法

该方法中, 决策者首先对每个目标函数提出预定的目标水平  $z_i (i=1, \dots, m)$ , 并根据每个目标水平得到偏移变量。每个目标函数联合其目标水平构成目标。对于最小化问题, 目标的形式为  $f_i(x) \geq z_i$ 。接着超过目标水平的值  $\sigma_i$  被最小化。

针对权重的方法, 需要做的是将偏移变量最小化, 多目标优化问题被转化为如下的求解问题:

$$\begin{aligned} \min \quad & \sum_{i=1}^m w_i \sigma_i \\ \text{s.t.} \quad & f_i(x) + \sigma_i \geq z_i, \quad i=1, \dots, m \\ & \sigma_i \geq 0, \quad i=1, \dots, m \end{aligned} \quad (2.13)$$

对基于权重的目标规划, 如果目标水平就是 Pareto 参考点, 或偏移变量均为正, 则可以保证得到的解为 Pareto 解。

该方法容易理解, 决策者容易决策, 但是权重参数仍然很难正确设置, 并且没有什么实际的意义。

用基于偏好的传统方法解决多目标优化问题可以充分利用用户掌握的决策信息, 直接得到符合决策条件的优化结果, 避免了优化后的决策步骤。此外传统方法还有易于实现, 运算速度快等优先。但是传统方法有几个重大的缺陷:

(1) 由于实际问题中不同目标的物理意义和度量单位往往不同, 目标之间不能直接进行比较或加权。尽管可以用目标函数的无量纲化处理来解决, 但这又增加了算法的复杂性, 而且会引起目标空间的改变导致无法正常利用决策信息。

(2) 这些算法要求用户提供精确的决策信息, 但常常无法提供确定符合实际需要的决策信息, 无法精确的建立优化问题的数学模型。

(3) 大多数算法只能得到一个局部最优解, 为了避免陷入局部最优, 扩大邻域大小是常见的方法, 但是随着邻域的增大, 算法的复杂性也会呈现出指数次方的增加。



(4)许多传统方法仅能运用于相应的小的问题集,推广性较差。对于一个现实世界的优化问题,必须尝试很多不同的方法,甚至要发明相应的新的方法来解决,这显然是不现实的、繁琐的。我们需要另外的方法来克服这样的困难。

#### 2.1.4 产生式方法

产生式方法在优化过程中不考虑目标之间的相互关系,通常没有偏好信息可供利用,多个目标函数是同时进行优化,而且通常用支配非支配的概念来区分非劣解和其它解。产生式方法最终会产生一系列的非劣解让决策者从中选择他们需要的方案。

产生式方法可以通过一些简单的途径实现,如线性加权法中改变权值系数而获得一系列非劣解集。但是由于基于偏好的方法本身存在的缺陷,这些方法都不能获得较好的非劣解集,易于陷入局部最优,特别是对非凸函数问题的解决效果更差。

使用产生式方法进行多目标优化,只需对问题进行一次即可,因为只要多目标优化问题的模型确定后,其非劣解集就是确定的,不像基于偏好的方法,如果决策者的偏好有所改变,则必须根据新的偏好信息再次进行优化。产生式方法的缺陷是往往需要付出很大的计算代价,这在高维复杂问题上的表现更明显。

各种现代优化算法的兴起,为多目标优化提供了更有效的解决方法,这类算法大都属于启发式算法<sup>[19]</sup>,它们都是通过模拟自然界中的各种现象发展起来的优化求解算法。现代启发式算法的优点是它可以有效的避免产生局部最优,同时它的推广性很强,鲁棒性很高。例如,虽然线性加权法可以比现代启发式算法更好的解决一些多目标线性规划问题,但是该方法也只限于解一些凸函数问题,而像粒子群这样的算法可以适用于解很多其它的问题。也就是说,现代启发式算法不会束缚到一个具体的问题域,同时像粒子群算法、遗传算法等群体搜索算法是同时搜索可能的解,从而在一次运行中就可以得到 Pareto 解集,传统算法却往往要运行多次,此外,现代启发式算法对所求问题的 Pareto 阵面的形状和连续性并不敏感,这样的问题常常在现实世界中遇到,这样的性质显然更适合于处理实际问题。

模拟退火算法<sup>[20]</sup>(Simulated Annealing, SA)是一种典型的现代启发式算法。算法是1982年由 Kirkpatrick 等人提出来的一种随机优化算法,它模拟了物理中金属的降温过程,该算法也被改进用于求解多目标优化问题,它具有很强的全局搜索能力,对搜索非劣解集非常有利。算法中有当前点  $v_c$ , 根据该点评价其邻域中的某一点  $v_n$ 。如果  $v_n$  较好,则用其代替  $v_c$ ; 否则,则会按照概率  $p$  来替换  $v_c$ 。算法求得的解与初始解无关;模拟退

火算法具有渐近收敛性,已在理论上被证明是一种以概率 1 收敛于全局最优解的全局最优化算法,并且该算法具有并行性。

禁忌算法也是一种现代启发式算法。每次迭代中,也是由某当代点的邻域来得到下一代最优可能点。不同的是,该算法保存一个禁忌表,在禁忌表中出现的点不允许出现在下一代点中,这样就可以避免算法陷入局部最优。

另一类很重要的算法是演化算法,主要包括遗传算法和粒子群算法。演化算法是基于种群操作的计算技术,为概率算法,通过在代与代之间维持由潜在的解组成的种群来实现多向性和全局搜索,这种从种群到种群的方法在搜索非劣解时非常有用。此外演化算法不需要许多数学上必备条件如函数的可导性、目标空间的凸性等要求,可处理各种类型的目标和约束,因而也就能处理各种复杂的实际问题,其适用范围远比传统算法要宽。

### (1)多目标遗传算法

用遗传算法求解多目标优化问题的一个特殊情况就是根据多个目标确定个体适应度。典型的适应度分配机制包括向量评价方法、基于 Pareto 的方法等。

向量评价遗传算法是第一种用于多目标优化的遗传算法,该方法采用了向量形式的适应度而不是标量形式的适应度来评价个体

基于 Pareto 的多目标遗传算法是目前应用最广泛、最具代表性的一类多目标遗传算法。它们的流程基本相同。

首先初始化种群,然后基于 Pareto 最优的概念对种群分类,按照分类评价种群中的个体并赋予适应度值,之后根据适应度按概率选择个体进行杂交和变异,生成新的种群,再次进行评价并循环下去直到满足收敛条件。基于 Pareto 的多目标遗传算法之间的区别主要体现在分类和评价的过程中,它们可以分为 Pareto 排序和 Pareto 竞争两类。

### (2)多目标粒子群算法

粒子群算法(Particle Swarm Optimization, PSO)是基于群体的演化算法,于 1995 年由 Eberhart 和 Kennedy 提出<sup>[21]</sup>。PSO 的思想来源于人工生命和演化计算理论,它模拟的是鸟群的觅食行为。与遗传算法不同的是 PSO 没有交叉和变异操作,而是通过追随当前找到的最优解来搜索全局最优解。

用 PSO 来求解优化问题时,问题的每一个解称为“粒子”(Particle)或“主体”(Agent)。每个粒子都有自己的位置和速度(决定前进的方向和距离),还有一个由目标函数决定的适应度。寻优过程中每次迭代并不是完全随机的,如果找到了一个较好的解,算法将会以此为依据来寻找下一个解。

## 2.2 粒子群算法

### 2.2.1 基本粒子群算法

粒子优化算法源于对鸟类捕食行为的模拟。设想有这样一个场景:一群鸟在随机搜索食物不知道食物放在何处,在整个搜索区域中只有一块食物。所有的鸟都都不知食物在何处,但它们知道自己当前的位置距离食物还有多远。那么找到食物的最优策略是什么呢?最简单有效的方法就是搜寻目前离食物最近的鸟的周围区域。

粒子群优化算法从这种模型中得到启示并用于解决优化问题。在粒子群优化算法中,优化问题的每一个解都被看作是搜索空间中的一只鸟。我们称之为“粒子”(Particle)或“主体”(agent)。每个粒子都有自己的位置和速度(决定飞行的方向和距离),还有一个由被优化函数决定的适应值,并且知道自己到目前为止发现的最好位置(pbest)和现在的位置  $X_i$ 。这个可以看作是粒子自己的飞行经验。除此之外,每个粒子还知道到目前为止整个群体中所有粒子发现的最好位置(gbest)(gbest是在pbest中的最好值)。这个可以看作是粒子的同伴的经验。每个粒子使用下列信息改变自己的当前位置:

- (1) 当前位置;
- (2) 当前速度;
- (3) 当前位置与自己最好位置之间的距离;
- (4) 当前位置与群体最好位置之间的距离。

粒子群算法是一种群优化算法。粒子群可以认为是粒子在D维空间内,按一定规律传递信息,并根据信息的变化改变自身状态所产生的自组织行为。图2.1是粒子群算法粒子间信息传递的示意图。

粒子群的信息主要来自由各粒子的个体极值构成的矩阵:  $P=(p_1, p_2, \dots, p_n)$ 。基本粒子群算法中粒子从P中提取的信息有群体最优位置全局极值P和每个粒子自身经验最优位置个体极值P。群体最优位置使得粒子能够快速收敛形成粒子群,并对全局极值的邻域进行搜索;个体自身经验最优位置保证粒子不至于过快收敛到群最优,而陷入局部极小点,使得粒子能够在一次迭代中对个体极值和全局极值之间的区域进行搜索。

粒子群算法是一种群搜索算法,它之所以有高效的搜索性能,是因为群体的合作。正如文献所说,“社会行为有两个主要的目的:一是每个个体能够在搜索食物的过程中协助其它群内的成员,二是群体合作能够提高搜索效率。”换言之,每个粒子能够向群体提供信息并且每个粒子又能够协助其它粒子进行搜索。

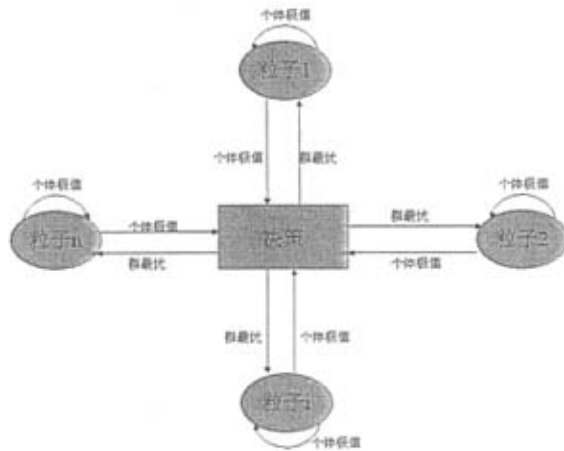


图 2.1 粒子信息传递示意图

Fig. 2.1 Particle information transmission

假设在一个  $D$  维的目标搜索空间中, 有  $m$  个粒子组成一个部落, 其中第  $i$  个粒子表示为一个  $D$  维的向量  $x_i=(x_{i1}, x_{i2}, \dots, x_{id}), i=1, 2, \dots, m$ , 即第  $i$  个粒子在  $D$  维的搜索空间中的位置是  $x_i$ 。换言之, 每个粒子的位置就是一个潜在的解。将  $x_i$  带入一个目标函数就可以计算其适应值, 根据适应值的大小衡量  $x_i$  的优劣。第  $i$  个粒子的“飞翔”速度也是一个  $D$  维向量, 记为  $v_i=(v_{i1}, v_{i2}, \dots, v_{id})$ 。记第  $i$  个粒子迄今为止搜索到的最优位置为  $p_i=(p_{i1}, p_{i2}, \dots, p_{id})$ , 整个粒子群迄今为止搜索到的最优位置为  $p_r=(p_{r1}, p_{r2}, \dots, p_{rd})$ 。

采用下列公式对粒子操作:

$$v_{id}=w \cdot v_{id} + c_1 \cdot r_1 \cdot (p_{id} - x_{id}) + c_2 \cdot r_2 \cdot (p_{gd} - x_{gd}) \quad (2.14)$$

$$x_{id}=x_{id} + v_{id} \quad (2.15)$$

在式(1)、(2)中,  $i=1, 2, \dots, m, m$  是该群体中粒子的总数;  $v_{id}$ : 迭代粒子  $i$  飞行速度矢量的第  $d$  维分量;  $p_{id}$ : 粒子  $i$  个体最好位置  $p_{best}$  的第  $d$  维分量;  $p_{gd}$ : 群体最好位置  $p_{best}$  的第  $d$  维分量;  $c_1, c_2$ : 权重因子;  $\text{rand}()$ : 随机函数, 产生  $[0, 1]$  的随机数;  $w$ : 惯性权重函数。

公式(1)主要通过三部分来计算粒子  $i$  新的速度: 粒子  $i$  前一时刻的速度, 粒子  $i$  当前位置与自己最好位置之间的距离, 粒子  $i$  当前位置与群体最好位置之间的距离。粒子  $i$  通过公式(2)计算新位置的坐标。通过式(1), (2)粒子  $i$  决定下一步的运动位置。在图 2.2 中, 以两维空间为例描述了粒子根据公式(1)、(2)从位置  $x^k$  到  $x^{k+1}$  移动的原理。

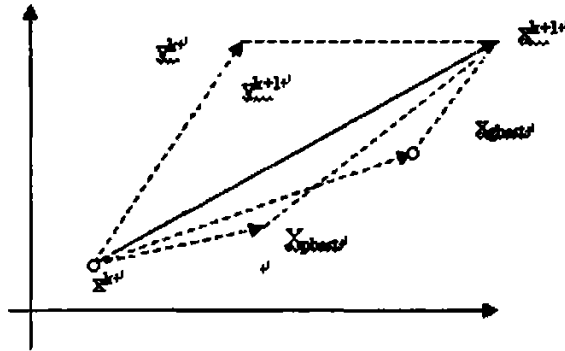


图 2.2 粒子移动示意图

Fig. 2.2 Particle mobile

如果从社会学的角度来看公式(1)，其中第二部分(粒子  $i$  当前位置与自己最好位置之间的距离)为“认知(cognition)”部分，表示粒子的动作来源于自己经验的部分；第三部分(粒子  $i$  当前位置与群体最好位置之间的距离)为“社会(social)”部分，表示粒子的动作来源于群体中其它粒子经验的部分，表现为知识的共享和合作。粒子就是通过自己的经验和同伴中最好的经验来决定下一步的运动。这与人类的决策何其相似，人们通常也是通过综合自身已有的信息和从外界得到的信息来作出决策的。

最后，列出基本粒子群算法的主要步骤(基本粒子群算法框图如图 2.3 所示)：

- (1) 初始化一个粒子群，即随机产生各粒子的初始位置和速度；
- (2) 根据目标函数评价每个粒子的适应度；
- (3) 对每个粒子，将其适应度与该粒子经历过的个体极值作比较，如果较好，则将该位置作为当前的最好位置  $pbest$ ；
- (4) 对每个粒子，将其适应度与整个粒子群经历过的全局极值作比较，如果较好，则重新设置  $gbest$  的值；
- (5) 根据速度和位置变化公式计算粒子新的速度和位置；
- (6) 如达到迭代终止条件(通常为一个预设的最大迭代代数或最小误差阈值)，则程序终止，否则返回(2)进行新一轮迭代。

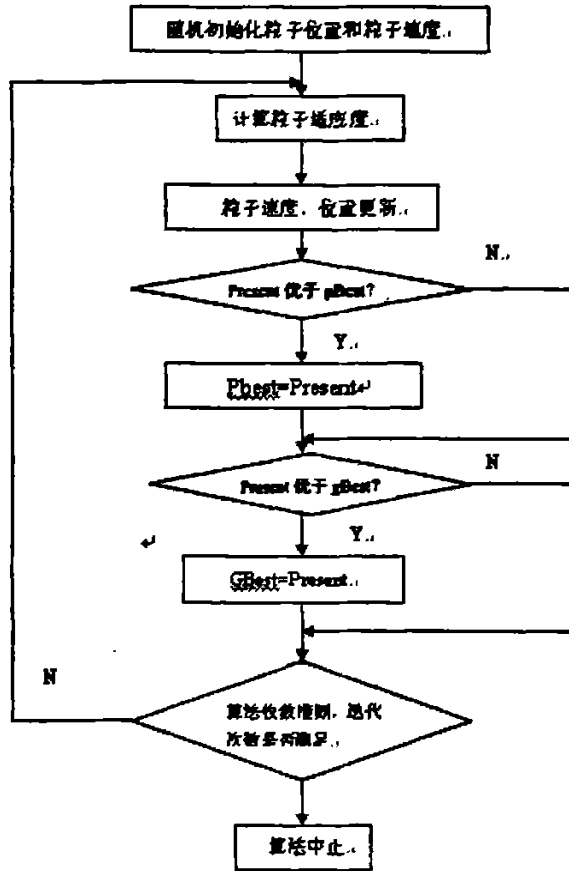


图 2.3 基本粒子群算法框图

Fig. 2.3 Elementary particle swarm algorithm

### 2.2.2 多目标粒子群算法

虽然粒子群优化算法在许多单目标优化问题中的成功应用说明了 PSO 算法的有效性，但是粒子群优化算法还不能直接应用于多目标优化问题，因为多目标优化问题和单目标优化问题是有本质的区别的：前者一般是一组或几组连续解的集合，而后者只是单个解或一组连续的解。另外，遗传算法在多目标优化问题中的成功应用以及 PSO 算法和遗传算法的相似性，说明 PSO 算法可能是一种处理多目标优化问题的方法。但是 PSO 算法不是像遗传算法那样在染色体间共享信息从而使整个种群逐步移向好的区域，PSO 算法中的信息是由最好的粒子给出的，其他个体跟着最好粒子快速向一点收敛，因此直接用 PSO 算法处理多目标优化问题，将很容易收敛于非劣最优域的局部区域。另外还需要对其进行很多改进和扩展，这些需要考虑的地方包括：

- (1) 两个粒子之间如何判断哪个更优;
- (2) 个体极值和全局极值的选取;
- (3) 如何保持算法的分布性;
- (4) 是否施加对粒子的扰动;

为了解决上述问题,一些研究者提出了他们的改进方案,下面将介绍多目标粒子群算法中几个最重要的改进方案:

Moore 和 Chapman 提出了主要是基于 Pareto 支配关系的多目标粒子群算法<sup>[22]</sup>,作者在文章中强调了粒子和种群(分别作为认理解部分和社会部分)之间作用的重要性。但是,他们并没有采用其它任何措施来保持算法的分布性。

Parsopoulos 和 Vrahatis 提出了一种目标聚合的多目标粒子群算法<sup>[23]</sup>,采用固定的或自适应的权重组合将多目标优化问题转换为单目标优化问题,然后采用基本粒子群算法进行求解。这种算法的缺点和传统多目标算法类似,一次运行只能获得一个 Pareto 最优解。

Hu 和 Eberhart 提出了一种动态近邻法<sup>[24]</sup>来求解多目标优化问题,采用动态近邻策略来为粒子选取最优经验,但是一次只能对一个目标寻优,其本质还是用一维的方法来处理多个目标。

Ray 和 Liew 提出了模拟真实群体行为的多目标粒子群算法<sup>[25]</sup>,在这个算法中引用了进化算法中 Pareto 支配和进化技巧,运用拥挤策略维持种群多样性和多层筛选产生全局最优经验。

Fieldsend 和 Singh 提出了一种“支配树”的数据结构<sup>[14]</sup>,用其来存储精英解以及为粒子选取最优经验,指导粒子飞行。

Mostaghim 等人提出了一种 Sigma 方法<sup>[16]</sup>,通过比较粒子与当前非支配解的 sigma 值来为粒子选取全局最优经验。

Li 的算法<sup>[26]</sup>主要将 NSGA-2 的主要思想移植到 PSO 算法中,该算法的结果与 NSGA-2 相比差不多,在某些例子中甚至比 NSGA-2 更好。

Coello 和 Lechuga 提出了一种基于网格的多目标粒子群算法(Multiple Objective Particle Swarm Optimization, MOPSO)<sup>[27]</sup>,将目标空间分割成网格,从含非支配解较少的网格中随机选取一个解作为当前粒子的全局最优经验。

Sierra 和 Coello<sup>[28]</sup>提出一种新的基于 Pareto 支配关系的多目标粒子群算法,并且设计了新的变异算子应用于不同的粒子。为了收集到固定数目的粒子,在算法的最后一代使用  $\epsilon$  支配踢掉多余的粒子。

Salazar-Lechuga, M 和 Rowe, J.E 的算法<sup>[29]</sup>的算法使用外部集保留非支配粒子, 而外部集更新采用的是支配关系及适应度共享的策略。

将粒子群算法用于解决多目标优化问题是近几年逐步发展起来的一个新兴的研究方向, 在这方面, 不少研究人员进行了很多有益的尝试, 也提出了很好的解决办法, 上述这些算法都有自己独特的方法或特点, 但是从他们的发表的论文中我们可以发现它们都存在着一个共同的缺点, 这些算法的分布性和收敛性还有待进一步提高。

### 2.2.3 粒子群优化算法与进化算法的比较

粒子群优化算法扎根于一些交叉学科, 包括人工生命、进化计算和群论等。粒子群优化与传统的进化算法存在一些相似和相异之处, 下面对两者作简单的比较:

粒子群算法和进化算法的相同点:

- (1) 都属于仿生算法。粒子群算法主要模拟鸟类觅食、人类认知等社会行为而提出的; 进化算法主要借用生物进化中“适者生存”的规律。
- (2) 都属于全局优化方法。在解空间都随机产生初始种群, 因而算法在全局的解空间进行搜索, 且将搜索重点集中在性能高的部分。
- (3) 都属随机搜索算法, 通过采用概率变换规则搜索空间求解。粒子群中认知项和社会项前都加有随机数; 而进化算法的进化操作都属于随机操作。
- (4) 隐含并行性。搜索过程是从问题的一个集合开始的, 而不是从单个个体开始, 具有隐含并行搜索特性, 从而减小了陷入局部极小的可能性。并且由于这种并行性易在并行计算机上实现, 以提高算法性能和效率。
- (5) 根据个体的适配信息进行搜索, 因此不受函数约束条件的限制, 如连续性、可导性等。
- (6) 对高维复杂问题, 往往会遇到早熟收敛和收敛性差的缺点, 都无法保证收敛到最优点。

粒子群算法和进化算法的不同点:

- (1) 粒子群算法有记忆, 好的解的知识所有粒子都保存, 而进化算法, 以前的知识会随着种群的改变被破坏。
- (2) 粒子群算法的粒子仅仅通过当前搜索到最优点进行共享信息, 所以很大程度上这是一项信息共享机制。而进化算法中, 染色体之间相互共享信息, 使得整个种群都向最优区域移动。



- (3) 进化算法的编码技术和进化操作都比较简单，而粒子群算法相对于进化算法，没有重组和变异操作，粒子只是通过内部速度进行更新，因此原理更简单、参数更少、实现更容易。
- (4) 在收敛方面，进化算法已经有了较成熟的收敛性分析方法，并且可对收敛速度进行估计，而粒子群算法这方面的研究还比较薄弱。尽管已经有简化确定性版本的收敛性分析，但将确定性向随机性的转化尚需进一步研究。
- (5) 在应用方面，粒子群算法主要应用于连续问题，包括神经网络训练和函数优化等，而进化算法除了连续问题外，还可应用于离散问题，比如 TSP 问题、货郎担问题、工作车间调度等。

### 3 改进的多目标粒子群算法

本文主要针对利用粒子群算法求解多目标优化问题进行研究，并提出了一种适用于求解多目标优化问题的多目标粒子群算法。

改进的多目标粒子群算法基于 Pareto 支配关系构造非支配解集合，采用外部集保存当前找到的非支配解集，将  $\epsilon$  支配概念用于更新外部集合，使算法能保持较好的分布性，通过引入半可行域的概念设计选择算子对约束条件进行处理以增加求解精度，应用拥挤制度和禁忌算法来改进全局极值和个体极值的选取方法避免陷入局部非劣最优解，并且提出了随机选择策略分组法这个构造非支配集构造方法加快算法运行效率。在本章，先列出改进的多目标粒子群算法的主要流程，然后再逐项介绍算法中采用的每项技术，最后利用标准的测试函数进行实验分析，验证了本文所提出的改进算法的可行性，实用性和先进性。

#### 3.1 改进的多目标粒子群算法的主要流程

多目标混合粒子群算法流程：

(1)初始化粒子群，种群大小为 N：

For i=1 To N

(a) Pop[i]的初始位置随机产生；

(b) V[i]=0; (V[i]是粒子 i 的速度)

(c) Pbest[i]初始化为粒子 i 本身; (Pbest[i]保存粒子 i 的个体极值)

(d) Gbest[i]初始化为粒子 i 本身; (Gbest[i]保存粒子 i 的全局极值)

(2)应用选择算子，得到足够多的有利个体；

(3)对粒子群中每个粒子按公式 3.1 更新其速度，按公式 3.2 更新其位置：

在公式 3.1 中，常数  $c_1$  和  $c_2$  控制个体极值和全局极值对粒子更新的影响程度，参数 W 称为惯性权重， $r_1$  和  $r_2$  是[0,...,1]之间的随机数。

$$v[i] = w \times v[i] + c_1 \times r_1 (Pbest[i] - Pop[i]) + c_2 \times r_2 (Gbest[i] - Pop[i]) \quad (3.1)$$

$$Pop[i] = Pop[i] + v[i] \quad (3.2)$$

(4)计算每个粒子的适应度函数值；

(5)求非支配集：用随机选择策略分组法找出种群中的非支配粒子进入非支配解集；

(6)更新个体极值，(在这里使用支配概念。对每个粒子，若粒子 i 的当前位置支配其个体极值位置，则更新其个体极值；若两者为非支配关系，当两者均在可行域或半可行域时，随机选取个体极值；当两者一个在可行域另一个在半可行域或一个在半可

行域另一个在非可行域时, 选取前者; 当两者均在非可行域时, 求其  $d$ ,  $d$  小者为个体极值);

(7)更新外部集, 将群体的非支配集按  $\varepsilon$  支配关系插入外部集;

(8)更新全局极值, 利用拥挤机制和禁忌算法在外部集中随机选择一粒子作为粒子  $i$  的全局极值;

(9)按公式对粒子进行迭代, 转至(2)直至满足中止条件退出。

外部集保存的就是算法每代运行的最好结果, 在算法迭代运算完成后, 外部集中的所有粒子就是算法最后得到的结果。

图 3.1 展示了多目标粒子群算法中的主要群体、施加在它们上的操作以及它们相互之间的关系。

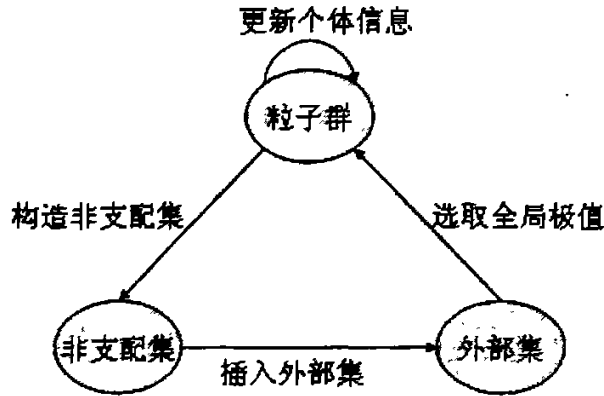


图 3.1 多目标粒子群算法的演示图

Fig. 3.1 Multi-objective particle swarm algorithm

在算法中, 一共设置了三个集合分别用于保存粒子群、非支配集、外部集。粒子群是执行搜索的主体, 非支配集和外部集是保存搜索结果的主体。算法开始运行时, 先随机初始化粒子群及相关参数, 然后求出粒子群中的所有非支配粒子并插入非支配集, 非支配集代表了算法在这一代运行中搜索到的最好的那部分粒子, 但是还必须分离出算法到目前为止所找到的最优的那部分粒子, 所以还需要将每代得到的非支配粒子插入外部集中, 而且全局极值应该是算法找到的解的代表, 显然把外部集作为全局极值的候选集合非常恰当, 粒子群又在极值的引导下不断搜索更优的解, 然后又进入下一个循环。

从图 3.1 可以发现, 算法中应该设计以下主要操作: 对约束条件处理、外部集的更新、个体更新、个体极值以及全局极值的选取、构造非支配集合, 下面将对算法中这几个主要步骤做详细说明。

## 3.2 改进的多目标粒子群算法的关键算子

### 3.2.1 设计选择算子对约束条件进行处理

用进化算法求解约束优化问题, 最基本的思想就是首先设法把个体带入可行域, 然后再在可行域内找到尽可能好的解。目前, 使用最广泛地对约束条件的处理方法是惩罚函数法。基于惩罚函数的进化算法一般来说都包含许多惩罚系数, 在实际应用中只有正确设置这些系数才可能获得可行解, 而要获得适当的惩罚系数则需要大量的实验为基础。作为改进文献<sup>[30]</sup>中介绍了不需要惩罚系数而用竞争选择的方法直接比较个体的优劣的算法, 之后又有许多基于这种思想的进化算法被提出。而这些算法虽然避免了引入惩罚系数, 但为了度量个体对约束的违反程度或保持进化群体的多样化, 仍然要引入新的参数。文献<sup>[31]</sup>中在竞争选择方法的基础上重新设计了不需要惩罚系数的基于惩罚函数的进化算法, 取得了不错的效果。本文引入半可行域的概念, 提出了竞争选择的新规则, 并对文献<sup>[31]</sup>中设计的适应度函数做出了改进, 设计了选择算子用于对约束条件进行处理。

竞争选择的概念源于进化群体中个体之间的竞争关系。具体到约束优化问题中, 对两个个体的比较主要按照下面三条规则来操作:

- (1) 任何可行解都优于不可行解;
- (2) 对两个可行解而言, 目标函数值较好的解为优;
- (3) 对两个不可行解而言, 根据事先定义好的度量约束违反程度的惩罚函数值判断, 违反约束程度小者为优。

求解约束优化问题难点在于对约束条件的处理。对同一个问题, 如果放宽约束条件, 问题会简单一些。如果没有约束条件时问题将比较简单。同理, 大部分约束优化问题的解是在约束边界附近取得的。如果优化问题的解不是在约束边界附近取得, 说明此时的约束条件比较宽松或根本不起作用, 此时目标函数的最优解已与约束条件无关。

对许多问题, 在最优解附近不可行解的目标函数值很可能优于位于可行域  $D$  内部的一个可行解的适应度值。因此, 在可行域边界附近的不可行解用目标函数来评价要比适应度函数更有效。基于以上原因<sup>[32]</sup>, 可引入半可行域概念。

定义 3.1.1 对搜索空间的任一点  $x$  和可行域  $D$  之间的距离定义为

$$d(x, D) = \max\{0, g_{\max}(x)\}$$

$$g_{\max}(x) = \{g_i(x), i = 1, 2, \dots, m\}$$

显然  $d(x, D)$  描述了给定搜索空间中一点  $x$  与可行域  $D$  之间的关系:

If  $d(x, D) = 0$  then  $x \in D$  otherwise  $x \notin D$ .

定义 3.1.2 对给定的数  $\varepsilon$  ( $\varepsilon \in R^+$ ) 存在  $0 < d(x, D) \leq \varepsilon$ , 即满足  $0 < g_i(x) \leq \varepsilon$  的点  $x$  构成的解空间区域  $H$ , 称为半可行域,  $\varepsilon$  ( $\varepsilon \in R^+$ ) 称为半可行域的宽度。

定义 3.1.3 半可行域内的点称为半可行解或半可行个体。

事实上, 半可行域的引入就是对约束条件的放宽。这样可以更好地利用可行域边界附近的不可行解的有用信息。因此, 其半可行解有如下性质: 1、半可行解优于不可行解, 但劣于可行解。2、在半可行域内的个体可由多目标函数决定的 Pareto 支配来判定其优劣。

为了有效利用半可行域的概念, 针对 PSO 算法本文设计了 PSO 选择算子。选择算子的主要作用有两个: 其一, 形成一个半可行域, 保证 PSO 算法中全局极值  $gBest$  一定在半可行域或可行域中。因为 PSO 算法中群体进化的有用信息主要由  $gBest$  给出, 因此保证了正确进化的方向; 其二, 随着迭代不断增加群体中的有利个体(可行解和半可行解)的比例, 当达到最大比例后, 选择算子将逐渐减小半可行域的宽度。因为随着有利个体的不断增加, 在已放宽的约束条件下, 算法必将向此时的约束最优点收敛, 此时半可行域的宽度将减小, 算法又将向另一点收敛, 如此循环, 直至半可行域宽度为零, 所有个体全部进入可行域, 收敛于约束最优点。

具体方法:

给定群体规模  $N$ , 有利个体比例阈值  $R$ , 初始化  $\varepsilon = 0$ 。

(1) 计算有利个体的比例  $rate$ ;

(2) 如果  $rate < R$ , 则  $\varepsilon = \varepsilon + \delta$ , 转(1);

(2) 如果  $rate \geq R$ , 结束;

(3) 如果  $rate \geq R_{max}$  并且  $\varepsilon = 0$ , 结束;

(3) 如果  $rate \geq R_{max}$  并且  $\varepsilon > 0$ , 则  $\varepsilon = \varepsilon - \delta$ , 转(1), 如果  $rate < R_{max}$ , 则  $\varepsilon = \varepsilon + \delta$ , 转(1), 结束;

其中  $\delta$  应根据  $\varepsilon$  的值选取, 一般来说, 应该比  $\varepsilon$  小几个数量级。每次迭代后  $R$  值均增大一定比例。

### 3.2.2 外部集合

在种群之外设置一个外部集合的目的是:

- (1) 为了将算法迭代运行中找到的  $\varepsilon$  支配解保存下来, 用来引导算法更快的向非劣最优区域逼近;
- (2) 作为粒子群中每个粒子的全局极值的候选集合;
- (3) 外部集保留的最终解集就是算法求解的结果。

为了让外部集对种群起到很好的引导作用同时让最后的解集保持良好的分布性, 我们就必须考虑在外部集更新时如何使外部集中的粒子分布均匀, 本文采用文献<sup>[33]</sup>采用的  $\varepsilon$  支配更新外部集合

#### 定义 3.1.4 $\varepsilon$ 支配

令  $f, g \in R^m$ , 则称  $f$   $\varepsilon$  支配  $g$  ( $\varepsilon > 0$ ), 记为  $f \succ_{\varepsilon} g$ , 如果

1.  $\forall i \in \{1, \dots, m\}: \left\lfloor \frac{f_i}{\varepsilon_i} \right\rfloor \leq \left\lfloor \frac{g_i}{\varepsilon_i} \right\rfloor$
2.  $\forall j \in \{1, \dots, m\}: \left\lfloor \frac{f_j}{\varepsilon_j} \right\rfloor < \left\lfloor \frac{g_j}{\varepsilon_j} \right\rfloor$

算法迭代运行时, 先更新整个粒子群, 找出粒子群中的所有非支配粒子作为候选集, 当一个非支配粒子不被外部集中已有的任何粒子  $\varepsilon$  支配时将它插入外部集, 否则被丢弃, 另外, 如果外部集中的某个粒子被新加入的非支配粒子  $\varepsilon$  支配, 则将该粒子从外部集中剔除。以此保证了解集的良好分布性。

#### 3.2.3 个体极值和全局极值的选取

个体极值和全局极值在粒子群算法中非常重要, 尤其对多目标优化问题而言两者更直接影响到算法的收敛速度和解集的分布性。且恰当的选取能避免算法过早的陷入局部最优。对这两个个体的选择, 按照竞争选取的规则进行。但在引入半可行域的概念后, 根据其性质本文对选取规则改进如下:

- (1)任何可行解都优于半可行解;
- (2)任何半可行解都优于不可行解;
- (3)对两个可行解或半可行解, 以支配关系来选取最优点;
- (4)对两个不可行解, 根据事先定义好的  $d(x, D)$  来判断,  $d(x, D)$  值小者为优。

对于个体极值, 我们基于上述的选取规则来判断一个粒子找到的最优位置。当两者同为可行解或半可行解且是非支配关系时, 以 50% 的概率进行替换。

对于全局极值, 外部集就是我们的候选集合, 算法开始运行及每次迭代运行时每个粒子的全局极值均在外部集中随机选取。全局最优值的选取在多目标粒子群算法中非常重要, 直接关系到粒子的更新更直接影响到算法的收敛速度和解集的分布性, 恰当的选取可以避免算法过早的陷入局部最优。过于极端的选择会使迭代产生的外部集中的粒

子过于集中,影响解集的分布性,这就要求在从外部集中随机选择粒子的全局最优值时,采用合理的方法,使最后的结果解集分布均匀。

本文前期采用拥挤机制用于粒子的全局最优值的选取,选用拥挤距离比较长的粒子做为全局最优值,使粒子在迭代过程中尽可能的在解集分布少的区域寻找最优值,优化解的分布性。随着迭代的进行,外部集中粒子的分布日趋均匀,此种情况下拥挤机制的作用下降,改为采用禁忌算法用于粒子的全局最优值的选取,使全局极值的选取尽可能的均匀,以此来保持解的分布性。

确定个体拥挤度的方法如下:

(1)确定外部集  $I$  中个体的数量  $l = |I|$ ,对每个个体  $i$ ,设其初始拥挤距离  $I[i]_{distance} = 0$ ,根据个体  $i$  的每个子目标值对  $I$  中个体进行排序,即  $I = sort(I, m)$  ( $m$  为第  $m$  个目标函数)。

(2)设外部集  $I$  中边界个体拥挤距离  $I[1]_{distance} = I[l]_{distance} = \infty$ ,对于外部集中第 2 个个体到第  $(l-1)$  个个体,其拥挤距离为:

$$I[i]_{distance} = I[i]_{distance} + (I[i+1].m - I[i-1].m) / (f_m^{max} - f_m^{min}) \quad (3.3)$$

禁忌(Tabu Search)算法是一种亚启发式(meta-heuristic)随机搜索算法,它从一个初始可行解出发,选择一系列的特定搜索方向(移动)作为试探,选择实现让特定的目标函数数值变化最多的移动。为了避免陷入局部最优解,TS 搜索中采用了一种灵活的“记忆”技术,对已经进行的优化过程进行记录和选择,指导下一步的搜索方向,这就是 Tabu 表的建立。Tabu 表中保存了最近若干次迭代过程中所实现的移动的反方向移动,凡是处于 Tabu 表中的移动,在当前迭代过程中是不允许实现的,这样可以避免算法重新访问在最近若干次迭代过程中已经访问过的解群,从而防止了循环,帮助算法摆脱局部最优解。另外,为了尽可能不错过产生最优解的“移动”,TS 搜索还采用“释放准则”的策略。

为了克服多目标粒子群算法中的早熟问题,并提高它的运算效率,在迭代运算后期,为每个粒子引入两个记忆装置,即两个禁忌表,第一个禁忌表中存放粒子的历史个体极值,另一个表中存放粒子的历史全局极值及选取此全局极值时粒子的迭代次数。由于禁忌搜索的引入,使之拥有了记忆功能,从而从而具有较强的“爬山”能力。它能在搜索过程中跳出局部最优解,转向其它区域进行搜索,从而使获得更好解的概率大大增加。因为算法使用了禁忌表,它可以限制适应值相同的子代出现的次数,因此可使群体中尽可能保持解集的多样性,从而避免算法早熟。

选取全局极值的方法:

- (1) 如果粒子选取的全局极值在个体极值禁忌表中存在, 则重新选取粒子的全局极值
- (2) 随机从外部集中选择  $k$  个非支配解, 比较其拥挤距离, 得到具有最高拥挤距离的非支配解与具有最低拥挤距离的非支配解的比值  $q$ , 如果  $q < \epsilon$ , 则按禁忌算法思想进行全局最优值的选取, 否则具有最高拥挤距离的非支配解作为粒子的全局最优值。

### 3.2.4 随机选择交换分组法构造非支配集

构造非支配解集是多目标进化算法中的一个重要步骤, 为了提高构造非支配集的效率, 减少算法的时间复杂度, 我们提出了一种随机选择策略交换分组的构造非支配集算法。算法先对每个粒子进行标记, 位于可行域内的粒子标记为 1, 位于不可行域和半可行域内的粒子标记为 0。然后对种群中的粒子进行排序, 从种群中的第一个粒子开始, 如果它标记为 0, 则将它与种群中从后向前数最后一个标记为 1 的粒子进行交换, 并记住被交换粒子的位置  $i$ , 如果它标记为 1 则其位置不变, 比较完一个粒子则依序对下一个粒子进行比较, 并更新被交换粒子的位置  $i$ , 当比较粒子的位置即为  $i$  时, 比较停止, 此时种群被分割成两部分, 前半部分为可行域中的粒子后半部分为不可行域和半可行域中的粒子。由选择算子中的概念可知, 我们只需比较可行域中的粒子就可找到种群中的非支配解集。

对种群的前半部分即标记为 1 的粒子群进行比较, 算法每一次循环都从种群中选择一个个体  $x$  (第一次为随机选择, 从第二次循环开始, 如上一次循环中存在个体  $k$  支配  $x$ , 则  $k$  作为这次循环的划分粒子  $x$ , 中止上次循环比较, 直接进行此次循环比较), 种群中其它个体依次与  $x$  进行比较, 通过一趟比较将种群分割成两部分, 种群的后半部分是被  $x$  支配的个体, 前半部分是支配  $x$  或者与  $x$  不相关的个体, 若  $x$  不被其它任何一个个体支配则将  $x$  并入到非支配集, 接着再对前半部分重复上述过程直到前半部分为空。

算法流程如下:

```
Function Nd
BEGIN
  Q=Pop;
  WHILE(Q ≠ ∅) DO
  BEGIN
    x ∈ Q, 令 Q=Q - {x};
    x_is_nodominated=T;
    FOR(y ∈ Q)DO
```



```

BEGIN
  IF(x > y) THEN Q=Q-{y}
  ELSE IF(y > x) THEN x_is_nodominated=F, x=y;
  END
  IF(x_is_nodominated) THEN Nd=Nd∪{x};
  END
END

```

现对算法进行时间复杂度分析，设种群前半部分的大小为  $n$ ，种群中共有  $m$  个非支配个体。先考察几种特殊情况。第一种特殊情况：第一次比较时产生了一个非支配个体，同时该非支配个体清除了  $(n-m)$  个支配个体，后  $(m-1)$  次比较产生每次产生一个非支配个体，共  $(m-1)$  个非支配个体。这种情况的时间复杂度为： $(n-1)+[(m-2)+(m-3)+\dots+1]=(n-1)+(m-1)(m-2)/2$ ，这是最好的一种情况。第二种特殊情况：前  $(n-m)$  次参加比较的  $(n-m)$  个个体均为支配个体，最坏情况是每个个体不从中清除任何其它支配个体，后  $m$  次比较每次产生一个非支配个体，总共  $m$  个非支配个体。这种情况的时间复杂度为： $(n-1)+(n-2)+\dots+1=n(n-1)/2$ ，这是最坏的一种情况。第三种特殊情况：前  $(m-1)$  次比较产生了  $(m-1)$  个非支配个体，且这  $(m-1)$  个非支配个体没有从构造集中清除一个支配个体，当进行第  $m$  次比较时，产生了第  $m$  个非支配个体，且此时第  $m$  个非支配个体从构造集中清除了所有的支配个体，共  $(n-m)$  个。这种情况的时间复杂度为  $(n-1)+(n-2)+\dots+(n-m)=(2n-m-1)m/2 < nm$ 。

此算法的性能取决于划分的对称性。采用随机的划分基准的选择，从而可以期望划分是较对称的。另外采用了如果上一次循环划分粒子的支配粒子存在则其作为新循环的划分粒子，中止上次循环后面粒子不比较的方法。这样可以更好的减少算法的时间复杂度。因为有效的减少了与划分粒子之间为非支配关系而被划分粒子的支配粒子所支配的粒子的比较次数。

一般情况下，设总共进行了  $k$  次比较， $m \leq k \leq n$ ，每次比较自然淘汰了一个个体，共计淘汰了  $k$  个个体，其中有  $m$  个非支配个体，有  $(k-m)$  个支配个体。 $k$  次比较从构造集中清除了  $(n-k)$  个支配个体，假设每次比较清除了这  $(n-k)$  个支配个体的概率相同，即为  $(n-k)/k$ ，则  $k$  次比较的时间复杂度

为： $(n-1)+(n-2-(n-k)/k)+\dots+(n-k-(k-1)(n-k)/k)=k(n-1)/2+(n-k)/2 < n^2$

故时间复杂度  $T(n)=O(kn)$ ，故本文讨论的构造非支配集的方法比文献<sup>[34]</sup>的  $O(n^2)$  也要好一些。

### 3.3 实验

#### 3.3.1 算法评价方法

评价一个多目标进化算法的性能可以从收敛性、分布性和时间效率三个方面考虑。本文采用了三个量化评价标准：

1. **Generational distance (GD)**: , 用来估计算法的最终解集与全局非劣最优区域的趋近程度, 计算如下:

$$GD = \sqrt{\sum_{i=1}^n d_i^2} / n \quad (3.4)$$

$n$  是解集中个体的数目,  $d_i$  是每个个体到全局非劣最优解的最小欧几里得距离。GD 的值越小就说明解集越靠近全局非劣最优区域, 如果  $GD = 0$  就说明算法的解都在全局非劣最优区域上, 这是最理想的情况。

2. **Spacing (SP)**: 通过计算解集中每个个体与邻居个体的距离变化来评价解集在目标空间的分布情况, 其函数定义如下:

$$SP = \sqrt{\sum_{i=1}^n (\bar{d} - d_i)^2 * 1/(n-1)} \quad (3.5)$$

$n$  是解集中个体的数目,  $d_i$  是每个个体到全局非劣最优解的最小欧几里得距离。 $\bar{d}$  是所有  $d_i$  的平均值。如果  $SP=0$  说明解集中所有个体之间的距离都相等, 分布均匀, SP 的值越小说明解集分布越均匀。

3. **CPU-Time**: 算法运行时间(秒为单位)。

#### 3.3.2 测试函数及实验结果

本文选择了两个有约束 F1、F2 和一个无约束 F3 三个标准测试函数来检测算法的有效性。三个测试函数均在文献<sup>[34][35][36]</sup>中用到, 是用来测试多目标进化算法的标准测试函数。

$$\begin{aligned} \text{F1: } \quad & \min f_1(x) = 2 + (x_1 - 2)^2 + (x_2 - 2)^2 \\ & \min f_2(x) = 9x_1 - (x_2 - 1)^2 \\ & g_1(x) = x_1^2 + x_2^2 \leq 225 \\ & g_2(x) = x_1 - 3x_2 + 10 \leq 0 \\ & \text{其中 } x_1, x_2 \in [-20, 20] \end{aligned}$$

实验循环迭代100次, 粒子数取100, 100次测试的平均结果:  $GD=0.008241$ ,  $SP=0.00193$ ,  $time=14.83s$ 。对比文献<sup>[37]</sup>中线性加权、MOPSO所得结果如图3.3、3.4, 本

文改进的算法优化示意图中的点集分布均匀、粒子数量多。可看出本文提出的方法在逼近程度、分散性上均有较大优势，但在运算时间上略长。如图3.2为100次运算中一次典型优化结果。

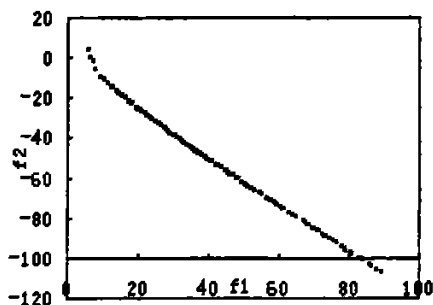


图 3.2 改进的 MOPSO 优化示意图

Fig. 3.2 Improved MOPSO

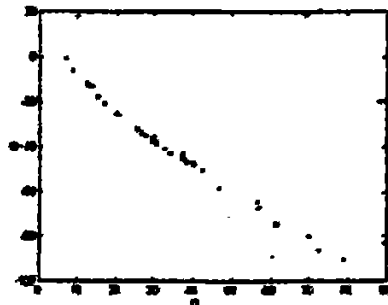


图 3.3 MOPSO 优化示意图

Fig. 3.3 MOPSO

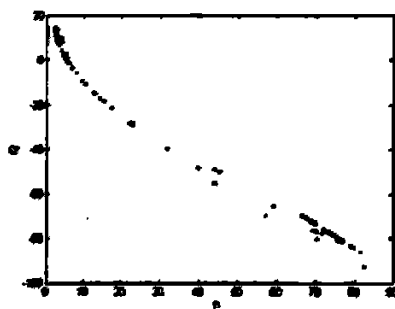


图 3.4 线性加权优化示意图

Fig. 3.4 The weighted linear

当不使用选择算子对 F1 进行迭代优化时，100 次后得到的外部集中平均仅有 73 个解，对比使用选择算子时得到的 132 个解，可得出选择算子对解决有约束问题有较大的帮助。

$$\begin{aligned}
 \text{F2: } & \min f_1(x) = x_1 \\
 & \min f_2(x) = x_2 \\
 & g_1(x) = (x_1 - 0.5)^2 - (x_1 - 0.5) \leq 0.5 \\
 & g_2(x) = 1 + 0.1 \cos(16 \tan^{-1}(x_1/x_2)) - x_1^2 - x_2^2 \leq 0 \\
 & \text{其中 } x_1, x_2 \in [0, 3.14]
 \end{aligned}$$

实验循环迭代100次，粒子数取100，100次测试的平均结果：GD=0.0373，SP=0.0593，time=10.23s。对比文献<sup>[36]</sup>中MOPSO、NSGA II所得结果如图3.6、3.7所示，本文改进

的算法优化示意图中的点集分布均匀、粒子数量多。可看出本文提出的方法在逼近程度、分散性上均有较大优势，但在运算时间上略长。图3.5为100次运算中一次典型优化结果。

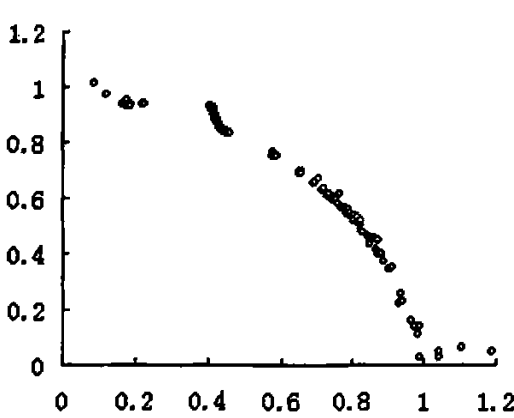


图 3.5 改进的 MOPSO 优化示意图

Fig.3.5 Improved MOPSO

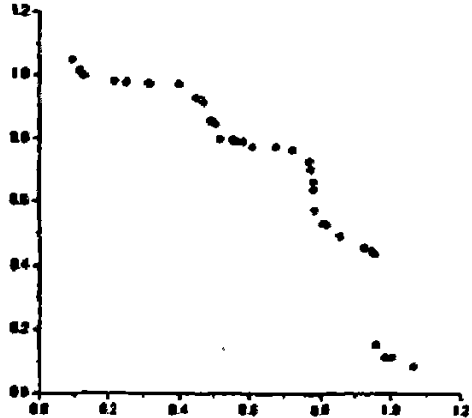


图 3.6 MOPSO 优化示意图

Fig 3.6 MOPSO

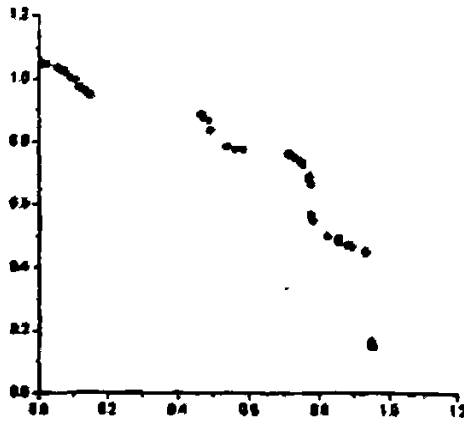


图 3.7 NSGA II 优化示意图

Fig.3.7 NSGA II

当不使用选择算子对 F2 进行迭代优化时，100 次后得到的外部集中仅有 50 个解，对比使用选择算子时得到的 102 个解，再次验证了选择算子对处理有约束问题有较大的帮助。

$$\begin{aligned}
 \text{F3: } & \min f_1(x) = 1 + (A_1 - B_1)^2 + (A_2 - B_2)^2 \\
 & \min f_2(x) = (x_1 + 3)^2 + (x_2 + 1)^2 \\
 & A_1 = 0.5 \sin 1 - 2 \cos 1 + \sin 2 - 1.5 \cos 2
 \end{aligned}$$

$$A_2=1.5\sin 1-\cos 1+2\sin 2-0.5\cos 2$$

$$B_1=0.5\sin x_1-2\cos x_1+\sin x_2-1.5\cos x_2$$

$$B_2=1.5\sin x_1-\cos x_1+2\sin x_2-0.5\cos x_2$$

$$x_1, x_2 \in [-3.14, 3.14]$$

实验循环迭代 100 次，粒子数取 100，100 次测试的平均结果：GD=0.00148，SP=0.0354，time=9.96s。对比文献<sup>[30]</sup>中 MOPSO-fs、MOPSO、NSGA-II 和 PAES 方法的结果可看出本文提出的方法在逼近程度、分散性上均有较大优势，如表 3.1 所示。但在运算时间上差不多。图 3.8 为 100 次运算中一次典型优化结果。

表 3.1 运算结果比较表

Tab. 3.1 Operational results

| 算法 | OMOPSO  | MOPSO-fs | MOPSO   | NSGA-II | PAES    |
|----|---------|----------|---------|---------|---------|
| GD | 0.00148 | 0.00899  | 0.01195 | 0.00346 | 0.02144 |
| SP | 0.0354  | 0.10479  | 0.14831 | 0.09264 | 0.22917 |

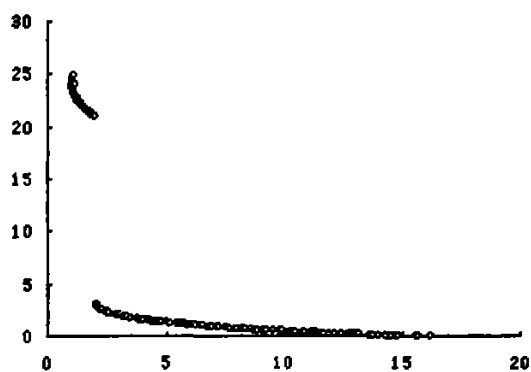


图 3.8 改进的 MOPSO 优化示意图

Fig. 3.8 Improved MOPSO

### 3.4 结论

本文设计的一种解约束问题的多目标混合粒子群算法，采用选择算子来处理约束条件，在选取全局极值和个体极值的过程中加入禁忌算法和拥挤机制，构造非支配集时采用随机选择策略交换分组法，用外部精英集来保存结果。实验表明，本文算法具有较好的收敛性和保持解的分布性，能够有效的解决无约束和有约束多目标优化问题。

## 4 营养配餐计算模型的研究

### 4.1 引言

为了保证正常的生长发育和日常生活需求，人们每日必须摄入含有足够营养素的食物，以满足人体对能量和营养素的正常需求及特殊生理、劳动条件下的营养要求。

合理的膳食营养是维持人体健康的必要条件。人体中营养素缺乏会导致营养不良症，而某些营养素过量不仅会造成浪费，而且会影响其它营养素的吸收，引起各种疾病的产生。人体所需要的能量和各种营养只能通过每天的膳食才能得到补充。我国的膳食结构基本属于“高谷”类型，总体营养水平偏低，尤其摄入的蛋白质质量少，动物蛋白摄入量仍低于世界平均水平，由于各地经济发展、生活水平差距较大，造成了营养过剩与营养不良并存的局面。目前，已知人体必需的营养物质约为五十种左右。但没有一种食物能够按照人体所需要的量和所希望的配比提供营养素。为满足人体的需要，必须摄取多种食物以实现合理的膳食搭配，使人体的营养需要与膳食之间建立平衡。这就要求对人们的膳食结构进行优化，即进行营养配餐。

随着我国经济的发展和社会的进步，人民生活水平逐步提高，人们的饮食已经由温饱型转向营养型。而且不同类的人群对营养的需求不同。随着人类的不断向前发展，越来越要求我们要满足各类人群的差异化、特殊化。例如糖尿病人需要控制糖的摄入量，肥胖人要控制热量的需求量，老人要优先补充维生素等。此外，在满足营养需求的基础上，某人想多食用他喜欢的某种或某类食物等。

我们知道，不同类型的人由于年龄、体重、劳动强度、性别的差异，每日所需摄入的营养标准不同。根据中国营养学会 2000 年编著的《中国居民膳食营养素参考摄入量—推荐的每日膳食中营养素供给量》的标准，按热量、蛋白质、脂肪、钙、铁、锌、硒、视黄醇当量、维生素 E、硫胺素、核黄素、烟酸、抗坏血酸等营养素配餐。笔者以此为普通人群设计营养配餐模型。

为了满足各类人群的多方面的需要，根据各类人群的偏好信息，笔者设计了灵活的营养配餐计算模型。利用多目标粒子群算法得到一组满足各类人群基本营养需求的配餐数据，再根据各类人群的偏好信息，选择出适合此类人群的一个配餐数据。尽量使完全不同类人群营养配餐结果不同。

## 4.2 营养配餐模型及应用实例

### 4.2.1 营养配餐模型

#### 多目标模型

该模型应用于多营养素的普通人群配餐问题。

按照多目标线性规划的步骤：

首先，确定目标约束方程

设  $A$  为  $n$  中食物营养成分组成的集合( $n$  的大小由配餐用户自己决定)，每种食物有  $m$  种营养素，用  $a_{ij}$  表示每  $j$  种食物中每克食物含有第  $i$  种营养素的实际数量，则有

$$A_{m \times n} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} = (a_{ij}), \quad i = 1, 2, \dots, m; \quad j = 1, 2, \dots, n$$

$A$  为约束系数矩阵；

$n$  种食物的实际重量组成的集合用  $X$ ，则  $X_{n \times 1} = (x_1, x_2, \dots, x_n)^T \in R^n$  为决策变量( $x_j$  表示第  $j$  种食物的重量  $x_j \geq 0$ )；

营养素总量用集合  $B$  表示，则  $B$  是期望值向量

$$B_{m \times 1} = (b_1, b_2, \dots, b_m)^T$$

然后，确定目标函数

$$\min z_i = \left| \sum_{j=1}^n a_{ij} \times x_j - b_i \right| \quad i = 1, 2, \dots, m \quad (4.1)$$

所以，共有  $m$  个目标函数，为多目标函数。

### 4.2.2 营养配餐实际应用模型

#### 1. 一餐营养配餐模型

首先，确定目标约束方程

设有  $n$  种食物，每种食物有 13 项营养成分：热量、蛋白质、脂肪、钙、铁、锌、硒、视黄醇当量、维生素 E、硫胺素、核黄素、烟酸、维生素 C。

(1) 约束系数矩阵  $A$ ：

$$A_{m \times n} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} = (a_{ij}), \quad i = 1, 2, \dots, m (m = 13); \quad j = 1, 2, \dots, n$$

(2) 决策变量 X:  $X_{n \times 1} = (x_1, x_2, \dots, x_n)^T \in R^n$ ,  $n$  为决策者预先给出的候选食物种数,  $x_j$  表示第  $j$  种食物的重量。

(3) 期望值向量 B:  $B_{1 \times 13} = (b_1, b_2, \dots, b_{13})^T$ , 其为一餐营养需求量。

然后, 确定目标函数

$$\min z_i = \left| \sum_{j=1}^n a_{ij} \times x_j - b_i \right| \quad i = 1, 2, \dots, 13 \quad (4.2)$$

一餐营养配餐模型分为无偏好信息和有偏好信息两种:

#### a. 无偏好信息

无偏好信息即对于上述 13 种营养素配平用户没有偏好, 对所选中的食物也没有偏好。

无偏好信息下, 在满足目标函数的基础上,  $\min z = \sum_{i=1}^{13} z_i / b_i$

#### b. 有偏好信息

有偏好信息即对于上述 13 种营养素配平用户有偏好, 或者对所选中的食物有偏好。

有偏好信息下, 对 13 种营养素有偏好的, 在满足目标函数的基础上, 需满足

$\max(\min) z_i = \sum_{j=1}^n a_{ij} \times x_j - b_i$ ,  $i$  即配平用户有偏好的营养素。此种偏好有实际的应用前

景。例如肥胖者对热量这种营养素有特殊的偏好, 在满足肥胖者基本营养需求的同时,

要求其摄入的热量尽可能的少。对所选中的食物有偏好的, 在满足目标函数的基础上,

需满足  $\max(\min) z = x_j$ ,  $j$  即配平用户有偏好的食物。此种偏好有实际的应用前景。例

如某个配平用户可能喜欢食用或者不喜欢食用某种食物, 这就要求我们在满足此配平用户营养需求的同时, 要求其喜欢的食物食物量尽可能的多或者不喜欢的食物食物量尽可能的少。

## 2. 一日营养配餐模型

首先, 确定目标约束方程

设有  $n$  种食物, 每种食物有 13 项营养成分: 热量、蛋白质、脂肪、钙、铁、锌、硒、视黄醇当量、维生素 E、硫胺素、核黄素、烟酸、维生素 C。

(1) 约束系数矩阵 A:



$$A_{m \times n} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} = (a_{ij}), \quad i=1,2,\dots,m(m=13); \quad j=1,2,\dots,n$$

(2) 决策变量 X:  $X_{n \times 1} = (x_1, x_2, \dots, x_n)^T \in R^n$ , n 为决策者预先给出的候选食物种数,  $x_j$  表示第 j 种食物的重量。

(3) 期望值向量 B:  $B_{1 \times 3} = (b_1, b_2, \dots, b_{13})^T$ , 其为一日营养需求量。早、中、晚三餐的配比为 3: 4: 3。  $B_1+B_2+B_3=B$ 。

然后, 确定多目标函数:

每日所需营养配平模型:

$$\min z_i = \left| \sum_{j=1}^n a_{ij} \times x_j - b_i \right| \quad i=1,2,\dots,13 \quad (4.3)$$

每餐所需营养配平模型:

$$\min z_{i1} = \left| \sum_{j=1}^n a_{ij} \times x_{j1} - b_{i1} \right| \quad i=1,2,\dots,13 \quad (4.4)$$

$$\min z_{i2} = \left| \sum_{j=1}^n a_{ij} \times x_{j2} - b_{i2} \right| \quad i=1,2,\dots,13 \quad (4.5)$$

$$\min z_{i3} = \left| \sum_{j=1}^n a_{ij} \times x_{j3} - b_{i3} \right| \quad i=1,2,\dots,13 \quad (4.6)$$

其中,  $x_{1j}: x_{2j}: x_{3j}=3: 4: 3; b_{i1}: b_{i2}: b_{i3}=3: 4: 3$

一餐营养配餐模型分为无偏好信息和有偏好信息两种:

a. 无偏好信息

无偏好信息即对于上述 13 种营养素配平用户没有偏好, 对所选中的食物也没有偏好。

无偏好信息下, 在满足目标函数的基础上,  $\min z = \sum_{i=1}^{13} z_i / b_i$

b. 有偏好信息

有偏好信息即对于上述 13 种营养素配平用户有偏好, 或者对所选中的食物有偏好。

有偏好信息下, 对 13 种营养素有偏好的, 在满足目标函数的基础上, 需满足

$\max(\min) z_i = \sum_{j=1}^n a_{ij} \times x_j - b_i$ , i 即配平用户有偏好的营养素。此种偏好有实际的应用前

景。例如肥胖者对热量这种营养素有特殊的偏好，在满足肥胖者基本营养需求的同时，要求其摄入的热量尽可能的少。对所选中的食物有偏好的，在满足目标函数的基础上，需满足  $\max(\min) z = x_j$ ， $j$  即配平用户有偏好的食物。此种偏好有实际的应用前景。例如某个配平用户可能喜欢食用或者不喜欢食用某种食物，这就要求我们在满足此配平用户营养需求的同时，要求其喜欢的食物食物量尽可能的多或者不喜欢食物食物量尽可能的少。

#### 4.3 实现营养配餐计算模型最优解方法

本文前面介绍的多目标优化方法可以分成四类：在优化前决策、在优化后决策、在优化过程中决策和无偏好信息决策。本文所采用的是在优化后决策，即在利用上文改进过的多目标粒子群优化算法优化后，再根据配餐用户的偏好信息进行优化。采用此种优化方式的好处是可以尽可能的与实际优化过程相符，摆脱了传统多目标优化的权重系数的影响，传统营养配餐多目标优化过程多采用利用权重系数多目标变成单目标，最后利用单目标求解的方式，此种方式要给出权重系数，实际应用中权重系数或者并无实际意义或者不好给出。

#### 4.4 营养配餐计算模型应用实例

笔者应用营养配餐计算模型进行实际配餐，具体如下：

实例 1. 配餐目标：以一餐膳食营养量标准进行配餐

表 4.1 营养配餐期望输出量

| 热量(kcal) | 800 | 锌(mg)   | 4   | 核黄素(mg) | 0.3 |
|----------|-----|---------|-----|---------|-----|
| 蛋白质(g)   | 34  | 硒(ug)   | 16  | 烟酸(mg)  | 9   |
| 脂肪(g)    | 18  | 视黄醇(ug) | 250 | 维 C(mg) | 20  |
| 钙(mg)    | 250 | 维 E(mg) | 8   |         |     |
| 铁(mg)    | 8   | 硫胺素(mg) | 0.2 |         |     |

配餐结果：

营养配餐实际输出结果，无偏好信息下由表 4.2 所示

表 4.2 营养配餐实际输出结果(食物重量)

| 馒头(g)  | 米饭(g) | 豆腐(g) | 土豆(g) | 小白菜(g) | 肉鸡(g) |
|--------|-------|-------|-------|--------|-------|
| 125.46 | 250   | 93.94 | 20.04 | 62.5   | 27.58 |

根据表 4.3, 分别计算 125.46(g)馒头, 250(g)米饭, 93.94(g)豆腐, 20.04(g)土豆, 62.5(g)小白菜, 27.58(g)肉鸡的营养成分含量, 如表 4.4 所示:

表 4.3 营养配餐所选食物营养成分(每百克食物所含营养成分量)

Tab. 4.3 Nutrients of Nutrition Catering selected food

|          | 馒头    | 米饭    | 豆腐   | 土豆   | 小白菜   | 肉鸡    |
|----------|-------|-------|------|------|-------|-------|
| 热量(kcal) | 233.0 | 114.0 | 98.0 | 76.0 | 15.0  | 389.0 |
| 蛋白质(g)   | 7.8   | 2.5   | 12.2 | 2.0  | 1.5   | 16.7  |
| 脂肪(g)    | 1.0   | 0.2   | 4.8  | 0.2  | 0.3   | 35.4  |
| 钙(mg)    | 18    | 6.0   | 138  | 8.0  | 90.0  | 37.0  |
| 铁(mg)    | 1.9   | 0.3   | 2.5  | 0.8  | 1.9   | 1.7   |
| 锌(mg)    | 1.01  | 0.47  | 0.63 | 0.37 | 0.51  | 1.1   |
| 硒(ug)    | 9.7   | 0.0   | 1.55 | 0.78 | 1.17  | 5.4   |
| 视黄醇(ug)  | 0.0   | 0.0   | 5.0  | 5.0  | 280.0 | 226.0 |
| 维 E(mg)  | 0.86  | 0.0   | 6.7  | 0.34 | 0.7   | 0.0   |
| 硫胺素(mg)  | 0.05  | 0.02  | 0.05 | 0.08 | 0.02  | 0.07  |
| 核黄素(mg)  | 0.07  | 0.03  | 0.03 | 0.04 | 0.09  | 0.07  |
| 烟酸(mg)   | 0.0   | 1.7   | 0.3  | 1.1  | 0.7   | 13.1  |
| 维 C(mg)  | 0.0   | 0.0   | 0.0  | 27.0 | 28.0  | 0.0   |

[数据来源: 《食物成分表》人民卫生出版社 1991 年出版]

表 4.4 配餐事物具体所含营养成分量与总计

Tab. 4.4 Catering things contained specific nutritional components and the total volume

|          | 馒头    | 米饭    | 豆腐     | 土豆    | 小白菜   | 肉鸡    | 总计     |
|----------|-------|-------|--------|-------|-------|-------|--------|
| 热量(kcal) | 292.3 | 285.0 | 92.0   | 15.2  | 9.3   | 107.2 | 801    |
| 蛋白质(g)   | 9.75  | 6.25  | 11.35  | 0.4   | 0.94  | 4.61  | 33.3   |
| 脂肪(g)    | 1.26  | 0.5   | 4.52   | 0.04  | 0.19  | 9.72  | 16.23  |
| 钙(mg)    | 22.68 | 15    | 129.72 | 1.6   | 56.7  | 10.36 | 236.06 |
| 铁(mg)    | 2.40  | 0.75  | 2.35   | 0.16  | 1.20  | 0.48  | 7.34   |
| 锌(mg)    | 1.28  | 1.18  | 0.59   | 0.08  | 0.32  | 0.31  | 3.76   |
| 硒(ug)    | 12.22 | 0.0   | 1.44   | 0.156 | 0.73  | 1.458 | 16.004 |
| 视黄醇(ug)  | 0.0   | 0.0   | 1.0    | 4.7   | 176.4 | 63.28 | 245.38 |
| 维 E(mg)  | 1.08  | 0.0   | 6.30   | 0.07  | 0.44  | 0.0   | 7.89   |
| 硫胺素(mg)  | 0.06  | 0.05  | 0.045  | 0.016 | 0.012 | 0.018 | 0.201  |
| 核黄素(mg)  | 0.09  | 0.08  | 0.03   | 0.01  | 0.06  | 0.02  | 0.29   |
| 烟酸(mg)   | 0.0   | 4.25  | 0.36   | 0.22  | 0.44  | 3.67  | 8.94   |
| 维 C(mg)  | 0.0   | 0.0   | 0.0    | 5.4   | 16.4  | 0.0   | 22.2   |

由表 4.4 得到实际配餐的营养总量, 将其与期望值对比, 如表 4.5 所示:

表 4.5 营养配餐期望输出量与实际输出量对比

Tab. 4.5 Nutrition Catering expect output compared with the actual output

|          | 期望输出配餐量 | 实际输出配餐量 | 配餐偏差   | 误差(%) |
|----------|---------|---------|--------|-------|
| 热量(kcal) | 800     | 801     | 1      | 0.125 |
| 蛋白质(g)   | 34      | 33.3    | 0.7    | 2.05  |
| 脂肪(g)    | 18      | 16.23   | -1.77  | 9.83  |
| 钙(mg)    | 250     | 236.06  | -13.94 | 5.576 |
| 铁(mg)    | 8       | 7.34    | -0.66  | 8.25  |
| 锌(mg)    | 4       | 3.76    | -0.24  | 6.0   |
| 硒(ug)    | 16      | 16.004  | 0.004  | 0.025 |
| 视黄醇(ug)  | 250     | 245.38  | -4.62  | 1.848 |
| 维 E(mg)  | 8       | 7.89    | -0.11  | 1.375 |
| 硫胺素(mg)  | 0.2     | 0.201   | 0.001  | 0.5   |
| 核黄素(mg)  | 0.3     | 0.29    | -0.01  | 3.33  |
| 烟酸(mg)   | 9       | 8.94    | -0.06  | 0.66  |
| 维 C(mg)  | 20      | 22.2    | 2.2    | 11    |

营养配餐实际输出结果, 有偏好信息, 偏好信息为热量, 要求在满足各种营养需求的同时, 摄入的热量最少, 结果由表 4.6 所示

表 4.6 营养配餐实际输出结果(食物重量)

Tab. 4.6 Nutrition Catering actual output results

| 馒头(g)  | 米饭(g)  | 豆腐(g)  | 土豆(g) | 小白菜(g) | 肉鸡(g) |
|--------|--------|--------|-------|--------|-------|
| 120.65 | 181.26 | 110.76 | 17.78 | 68.64  | 39.61 |

根据表 4.3, 分别计算 120.65(g)馒头, 181.26(g)米饭, 110.76(g)豆腐, 17.78(g)土豆, 68.64(g)小白菜, 39.61(g)肉鸡的营养成分含量, 如表 4.7 所示:

由表 4.7 得到实际配餐的各种营养元素含量, 将其与期望值对比, 计算其配餐偏差及误差如表 4.8 所示。各种营养元素的误差量与无偏好信息比较大部分均有一定的增大, 部分营养元素的误差量增长较大, 这是由于在满足摄入热量最少的同时, 以误差总量为代价, 使得误差总量有了一定的增长。但各种营养素的误差范围在可允许的范围之内, 因此认为得到的实验数据是可行的, 有效的, 是符合实际应用的。与此类似, 可以选择任意一种营养元素作为偏好信息。

表 4.7 配餐事物具体所含营养成分与总计

Tab. 4.7 Catering things contained specific nutritional components and the total volume

|          | 馒头     | 米饭     | 豆腐     | 土豆    | 小白菜    | 肉鸡     | 总计     |
|----------|--------|--------|--------|-------|--------|--------|--------|
| 热量(kcal) | 281.11 | 206.63 | 108.54 | 13.51 | 10.29  | 154.08 | 774.16 |
| 蛋白质(g)   | 9.41   | 4.53   | 13.51  | 0.35  | 1.02   | 6.61   | 35.43  |
| 脂肪(g)    | 1.20   | 0.36   | 5.31   | 0.03  | 0.20   | 14.0   | 21.11  |
| 钙(mg)    | 21.71  | 10.87  | 152.84 | 1.42  | 61.77  | 14.65  | 263.26 |
| 铁(mg)    | 2.30   | 0.55   | 2.77   | 0.15  | 1.30   | 0.68   | 7.75   |
| 锌(mg)    | 1.22   | 0.85   | 0.70   | 0.07  | 0.35   | 0.44   | 3.63   |
| 硒(ug)    | 11.70  | 0.0    | 1.71   | 0.13  | 0.80   | 2.13   | 16.47  |
| 视黄醇(ug)  | 0.0    | 0.0    | 5.53   | 0.88  | 192.19 | 89.51  | 288.11 |
| 维 E(mg)  | 1.03   | 0.0    | 7.42   | 0.06  | 0.48   | 0.0    | 8.99   |
| 硫胺(mg)   | 0.06   | 0.04   | 0.05   | 0.01  | 0.01   | 0.03   | 0.20   |
| 核黄素(mg)  | 0.09   | 0.06   | 0.03   | 0.01  | 0.06   | 0.03   | 0.28   |
| 烟酸(mg)   | 0.0    | 3.08   | 0.33   | 0.19  | 0.48   | 5.18   | 9.26   |
| 维 C(mg)  | 0.0    | 0.0    | 0.0    | 4.8   | 19.21  | 0.0    | 24.01  |

表 4.8 营养配餐期望输出量与实际输出量对比

Tab. 4.8 Nutrition Catering expect output compared with the actual output

|          | 期望输出配餐量 | 实际输出配餐量 | 配餐偏差   | 误差(%) |
|----------|---------|---------|--------|-------|
| 热量(kcal) | 800     | 774.16  | -25.84 | 3.23  |
| 蛋白质(g)   | 34      | 35.43   | 1.43   | 4.20  |
| 脂肪(g)    | 18      | 21.11   | 3.11   | 17.27 |
| 钙(mg)    | 250     | 263.26  | 13.26  | 5.304 |
| 铁(mg)    | 8       | 7.75    | -0.25  | 3.125 |
| 锌(mg)    | 4       | 3.63    | -0.37  | 9.25  |
| 硒(ug)    | 16      | 16.47   | 0.47   | 2.93  |
| 视黄醇(ug)  | 250     | 288.11  | 38.11  | 15.24 |
| 维 E(mg)  | 8       | 8.99    | 0.99   | 12.37 |
| 硫胺素(mg)  | 0.2     | 0.20    | 0.00   | 0.0   |
| 核黄素(mg)  | 0.3     | 0.28    | -0.02  | 6.66  |
| 烟酸(mg)   | 9       | 9.26    | 0.26   | 2.88  |
| 维 C(mg)  | 20      | 24.01   | 4.01   | 18.22 |

营养配餐实际输出结果,有偏好信息,偏好信息为肉鸡,要求在满足各种营养需求的同时,食用的肉鸡量最多,结果由表 4.9 所示

表 4.9 营养配餐实际输出结果(食物重量)

Tab.4.9 Nutrition Catering actual output results

| 馒头(g)  | 米饭(g)  | 豆腐(g)  | 土豆(g) | 小白菜(g) | 肉鸡(g) |
|--------|--------|--------|-------|--------|-------|
| 110.37 | 150.02 | 100.83 | 27.31 | 46.34  | 56.64 |

根据表 4.3, 分别计算 110.37(g)馒头, 150.02(g)米饭, 100.83(g)豆腐, 27.31(g)土豆, 46.34(g)小白菜, 56.64(g)肉鸡的营养成分含量, 如表 4.10 所示。

表 4.10 配餐事物具体所含营养成分量与总计

Tab. 4.10 Catering things contained specific nutritional components and the total volume

|          | 馒头     | 米饭     | 豆腐     | 土豆    | 小白菜    | 肉鸡     | 总计     |
|----------|--------|--------|--------|-------|--------|--------|--------|
| 热量(kcal) | 257.16 | 171.02 | 98.82  | 20.75 | 6.95   | 220.33 | 775.03 |
| 蛋白质(g)   | 8.60   | 3.75   | 12.30  | 0.54  | 0.69   | 9.45   | 35.33  |
| 脂肪(g)    | 1.10   | 0.30   | 4.84   | 0.05  | 0.13   | 19.82  | 26.11  |
| 钙(mg)    | 19.87  | 9.00   | 139.15 | 2.19  | 41.70  | 20.96  | 232.87 |
| 铁(mg)    | 2.10   | 0.45   | 2.52   | 0.22  | 0.88   | 0.96   | 7.13   |
| 锌(mg)    | 1.12   | 0.71   | 0.64   | 0.10  | 0.24   | 0.63   | 3.44   |
| 硒(ug)    | 10.70  | 0.0    | 1.56   | 0.22  | 0.54   | 3.05   | 16.06  |
| 视黄醇(ug)  | 0.0    | 0.0    | 5.04   | 1.36  | 129.74 | 128.00 | 264.14 |
| 维 E(mg)  | 0.94   | 0.0    | 6.75   | 0.09  | 0.32   | 0.0    | 8.1    |
| 硫胺素(mg)  | 0.05   | 0.03   | 0.05   | 0.02  | 0.01   | 0.04   | 0.20   |
| 核黄素(mg)  | 0.08   | 0.05   | 0.03   | 0.01  | 0.05   | 0.04   | 0.26   |
| 烟酸(mg)   | 0.0    | 2.55   | 0.30   | 0.30  | 0.32   | 7.41   | 10.88  |
| 维 C(mg)  | 0.0    | 0.0    | 0.0    | 7.37  | 12.97  | 0.0    | 20.34  |

由表 4.10 得到实际配餐的各种营养元素含量, 将其与期望值对比, 计算其配餐偏差及误差如表 4.11 所示。各种营养元素的误差量与无偏好信息比较大一部分均有一定的增大, 部分营养元素的误差量增长较大, 这是由于在满足食用肉鸡量最多的同时, 以误差总量为代价, 使得误差总量有了一定的增长。但各种营养素的误差范围在可允许的范围之内, 因此认为得到的实验数据是可行的, 有效的, 是符合实际应用的。与此类似, 可以选择任意一种食物作为偏好信息进行决策求解, 使偏好的食物的食物量最多和最少, 在满足基本营养摄入的基础上, 满足不同人对食物口味的需求。使配餐结果更加的人性化, 多样化, 满足不同人群的多样需求。以此为基础, 也可选择任意两种和两种以上食物作为偏好信息进行决策求解。

表 4.11 营养配餐期望输出量与实际输出量对比

Tab. 4.11 Nutrition Catering expect output compared with the actual output

|          | 期望输出配餐量 | 实际输出配餐量 | 配餐偏差   | 误差(%)  |
|----------|---------|---------|--------|--------|
| 热量(kcal) | 800     | 775.03  | -24.97 | 3.12   |
| 蛋白质(g)   | 34      | 35.33   | 1.33   | 3.91   |
| 脂肪(g)    | 18      | 26.11   | 8.11   | 45.05  |
| 钙(mg)    | 250     | 232.87  | -67.13 | 26.852 |
| 铁(mg)    | 8       | 7.13    | -0.87  | 10.875 |
| 锌(mg)    | 4       | 3.44    | -0.56  | 14     |
| 硒(ug)    | 16      | 16.06   | 0.06   | 0.375  |
| 视黄醇(ug)  | 250     | 264.14  | 38.11  | 15.24  |
| 维 E(mg)  | 8       | 8.10    | 0.1    | 1.25   |
| 硫胺素(mg)  | 0.2     | 0.20    | 0.00   | 0.0    |
| 核黄素(mg)  | 0.3     | 0.26    | -0.04  | 13.33  |
| 烟酸(mg)   | 9       | 10.88   | 1.88   | 20.88  |
| 维 C(mg)  | 20      | 20.34   | 0.34   | 1.7    |

实例 2. 以一日膳食营养量标准进行配餐

表 4.12 营养配餐期望输出量

Tab. 4.12 Expect output of Nutrition Catering

|          |      |         |     |         |     |
|----------|------|---------|-----|---------|-----|
| 热量(kcal) | 2400 | 锌(mg)   | 15  | 核黄素(mg) | 1.2 |
| 蛋白质(g)   | 80   | 硒(ug)   | 50  | 烟酸(mg)  | 25  |
| 脂肪(g)    | 60   | 视黄醇(ug) | 800 | 维 C(mg) | 60  |
| 钙(mg)    | 800  | 维 E(mg) | 10  |         |     |
| 铁(mg)    | 20   | 硫胺素(mg) | 1.2 |         |     |

配餐结果:

营养配餐实际输出结果, 无偏好信息下由表 4.14 所示

表 4.14 营养配餐实际输出结果(食物重量)

Tab. 4.14 Nutrition Catering actual output results

| 馒头(g)  | 米饭(g)  | 烙饼(g)  | 菠菜(g) | 土豆(g)  | 肉鸡(g) | 鸡蛋粉   |
|--------|--------|--------|-------|--------|-------|-------|
| 229.87 | 550.81 | 215.57 | 56.02 | 171.28 | 96.98 | 51.61 |

根据表 4.13, 分别计算 229.87 (g)馒头, 550.81 (g)米饭, 215.57 (g)烙饼, 56.02 (g)菠菜, 171.28 (g)土豆, 96.98 (g)肉鸡 51.61(g)鸡蛋粉的营养成分含量, 如表 4.15 所示。

表 4.13 营养配餐所选食物营养成分(每百克食物所含营养成分量)

Tab.4.13 Nutrients of Nutrition Catering selected food

|          | 馒头    | 米饭    | 烙饼    | 土豆   | 菠菜    | 肉鸡   | 鸡蛋粉   |
|----------|-------|-------|-------|------|-------|------|-------|
| 热量(kcal) | 233.0 | 114.0 | 255.0 | 76.0 | 24.0  | 76.0 | 545.0 |
| 蛋白质(g)   | 7.8   | 2.5   | 7.5   | 2.0  | 2.6   | 2.0  | 43.4  |
| 脂肪(g)    | 1.0   | 0.2   | 2.3   | 0.2  | 0.3   | 0.2  | 36.2  |
| 钙(mg)    | 18    | 6.0   | 20.0  | 8.0  | 66.0  | 8.0  | 954.0 |
| 铁(mg)    | 1.9   | 0.3   | 2.4   | 0.8  | 2.9   | 0.8  | 10.5  |
| 锌(mg)    | 1.01  | 0.47  | 0.94  | 0.37 | 0.85  | 0.37 | 5.95  |
| 硒(ug)    | 9.7   | 0.0   | 7.5   | 0.78 | 0.97  | 0.78 | 39.1  |
| 视黄醇(ug)  | 0.0   | 0.0   | 0.0   | 5.0  | 487.0 | 5.0  | 525.0 |
| 维 E(mg)  | 0.86  | 0.0   | 1.03  | 0.34 | 1.74  | 0.34 | 11.56 |
| 硫胺素(mg)  | 0.05  | 0.02  | 0.02  | 0.08 | 0.04  | 0.08 | 0.05  |
| 核黄素(mg)  | 0.07  | 0.03  | 0.04  | 0.04 | 0.11  | 0.04 | 0.4   |
| 烟酸(mg)   | 0.0   | 1.7   | 0.0   | 1.1  | 0.6   | 1.1  | 0.0   |
| 维 C(mg)  | 0.0   | 0.0   | 0.0   | 27.0 | 32.0  | 27.0 | 0.0   |

[数据来源:《食物成分表》人民卫生出版社 1991 年出版]

表 4.15 配餐事物具体所含营养成分量与总计

Tab. 4.15 Catering things contained specific nutritional components and the total volume

|          | 馒头     | 米饭     | 烙饼     | 菠菜     | 土豆     | 肉鸡     | 鸡蛋粉    | 总计      |
|----------|--------|--------|--------|--------|--------|--------|--------|---------|
| 热量(kcal) | 478.13 | 627.93 | 549.71 | 13.44  | 130.17 | 377.26 | 281.27 | 2457.91 |
| 蛋白质(g)   | 14.25  | 13.77  | 16.16  | 1.45   | 3.42   | 16.19  | 22.39  | 87.63   |
| 脂肪(g)    | 2.75   | 1.10   | 4.95   | 0.16   | 0.34   | 34.33  | 18.68  | 62.31   |
| 钙(mg)    | 133.33 | 33.05  | 43.12  | 36.98  | 13.71  | 35.89  | 492.37 | 788.45  |
| 铁(mg)    | 3.91   | 1.66   | 5.18   | 1.63   | 1.38   | 1.65   | 5.42   | 20.83   |
| 锌(mg)    | 0.92   | 2.59   | 2.03   | 0.48   | 0.64   | 1.07   | 3.08   | 10.80   |
| 硒(ug)    | 16.55  | 0.0    | 16.16  | 0.54   | 1.33   | 5.23   | 20.17  | 59.98   |
| 视黄醇(ug)  | 0.0    | 0.0    | 0.0    | 272.83 | 8.57   | 219.19 | 270.96 | 771.55  |
| 维 E(mg)  | 0.21   | 0.0    | 2.23   | 0.98   | 0.59   | 0.0    | 5.97   | 9.98    |
| 硫胺素(mg)  | 0.05   | 0.12   | 0.05   | 0.03   | 0.14   | 0.07   | 0.03   | 0.49    |
| 核黄素(mg)  | 0.05   | 0.17   | 0.09   | 0.07   | 0.07   | 0.07   | 0.21   | 0.73    |
| 烟酸(mg)   | 0.0    | 9.37   | 0.0    | 0.34   | 1.87   | 12.71  | 0.0    | 24.31   |
| 维 C(mg)  | 0.0    | 0.0    | 0.0    | 17.92  | 46.24  | 0.0    | 0.0    | 64.16   |

由表 4.15 得到实际配餐的各种营养元素含量, 将其与期望值对比, 计算其配餐偏差及误差如表 4.16 所示。各种营养元素的误差量与一餐无偏好信息比较大部分均有一定的



增大,部分营养元素的误差量增长较大,这是由于所选择的食物量增多造成了误差的增大。但各种营养素的误差范围在可允许的范围之内,因此认为得到的实验数据是可行的,有效的,是符合实际应用的。

表 4.16 营养配餐期望输出量与实际输出量对比

Tab. 4.16 Nutrition Catering expect output compared with the actual output

|          | 期望输出配餐量 | 实际输出配餐量 | 配餐偏差   | 误差(%) |
|----------|---------|---------|--------|-------|
| 热量(kcal) | 2400    | 2457.91 | 57.91  | 2.41  |
| 蛋白质(g)   | 80      | 87.63   | 7.63   | 9.53  |
| 脂肪(g)    | 60      | 62.31   | 2.31   | 3.85  |
| 钙(mg)    | 800     | 788.45  | -12.55 | 1.56  |
| 铁(mg)    | 20      | 20.83   | 0.83   | 4.15  |
| 锌(mg)    | 15      | 10.80   | -4.20  | 28    |
| 硒(ug)    | 50      | 59.98   | 9.98   | 19.96 |
| 视黄醇(ug)  | 800     | 771.55  | -28.45 | 3.55  |
| 维 E(mg)  | 10      | 9.98    | 0.02   | 0.2   |
| 硫胺素(mg)  | 1.2     | 0.49    | -0.71  | 59.1  |
| 核黄素(mg)  | 1.2     | 0.73    | -0.47  | 39.1  |
| 烟酸(mg)   | 25      | 24.31   | -0.69  | 2.76  |
| 维 C(mg)  | 60      | 64.16   | 4.16   | 6.93  |

结论:

- (1) 从表 4.5 可以看出,营养配餐的误差非常小,最大误差 11%,平均误差为 3.88%,其结果可信度高;
- (2) 从表 4.8 可以看出,营养配餐的误差非常小,最大误差 17.27%,平均误差为 6.31%,其结果可信度高;
- (3) 从表 4.11 可以看出,除脂肪、钙外,营养配餐的总体误差非常小,平均误差为 7.66%,其结果可信度高;
- (4) 从表 4.16 可以看出,除硫胺素、核黄素外,营养配餐的总体误差非常小,平均误差为 7.53%,其结果可信度高。

误差分析:在配餐过程中,由于硫胺素、核黄素基数很小,而实际配餐差额并不大,由于基数小,在计算过程中四舍五入影响计算精度造成。表 4.8,表 4.11 的误差额比表 4.5 的误差额大,是由于有偏好信息的加入,优先满足了偏好信息,使误差额比无偏好

信息时略大。表 4.16 的误差额比表 4.5 的误差额大，是由于多了一个决策变量，使算法误差加大。但从总体而言，配餐结果令人满意。

#### 4.5 本章小结

笔者的营养配餐计算模型为多目标模型。在实际应用中，根据中国营养学会 2000 年编著的《中国居民膳食营养素参考摄入量—推荐的每日膳食中营养素供给量》的标准，按热量、蛋白质、脂肪、钙、铁、锌、硒、视黄醇当量、维生素 E、硫胺素、核黄素、烟酸、抗坏血酸等营养素配餐，提出了适用于各类人群的一餐营养配餐模型和一日营养配餐模型。

笔者提出了新的营养配餐模型，与传统的营养配餐模型最大的不同之处：

传统的营养配餐模型多为经济目标函数，主要考虑食物的价格因素；随着人民生活水平逐步提高，人们的饮食已经由温饱型转向营养型，而且不同类的人群对营养的需求不同，随着人类的不断向前发展，越来越要求我们要满足各类人群的差异化、特殊化。例如糖尿病人需要控制糖的摄入量，肥胖人要控制热量的需求量，老人要优先补充维生素等。此外，在满足营养需求的基础上，某人想多食用他喜欢的某种或某类食物等。为了满足各类人群的多方面的需要，根据各类人群的偏好信息，笔者设计了灵活的营养配餐计算模型。尽量满足各类人群营养配餐需求。

由于营养配餐计算模型属于多目标线性规划问题，实际求解过程复杂，笔者将多目标粒子群算法应用于求多目标线性规划的最优解。

最后，笔者应用营养配餐计算模型进行实际配餐，分别以一餐营养量和一日营养量为目标，应用集成到决策系统中营养配餐计算模型进行配餐，从总体上看，配餐结果令人满意。

## 5 营养配餐决策支持系统的设计与实现

### 5.1 引言

我国用计算机进行营养配餐和菜肴评价大约兴起于上世纪 90 年代。北方工业大学和北京营养研究所于 1991 年研制的“大学生营养配餐电脑系统”；四川一研究所研制的“卫生导餐系统”于 1997 年通过国家鉴定；中国保健食品研究与应用多媒体信息系统的建立，其应用 web 技术建立保健食品查询系统；陈艳秋等研究的营养膳食分析与配制营养软件的设计及应用等。除各大学校及科研机构对电脑配餐系统进行研究实现外，有许多公司企业也进行这方面的研究，包括营养博士配餐软件，“配餐大师”配餐软件等有一系列的软件问世。

虽然这方面的研究进行了很长时间，但还存在着各种各样的缺陷：

- 1、功能单一，多为单机版，基于 web 或群体机关开发的人很少
- 2、界面不够友好，人机交互不够充分
- 3、系统设计不够合理，多为两层 c/s 结构
- 4、配菜时应用的优化模型不够合理，多采用原料的最优价格，采用单纯行法求解。
- 5、营养膳食指南和原料菜色多一成不变，不方便添加、删除和修改

在本章中先介绍营养配餐决策支持系统的设计和开发方面的研究工作，主要介绍营养配餐模块的设计与实现。

### 5.2 营养配餐决策支持系统的整体设计

#### 5.2.1 系统的整体架构设计

本系统采用分布式多层架构 DSS 系统设计，为利用 javabean 和 EJB 组件模型开发的基于 web 服务的一款软件。

本系统体系结构采用 B/S 模式，即 Browser/Server(浏览器/服务器)结构，就是只安装维护一个服务器(Server)，而客户端采用浏览器(Browse)运行软件。它是随着 Internet 技术的兴起，对 C/S 结构的一种变化和改进。在 B/S 体系结构系统中，用户通过浏览器向服务器发出请求，服务器对浏览器的请求进行处理，将用户所需信息返回到浏览器。而其余如数据请求、加工、结果返回以及动态网页生成、对数据库的访问和应用程序的执行等工作全部由 Web Server 完成。随着 Windows 将浏览器技术植入操作系统内部，

这种结构已成为当今应用软件的首选体系结构。B/S 结构的主要特点是分布性强、维护方便、开发简单且共享性强、总体拥有成本低。具体体系结构示意图如图 6.1 所示。



图 5.1 系统体系结构

Fig. 5.1 Systematic system structure

这种三层体系结构具备客户/服务器结构的分布计算特性,用户访问的信息存储于数据库中,Web 服务器则负责信息显示、数据处理和逻辑运算,作为整个系统的控制核心,服务期端还需要考虑流量负荷、系统安全性以及响应速度等问题。本系统的实现则采用个人主机充当服务器,模拟用户远程访问本机页面,并出发业务逻辑,数据库采用 SQLServer 数据库。

### 5.2.2 系统设计目标

本系统主要面向广大的营养配餐者,为用户提供全方面、准确的营养配餐信息。系统提供两种营养配餐模式:(1)一餐营养配平模型;(2)一日营养配平模型。在所有配平模式下,用户选择所想要食用的食物,作为营养配平的决策变量,输入配平者的基本信息,以便于从营养库中查询配餐者所需的营养信息作为期望值,输入偏好信息作为决策支持,建立营养配餐计算模型,经求解计算得出用户食用食物的食物量,给配平者作为信息参考。同时,系统可以使管理员利用一定的管理权限通过页面直接对数据库中的数据进行添加、查找、删除。方便了管理员对成菜营养信息、人体营养信息、材料信息等信息的管理。本系统在界面设计上注意美观大方、操作简单、界面友好,以方便普通用户的操作。

### 5.2.3 系统的整体框架

如图 5.2 为营养配餐 DSS 的总体结构框架图,该 DSS 由以下模块组成:数据库管理模块,人机交互模块和模型库管理系统。

- (1) 数据库管理模块主要是管理和维护数据库的数据:数据库包括
- 原料营养库:存放常用食物的各种营养成分;
  - 成菜库:用来存放各种成菜的构成原料;
  - 成菜营养库:用来存放各种成菜的营养含量;

成菜价格库：用来存放成菜的价目表；

人体供给量标准库：用来存放各类人群的营养标准；

配方库：用来存放经系统决策后提供的菜谱数据；

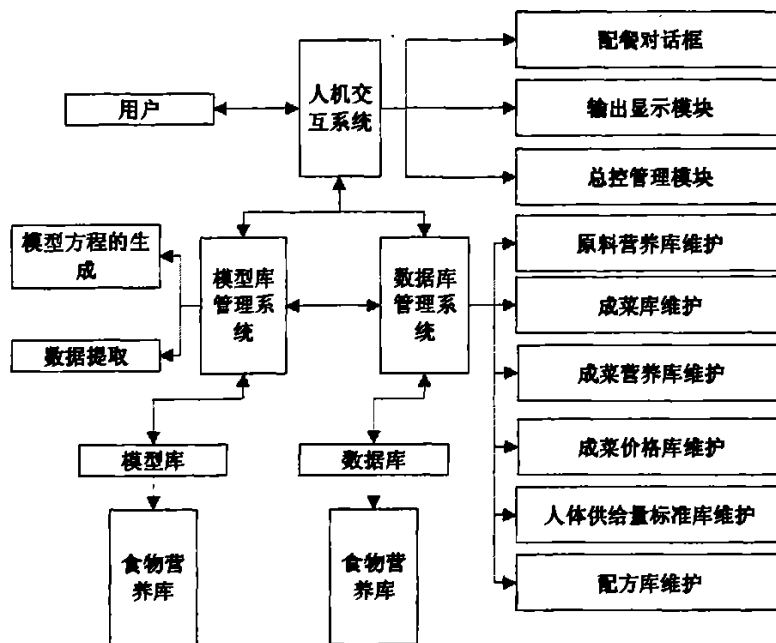


图 5.2 总体结构框架图

Fig. 5.2 Overall framework structure

- (2) 模型库管理系统，一是管理各种模型，二是他要从其他模块中接受相关数据，根据模型生成模型方程，求得配餐结果。模型库中存放了营养配餐决策的各种模型：包括单目标模型，多目标模型，营养评价模型和成菜营养计算模型，模型的接口数据来自于用户输入的信息和从数据库中取得的数据，模型字典存放模型的名字和模型程序名等说明性信息。
- (3) 人机交互系统：给用户提供的会话，数据的录入、更改与删除，显示结果。

### 5.3 营养配餐模块的设计与实现

#### 5.3.1 营养配餐模块设计

营养配餐模块主要包含如下几个过程：

- (1) 数据输入：通过会话屏幕输入各种参数和用户信息，主要包括配餐者所选择的

食物，配餐类型，偏好信息，配餐者的性别、年龄、劳动强度等决定配餐者营养需求的信息。

(2) 目标方程的形成：根据用户信息从数据库中提取相应的数据，从营养成分数据表和供给标准表中查到有关数据生成配餐多目标方程，自动形成目标规划方程。

(3) 求解：启动计算模型，应用多目标粒子群优化算法及偏好信息，求出满足配餐者需求的最优解。输出配餐结果并把菜单结果存入配方库。

营养配餐模块流程如图 5.3 所示。

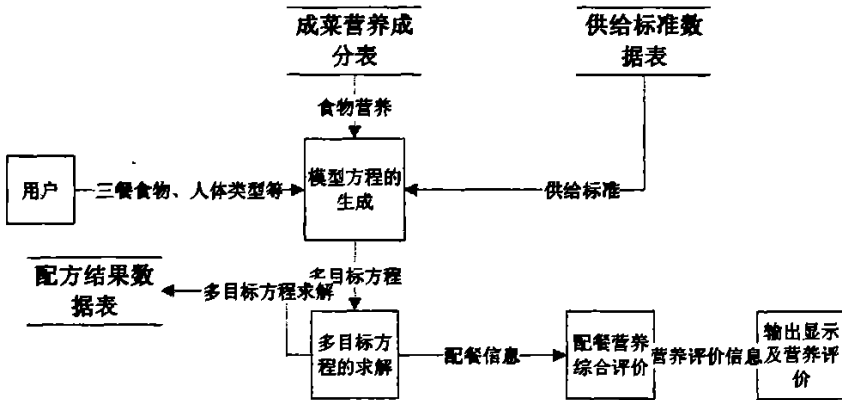


图 5.3 营养配餐模块数据流图

Fig. 5.3 flow chart of Nutrition Catering Module

### 5.3.2 营养配餐模块功能实现

(1) 数据输入：通过会话屏幕输入各种参数和用户信息，主要包括配餐者所选择的食物，配餐类型，偏好信息，配餐者的性别、年龄、劳动强度等决定配餐者营养需求的信息，其主要界面如下图 5.4 所示：



图 5.4 数据输入示意图

Fig. 5.4 Data input

配餐者在左下方选择要选择的食品，可选择食品列表中的可选食品的各种信息均事先存储在数据库中，管理员可通过调用数据库管理模块对其进行添加、删除或者修改。点击提交按钮提交。提交所选食品后的处理结果如图 5.5 所示。



图 5.5 数据输入示意图 2

Fig. 5.5 Data input 2

提交的所选食品信息即确定了营养配餐计算模型的决策变量，再次提交会从原料数据库中查找所选食品的营养含量值，并传值给营养配餐计算模型中的多目标方程。然后提交配餐者个人信息以此从人体供给量标准库查找符合用户个人信息的供给标准，作为营养配餐计算模型中多目标方程中的期望值。根据偏好信息，从模型库中选择合适的计算模型进行营养配平，得到最终的配餐结果如图 5.6 所示。



图 5.6 配餐结果示意图

Fig. 5.6 Catering results

## 结 论

### 论文工作总结

在科学研究、工程实践及社会生产和经济发展中的许多优化问题都是多目标优化问题。在同样条件下,经过优化技术的处理,对系统效率的提高、能耗的降低、资源的合理利用及经济效益的提高等均有显著的效果。优化技术存在较广泛的应用前景。

由于多目标优化问题不存在唯一的全局最优解,所以求解多目标优化问题实际上是寻找一个解的集合(Pareto 最优解集)。进化算法的出现为那些难以找到传统数学模型的多目标优化难题指出了一条新的出路。本文研究的粒子群优化算法(particle swarm optimization)是在 1995 年由美国社会心理学家 Kennedy 和电气工程师 Eberhart 共同提出的。它收敛速度快、易于实现且仅有少量参数需要调整,因而一经提出就成为进化计算领域的一个新的研究热点,目前已经被广泛应用于多目标优化问题中。

本文主要有以下四个创新点:

1. 大多数多目标优化问题均含有约束条件,大部分约束优化问题的解是在约束边界取得,因此求解约束优化问题的难点在于对约束条件的处理。本文引入半可行域的概念,改进了适应度函数的求解方式,用以处理约束条件。从而使约束条件的处理更为合理,满足实际的求解需要。

2. 全局最优值的选取在多目标粒子群算法中非常重要,直接关系到粒子的更新更直接影响到算法的收敛速度和解集的分布性,恰当的选取可以避免算法过早的陷入局部最优。本文采用外部集保存当前找到的最优解集,从外部集中随机选择粒子的全局最优值,前期采用拥挤机制用于粒子的全局最优值的选取,在外部集中粒子的分布日趋均匀的情况下采用禁忌算法用于粒子的全局最优值的选取。从而使全局最优值的选取更为合理,保持了解的分布性。

3. 在算法的每次迭代过程中均须构造非支配集,因此非支配集的构造效率直接影响到整个算法的时间复杂度,传统的构造非支配集的方法时间复杂度较高,本文采用了随机选择策略交换分组的非支配集构造方法,尽量以上一次循环的划分粒子的支配粒子作为此次循环的划分粒子,提高了构造非支配集的效率,减少了算法的时间复杂度。

4. 在实际应用中,多目标粒子群算法的应用范围较小。本文提出了新的营养配餐计算模型,将提出的新的多目标粒子群算法应用到新的营养配餐模型中,针对营养配餐的特点设计并实现了基于 Web 的新的营养配餐模型。通过与传统营养配餐模型和求解



方式的实验结果进行比对，验证了本文提出的新的多目标粒子群算法的可行性和有效性，拓展了新的多目标粒子群算法的应用领域。

## 论文展望

多目标粒子群优化算法理论研究及营养配餐决策支持系统的设计实现涉及多方面的理论、方法和技术，本论文对其中的一些关键问题进行了有益的研究和探讨，但还有许多问题需要进一步的研究。

1. 多目标粒子群优化的改进设计中，在全局极值的选取时，需根据外部集解的分布情况来判断应用拥挤机制还是禁忌算法，本文中没能给出一个适用于普遍意义上的判定值，只能根据实验经验给出一个大体值。普遍的恰当的判定值还需要进一步的研究得出。

2. 将多目标粒子群优化算法应用到营养配餐计算模型中，随着选取食物种数的增加，计算结果的误差加大，即将多目标粒子群优化算法应用到多维求解中还有待进一步的研究。

3. 营养配餐计算模型中多目标优化函数的建立，由于笔者的专业知识有限，不能给出十分合适的配餐期望值和配餐误差。应由专业人员指导，给出合适的配餐误差及给每类特定人群设定专业的满足需求的配餐期望值。根据营养配餐的特殊需要，给出食物相生相克的属性，应用到营养配餐当中。得到的配餐结果过于具体，应用到实际的日常饮食中怎样处理，还有待研究。

虽然本文的研究工作取得了一定的成绩，但是由于粒子群优化算法及营养配餐是一个相对崭新的研究领域，在理论和应用中还有很多方面需要进一步的研究。另外，由于本人的领域知识有限，可用的数据和实验工具的匮乏，所以还需进一步进行学习，努力解决在理论和实际应用中遇到的难题。

## 参 考 文 献

- [1] 玄光男, 程润伟 著, 于歆杰, 周根贵 译, 遗传算法与工程优化, 清华大学出版社, 2004
- [2] Pareto V. Cours. DEconomies Politique, volume I and II. Rouge, Lausanne, 1896
- [3] 汪晗. 基于进化计算的多目标优化与决策方法研究[硕士学位论文]. 中国人民解放军国防科学技术大学, 2002
- [4] J. D. Schaffer. Multiple Objective Optimization with Vector Evaluated Genetic Algorithms. In Proceedings of the 1<sup>st</sup> Int. Conf. on Genetic Algorithms, 1986, 93-100.
- [5] Joshua D. Knowles and David W. Corne. The Pareto Archived Evolution Strategy: A New Baseline Algorithm for Multiobjective Optimization. In 1999 Congress on Evolutionary Computation, Washington, D. C., July 1999, 98-105.
- [6] Eckart Zitzler and Lothar Thiele. An Evolutionary Algorithm for Multiobjective Optimization: The Strength Pareto Approach. Technical Report 43, Computer Engineering and Communication Networks Lab (TIK), Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, May 1998
- [7] Kalyanmoy Deb, Samir Agrawal, Amrit Pratab, and T. Meyarivan. A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II. KanGAL report 200001, Indian Institute of Technology, Kanpur, India, 2000
- [8] Zitzler E, Laumanns M, Thiele L. SPEA2: Improving the Strength Pareto Evolutionary Algorithm. Technical Report TIK-Report 103, Swiss Federal Institute of Technology Zurich (ETH), May 2001.
- [9] Mikkel T. Jensen. Reducing the Run-Time Complexity of Multiobjective EAs: The NSGA-II and Other Algorithms, IEEE Transactions on Evolutionary Computation, Vol. 7, No. 5, October 2003, 503-515.
- [10] James Kennedy, Russell C. Eberhart, Swarm intelligence. Morgan Kaufmann Publishers Inc., San Francisco, CA, 2001.
- [11] K. E. Parsopoulos, M. N. Vrahatis. Particle Swarm Optimization Method in Multiobjective Problems. Proceedings of the 2002 ACM symposium on Applied computing. Publisher ACM Press New York, NY, USA, 2002, 603 - 60.
- [12] Xiaodong Li. A Non-dominated Sorting Particle Swarm Optimizer for Multiobjective Optimization. In Erick Cant et al. (editors), Genetic and Evolutionary Computation-GECCO 2003, Springer, Lecture Notes in Computer Science Vol. 2723, July 2003, 37-48.
- [13] Gregorio Toscano-Pulido and Carlos A. Coello Coello. Using Clustering Techniques to Improve the Performance of a Multi-Objective Particle Swarm Optimizer. In Kalyanmoy Deb et al. (editors), Genetic and Evolutionary Computation-GECCO 2004. Proceedings of the Genetic and Evolutionary Computation Conference. Part I,

Springer-Verlag, Lecture Notes in Computer Science Vol. 3102, Seattle, Washington, USA, June 2004, 255-237.

- [14] Fieldsend, J.E., Singh, S. A Multi-Objective Algorithm Based upon Particle Swarm Optimization, an Efficient Data Structure and Turbulence. In: Proceedings of the 2002 U.K. Workshop on Computational Intelligence, Birmingham, UK, 2002, 37-14.
- [15] Thomas Bartz-Beielstein, Philipp limbourg, et al. Particle Swarm Optimizers for Pareto Optimization with Enhanced Archiving Techniques. In Proceedings of the 2003 Congress on Evolutionary Computation (CEC' 2003), IEEE Press, Canberra, Australia, December 2003, 1780-1787.
- [16] Mostaghim, S., Teich, J.. Strategies for Ending Good Local Guides in Multi-objective Particle Swarm Optimization (MOPSO). In: 2003 IEEE Swarm Intelligence Symposium Proceedings, Indianapolis, Indiana, USA, IEEE Service Center, 2003, 26-33.
- [17] Vilfredo Pareto. Cours D' Economie Politique, volume I and II. F. Rouge, Lansanne, 1896
- [18] Kaisa Miettinen. Some Methods for Nonlinear Multi-objective Optimization. Evolutionary Multi-Criterion Optimization First International Conference, EMO2001 Zurich, Switerland, March 2001
- [19] Sarah Waziruddin, David Brogan, Paul Reynolds Jr., Coercion through Optimization: A Classification of Optimization Techniques, Proceedings of the Simulation Interoperability Workshop (SIW), 2003
- [20] Kirkpatrick, S., C.D. Gelatt Jr., M.P. Vecchi. Optimization by Simulated Annealing. Science, 220, 4598, 1983, 671-680.
- [21] James Kennedy, Russell Eberhart, Particke Swarm Optimization, IEEE International Conference on Neural Networks, 1995
- [22] J. Moore and R. Chapman. Application of Particle Swarm to Multi-objective Optimization: Dept. Computer. Sci. Software Eng., Auburn Univ., 1999.
- [23] Parsonopoulos K. E., Vrahatis M. N. Particle Swarm Optimization Method in Multiobjective Problems . Proceeding sof the 2002 ACM Symposium on Applied Computing (SAC 202) 2002:603-607.
- [24] X. Hu, Eberhart R. C. Multiobjective Optimization Using Dynamic Neighborhood Particle Swarm Optimization . In Proceedings of the IEEE World Congress on Computational Intelligence, Hawaii, USA, 2002:16 66-1670
- [25] Ray T. and Liew K. M. A swarm metaphor for multiobjective design optimization, Eng. Opt., vol. 34, no. 2, pp. 141 - 153, Mar. 2002

- [26] X. Li et al.. "A nondominated sorting particle swarm optimizer for multi-objective optimization," in Lecture Notes in Computer Science, vol. 2723. Proc. Genetic and Evolutionary Computation-GECCO 2003. Part I, E. Cantu-paz et al., Eds.. Berlin, Germany, July 2003, pp. 37-48.
- [27] Coello C. A., Lechuga M. S. MOPSO: A proposal for multiobjective particle swarm optimization. Proceedings of the IEEE congress on Evolutionary Computation (CEC 2002), Honolulu, Hawaii, USA, 2002
- [28] M. R. Sierra, Coello. C., "Improving PSO-Based Multi-objective Optimization Using Crowding, Mutation and  $\epsilon$ -Dominance," Third International Conference, EMO 2005, pp. 505-519, Springer. Lecture Notes in computer Science Vol. 3410, Guanajuato, México, March 2005
- [29] Salazar-Lechuga, M., Rowe, J.E.: Particle swarm optimization and fitness sharing to solve multi-objective optimization problems. In : Congress on Evolutionary Computation(CEC' 2005). Volume 2. (2005) 1204-1211
- [30] Deb K, Agrawal S. A niched-penalty approach for constraint handling in genetic algorithms [C]. In: Montana D, ed. Proceedings of thd ICANNGA-99. Portoroz:Slovenia, 1999:234-239.
- [31] Kalyanmoy D. Efficient constraint handling method for genetic algorithm [J]. Computer Methods in Applied Mechanics and Engineering, 2000, 186(2):311-338.
- [32] Rechian Y, Fung K, TANG Jia-fu, et al. Extension of a hybrid genetic algorithm for nonlinear programming problems with equality and inequality constraints [J]. Computers & Operations Research, 2002, 29: 261-274.
- [33] Laumanns, M., Thiele, L., Deb, K. and Zitzler, E., "Combining Convergence and Diversity in Evolutionary Multi-objective Optimization," Evolutionary Computation 10 (2002) 263 - 282 [27]
- [34] K. Deb, A. Pratab, S. Agarwal, and T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGAII, IEEE Trans. Evol. Comput., vol. 6 Apr. 2002., pages 182-197.
- [35] Srinivas, N. and Deb, k. (1995). Multi-Objective function optimization using nondominated sorting genetic algorithms, Evolutionary Computation (2), 221-248.
- [36] Masahiro Tanaka, Hikaru Watanabe, Yasuyuki Furukawa, GA-based Decision Support System for Mufti-Criteria Optimization, IEEE International Conference on Systems, Man and Cybernetics-2, Intelligent Systems for the 21st Century, 1995
- [37] 张敏慧. 改进的粒子群计算智能算法及其多目标优化的研究应用研究(硕士学位论文). 杭州:浙江大学, 2005
- [38] 朱建才. 多目标优化方法库的开发与应用研究(硕士学位论文). 西安:西北工业大学, 2006

## 攻读硕士学位期间发表学术论文情况

陈绍新, 周宽久. 一种改进的解约束问题的多目标混合粒子群算法. 2007 年研究生学术交流会——通信与信息技术会议论文集

注: 发表论文属于学位论文的第三章 3.1、3.2、3.3 小节。

## 致 谢

值此本文完成之际，谨向曾帮助过我的老师、同学以及亲友一并表示衷心的感谢！

首先感谢我的导师周宽久老师，本文从选题、设计、撰写及整个研究过程中一直是在周宽久老师的精心指导和严格要求下进行的。周宽久老师渊博的知识、敏锐的思维、严谨的治学态度、一丝不苟的工作作风使我终生受益非浅，也为我今后的工作与学习树立了典范。

在此，也对软件学院的所有老师表示深深的谢意。在软件学院学习的两年时间里，在你们的谆谆教诲下不但使我具有了提出问题、分析问题、解决问题的能力，你们高尚的师德、卓绝的学识更让我学会了在今后的工作生活中怎样去为人做事。

此外，还要感谢远方的父母，是你们从小到大始终如一的给予我无微不至的关怀，同时也为了我放弃了许多，失去了许多，如果没有你们的辛勤劳作、默默的支持以及对我孜孜不倦的教诲，就没有我的今天，再次向你们表示最由衷的谢意。

感谢培育了我的母校，感谢和我朝夕相处的同学们，我们共同营造的那份轻松与欢乐我将永远回忆与珍惜，衷心祝你们学业、事业有成！

转眼即将离开培养我多年的母校，踏上新的工作岗位。我会永远记住这一段难忘的光阴！在以后的工作、生活中，不断求索，领悟人生的真谛。

作者: [陈绍新](#)  
学位授予单位: [大连理工大学](#)

## 相似文献(10条)

1. 期刊论文 [贺益君, 俞欢军, 成飙, 陈德钊, HE Yijun, YU Huanjun, CHENG Biao, CHEN Dezhao](#) [多目标粒子群算法用于补料分批生化反应器动态多目标优化](#) - [化工学报](#)2007, 58 (5)

多目标优化是过程系统工程的重要课题,通常以加权或约束方式将其转换为单一目标,未能反映多目标间的复杂关系,不利于随时根据需求作出有效的决策.基于群智能的粒子群算法具有全局优化性能,且易于实现.为使其适于多目标优化,应拓展功能,实施改造.以Pareto支配概念评价种群个体的优劣,设计了确定局部最优点和全局最优点的操作.又利用各粒子的局部最优点信息进行速度更新,以加强种群的多样性,避免因早熟而陷于局部最优.还设置了外部优解库,并通过分散度计算,以适当的策略进行更新,使之逐步均匀地逼近于Pareto最优解集.由此构建一种多目标粒子群优化算法(multi-objective particle swarm optimization, MOPSO),并用于补料分批生化反应器的动态多目标优化,取得了满意的结果.可基于所获得的Pareto最优解集,分析目标间的关系,为合理决策提供有效的支持.经与NSGA-II比较, MOPSO算法具有更为优良的性能.

2. 期刊论文 [马琳, MA Lin](#) [面向安全服务重组多目标优化的粒子群遗传算法](#) - [中南林业科技大学学报](#)2007, 27 (5)

遗传算法具有快速随机的全局搜索能力,但局部搜索能力差,易陷入早熟收敛,迭代效率低.粒子群算法采用速度—位置模型,可以较快收敛到指定精度.将粒子群算法与遗传算法融合,采用多目标遗传算法得出初步的优化结果,并将其作为粒子,利用粒子群算法强化局部搜索,加快收敛速度,仿真结果证明了该算法的优越性.在CSSM对底层安全服务的重组时利用粒子群和遗传算法的结合(GAPSO),能够提高效率.

3. 学位论文 [何宽](#) [基于粒子群算法的多目标优化方法研究](#) 2007

工程设计、软件硬件设计、生产管理、金融投资等领域所遇到的问题往往需要同时考虑多个目标在某种意义下的最优问题,这种含有多个目标的最优化问题被称为多目标优化问题.因为这类问题的各个目标之间常常是相互冲突或不可比较的,所以解决多目标优化问题通常很难.各目标之间通过决策变量相互制约,对其中一个目标优化必须以其它目标作为代价,而且各目标的单位又往往不一致,很难客观地评价多目标问题的优劣性.解决多目标优化问题很困难,而现实生活的很多问题都是多目标问题,因此研究处理多目标优化问题的方法是非常重要的.

基于种群的、可以同时搜索到多个Pareto前端的群体智能优化算法——粒子群优化算法,作为一种有效的多目标最优化工具,已经得到广大学者的认可.

本文将采用粒子群算法处理多目标优化问题,通过改进的粒子群算法使其适用于多目标优化问题,通过将多目标问题转化为单目标问题,借鉴遗传多目标优化方面的优秀算法模拟退火算法和粒子群优化算法的思想,提出相应的混和形粒子群算法(HMOPSO)方法.

模拟退火算法来源于固体退火原理,将固体加热至充分高温,再让其徐徐冷却,加热时,固体内部粒子随温度升变为无序状,内能增大,而徐徐冷却时粒子渐趋有序,在每个温度都达到平衡态,最后在常温时达到基态,内能减为最小,本文将模拟退火算法的思想应用到粒子群算法中,使得粒子群算法能够快速收敛.

为了避免算法过早收敛,加入变异操作以增强算法的全局探测能力.对每个粒子的pbest进行综合加权,能够促进粒子加快收敛,在这里我们采用的加权方式为通过基本的粒子群优化算法来产生一组新的个体,然后再独立地进行杂交运算和高斯变异运算,通过对所产生出的各个个体分别进行模拟退火,以其结果作为下一代群体中的个体.在每次进化中,杂交运算依据杂交概率选取指定数量的粒子放入一个池中.池中的粒子随机地两两杂交,产生同样数目的孩子粒子,并用孩子粒子代替父母粒子,以保持种群的粒子数目不变.

计算个体的适应函数值对于多目标优化问题很重要的议题.为分辨个体之间的优劣,本论文采用了Ho提出的以Pareto理论为基础的通用化且不受尺度因素影响的评分函数(GPSISF),对于每个基于Pareto理论来评估解的好坏,没有权重加总法需要决定权重值的困难,也不会受到人为主观判断的影响.由于各目标函数值尺度适应值不尽相同,在权重加总法中需要考虑到尺度因素,以免使得权重设定失之准确.以评分方式有效办法不同解的优劣程度:取代传统排名法可能不同的解给予相同的排名,以及距离法有尺度因素影响的缺点,以精确地分解的优劣程度.

将HMOPSO算法应用于复杂数值函数ZDT1, ZDT3, ZDT4, ZDT6,进行测试,验证了该算法的可行性,和正确性.研究结果丰富了多目标粒子群优化算法理论,提高了算法性能,拓宽了算法的应用领域和实现形式,为算法在复杂系统中的应用提供了有益的指导.

4. 期刊论文 [王小刚, 李明杰, 王福利, 胡伦, WANG Xiao-gang, LI Ming-jie, WANG Fu-li, HU Lun](#) [一种新的多目标粒子群算法的研究与应用](#) - [东北大学学报\(自然科学版\)](#) 2008, 29 (10)

基于多目标粒子群优化算法的研究成果,以提升多目标优化方法性能为目标,结合铜电解槽稳态优化工程实际,提出了一种新的多目标优化粒子群算法.该方法采用可变的外部集策略和快速排序方法来减少优化计算量,采用拥挤度算子和强支配关系保证结果良好的分布性.测试函数的仿真对比分析及对铜电解实际过程数据的优化运行结果均表明该方法在收敛性、解的分布性及计算效率方面具有良好品质,特别是在求解三目标优化问题时的突出表现,表明这种算法在多目标优化领域具有更广泛的适用性.

5. 会议论文 [吴晶, 盛四清](#) [基于集成粒子群算法的配电网优化规划](#) 2009

配电网规划是一个大规模、复杂的、具有非线性离散变量和多约束的多目标数学优化问题.与输电网相比,配电网的发展受社会经济各方面外部环境因素的影响更大.

本文采用资金投入、网络损耗、供电可靠性和对环境影响等多目标函数建立数学模型.基本粒子群算法无法对多目标优化问题求解,故本文通过将多目标函数和不等约束条件分成多个子目标,利用模糊化和粗糙集理论确定子目标之间的关联度,从而得到表征对各目标相对重要的权重系数,进而将追求多目标优化转化为寻求各目标总体满意程度的最优.结合改进的PSO全局寻优算法和Powell11局部寻优算法对配电网进行优化规划,克服单用粒子群算法求解问题早熟收敛的不足,增加算法的开发能力,提高局部寻优能力.算例表明,该方法具有一定可行性和有效性.

6. 期刊论文 [徐刚, 瞿金平, XU Gang, Qu Jin-ping](#) [一种用于多目标优化的混合粒子群优化算法](#) - [计算机工程与应用](#) 2008, 44 (33)

将粒子群算法与局部优化方法相结合,提出了一种混合粒子群多目标优化算法(HMOPSO).该算法针对粒子群局部优化性能较差的缺点,引入多目标线搜索与粒子群算法相结合的策略,以增强粒子群算法的局部搜索能力. HMOPSO首先运行PSO算法,得到近似的Pareto最优解;然后启动多目标线搜索,发挥传统数值优化算法的优势,对其进行进一步的优化.数值实验表明, HMOPSO具有良好的全局优化性能和较强的局部搜索能力,同时HMOPSO所得的非劣解集在分散性、错误率和逼近程度等量化指标上优于MOPSO.

7. 会议论文 [傅调平, 陈建华, 李刚强, 倪兴勇](#) [一种改进的多目标粒子群算法PMOPSO及其在舰艇航路优化中的应用](#) 2007

设计实现了一种改进的航路优化多目标粒子群算法.分析了军事航海的特殊性,提出基于多目标优化的舰艇航路优化模型.针对常见的PSO算法普遍存在的不足,提出一种基于排挤策略的航路优化多目标粒子群算法PMOPSO.算法采用解的占优比较法更新局部最优P<sub>l</sub>,提出基于排挤策略方法选择全局最优P<sub>g</sub>,有力地维护了种群的多样性,最后采用不可行度方法来处理多约束.实验结果表明:算法在Pareto解集的代间距、分散性和错误率指标上有了较大提高,能较好解决舰艇航路优化问题.

8. 期刊论文 [吴小虎, 徐琪, WU Xiaohu, XU Qi](#) [基于自适应网格粒子群算法的多目标配送优化模型](#) -公路交通科技

2010, 27(5)

对配送方案的选择提出多目标优化,在满足客户需求的前提下,力求成本最低和各配送中心负荷均衡,建立多目标规划模型.运用粒子群算法对解空间粒子进行局部和全局的搜索,再运用自适应网格算法对非劣解外部集进行更新和维护,保持其规模.实证表明,采用基于自适应网格的多目标粒子群算法对该模型进行求解能够得到均匀分布于解空间的Pareto前沿.结果表明两目标具有一定的悖反关系,据此选择满意解.

9. 期刊论文 [王正初, 赵燕伟, Wang Zhengchu, Zhao Yanwei](#) [基于粒子群算法的多目标模糊物元优化研究](#) -机械制造

2007, 45(9)

针对产品设计优化过程中多目标性、模糊性、不确定性等特性,提出基于粒子群算法的多目标模糊物元求解策略.建立了多目标模糊物元优化模型,定义了物元的各个权重,根据某种优化准则,将多目标优化问题转化成了单目标问题的优化.运用粒子群算法作为求解策略,并提出新的改进方法,给出模糊物元多目标优化设计问题的改进自适应粒子群算法(MAPSO)求解过程,并将它与其他方法进行比较,结果显示该算法具有较高的执行效率.

10. 学位论文 [黄婉平](#) [自适应粒子群优化算法及其应用研究](#) 2006

参数优化是许多科学、工程问题以及社会经济活动中的重要研究内容.国内外学者已经针对这一问题提出了大量进化算法,如遗传算法、免疫算法、粒子群算法等.粒子群算法作为一种新的进化算法,不依靠遗传算子来操作个体,而是依靠个体间的信息交换来达到群体的共同演化,所有的微粒都有调整自身速度和记忆经历过的最好位置的能力.该算法已成功地解决了许多工程实际问题,并取得了很好的优化效果.

本文在分析粒子群优化算法基本原理的基础上,针对标准粒子群算法易陷入局部解的不足,提出了变异操作和线性递增的改进方案——自适应粒子群算法,进而对该算法在FIR数字滤波器的优化设计、多目标满意优化以及鲁棒PID参数整定方面进行了应用研究,并通过仿真实验验证了自适应粒子群算法的有效性和优越性.

本文的主要研究工作与贡献如下: 1. 简要的回顾了计算智能的理论和和技术发展史,叙述了粒子群算法的研究背景,总结了多目标优化的传统解决方法以及多目标满意优化的特点,并介绍了鲁棒PID控制的研究背景及其现状.

2. 对粒子群算法进行详尽的分析和综述,将粒子群算法应用在频率抽样技术中,提出一种新的FIR滤波器优化设计方法,实例验证了这种优化设计方法的可行性及优越性.

3. 针对粒子群算法在参数设置方面缺乏理论依据,以及在收敛过程中易陷入局部解的问题,分析了粒子群中独立粒子的运动轨迹,以及整个粒子群系统的稳定性,在此基础上引入自然界“优胜劣汰”的思想,提出新的参数设置方法,仿真结果证明新方法具有良好的自适应能力.

4. 在介绍满意优化方法的一般定义和传统的多目标优化方法的基础上,通过多目标满意优化方法寻求“满意解”来代替“最优解”,提出了一种基于自适应粒子群算法的多目标满意优化策略,在满足有效解或者弱解有解的条件下,使相互矛盾的各个目标得以兼顾.

5. 从分析LQR指标入手,结合minmax原则,将自适应粒子群算法应用于PID参数整定中,提出新的PID参数整定方法.仿真结果表明了该整定方法所设计的控制器具有良好的鲁棒性.

本文链接: [http://d.g.wanfangdata.com.cn/Thesis\\_Y1226632.aspx](http://d.g.wanfangdata.com.cn/Thesis_Y1226632.aspx)

授权使用: 上海海事大学(wf1shyxy), 授权号: a5b65ce5-0a45-4df0-9151-9df700025a4b

下载时间: 2010年9月20日