

中国海洋大学
硕士学位论文
装箱问题方法研究及其集成应用
姓名：韩运实
申请学位级别：硕士
专业：计算机应用技术
指导教师：丁香乾
20040701

装箱问题方法研究及其集成应用

摘 要

装箱问题就是将不同尺寸的物品摆放入有一定容量的容器中,以获得某种最佳的效益。装箱问题广泛地用于机械生产和交通运输等行业当中。对该问题的求解方法的研究无论是在理论上,还是在实践中都具有一定的意义。装箱问题属于组合最优化问题和 NP 完全问题,具有高度复杂性,用一般的数学方法根本无法求解,目前只能用启发式方法解决装箱问题,因此,从六十年代开始,陆续提出的都是近似算法。

文中首先对装箱问题的种类、求解算法,以及装箱问题的研究现状进行了综述和分析。重点放在研究用于解决装箱问题的各种启发式算法,在总结概括了当前主要的装箱优化算法的基础上,指出构造式启发式算法与遗传算法相结合,是改进装箱算法的一种有效的途径。然后依据该混合遗传算法,采用 Delphi 语言开发了用于解决任意尺寸三维物体装载的智能装载系统,并详细说明了系统的整体结构、设计过程以及数据流程图,对部分系统运行结果进行了分析。并且给出了该算法和第二章给出的构造启发式算法的性能比较。

遗传算法是借鉴生物的自然选择和遗传进化机制而开发出的一种全局优化自适应概率搜索算法,对于非常复杂、高度非线性大型系统的优化求解,表现出比其它传统优化方法更加独特和优越的性能,是 21 世纪有关智能计算中的核心技术之一。利用简单的编码技术和繁殖机制来表现复杂的现象,从而解决非常困难的问题。特别是它不受搜索空间的限制性假设的约束,不要求诸如连续性、导数存在和单峰等的假设。

文中综述了遗传算法的基本原理和方法,分析了遗传算法的编码、适应度函数、选择算子、交叉算子和变异算子在整个遗传算法的运算过程中的作用。针对装箱问题对编码方式作了改进,采用数字符号编码方式。本文采用了跨世代精英选择策略来保持种群中的优良个体,使得产生的遗传算法更加有效,同时,提出了与装箱问题相适应的交叉和变异策略。但是单一地使用一种算法不易得到十分满意的结果,因此在遗传算法的搜索策略和变动策略中应用了启发式方法,引导

搜索方向。结果表明应用该混合方法缩短了计算时间，同时使得问题的解得到了很好的改善。

最后，总结了本文的工作并展望了进一步的研究方向。

关键词：装箱问题，启发式方法，遗传算法，自动装载系统

The research and integrated applications of approaches to solving bin packing problem

Abstract

The bin packing problem is to put some objects with various sizes into a defined space in order to obtain the specified optimal benefit. Bin packing is a combinatorial optimization and NP-complete problem and widely applied to the mechanical manufacture and traffic transportation industries. To study approaches to solving bin packing problem has great significance not only in theory but also in practice. Up to now the heuristic algorithms is the only way to solve the bin packing problem because of its high complication, so approximate algorithms were presented in succession from 1960.

It begins by summarizing the bin packing problem in sorts and their algorithms, as well as analysis on the current research on the bin packing problem. The main work of this dissertation lies on applying heuristic algorithms to solve the bin packing problems. After a brief review of current packing algorithms, an efficient approach for bin packing which is genetic algorithm combined with heuristic method is indicated. An intelligence loading system employing Delphi language is developed on the basis of the previous hybrid algorithm, which is propitious to general bin packing problem. Then the whole system structure is analyzed, designing process and data flow chart is illuminated and part of running results is analyzed. Comparable performance of this hybrid algorithm and the construction heuristic algorithm which is described in chapter 2 is presented.

Genetic Algorithm is a method for searching for the optimum solution to a complex problem, based on the mechanics of natural selection, the process of evolution. It is flexible and robust. Its goal is to achieve similar breadth of performance by abstracting nature's adaptation algorithm of choice in artificial form. Genetic Algorithm is basically an automated, artificial intelligent approach to trial and error. In the aspect of solving large, non-linear and poorly understood problems where

expert knowledge is scarce or difficult to encode and traditional methods fail, GA has great advantages. It is one of kernel techniques related with intelligent computing in 21 century.

In this dissertation, the basic principle of genetic algorithms is summarized and the erect of coding is analyzed, fitness function, selection operators, crossover operators and mutation operators in the genetic process of genetic algorithms. Numerical sign coding is improved to solve bin packing problems. According to this coding, genetic algorithm with Cross generational Elitist Model to make new algorithm more effective is put forward. At the same time, crossover and mutation are developed to construct genetic algorithm of bin packing problems. It is not easy to get the satisfactory result just use one of these algorithms signally, so heuristic approach is utilized in search tragedy and movement tragedy in order to guide research direction. It has been proved that hybrid algorithm shortens the computation time and improves the solution.

At last, a simple prospect for the possible application is given.

Key words: Bin packing problem, Heuristic method, Genetic algorithm, intelligence loading system

1 第一章 绪论

1.1 课题的提出

装箱问题是复杂的离散组合最优化问题^[1]。所谓组合优化,是指在离散的、有限的数学结构上,寻找一个满足给定条件,并使其目标函数值达到最大或最小的解。一般来说,组合优化问题通常带有大量的局部极值点,往往是不可微的、不连续的、多维的、有约束条件的、高度非线性的 NP 完全问题。装箱问题也不例外,同许多组合最优化问题,如旅行商问题、图的划分问题等一样属于 NP-HARD^[2,3]问题。经典的装箱问题要求把一定数量的物品放入容量相同的一些箱子中,使得每个箱子中的物品大小之和不超过箱子容量并使所用的箱子数目最少。

从 20 世纪 70 年代初开始,装箱问题就引起了广泛的探讨和研究^[4,5]。然而装箱问题可以追溯到 1831 年高斯(Gauss)开始研究布局问题^[7],因为装箱问题和布局问题本质上是一样的,到现在已有百余年的历史,在文献上始见于 1930 年 Kantorovich 的俄文文章(1960 年译为英语刊登在 Management Science^[8]上)。虽然经过几代人的努力,但迄今尚无成熟的理论和有效的数值计算方法。由于目前 NP 完全问题不存在有效时间内求得精确解的算法,装箱问题的求解极为困难,因此,从 70-80 年代开始,陆续提出的装箱算法都是各种近似算法,如下次适应、首次适应、降序下次适应和调和算法等^[5]。

装箱问题广泛存在于工业生产,包括服装行业的面料裁剪、运输行业的集装箱货物装载、加工行业的板材型材下料、印刷行业的排样和现实生活中包装、整理物件等。在计算机科学中,多处理器任务调度、资源分配、文件分配、内存管理等底层操作均是装箱问题的实际应用^[4],甚至还出现在一些棋盘形、方块形的数学智力游戏中。装箱问题的研究文献分布面很广,在运筹学、计算机辅助设计、计算机图形学、人工智能、图像处理、大规模集成电路逻辑布线设计、计算机应用科学等诸多领域都有装箱问题最新的研究动态和成果出现,从这个角度来讲,布局问题涉及到了工业生产的方方面面,也足以证明了装箱问题的应用前景日趋广泛而重要。

从商业上的应用来看,至今尚没有成熟的用于工业生产商品化软件,在学术研究方面,各学术团体对装箱问题的研究是一维、二维多于三维,且多为不带性能约束的装箱问题,要实现任意约束条件下的三维物体的装载问题,至今尚无成功的通用的方法。虽然市面上已用一些智能化的货物装载软件,但是这些方法大多数因货物的属性和问题复杂程度的不同而施加不同的限制。除此之外,对于装箱问题的大多数研究都是在运筹学(Operation Research)领域完成的。这些方法的缺点是因为它们一般都是启发式的方法,所以只能应用在专门的一类问题当中。但是在工程实际中,由于解决问题的要求和目标的多样性,需要寻求一种能以有限的代价来解决搜索和优化的通用方法,遗传算法(Genetic algorithm,简称 GA)作为模拟自然界生物进化原理的一种随机性全局优化概率搜索算法,具有很强的全局搜索能力,特别适合求出问题的近似最优解,用 GA 解决装箱问题是一可行思路^[9]。传统的遗传算法有许多成功应用的案例,但也存在一些缺点:①收敛速度慢,②可能陷入局部最小值。因此在装箱问题的应用中,除了对基本遗传算法有很多改进外,还需要将遗传算法和各种优化算法相融合,提高算法的收敛速度和对各种实际问题的搜索空间的大小变化适应能力。

随着国内物流业的发展,物流相关技术的应用和发展受到越来越多的重视。装箱问题作为物流配送过程中的一个关键性的技术,对提高配送业务的自动化水平、提高货物装载的优化程度、提高配送业务的工作效率和规范业务流程等方面都有重要的意义。

针对上述原因,本文选择装箱问题方法研究及集成应用这一课题,以解决当前物流配送环节面临的问题。旨在寻求面向现代物流企业实用的优化配送算法,建立现代物流配送系统,以期达到对现代货物配送装载业务有一个实用、有效的技术解决方案目的。

1.2 装箱问题分类

装箱问题涉及多学科、多领域的知识,属于复杂的组合最优化问题,在生产实践中广泛出现并亟待解决。本文从不同角度可以把装箱问题分为以下几类:

1.2.1 按照装箱物体所属装箱空间对装箱问题的分类

1. 一维装箱问题

经典的一位装箱问题是这样的：设给定一正数 c 和一组数 $L = \{a_1, a_2 \cdots a_n\}$, $0 < a_i \leq c, \forall i$, 问题是要寻求一种分拆方法 A 将 L 分成一些互不相交的子集 B_j , 使得 $L = B_1 \cup B_2 \cup \cdots \cup B_m$ 且满足 $\sum_{a_i \in B_j} a_i \leq c, \forall j$, 并使 m 为满足这一要求的最小整数。该问题有着广泛的实际背景, 如:

1) 用户送来一张订货单, $L = \{a_1, a_2 \cdots a_n\}$, a_i 表示第 i 个货件的长度, 货件可能是钢条、铅管、电缆、原装纸卷等一维线材。假如厂家所供应原材料的标准长度为 c , 同时满足 $a_i \leq c, \forall i$, 试想如何能以最少量的原材料截出用户所需的货件。

2) 有 n 个零件需要加工, 假定每个零件只需在一台机器上加工一次即可。设第 i 个零件所需的加工时间为 a_i , 要求所有零件在规定时间内 c 内全部加工完毕, 试问如何能以最少数的机器和人力在规定时间内完成所需加工任务。规定所有机器的性能皆相同。

2. 二维装箱问题

二维装箱问题有下述两类:

1) 给定 n 个矩形 $L = \{a_1, a_2 \cdots a_n\}$ (其中 $a_i = (x_i, y_i)$; $x_i, y_i \in (0, 1], i = 1, 2 \cdots n$) 装入一宽度为 1, 高无限大的矩形 A (箱子), 要求将 n 个矩形 $a_1, a_2 \cdots a_n$ 互不叠迭地装填入 A 中, 使布局的总高度最小。对每个小矩形 a_i 装填入 A 后的姿态也有直角装填, 定向等限制;

2) 给定 n 个矩形 $L = \{a_1, a_2 \cdots a_n\}$ 及无限多大小都相等的大矩形(箱子), 要求将 $a_1, a_2 \cdots a_n$ 直角定向装填入箱子中, 使所占用的箱子个数最少。

对第一类二维装箱问题, 若设小矩形的高度都相等, 限制装填方式为直角定向, 则问题蜕化为一维装箱问题; 对第二类问题若设小矩形之宽与箱子宽相等, 也退化为一维情形。

这类问题在实际中有着广泛应用。如应用于板类(金属板,木板,玻璃、大理石板等)切割行业,服装剪裁行业等中的下料问题;建筑业中的房间布局,工业中的模板布局(Template Layout)、新闻报纸、广告等版面布置,集成电路布图设计及其它一些工程设计等。

3. 三维装箱问题

与二维装箱问题相似,三维装箱问题也有下述两种形式

1) 给定 n 个矩形体 $L = \{a_1, a_2 \cdots a_n\}$ (其中 $a_i = (x_i, y_i, z_i)$; $x_i, y_i, z_i \in (0,1]$, $i=1,2,\cdots,n$) 装入一底为 1×1 , 高为正无穷的柱形箱子 A , 要求将 n 个矩形体 $a_1, a_2 \cdots a_n$ 互不叠迭地装填入 A 中, 使布局的总高度 Z 最小。对每个矩形体 a_i 装填入 A 后的姿态也有直角装填, 定向等限制。

2) 给定 n 个矩形体 $a_1, a_2 \cdots a_n$ 及无限多大小都相等的箱子, 要求 $a_1, a_2 \cdots a_n$ 直角定向装填入箱子中, 使所占用的箱子个数最少。

对第一类三维装箱问题, 若设矩形体的高度都相等, 限制装填方式为直角定向, 则问题蜕化为二维装箱问题; 对第二类问题若设矩形体之宽与箱子宽相等, 也退化为二维情形。

因此三维装箱问题可以看作是一维、二维装箱问题的一个泛化^[13, 14]。三维装箱问题有着广泛的应用背景, 如运输行业的集装箱货物装载、飞机装舱、码头装货、列车装箱, 机械行业中箱体和钟表等布局, 航空、航天工业中导弹仓的布局, 以及建筑、电子、造船业、纺织业等诸多领域。

1.2.2 按照装箱物体的形状对装箱问题的分类

1. 规则物体的装箱

包括二维规则物体的装载和三维规则物体的装载。规则物体是指具有规则外形的物体, 例如圆柱体、矩形体等。在以前及现在的装载研究中, 研究较多的仍然是规则体的装载问题, 如: 工业应用中的底盘装载问题; 三维布局中的集装箱的货物摆放问题。

货物底盘装载问题广泛存在于工业生产和交通运输中。由于箱子的大小和形状完全相同, 且箱子的边平行于底盘的边, 因此该问题也可简化为二维问题。对

于该问题, 有的学者使用精确的方法求解^[10]。

在运输生产中, 常见的集装箱装载要求有两类, 一是集装箱数量无限, 而待装的货物有限, 要使集装箱利用率最高; 二是集装箱数量固定, 待装货物数量无限, 远远超过已有集装箱的承载能力, 一般是其几倍, 要求在有限的集装箱空间内尽可能地多装货物, 使集装箱利用率最高, 生产实际中更多地遇到的是第 2 类问题。集装箱布局要考虑货物重量、空间利用率、货物易碎性、以及吊装的可能性等等, 为多目标多约束优化问题。

2. 不规则物体的装箱

包括二维不规则物体的装载和三维不规则物体的装载。不规则物体是指具有任意几何形状的物体。不规则物体的装箱问题在工业生产中大量存在, 但同时也是难度最大的装箱问题。二维不规则物体的装箱如服装样本的下料、皮革下料等。三维不规则物体的装箱如向具有任意几何形状的容器中放置任意几何外形的装箱物体, 并满足特定的约束条件, 达到装箱目标最优。从文献资料中可以看出, 许多研究者利用人工智能原理求解该复杂问题^[11]。该问题的求解算法基本上是启发式的。

1.2.3 按照装箱物体达到情况对装箱问题的分类

1. 在线装箱问题

如果一个装箱算法在装入物品 α_i 时, 只利用 α_i 前面物品 $\alpha_j, 1 \leq j \leq i$ 的信息, 而不知道后继元素的任何信息, 即按照元素到达顺序随到随装, 则称该算法为在线(on line)算法。这种情况下的装箱问题称为在线装箱问题^[12]。在实际应用中, 往往要求装载具有在线特性。例如对从传送带上下来的物体进行装载。

很多一维、二维在线装箱问题都采用层的思想: 物品的尺寸不能提前知道。层由沿箱子宽度排列的一系列物品组成。也就是说箱子是由层中最大高度的物品来分割的。典型方法包括 next-fit decreasing height(NFDH)^[13,14,15], first-fit decreasing height(FFDH)^[13,15], best-fit decreasing height(BFDH)^[13], improved first-fit decreasing height(IFFD) and harmonic algorithms^[13,15]。尽管这些方法都很相似, 但是每个算法都是用不同的方法决定将物品放在哪里。

和二维装箱算法相似的一些算法被应用到三维装箱问题中。在线最流行的解

决三维装箱方法之一是 shelf method^[16]。该方法也是通过聚组相近尺寸物品来提高装载利用率的。

2. 离线装箱问题

物品装载以前就已得到所有物品信息，之后统一处理所有物品的近似算法称为离线(off line)装箱算法。这种情况下的装箱问题称为离线装箱问题。在现代化的物流配送中，很多都要求按订单送货，因此物品的信息提前都是知道的。该问题广泛地出现在铁路货车车厢装载、汽车车厢装载、轮船配载、集装箱装载等情况中。

分支定界方法^[17]、遗传算法^[18,19]和模拟退火方法^[20]都是用来解决离线的装箱问题的常用算法。分支定界方法利用递归方法寻找问题的可行解，采取了必要的限制条件，设法排除可行域中大量非最优解区域，从而能够有效求解一些规模较大的问题。在遗传算法中，问题模型是数字串，称为染色体。一个染色体的存活概率被称为适应度函数。这个函数是衡量染色体所代表的解接近于最优解的程度。在执行遗传算法之前，给出一群“染色体”，也即是“假设解”。然后，把这些“假设解”置于问题的“环境”中，并按“适者生存”的原则，从中选择出较适应环境的“染色体”进行复制，再通过交叉，变异过程产生更适应环境的新一代“染色体”群。这样，一代一代地进化，最后就会收敛到最适应环境的一个“染色体”上，它就是问题的最优解。模拟退火方法源于对固体退火过程的模拟，它与局部搜索算法的最大不同是接受新解的准则不同，允许目标函数在有限范围内变坏，由一控制参数 t （温度）决定。开始时 t 值大，可能接受较差的恶化解，随着 t 值的减小，只能接受较好的恶化解，最后在 t 值趋于零时，就不再接受任何恶化解了。使算法既可以从局部极小的“陷阱”中跳出，更有可能求得组和优化问题的整体最优解，又不失简单性和通用性。

1.3 装箱问题研究现状

装箱问题是一个典型的 NP 完全问题，即在有限时间内找不到问题的最优解，用传统的搜索技术求解这类问题，就会出现“组合爆炸”的现象，在实践中，人们一般用启发式算法来求解。因此从六七十年代开始研究装箱问题，提出的各种算法都是近似算法。

关于一维装箱问题的研究结果是丰富的。在经典的一维装箱算法中, 最早的一个著名的结果是 Johnson^[24]得到的, 他处理的算法是递减首次适合算法 FFD。Johnson 在 1973 证明了 $R_{FFD}^{\infty} = 11/9 \approx 1.222\cdots$ 。随后, Garey 与 Johnson (1985) 设计出改进的 FFD, 具有 $R_A^{\infty} = 71/60 = 1.1833\cdots$ 。这个不断改进的 R_A^{∞} 的过程已达到一个惊人结局, Lueker 等发现对任意小的 $\varepsilon > 0$ 存在着具有 $R_{A(\varepsilon)}^{\infty} < 1 + \varepsilon$ 的算法 $A(\varepsilon)$, 在此基础上只要稍做技术改进, 便可构造 $R_A^{\infty} = 1$ 的算法。然而不幸的是, 上述研究进程中, 随着算法的性能越来越好, 其实践性也变得越来越差, 方法不易实行。

二维装箱问题的研究, 主要从实际工作角度出发, 设计一些实用方法, 虽然也有些理论结果, 但并不太深入, 主要是问题太难, 使人无从下手。目前就二维空间上的在线装箱问题而言, 最坏性能比最好结果为 J. Csirik 和 A. Van Vliet 提出的一种叫 ROUND 的算法^[28], 该算法用来解决多维在线装箱问题, 并给出了最坏性能比的严格证明 $R_{round}^{\infty} = 2.85958$ 。

三维装箱问题由于其复杂性和难度大, 因此研究相应较少, 且各种算法都是基于一定的假设之下, 因此这些算法在具有各种复杂约束条件的实际应用中就会受到很大的限制。目前为止, 最好的三维装箱算法是文献[27]中的 level-strip 算法, 渐近性能比可任意逼近 2.89。

大多数前人的研究工作都是集中在一维、二维装箱问题的解决方案上而不是三维。部分原因归结于一维、二维装箱问题有着大量的实际应用背景, 同时也归结为三维装箱问题的复杂性上。一维、二维装箱问题已经是 NP-complete 问题, 同样三维装箱问题也是。然而, 在进行三维装箱时, 算法设计难度的增加和一维、二维装箱不是一个数量级别的。例如: 在装载时, n 个二维物品, 所有可能的组合是: $2^n \times n!$, 而三维的情况下, 所有可能的组合是: $3^n \times n!$ 。维数的增加导致问题空间的急剧增加, 因此有效算法的设计难度也会大大增加。

1.4 本文的意义和主要研究工作

1.4.1 本文的研究内容及意义

本文从应用的角度对装箱问题算法作了认真分析和研究,其中比较详细地介绍启发式算法和遗传算法在装箱问题的应用,然后将其集成应用到求解三维装箱问题中,从而完成家电行业智能化配车系统的研发。

文本研究的意义如下:

1. 智能化配车作为物流配送过程中的一个关键性的技术,对提高配送业务的自动化水平、提高车辆装载的装载率和装载准确率、提高配送业务的工作效率、降低货物成本、规范业务流程、改变依赖装车人员经验进行车辆装载的传统模式方面都有重要的意义。尤其对竞争日益白热化的家电行业来说,优秀的物流配送环节对于提高企业的物流竞争,甚至是企业的竞争力都是非常有帮助的。
2. 对于装箱问题的其他应用方面:线材(钢条、铅管、电缆、原装纸卷等)下料、板类(金属板,木板,玻璃、大理石板等)切割以及运输行业的集装箱货物装载、飞机装舱、码头装货、列车装箱,提供一个很好的借鉴经验和参考依据。
3. 装箱问题属于经典的 NP 完全问题,其方法的研究对其他 NP 完全问题的研究也具有很大的参考价值。

1.4.2 本文的内容安排及解决的关键问题

本文重点对启发式算法和遗传算法进行了改进和研究,二者相结合开发了家电行业智能化配车系统。全文安排如下:

第一章首先综述装箱问题,然后对其进行分类,接着对装箱问题的研究现状进行了具体论述,在此基础上,阐述了本文的研究内容及其意义。

第二章首先对启发式算法作了总结,然后介绍了在装箱问题中经常用到的算法,并对其性能作了简单分析。

第三章对遗传算法做了详细介绍,对其在装箱问题中的应用进行了研究。

第四章介绍利用启发式算法和遗传算法集成而成的算法完成的系统，该章介绍了系统研究背景、研究内容、系统设计方案及核心算法的实现，并给出了一个实例演示。随后给出了系统部分运行结果并简单的分析了待装物体的属性与箱体利用率的关系。

第五章总结了本文的工作并展望了进一步的研究方向。

本课题所解决的关键问题如下：

1. 对遗传算法的搜索方法进行改进，提高搜索速度、对实际问题的搜索空间的大小变化适应能力、系统的健壮性。
2. 启发式算法和遗传算法的动态集成，使之既可以克服遗传算法相对盲目性，使搜索方向相对明确，又可以克服启发式算法的搜索空间的相对局限性，拓宽搜索的空间范围。
3. 开发智能化装载系统：设计数据维护、装载图形、数据输出模块。

2 第二章 启发式算法在装箱问题中的应用

装箱问题是应用背景较强的离散组合最优化问题。即使是最简单的一维装箱问题,也属于 NP 完全问题。70 年代末,根据 Garey 和 Johnson^[21]关于 NP 完全性理论的研究表明:在有限合理的时间,不可能求得大规模 NP 完全问题的精确解,问题的求解只能依赖于各种启发式方法。

启发式算法的严格定义为:考虑某类问题 P , 设其所有实例集合为 D_p , 对于每一个实例 $I \in D_p$, 有一个相应的待用解集合 $S_p(I)$, 若对任意给定的实例 $I \in D_p$, 算法 A 总能找到一个待用解集 $\sigma \in S_p(I)$, 则称算法 A 为问题 P 的一个启发式算法。

就优化机制与行为而分,目前装箱问题中常用的启发式算法主要分为:经典算法、构造型算法、计算智能算法^[23]和混合型算法等。其中计算智能算法包括:遗传算法^[18,19]、模拟退火算法^[20]、禁忌搜索^[22]等方法。本章的启发式算法只包括经典算法、构造型算法等传统的装箱优化算法。

2.1 启发式方法概述

在计算机科学、人工智能、智能工程、运筹学及其它一些工程学科中,我们经常遇到启发式算法(Heuristic)这个词。在不同学科中,对该词的定义可能不同,并且对它的研究也是分头进行的。然而,通过研究发现,启发式算法具有许多共同的特点,由于各学科在启发式算法研究上的互不理解、互不通气,造成了研究上的重复,并且错过了许多固有的合作机会。

生活经验告诉我们,使用“窍门”求解问题,往往能达到事半功倍的目的。这种窍门来自经验,与求解的问题密切相关,尽管不保证一定奏效,但好的窍门常常一用就灵,在大多数情况下能很快解决问题。这就是说,对许多任务来说,有可能使用与任务或问题有关的信息而大大减少搜索的代价,并找到比较满意的解。这种窍门实质上就是启发式方法。

启发式算法是和问题求解及搜索相关的,也就是说,启发式算法是为了提高搜索效率才提出的。Pearl J. 1984 年在其专著“*Heuristics: intelligent search*”

strategies for computer problem solving”中对启发式算法进行了详细的讨论。他认为:所谓启发式算法是指一组指导算法搜索方向的、建议性质的规则集,通常按照这个规则集,计算机可在解空间中寻找到一个较好解,但并不能保证每次都能找到较好的解,更不能保证找到最优解。换句话说,所谓启发式算法就是那些从大量的实验数据来看,算法的实际计算性能较好,但理论上证明这些算法的最坏解题性能并不好,或者理论上并不能证明这些算法具有优良的解题性能。

这种算法的本质是部分地放弃算法“一般化,通用化”的概念,把所要解的问题的具体领域知识加进算法中去,以提高算法的效率。例如:宽度优先搜索法几乎可以用于一切搜索问题,如九宫图、河内塔、旅行销售商、华容道、以至魔方等等。但在实际使用时,效率也许低得惊人,甚至根本解不出来(如魔方问题)。但如果我们为每类问题找出一些特殊规则,和宽度优先法配合起来使用,那结果就可能完全不一样了。

由于启发式算法的设计不依赖于纯数学中优化理论的突破与发展,因此其研究和应用在最近 20 多年来获得了突飞猛进的发展。尤其是在复杂的组合优化问题上,各种各样的启发式算法更是层出不穷。但正是由于启发式算法缺乏严格的理论推导和数学证明,决定了它的设计具有很强的随意性及创造性。

段国林在“启发式算法及其在工程中的应用”中将启发式算法归纳为七类^[29]:

- (1) 构造法 (Construction)。一个优化问题的解是由若干个构造元素组成的,构造性启发式算法通过一个一个地增加解的构造元素来求得一个可行解。“贪婪法”在每一步都寻找最大的改进,其中包含了大量构造性的启发式算法。在大多数构造性启发式算法中,直到算法结束才会找到可行解。构造性算法的循环次数与问题解的构造元素个数成正比,而与解空间的大小无关,因此,其计算速度通常很快。
- (2) 改进法 (Improvement)。该算法从一个可行解开始,通过在其临域内的搜索如交换、合并结构元素等来改进解的质量。一般来讲,在整个搜索过程中,解一直处于可行状态。将构造法和改进法组合起来使用是很常见的,先利用构造法得到初始解,然后利用改进法搜索临域内的最优点。这种组合法非常普遍。
- (3) 数学规划法 (Mathematical programming)。该方法在问题的数学优

化模型及其精确求解方法的基础上,修改求解方法以得到问题有效的启发式算法。

- (4) 分解法 (Decomposition)。该法指求解一系列容易求解的小问题,一个问题的输出是下一个问题的输入,然后将这些解归纳、合并成一个解。许多规划 (Schedule) 问题的启发式算法使用了分解法。
- (5) 分割法 (Partitioning)。分割法将一个问题分割为几个子问题,然后独立地解决每个子问题。这些子问题的解再合并成整个问题的解。
- (6) 解空间限制法 (Solution space restriction)。该法的思想是限制解的构造,以使问题变得容易求解。在某种意义上,所有的启发式算法都是限制法,然而在这里,是指明确地约束解空间的方法。典型的限制法只允许算法在具有特殊性质的解中搜索。
- (7) 松弛法 (Relaxation)。这种方法与限制法相反,它是指为了得到容易处理的问题而扩展解空间。

虽然把启发式算法分成以上 7 类,但在实际应用中很难明确划分使用的是某一种方法。根据开发者的创造性、经验及设计/修改新的解题方法的技巧艺术,可能将以上几种方法结合起来使用。

2.2 装箱问题求解方法分类

根据装箱问题在各个不同的应用领域的具体特点,结合各领域中的成功经验,可以设计出具有领域特色的启发式装箱求解算法。现有的大多数装箱求解算法都是启发式的,它的设计充分发挥了设计者的创造性,启发式装箱求解算法设计简单、灵活、计算速度快,是目前最重要的一种实用问题求解方法。启发式装箱求解算法大体分为以下几类:

1. 数学规划法 (Mathematical Programming)

数学规划法包括分枝定界法 (Branch-and-bound Algorithm)、动态规划法 (Dynamic programming)、整数规划法 (Integer programming) 和线性规划法 (Linear programming) 等。该方法利用某一优化问题的数学模型,通过修改该模型的精确求解过程得到有效的启发式算法。这四种方法中分枝定界法应用较广泛。该方法的基本思想是试图通过枚举解空间中的有限个解来获得 NP 完全问题

的局部最优解。它由分枝和定界两个操作步骤组成,分枝操作用于将规模较大的原始问题逐步分解为规模较小且易于求解的子问题;而定界操作主要用于评价各分枝的优劣,来减小搜索范围。二维切割问题中,“一刀切”问题是该方法的经典应用,如玻璃切割、纸张裁剪等。

由于数学规划法在进行问题的求解时要把问题抽象为一定的数学模型,所以当实际问题涉及一些不易用数学模型表达的约束条件时,采用各种近似方法得到的数学模型往往与原问题产生较大的偏差,影响问题的求解效果,并且当问题规模增大、解空间急剧增加时,它们的时间复杂性变得很坏。

2. 构造法 (Construction Algorithms)

装箱启发式算法中使用最多的方法是构造性算法。构造启发式算法通过一个一个地增加解的构造元素来求得一个可行解。构造性算法的循环次数与问题解的构造元素个数成正比,而与解空间的大小无关,因此其计算速度通常很快。

装箱问题中构造法基本上由两类规则组成。第一类为定序规则 (Ordering rules),它被用来确定待布局物体放入布局空间的先后顺序;第二类为定位规则 (Placement rules),它用来确定每一个布局物体在布局空间摆放的位置。由于定序规则和定位规则的不同,也就产生不同的构造布局方法。常用的定序规则有:

- (1)按照待装物体最短边边长递减的顺序进行定序;
- (2)按照待装物体最长边边长递减的顺序进行定序;
- (3)按照待装物体体积递减的顺序进行定序;
- (4)按照待装物体最小面积递减的顺序进行定序;
- (5)按照待装物体可行域递减的顺序进行定序;

许多学者对布局问题中定位规则进行了研究,提出了如下一些规则和策略:

- (1)占角策略,即将待装物体摆放在布局容器的某一角;
- (2)顺放策略,即从布局容器的某一角开始将待装物体沿着容器某一边摆放;
- (3)在底盘装载中,将待装物体先沿着边放置,最后摆放到底盘中心;
- (4)在三维规则装箱中,从布局容器的某一面墙开始,一层一层地布局;在某一面墙上,再确定一条边,最后归结为一个角;
- (5)在三维规则装箱中,先按“右、前、上”的顺序找寻剩余空间,然后按照“左、后、下”的顺序摆待装局物体;

启发式方法具有简便、灵活、计算速度快的优点；但它的明显缺陷是所产生的解与最优解的差异很难评价。由于构造法对于装箱问题这类复杂的组合最优化问题具有很强的实用性并能在较短的时间内给出一个令人较满意的解，因而，关于它的研究人们仍然会作进一步的努力。

3. 数值优化方法 (Optimal Algorithms)

数值优化方法具有较为成熟的理论和算法，广泛应用于工程设计领域并取得了许多有效成果，装箱问题是它的一个应用分支，但是数值优化方法依赖于数学模型，且只能找到局部最优解，它只适用于小规模物体的装箱问题。对于规模大的装箱问题用数学模型很难准确描述，即使能用简化的数学模型来描述，由于局部最优解数目的急剧增加，其求解质量也将严重变坏。此外，数值优化方法所得解的质量在很大程度上还依赖于初始解的选择。

4. 改进法 (Improvement Algorithms)

装箱启发式算法中的另一类算法为改进法。改进性启发式算法从一个可行解开始，通过其邻域内的搜索操作如：交换、合并、增加、删除结构元素等来改进解的质量。前面提到的优化算法、线性规划法以及下面将要介绍的遗传算法、模拟退火法也都可用来作为改进算法。

5. 现代优化方法：遗传算法 (Genetic Algorithm, GA), 模拟退火法 (Simulated Annealing, SA)

遗传算法是一种基于生物学进化原理的搜索算法。在解决高维空间、高复杂及非线性问题的优化中具有全局最优、效率高及易于并行计算等优点，有很强的解决问题的能力，但有着收敛速度慢和易陷入局部最优解的缺点。

由于一般组合优化问题与物质的退火过程具有很大的相似性，因此，模拟退火算法被广泛的用来解决组合优化问题，在这方面已有一些成功的应用实例，模拟退火算法可用来解决连续、离散等优化问题，尤其是解空间状态不良的问题。尽管模拟退火算法是一种有可能得到优化问题的全局最优解的问题求解方法，并且已经逐步成为一种用于优化问题求解的一般、通用的方法，但是这是以极其漫长的退火过程即问题求解过程为代价的。

除上述 5 种方法外，神经网络法^[30]、并行算法^[31]、基于知识的求解方法^[32]在装箱问题上的应用研究也越来越受到人们的重视。

2.3 启发式装箱算法现状综述

对于装箱问题，都是用性能比：即近似算法所使用的箱子数目与最少必须使用的箱子数目之比，来评价该近似算法的优劣；另外算法的时间复杂度、是否具有动态特性也是评价近似算法性能的重要依据。

对于一个给定的近似装箱算法 A ，令 $A(L)$ 表示该算法对输入物品序列 L 操作所使用的箱子数目， $OPT(L)$ 则表示最少必须使用的箱子数目，并用 $S(L)$ 表示输入物品序列中所有物品的大小之和。则算法 A 对输入序列 L 的性能比为 $R_A(L) = \frac{A(L)}{OPT(L)}$ 。在分析装箱问题近似算法时，平均性能比 $\overline{R_A}$ 是评测算法性能的一个重要方面，其定义如下

$$\overline{R_A} = \lim_{n \rightarrow \infty} \frac{E[A(L_n)]}{E[OPT(L_n)]}$$

因此装箱问题所关心的主要问题为如何构造装法 A ，使

$$A(L) \leq \alpha OPT(L) + \beta A(U)$$

对任意 n 个物件 $L = \{a_1, \dots, a_n\}$ 成立的常数 α 、 β 最小。不难看出，若 $\beta = 0$ ，则 α 即启发式算法 A 的绝对性能比；若 $\beta > 0$ ，则称 α 为 A 的渐近性能比。通常，将绝对性能比记为 R_A ，将渐近性能比记为 R_A^∞ 。显然总有 $1 < R_A$ ， $R_A^\infty < \infty$ ，且比值越接近 1 表示性能越好，该比值反映了由装法 A 所需的箱子数 $A(L)$ 与客观上所需的最少箱子数 $OPT(L)$ 接近或相差的程度。

1. 早期经典装箱算法

下次适应算法 FF 法 (First Fit Algorithm) 作为最先被研究的、速度最快的算法之一，最简单易行却也是性能最差的一种算法。该算法的思想是：设箱子 B_1, B_2, \dots 的容积 c 和 $L = \{a_1, \dots, a_n\}$ 已经给定。(以下皆设 $c=1$) 我们就按照给定的表 L 先将 a_1 装入 B_1 。若余下的空出能容下 a_2 ，则将 a_2 放入 B_1 ，否则将 B_1 送走，将 a_2 放入 B_2 。一般说来，若已将 a_i 放入 B_j ，在处理 a_{i+1} 时，若 B_j 还有空处能容

纳 a_{i+1} , 则将 a_{i+1} 放入 B_j 之中, 否则将 B_j 送走, 将 a_{i+1} 放入 B_{j+1} 之中。即按表中的顺序装箱, 若正在使用中的箱子容不下新送来的物件, 则将该箱子送走。。容易证明 (在 n 上施行归纳法): $NF(L) \leq 2OPT(L)$, $\forall L$ 。式中的系数 2 时不能改进的。因为对于 $L = \left\{ \frac{1}{2}, \frac{1}{2N}, \frac{1}{2}, \frac{1}{2N}, \dots, \frac{1}{2}, \frac{1}{2N} \right\}$ (共 $2N$ 个 $\frac{1}{2}$), 我们有 $NF(L) = 2OPT(L) - 1$ 。

NF 算法的最大缺点是: 不去利用已经使用过的箱子的空处。 FF 算法则是针对这一缺点设计的, 对于 FF 算法: 设 a_i 已装入 B_j 。在处理 a_{i+1} 时, 我们先观察已经放用过的箱子 $B_1 \dots B_j$ 中是否有能容下 a_{i+1} 的箱子, 若有, 则将 a_{i+1} 放入其中的编号最小者, 否则将 a_{i+1} 放入 B_{j+1} 。这样的装法文献里称为 FF 法 (First Fit Algorithm)。上述算法有 $FF(L) \leq \frac{17}{10}OPT(L) + 1$, $\forall L$, 式中的系数 $\frac{17}{10}$ 也是不可改进的, 在 [5] 中有例子说明, 可以有 $FF(L) > \frac{17}{10}OPT(L) - 8$ 。上述结果的证明是相当困难的。

若仔细检查一下 FF 算法, 则易发现: 这是由于在 L 中, 有可能较大的 a_i 放在后面, 前面已经用过的箱子虽然还有空余, 但不足容纳后面的物件, 只好启用新的箱子, 为了纠正这一缺点, 人们就设计出 FFD 算法。他是先将 L 中的 a_i 按从大到小的顺序排列, 即 $a_1 \geq a_2 \geq \dots a_n$, 然后施行 FF 算法, 使得可以用后面的物件去填补前面箱子的空缺。

关于装箱问题, 最早的一个著名结果是由 Johnson^[24]得到的。他所处理的算法是 FFD (First Fit Decreasing) 算法。对于这一算法, 他证明了 $FFD(L) \leq \frac{11}{9}OPT(L) + 4$ 对所有的 L 成立, 并证明了 $\frac{11}{9}$ 是最小的数, 即对任何 $\alpha < \frac{11}{9}$, 皆可找到一张表 L , 使得 $FFD(L) > \alpha OPT(L)$, 他的文章一直没有全文发表, 主要原因是文章太长, 达一百多页, 引理达 75 个之多。据 1985 年 Johnson 和 Garey 的文章中说, 原文的一条引理的证明是非常迂回曲折, 而且还多少有些靠不住 (and perhaps somewhat shaky)。在 1985 年的文章中, 他们对这一引

理给出了一个“简单”的证明，但也有八页之多。之后，Baker 给出了 $FFD(L) \leq \frac{11}{9}OPT(L)+3$ 的证明，越民义 1990 年给出了 $FFD(L) \leq \frac{11}{9}OPT(L)+1$ 的证明，李荣琉、越民义给出了 $FFD(L) \leq \frac{11}{9}OPT(L)+\frac{7}{9}$ 的证明^[25]。

如前所述，自 Johnson 的结果公布以后，相继出现了不少有关装箱问题的文章，在[7]中，Johnson 等人对 84 年以前的文献（共 110 篇）作了详尽的报道，当时对这一问题的研究正处于高潮时期，从 81 年到 84 年这三年之间，文献比过去翻了一番。

下面这张表摘自文献[26]。列举了各种早期经典的装箱算法的时间复杂度、渐近性能比等。值得注意的是，在早期研究过程中，随着算法的性能越来越好，其实践性也变得越来越差，方法不易实行。

Algorithm	Timing	R_A^∞
Worst Fit	$O(n \log n)$	2.0
Next Fit	$O(n)$	2.0
Best Fit	$O(n \log n)$	1.7
Almost worst fit	$O(n \log n)$	1.7
NF decreasing	$O(n \log n)$	1.691...
Revised FF	$O(n \log n)$	1.666...
Group fit grouped	$O(n)$	1.5
FF grouped	$O(n \log n)$	1.333...
First Fit	$O(n \log n)$	1.7
Iterated LFD	$O(n \log n)$	1.333...
FF decreasing	$O(n \log n)$	1.222...
BF decreasing	$O(n \log n)$	1.222...
Modified FFD	$O(n \log n)$	1.183...
Next-2 Fit		
AWF decreasing		

表 2-1 经典装箱算法性能比较

2. 二维装箱问题中的启发式算法

二维装箱问题的研究, 主要从实际工作角度出发, 设计一些实用方法, 虽然也有些理论结果, 但并不太深入, 主要是问题太难, 使人无从下手。文献[33]对二维装箱问题进行了综述:

早在 60 年代初, Codd (1960) 在多极程序设计系统 (multiprogramming system) 的研究中, 便讨论过二维装箱问题的应用。70 年代中期, Garey 与 Graham 在多处理系统 (multiprocessor system) 研究中, 也考虑过类似问题。其它文献也曾讨论过相关问题, 如 Erdos 与 Graham 证明了矩形区域内关于正方形群体的直角装填一般而言并不能达到最优, 基于 Meir 与 Moser 的早期工作, Kleifman 与 Krieger 考虑了关于正方形布局的最小外包络矩形的装箱问题等等。1980 年, Baker 等首次提出了第一类型的二维装箱问题, 并设计了 BL (Bottom-up Left-justified) 算法, 给出如下性能结果: 若群体以其宽度递降排列, 则有 $R_{BL} \leq 3$, 若群体为正方形, 则有 $R_{BL} \leq 2$ 。继之, Coffman 等 (1980) 提出下次适合递降高度 NFDH 及首次适合递降高度 FFDH 算法, 得出如下结果: 若群体以递降高度排列, 则 $R_{NFDH}^{\infty} = 2$, $R_{FFDH}^{\infty} = 1.7$ 。Baker 等 (1981) 在 NFDH 与 FFDH 算法基础之上, 设计了 $R_{SF}^{\infty} = \frac{5}{4}$ 的 UD 算法, Baker 与 Schwarz (1983) 注意到以往算法均需将群体按一定次序重排, 而这一手续在某些应用中难以实现, 针对该问题, 设计了按任意给定排列次序进行放置的 Shelf 算法, 其性能可任意接近 NFDH 及 FFDH 之性能。

Chung 等在 1982 首次研究了第二类二维装箱问题, 基于第一类二维问题的 FFDH 算法, 设计了求解该类问题的混合首次适合算法 HFF 具有性能 $2.022 \leq R_{HFF}^{\infty} \leq 2.125$; Berkey 与 Wang (1987) 将若干求解一维及第一类二维问题的方法推广到该类问题。通过大量实例比较了它们的实践情况。此外, Csiriu 等, Ong 等讨论了关于算法的或然性及平均性能分析。Barnes (1982) 以代数学的角度研究了装箱问题。引出了若干有意义的理论结果。

2. 三维装箱问题中的启发式算法

三维装箱问题由于其复杂性和难度大, 研究的就更少了, 且各种算法都是基于一定的假设之下, 理论贡献也不多。而且主要都是基于空间分解或层的思想。

下面从实用角度介绍一些三维装箱算法。

Li 和 Cheng^[27]在 1990 年提出了多项式近似算法 C_1 ，其渐近性能比 $R(C_1)=2.6875$ 。沈灏^[40]根据算法 C_1 的思想，进一步利用盒子底部为正方形的特点，提出了所谓“单元装箱法”使新算法的渐近性能比得到改进： $R(D) \leq 2.32015$ 。

杨传民等人^[34]通过全面枚举搜索方法来研究相同大小的立方体的装箱优化问题。Mannchen^[35]设计的树搜索算法，理论上对三维箱体布局有效，但由于三维箱体布局属于 NP 完全问题，随着布局箱体的增多，解空间急剧膨胀，因而计算效率较低。H. GEHRING^[36]提出按阶段填充，在深度方向按层布局的启发式算法，不仅考虑了空间利用率，而且还考虑了重心平衡。王金敏^[38]在其基于约束的布局求解理论与方法的研究中对三维布局问题中的底盘装载问题、约束底盘装载问题给出了启发式求解方法，研究了布局物体的定序方法、定位方法，以及布局中的装卸问题。何大勇，鄂明成等^[39]则在集装箱布局问题中提出了一种利用三叉树结构表达三维矩形物体布局状态空间的方法，将布局空间依次分解，定位规则与定序规则并用，给出了由计算机随机产生的 30 个长方体的布局结果，得出了集装箱的利用率随原始布局长方体的体积与集装箱体积的比值的变化趋势的实验曲线。

2.4 一种构造启发式装箱算法实例

2.4.1 约束条件及其处理

对算法中的约束条件处理方法就是引入不同变量分别表示货物的摆放方式、货物的配套、底置等级和优先级等属性。

2.4.2 装箱的启发式策略

- 1) 同种类型的货物一次性装载。
- 2) 当装载不满时应避免中间突出的情况，以免使空间过于破碎。如有中间突出的情况，应把突出的摆放换到边上，这样可以利用最外边的一段空间。如有小于最小尺寸的边，应尽量把此类情况转换为其它方式。

3) 在宽度方向求最优, 是为了维护空间完整度。

2.4.3 空间划分

因为启发式策略是将一种产品集中装载, 因此将一种产品装入后, 最多会产生如图 2-1 所示的六种子空间。

装载的顺序是: D-E-C-F-B-A, 即按空间体积的递减顺序装载, 增加空间利用率。

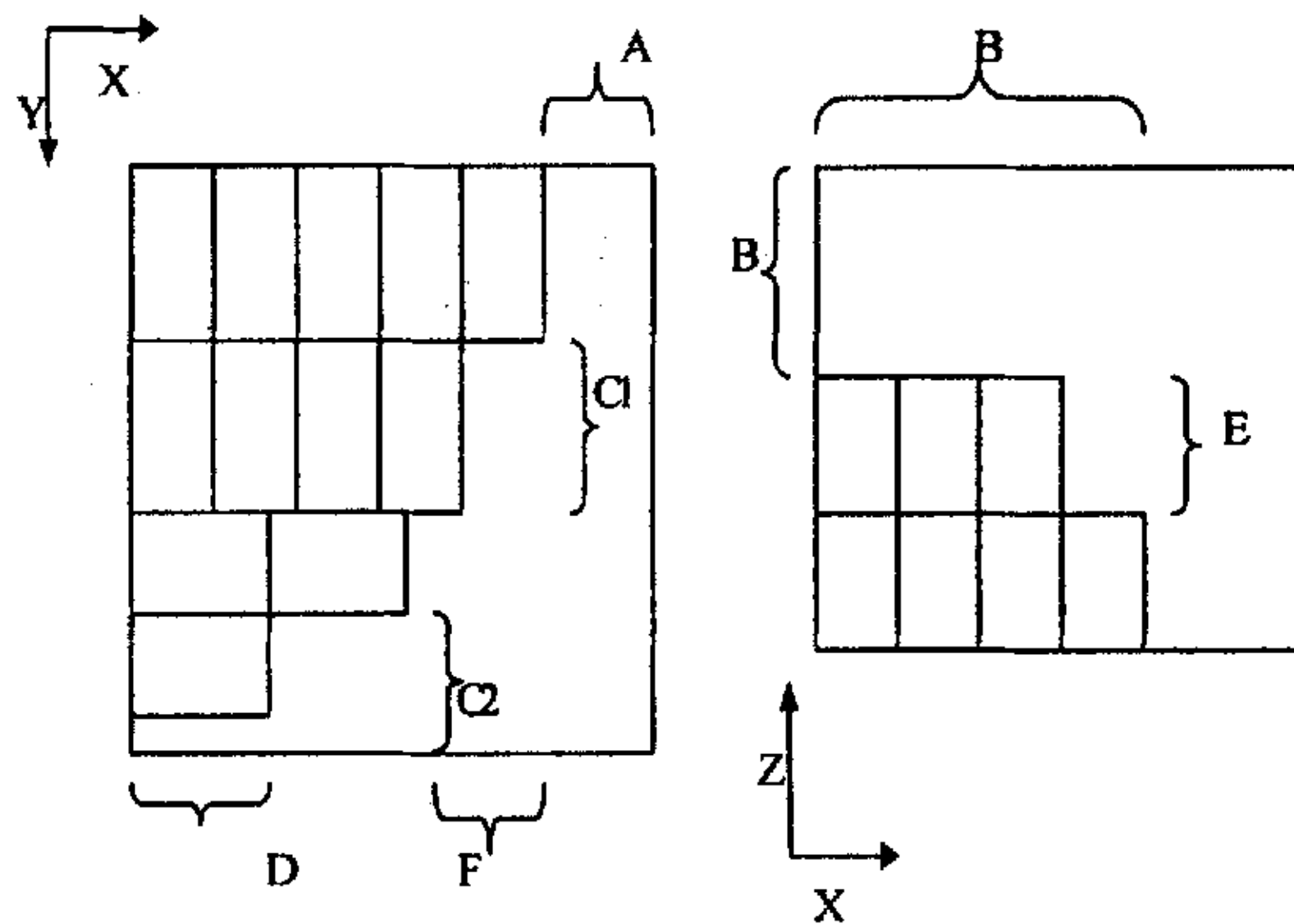


图 2-1 构造启发式装箱算法空间分解图

2.4.4 定序规则和定位规则

- 1、按照待装物体体积递减的顺序进行定序;
- 2、定位策略为: 占角策略, 即将待装物体摆放在布局容器的某一角;

3 第三章 遗传算法在装箱问题中的应用

3.1 遗传算法的起源与发展

遗传算法起源于对生物系统所进行的计算机模拟研究。虽然早在 50 年代初期,就有研究人员开始研究运用数字计算机模拟生物的自然遗传与自然进化过程,到 50 年代末期,已有一些这方面的论文发表,但是当时从事这方面研究的主要是一些生物学家,研究的目的主要是为了更深入地理解自然遗传与自然进化现象^[41]。进入 60 年代后,美国的 Michigan 大学的心理学及计算机科学教授 John H. Holland 在从事如何建立能学习的机器中注意到,学习不仅可以通过单个的生物体适应而且通过一个种群的许多代进化适应也能发生。Holland 教授开始认识到生物的自然遗传现象与人工自适应系统行为的相似性,他认为不仅要研究自适应的系统,还要研究与之相关的环境,因此他提出在研究和设计人工自适应系统时,可以借鉴生物自然遗传的基本原理,模仿生物自然遗传的基本方法。1967 年,他的学生 J. D. Bagley 在其博士论文中首次提出“遗传算法 (Genetic Algorithms)”一词。

到 70 年代初, Holland 提出了“模板定理” (Schema Theorem), 一般认为是“遗传算法的基本定理”,从而奠定了遗传算法研究的理论基础。1975 年, Holland 总结了自己的研究成果,发表了在遗传算法领域具有里程碑意义的著作《Adaptation in Natural and Artificial System》。在这本书中, Holland 为所有的适应系统建立了一种通用理论框架,并展示了如何将自然界的进化过程应用到人工系统中去。Holland 认为,所有的适应性问题都可以表示为“遗传”问题,并用“进化”方法来解决。以后, Holland 等人将该算法加以推广,应用到优化及及其学习等问题中,并正式定义为遗传算法。人们通常把 Holland 所提出的遗传算法简称为简单遗传算法或标准遗传算法,简记为 SGA。80 年代以后,遗传算法广泛应用到各种复杂系统的自适应控制以及复杂的优化问题中。

1985 年,在美国召开了第一届遗传算法国际会议,并且成立了国际遗传算法学会 (ISGA, International Society of Genetic Algorithms)。此会以后每两年举行一次。1989 年, Holland 的学生美国伊利诺大学 D. J. Goldberg 出版了

《Genetic Algorithm in Search, Optimization, and Machine Learning》，这本书对于遗传算法理论及其多领域的应用展开了较为全面的分析和例证。一般认为，这一时期的遗传算法从古典阶段发展到了现代阶段，本书则奠定了现代遗传算法的基础。1991 年，L. Davis 编辑出版了《Handbook of Genetic Algorithms》，其中包活了遗传算法在工程技术和社会生活中大量的应用实例。这本书为推广和普及遗传算法的应用起到了重要的指导作用。1992 年，Michalewicz 出版了另一本很有影响力的著作“Genetic Algorithms + Data Structures = Evolution Programs”，对遗传算法应用于最优化问题起到了推波助澜的作用。

从 80 年代开始，有关遗传算法的研究和应用日益普遍。目前，几乎所有领域的研究人员都尝试过遗传算法在各自专业领域的应用，并取得了丰硕的成果。在实际应用过程中，遗传算法也得以进一步完善和发展。

3.2 遗传算法概述

3.2.1 遗传算法的生物学基础

遗传算法的基本思想是基于 Darwin 进化论和 Mendel 的遗传学说的。

Darwin 进化论最重要的是适者生存原理。它认为每一物种在发展中越来越适应环境。物种每个个体的基本特征由后代所继承，但后代又会产生一些异于父代的新变化。在环境变化时，只有那些适应环境的个体特征方能保留下来。

Mendel 遗传学说最重要的是基因遗传原理。它认为遗传以密码方式存在细胞中，并以基因形式包含在染色体内。每个基因有特殊的位置并控制某种特殊性质。所以，每个基因产生的个体对环境具有某种适应性。基因突变和基因杂交可产生更适应于环境的后代。经过存优去劣的自然淘汰，适应性高的基因结构得以保存下来。

由于遗传算法是由进化论和遗传学机理而产生的直接搜索优化方法，故而在在这个算法中要用到各种进化和遗传学的概念。这些概念如下：

1、串 (String)

它是个体 (Individual) 的形式，在算法中为字符串，并且对应于遗传学中的染色体 (Chromosome)。

2、群体 (Population)

个体的集合称为群体，串是群体的元素

3、群体大小 (Population Size)

在群体中个体的数量称为群体的大小。

4、基因 (Gene)

基因是串中的元素，基因用于表示个体的特征。例如有一个串 $S=1011$ ，则其中的 1, 0, 1, 1 这 4 个元素分别称为基因。它们的值称为等位基因 (Alleles)。

5、基因位置 (Gene Position)

一个基因在串中的位置称为基因位置，有时也简称基因位。基因位置由串从左向右计算，例如在串 $S=1101$ 中，0 的基因位置是 3。基因位置对应于遗传学中的地点 (Locus)。

6、基因特征值 (Gen Feature)

在用串表示整数时，基因的特征值与二进制数的权一致

7、串结构空间 SS

在串中基因任意组合所构成的串的集合。基因操作是在结构空间中进行的。串结构空间对应于遗传学中的基因型 (Genotype) 的集合。

8、参数空间 SP

这是串空间在物理系统中的映射，它对应于遗传学中的表现型 (Phenotype) 的集合。

9、非线性

它对应遗传学中的异位显性 (Epistasis)。

10、适应度 (Fitness)

表示某一个体对于环境的适应程度。

3.2.2 遗传算法的基本原理

遗传算法是一种建立在自然选择和群体遗传机理基础上的自适应概率性搜索算法，它根据“适者生存，优胜劣汰”等自然进化规则演化而来。与传统搜索算法不同，遗传算法把优化问题的解的搜索空间映射为遗传空间，把每一可能的解编码为一个称为染色体的二进制串 (也有其它编码方法)，染色体的每一位称为基因。每个染色体 (对应一个个体) 代表一个解，一定数量的个体组成群体。

搜索时, 首先随机地产生一些个体组成初始群体(即问题的一组候选解), 初始种群产生之后, 按照适者生存和优胜劣汰的原理, 逐代演化产生出越来越好的近似解。在每一代, 按预先根据目标函数确定的适应度函数计算各个体对问题环境的适应度, 再根据个体适应度对个体对应的染色体进行选择, 抑制适应度低的染色体, 弘扬适应度高的染色体, 然后进行选择、交叉、变异等遗传操作产生进化了的新一代群体。如此反复操作, 一代一代不断向更优解方向进化, 最后得到满足某种收敛条件的最适应问题环境的群体, 从而获得问题的最优解。

遗传算法的形式化描述如下:

```
begin
     $0 \rightarrow t$ ;
    初始化  $P(t)$ ;
    评估  $P(t)$ ;
    while 不满足中止条件 do
        begin
            遗传运算(  $P(t)$  )  $\rightarrow C(t)$ ;
            评估  $C(t)$ ;
            选择(  $P(t)$ ,  $C(t)$  )  $\rightarrow P(t+1)$ ;
             $t+1 \rightarrow t$ 
        end
    end
```

其中 $P(t)$ 表示第 t 代种群, $C(t)$ 表示第 t 代种群的后代。

3.2.3 遗传算法的工作机理

遗传算法是建立在自然选择和群体遗传学机理基础上的随机、迭代、进化, 具有广泛使用性的搜索方法, 是借鉴自然选择和自然进化原理, 模拟生物在自然界中的进化过程所形成的一种优化求解算法。尽管这种自适应寻优技术可用来处理复杂的线性、非线性问题, 但它的工作机理却十分简单。标准遗传算法(Canonical Genetic Algorithms)的步骤如下:

1. 构造满足约束条件的染色体。染色体是根据问题特点, 选择适当的编码方式, 对应问题解的一种编码方案。
2. 随机产生初始群体。初始群体是一组染色体的集合, 其数量要根据问题的规

模及编码方案适当选择。

3. 计算每个染色体的适应度。适应度是反映染色体优劣的唯一标准。遗传算法就是要寻找适应度最大的染色体。在实际应用中即是最优解的编码形式,有时也可能是次优解。
4. 使用复制、交叉和变异等算子对初始群体进行变换产生新的群体。
5. 重复步骤 3、4 直到新的群体中存在满足条件的染色体终止。

3.2.4 遗传算法的特点

为了解决各种优化计算问题,人们提出了各种各样的优化算法,如单纯形法、梯度法、动态规划法、分枝定界法等。这些优化算法各有各的长处和限制,有各自不同的适用范围。与普通的搜索算法(如梯度算法)一样,遗传算法也是一种迭代算法。但是与其他一些优化搜索算法相比,遗传搜索要求两个存在明显冲突的方向之间的平衡:挖掘(Exploitation)最优解和探测(Exploration)搜索空间^[42]。爬山法(Hill Climbing)是尽可能地改进挖掘最优解策略的例子,但它忽视了对搜索空间的探测。随机搜索(Random Search)是一个注重探测搜索空间而忽视挖掘最优解的典型例子。遗传算法则是在总体上寻求挖掘和空间探测之间平衡的一类算法。

遗传算法是一类可用于复杂系统优化计算的鲁棒搜索算法。与其他一些优化算法相比,它主要有下述几个特点^[41]:

(1) 遗传算法以决策变量的编码作为运算对象(函数优化中有时也采用决策变量本身作为运算对象,可以视为简单的代数编码)。传统的优化算法往往直接利用决策变量的实际值本身来进行优化计算,但遗传算法不是直接以决策变量的值,而是以决策变量的某种形式的编码为运算对象。这种对决策变量的编码处理方式,使得我们在优化计算过程中可以借鉴生物学中染色体和基因等概念,可以模仿自然界中生物的遗传和进化等机理,也使得我们可以方便地应用遗传操作算子。特别是对一些无数值概念或很难有数值概念,而只有代码概念的优化问题,编码处理方式更显示出了其独特的优越性。

(2) 遗传算法直接以目标函数值作为搜索信息,并不利用梯度信息,无需求导运算。传统的数值优化算法不仅需要利用目标函数值,而且往往需要目标函数

的导数值等其他一些辅助信息才能确定搜索方向。而遗传算法仅使用由目标函数值变换来的适应度函数值,就可确定进一步的搜索方向和搜索范围,无需目标函数的导数值等其他一些辅助信息。这个特性对很多目标函数是无法或很难求导数的函数,或导数不存在的函数的优化问题,以及组合优化问题等,应用遗传算法时就显得比较方便。因为它避开了函数求导这个障碍。再者,直接利用目标函数值或个体适应度,也可使得我们可以把搜索范围集中到适应度较高的部分搜索空间中,从而提高了搜索效率。

(3) 遗传算法同时使用多个搜索点的搜索信息,是一种群体优化过程。传统的优化算法往往是从解空间中的一个初始点开始,进行最优解的迭代搜索。单个搜索点所提供的搜索信息毕竟不多,虽然搜索效率较高,但容易使搜索过程陷于局部最优解而停滞不前。遗传算法从由很多个体所组成的一个初始群体开始最优解的搜索过程,而不是从一个单一的个体开始搜索。对这个群体所进行的选择、交叉、变异等运算,产生出的乃是新一代的群体,在这之中包括了很多群体信息。这是遗传算法所特有的一种并行性。

(4) 遗传算法使用概率搜索技术。很多传统的优化算法往往使用的是确定性的搜索方法,一个搜索点到另一个搜索点的转移有确定的转移方法和转移关系,这种确定性往往也有可能使得搜索永远达不到最优点,因而也限制了算法的应用范围。而遗传算法属于一种自适应概率搜索技术,其选择、交叉、变异等运算都是以一种概率的方式来进行的,从而增加了其搜索过程的灵活性。虽然这种概率特性也会使群体中产生一些适应度不高的个体,但随着进化过程的进行,新的群体中总会更多地产生出许多优良的个体,实践和理论都已证明了在一定条件下遗传算法总是以概率 1 收敛于问题的最优解。当然,交叉概率和变异概率等参数也会影响算法的搜索效果和搜索效率,所以如何选择遗传算法的参数在其应用中是一个比较重要的问题。而另一方面,与其它一些算法相比,遗传算法的鲁棒性又会使参数对其搜索效果的影响会尽可能地低。

3.2.5 遗传算法改进方式综述

自从 1975 年 J. H. Holland 系统地提出遗传算法的完整结构和理论以来,众多学者一直致力于推动遗传算法的发展,对编码方式、控制参数的确定、选择方

式和交叉机理等进行了深入的探究,引入了动态策略以改善遗传算法的性能,提出了各种变形的遗传算法(Variants of Canonical Genetic Algorithms,简称VCGA)。其基本途径概括起来有下面几个方面:

1. 改进遗传算法的组成成分或使用技术,如选用优化控制参数、适合问题特性的编码技术、改进适应度定义的方法等。其中一个重要的对控制参数的改进是:引入了代沟(generation gap)概念控制个体更新百分比,即第 t 代有 $n(1-G)$ 个个体选择复制到第 $t+1$ 代群体中,代沟为遗传算法利用进化的历史信息提供了条件,加速了遗传算法的收敛过程。

2. 采用混合遗传算法:将其他优化技术与遗传算法混合使用,以提高算法的综合能力。例如,为了提高算法的全局优化能力,通常在选择机制中融合模拟退火技术^[50]。为提高算法在最优点附近的局部寻优能力,融合其他启发式搜索或传统优化技术^[51]。

3. 采用动态自适应技术,在进化过程中调整算法控制参数和编码粒度,提高算法的可靠性和对各类问题的适应能力^[49]。

4. 采用非标准的遗传操作算子:为了提高算法搜索能力,提出了多种新的遗传算子。例如倒序变异算子、灾变算子^[48]以及多种选择算子等。但是,多数新算子缺乏有效性的理论依据。

5. 采用并行遗传算法:模拟生物种群进化中的独立种族和小生境进化现象,各个独立的进化群体采用复杂的机制相互协作^[49,50],提高算法的寻优效果。

6. 新的进化机制:在遗传算法中引入新的生物进化机制,并试图通过解释其寻优机理来研究进化计算的理论基础。免疫算法^[46]和蚁群算法^[47]作为一种新的进化机制也开始应用于复杂优化问题的求解。

7. 分布式遗传算法:模拟生物进化中的基因隔离和基因迁移,将群体分成若干亚群,在各亚群内,按通常遗传算法进行遗传操作,每个亚群之间按一定的比例进行优良个体迁移。这样一方面保证了各亚群搜索的独立性和充分性,又能使各亚群共享优良基因信息,有利于收敛至全局最优解。

3.3 遗传算法中的基本实现技术

3.3.1 编码方法

在遗传算法中,首先要将搜索空间解的表示映射成遗传空间解的表示,这一操作称为编码,其相反操作称为解码。对编码的基本要求是,两个空间的解需一一对应且编码且尽量简明。遗传空间的解又称个体(individual)或染色体(chromosome)。个体通常由字符串表示,字符串的每一位称为遗传因子(gene)。多个个体组成一个种群(population)供遗传算法使用。值得一提的是,遗传算法不仅可以对数值符号集组成的字符串进行操作,还能够直接对结构对象进行操作。所谓结构对象泛指集合、序列、矩阵、树结构、图和表等各种结构形式的对象。遗传算法的这个特点使它具有广泛的应用领域。

编码是连接问题与算法的桥梁、应用遗传算法是要解决的首要问题,也是设计遗传算法时的一个关键步骤。编码方法除了决定个体的染色体排列形式外,还决定了个体从遗传空间的基因型变换到搜索空间的表现型时的解码方法;编码方法也影响到交叉算子、变异算子等遗传算子的运算方法。由此可见,编码方法在很大程度上决定了如何进行群体的遗传进化运算以及遗传进化运算的效率。一个好的编码方法,有可能会使得交叉运算、变异运算等遗传操作可以简单的实现和执行。反之,可能使得交叉运算、变异运算等遗传操作难以实现,也可能产生很多无效解。

Holland 的编码方法是二进制编码,但对于许多遗传算法的应用,特别是在工业工程中的应用,这种简单的编码方法很难直接描述问题的性质。近十年来,针对特殊问题,人们提出了其它编码方法。例如:格雷编码、浮点编码、符号编码。

二进制码编码、解码操作简单易行,不仅便于实现交叉、变异等进化操作,而且便于利用模式定理对算法进行理论分析。格雷码是二进制码的变形,具有与二进制码相似的特点,其编码、解码操作相对复杂,但基于格雷编码的算法局部搜索能力有较大提高。浮点编码方式编码长度等于参数向量的维数,达到同等精度要求的情况下,编码长度远小于二进制码和格雷码。此外,浮点编码使用的决策变量的真实值,不需反复进行数据转换。符号编码方法^[43]是指染色体编码串中

的基因值取一个无数值含义、而只有代码含义的符号集。这些符号可以是字符,也可以是数字。符号编码的主要优点是便于在遗传算法中用所求问题的专门知识及相关算法。

对于非二进制编码,染色体编码与问题的解之间有三个主要问题:①染色体的可行性;②染色体的合法性;③映射的唯一性。可行性是指染色体编码成为解之后是否在给定问题的可行域内。合法性是指染色体编码是否代表给定问题的一个解。

3.3.2 适应度函数

遗传算法中使用适应度来度量群体中各个个体在优化计算中有可能达到或接近于或有助于找到最优解的优良程度。适应度较高的个体遗传到下一代的概率就较大;而适应度较低的个体遗传到下一代的概率就较小。度量个体适应度的函数称为适应度函数。

遗传算法在进化搜索中基本不利用外部信息,仅以适应度函数(fitness function)为依据,利用种群中每个个体的适应度值来进行搜索。因此适应度函数的选取至关重要,直接影响到遗传算法的收敛速度以及能否找到最优解,合理的适应度函数能引导搜索向最优化方向进行,设计不当有可能会产生骗问题^[41]。此外,它对遗传算法的终止搜索条件也有影响。值得提出的是,作为一种随机搜索方法,遗传算法的适应度函数或目标函数不受连续可微的约束,而且函数的定义域可以是任意集合。对目标函数的唯一要求是对于输入可以计算出能加以比较的输出。当然适应度函数设计应尽可能简单,以减少计算时间和空间上的复杂性。

一般而言,适应度函数是有目标函数变换而成的。对目标函数值域的某种映射变换称为适应度的尺度变换(fitness scaling)。常用的变换方法有:线性尺度变换、方差缩放技术、乘幂尺度变换和指数尺度变换。基本遗传算法中采用的是线性尺度变换。

3.3.3 选择算子^[43-45]

选择源于对自然界优胜劣汰竞争原则的模拟。在生物的遗传和自然进化过程中,对生存环境适应程度较强的个体,将有更多的机会遗传到下一代,繁殖下一

代的数目较多；而对环境适应程度较差的个体将遗传到下一代的机会就较少，甚至被淘汰。这样，就更容易产生对环境适应能力较强的后代。

遗传算法中所谓选择运算即从上一代种群中选择一定量染色体到新一代种群的方法。优胜劣汰的选择机制使得适应值大的解有较高的存活概率，这是遗传算法与一般搜索算法的主要区别之一。发展各种选择算子的目的是为了减少基因缺失，提高遗传算法的全局收敛性和搜索效率。

选择策略对算法性能的影响会起到举足轻重的作用。不同的选择策略对算法的性能也有较大的影响。不同的选择策略将导致不同的选择压力，即下一代中父代个体的复制数目的不同分配关系。较大的选择压力使最优个体具有较高的复制数目，从而使得算法收敛速度较快，但也较容易出现过早收敛现象。相对而言，较小的选择压力一般能使群体保持足够的多样性，从而增大了算法收敛到全局最优的概率，但算法的收敛速度一般较慢。

选择算子的设计的内容就是确定如何从父代群体中按某种方法选取个体遗传到下一代群体中。常用的选择策略包括：比例选择、最优保存策略、确定式采样选择、无回放随机选择、无回放余数随机选择。比例选择方法简单，但容易出现早熟收敛，速度慢现象。无回余数随机选择降低了选择误差，保证优秀个体的遗传，但操作相对复杂。确定式采样选择降低了选择误差，操作相对于无回余数随机选择简单。基本遗传算法中采用的是比例选择。

3.3.4 交叉算子^[43-45]

交叉起源于模拟自然界有性繁殖过程中的分裂和重组过程。经过这个过程，生物物种性状得到传递和延续。一般，子代个体具有父母双方的优点，因而表现出比父代更优秀的性状。在遗传算法运行过程中，交叉的作用也在于此，交叉的结果是个体更趋近最优解。

遗传算法中所谓交叉运算，是指对两个相互配对的染色体按某种方式相互交换其部分基因，从而形成两个新的个体。交叉运算是遗传算法区别于其他进化算法的重要特征，它在遗传算法中起关键作用，是产生新个体的主要方法。遗传算法中，在交叉运算之前还必须对群体中的个体进行配对，目前常用的配对策略是随机配对。然后在配对染色体中随机选择杂交位置进行杂交。杂交的目的是为了

产生适应值更高的染色体,从而使各代染色体都向前进化。通常交叉的方式组合出新个体,在串空间进行有效搜索,同时降低对有效模式的破坏概率。

交叉算子的设计包括两个方面的内容:①如何确定交叉点的位置?②如何进行部分基因的交换?设计的指导原则是既不要太多的破坏表示优良性状的基因,又要产生一些较好的新个体模式。常用的交叉方法有单点交叉、双点交叉、多点交叉、均匀交叉和算术交叉。其中前三者适合于二进制编码方式,后两种适合浮点数编码方式。基本遗传算法中采用的是单点交叉。

3.3.5 变异算子^[43-45]

变异起源于对生物进化过程中基因突变的模拟。自然界中丰富多彩的物种正是基因突变的功劳。基因突变的概率虽然很小,但是突变结果有时却能带来非常优秀的性状,当然也会有不适应环境的个体的产生。

遗传算法中所谓变异运算,是指将个体编码串中的某些基因值用其它基因值来替换,从而形成一个新的个体。变异操作的主要目的是改善算法的局部搜索能力,并维持群体的多样性,防止出现早熟现象。当产生的后代并不比前辈更为优秀且又未达到可行性解,就会产生不成熟的收敛。为了克服这种情况,只有依赖于变异。当算法陷入局部搜索时,可以通过变异算子,使个体跳出局部空间,寻求全局最优。变异的过程通常是在一定概率下进行的,而且只变动染色体中某一个或几个基因。遗传算法中的变异运算是产生新个体的辅助方法,但它是必不可少的一个运算步骤,因为它决定了遗传算法的局部搜索能力。交叉运算和变异运算的相互配合,共同完成对搜索空间的全局搜索和局部搜索。

变异运算的设计包括两方面:①如何确定变异点的位置?②如何进行基因值替换?下面介绍几种常用的变异操作方法。常用的变异方法有基本位变异、均匀变异、非均匀变异、二元变异、高斯变异。均匀变异适合于遗传算法运行的初期阶段,它可以在整个搜索空间随即游动,能够增加种群的多样性。不利于重点区域内的搜索。而非均匀变异这可以克服这个缺点。高斯变异也是提高重点区域局部搜索性能的变异方法。其中基本位变异适合于二进制编码方式,后四种适合浮点数编码方式。基本遗传算法中采用的是基本位变异。

3.3.6 控制参数

在进行遗传算法程序时,要对一些参数做事先选择,它们包括群体规模、交叉概率、变异概率、最大进化代数、染色体的长度等,这些参数对遗传算法性能有很重要的影响。这些参数的选择和设定目前尚无统一的理论指导,多数都视具体问题而定。对一个具体问题,要想获得最优的参数设置,通常采取试验法和二级演化算法(meta-EA)两种方法之一,虽然这两种方法能就某一类问题获得较佳的参数设置,但由于它们的计算量都比较大,从而在计算实际问题时用得并不是很多。常用的方法是根据经验设定控制参数值,然后多执行算法几次以获得问题的最优或近似最优解。

1. 群体规模 M

即群体中个体的数目。群体规模影响到遗传算法的最终结果以及遗传算法的执行效率。一般来讲,选择较大数目的初始种群可以同时处理更多的解,因而容易找到全局最优解,其缺点是增加了每次迭代的时间,计算复杂度高。而当种群规模 M 较小时,算法的运算速度是提高了,但却降低了群体的多样性,容易陷入局部最优解,导致早熟现象。一般取为 20~100^[44]。

2. 交叉概率 p_c

交叉概率的选择决定了交叉操作的频率。如果频率过小,会使搜索过程缓慢,以至停滞不前。频率越高,产生新个体的速度就越快,就可以越快的收敛到最有希望的最优解区域,因此一般选取较大的交叉率,但太高的频率可能导致具有高度适应度的个体结构很快就会被破坏,过早的收敛,一般取值 0.4~0.9^[44]。

3. 变异概率 p_m

变异概率的选取一般受种群大小、染色体长度等因素影响,通常选取很小的值,一般取 0.001~0.1^[44]。若选取高的变异率,虽然增加样本的多样性,但也有可能破坏掉很多较好的模式,那么遗传算法就变成了纯粹的随机搜索算法。种群大小及染色体长度越大,变异率选取越小。但变异率太小,就不容易产生新的个体结构,抑制早熟现象的能力也就会很差了。

4. 最大进化代数

最大进化代数作为一种模拟终止条件,一般是具体问题而定,取 100~500^[44]。

5. 染色体的长度

染色体的长度主要决定于问题求解精度的要求,精度越高,染色体长度越大,搜索空间也就越大,相应的要求种群大小设置的大一些。受计算机硬件的限制,这个长度也有个上限,超过此限度,算法的运行速度将难以忍受。

上述运行参数的取值只是一般情况下的常用取法,并非固定不变的。在具体情况下可适当调整,只要是提高算法效率的取值就是可行的。

3.4 遗传算法在装箱问题中的应用

遗传算法 (Genetic algorithm, 简称 GA) 作为模拟自然界生物进化原理的一种随机性全局优化概率搜索算法,具有很强的全局搜索能力,特别适合求出问题的近似最优解。由于遗传算法在组合优化问题中的操作直观,表现力强,以及在 TSP 中的成功运用经验,因而用遗传算法解决装箱问题是一可行思路^[53]。

早在 1984 年就有学者将遗传算法应用于求解装箱问题。但是对于装箱问题这类离散的组合最优化问题,往往无法直接采用简单遗传算法进行求解。一般来说,遗传算法对解的编码方案大多采用二进制编码形式,但是对于装箱这类排序问题显然是行不通的,一是物品的坐标用二进制表示编码太长,会导致搜索空间急剧扩大,算法性能降低,二是坐标值的调整变化描述了待布入物品本身位置的改变,可能会出现待布入物品交叉的现象,而处理交叉问题计算量特别。并且经典的交叉、变异操作常会产生无意义的解。因此,必须对简单的遗传算法进行适当的改进以适应装箱问题的特殊情况。此外为了提高遗传算法的收敛性和性能,还可以和其他优化算法集成使用或引入新的进化机制。一些学者对此进行了研究并取得了一定成果。

方平,李娟^[54]提出了两种求解装箱问题的遗传算法。一种是简单遗传算法,它采用等长度字符代码编码方法,使用常规的遗传操作算子。另一种是混合遗传算法,它综合运用解装箱问题的 FFD 近似算法和简单遗传算法。由这两种遗传算法所得到的装箱方案较一些近似算法所得到的装箱方案都要好。曹先彬,刘克胜等^[52]利用一种免疫遗传算法 (IGA: Immune Genetic Algorithm) 研究装箱问题的求解,该算法明显优于其他几种用于装箱问题的算法:穷举法在箱子数目大时出现的“组合爆炸”,启发式搜索法又要依赖于个人经验,并且同样不可避免地存

在“组合爆炸”，神经网络方法效果明显，但收敛速度慢，易陷入局部最优。而免疫遗传算法则是以遗传算法为基础，同时引入生物免疫机制中抗体的多样性保持策略以提高算法的群体多样性，取得了令人满意的布局效果。卜雷等^[55]从缩小搜索区域、增强算法的收敛性以及缩短计算时间的角度出发，提出将模拟退火思想引入遗传算法，应用遗传模拟退火算法求解零担货物在集装箱中的序贯装箱优化问题，并通过实际数值计算表明该算法优于传统的计算方法。何大勇^[56]等用遗传算法解决复杂集装箱装载问题并给出了有效的编码形式和解码运算。算例求解结果显示出很好的效果。

4 第四章 家电行业智能化配车系统中的装箱方法集成

家电行业是最早尝试市场化运作的行业，也是目前我国市场化程度最高的行业。从20世纪80年代中后期开始，家电企业率先摆脱了计划经济体制下的商业运作模式，大胆探索，最终涌现出了一大批优秀的国际知名的家电企业和一大批年富力强的管理人才，他们既了解中国的国情，又能灵活运用管理理论，在他们的努力和实干下，家电业成为国内一个朝气蓬勃的行业。据海关总署的统计数据，2003年前十个月中国家用电器出口已经突破一百亿美元大关，比去年同期增长百分之三十九点七。主要大家电产品均保持大幅度的出口增长，其中空调器、微波炉、电冰箱、洗衣机、电风扇、电饭锅等产品的出口增幅均超过两位数。可见家电制造业的发展是快速的。同时家电出口亦十分喜人，仅彩电一项，去年出口超过一千万台。由此可见，中国家电制造业已成为中国制造业中具有强劲实力、与国际先进技术差距较少并已参加到国际竞争前沿的重要产业。

但在辉煌的成绩下，亦存在着增长放缓、产能过剩、利润降低、竞争加剧等较深层次的问题。近年来，中国家电业感到压力骤增，一方面国内市场价格战经久不息造成企业效益普遍下滑，而商业资本还想进一步挤压上游厂商利润空间；另一方面，入世后跨国公司开始强势进入，并逐步蚕食国内企业的市场份额。如今的家电业竞争，已经不同于几年前的传统因素类似于原料资源、生产技术、市场销售，资金筹措及内部管理机制建设等普通竞争了，物流竞争开始占有重要地位。谁能更快速更有效更经济地完成物流环节，谁就在竞争中占有先机。向现代物流这个大方向迈进，进一步整合优化企业物流链条，已经成为大部分中国家电企业的共同目标。

4.1 家电行业智能化配车系统的研究背景

近年来，中国家电行业压力骤增，一方面国内市场价格战经久不息造成企业效益普遍下滑，而商业资本还想进一步挤压上游厂商利润空间，另一方面，入世后跨国公司开始强势进入，并逐步蚕食国内企业的市场份额。内外交困的局面下，家电企业意识到要想突出重围，除了加紧对自身资源的整合改造，提高核心竞争力以外别无他途。如今的家电业竞争，已经不再局限于生产成本、营销手段等传

统方式,物流竞争开始占有重要地位。谁能更快速更有效更经济地完成物流环节,谁就在竞争中占有先机。而家电行业的物流配送流程中,由于运输量大、产品种类繁多、而且要求按订单发运有时还要求考虑提货地点,用手工方式操作工作量很大,配送人员的工作效率和配送的准确率、车辆空间利用率都受到限制。导致产品运输成本居高不下,不利于提高企业竞争力。在家电行业车辆货物装载中遇到具体的问题表现在:

- 信息化程度普遍不高,手工装箱,没有先进的科学技术作依据,装箱方案带有很大的主观性,往往会造成车辆的空间利用率普遍偏低,浪费严重的现象,导致产品的运输成本居高不下,因而会增加产品成本,在竞争日益白炙化的家电行业,高成本就意味着被市场淘汰
- 仅凭经验装载,往往由于货物数量种类繁多、车辆数量规格繁多再加上装载时约束条件众多,就会导致顾此失彼,需要多次手工重装,增加了商品的损耗率,降低了配载人员工作效率
- 装箱方案不当,导致货物滞留,拉长了配送时间,致使物流运输渠道不畅通,容易贻误商机,不利于提高企业在市场上的综合竞争力,不利于行业整体物流的发展
- 业务操作流程不规范,企业整体运作效率不高,不利于企业管理水平的提高。

青岛海伊鑫产业技术有限公司是由国内专家联合投资组建的集电子信息技术产业、海洋生物技术产业、科、工、贸为一体的科技型企业。公司有着丰富的专家人才库,在科技成果产业化、技术对接、资源整合方面有着宝贵的经验,同时在电子信息技术、海洋生物技术等相关领域与国内数家知名企业进行了成功的合作,曾获国家科技部创新奖和进步奖。中国海洋大学信息工程中心是青岛市高新技术企业,是国家 863 计划 CIMS 专题工作组授权的 863 计划 CIMS 服务咨询公司。公司有丰富的企业数据资源管理系统实施经验和成功案例,在办公自动化和人工智能研究方面成果颇丰。同时,公司在家电行业的系统开发研究过程中已积累了大量的知识,并取得了可喜的成绩。

正是因为市场竞争的压力,使得家电行业物流配送在车辆配载方面提出了更高的要求。青岛海伊鑫产业技术有限公司和中国海洋大学信息工程中心合作,抓

住市场脉搏，凭借雄厚的技术实力，共同研制家电行业的智能化配车系统。

4.2 家电行业智能化配车系统的研究内容

配车问题即多约束条件下的三维装箱问题，广泛地出现在铁路货车车厢装载、汽车车厢装载、轮船配载、集装箱装载等情况中，是一个复杂的组合优化问题，从计算复杂性理论来讲，属于 NP-HARD 问题。问题的提法是：将一批待装箱体(长方体)装入长方体容器中，目标是使容器空间利用率和重量利用率达到最高。同时要考虑到约束有：(1)箱体本身的承重性(易碎性)；(2)箱体搬运的难易性；(3)一些货物必须隔离；(4)不允许超过最大承重量；(5)重心与几何形心偏差不应太大；(6)货物码放的稳定性等等。

家电行业的智能化车辆装载问题，由于其行业特色及装箱本身的约束条件，求解非常困难，单一技术难以胜任。因此系统应现实需求设计，是集遗传算法、动态规划技术、启发式算法、空间分割、布局等多种优化方法相结合而成的智能化车辆装载平台。系统基于 Windows 平台设计采用面向对象的技术实现的，有两种运行方式：一是独立运行，即作为一个独立系统，脱离其它 MIS 系统而运行；一是内嵌式运行，即嵌入到企业原有的 MIS 系统，利用原有系统的业务数据进行运算，并提供运算结果给 MIS 系统。整个系统的运行都在客户端，充分的利用客户端资源，从而减轻服务器端负担，减少网络传输，相对提高了系统的科学计算速度。系统把传统的启发式算法嵌入到遗传算法中构造了一个混合式算法。其中，遗传算法被用于个体中的全局搜索，而启发式算法在染色体中实行局部寻优，同时在货物的装载中应用到动态规划技术、空间分割、布局等优化技术，从而使系统在满足各种约束条件的基础上，高性能、高效率的最大化利用车辆装载空间。

系统基于多种优化技术，综合考虑家电行业特色、货物装载方式、承重能力、稳定性、到站顺序等多方面约束条件，辅助配车人员更为合理，更为经济，更为高效地展开工作。从而真正实现：

- 1) 能准确地根据待装货物的数量、规格、装载方式、到站顺序、套装与否、车厢的规格、数量等条件自动分析，计算车辆的装载顺序和货物的装载顺序，制定出可行的装载计划。实现最优化装车。
- 2) 能虚拟出装箱过程，步进式显示装车作业流程，并可以打印出装车清单

和装车步骤分解图，为现场装载提供决策支持。

- 3) 算法严密可靠，充分考虑到货物的形变因素、货物摆放要求和堆叠控制。
- 4) 对组合装车方案中的各车单独显示以方便理解。
- 5) 自定义货物、集装箱尺寸，解决多种货物多车厢的装载问题。
- 6) 三维立体旋转功能，自由切换观察角度，更清晰地展现货物在集装箱内的摆放情况。
- 7) 提高配载效率和配载准确率，降低产品的运输成本，提高工作效率。

建立较为科学的配车模式，规范业务流程，提高家电行业的管理水平。

4.3 系统总体设计方案

4.3.1 系统的总体架构

系统分独立式和内嵌式两种运行方式。

独立式是指系统可不需要其它系统的支持单独运行。这种运行方式下，系统独立维护货物规格属性、车厢规格属性等基础数据。对所列出的货物和车厢，使用优化算法进行计算，生成优化的配载解决方案，并可对解决方案进行打印输出。

这种方式的优点是安装方便，使用简单，不需要借助其它系统即可以有效运行，适合于数据规模小、数据离散的应用。

内嵌式是指在用户已拥有成熟的业务管理系统的情况下，配车系统嵌入到管理系统内，与管理系统一起运行。这种方式下，由管理系统提供货物规格属性、车厢属性、装载地点等基础数据，并提供需要配载的货物列表、车厢列表。配车系统使用优化算法对其进行计算，生成适合的配载解决方案。并可把解决方案返回提交给管理系统或直接打印。

这种方式的优点是集成度高，与企业业务结合严密，对大数据量的应用，运行效率高。同时，因其与业务管理系统相互结合，对整个管理系统运行效率的提高有很大的帮助。其缺点是安装复杂，需要定义多个数据接口，常常需要根据实际情况作二次开发。系统整个架构如下图：

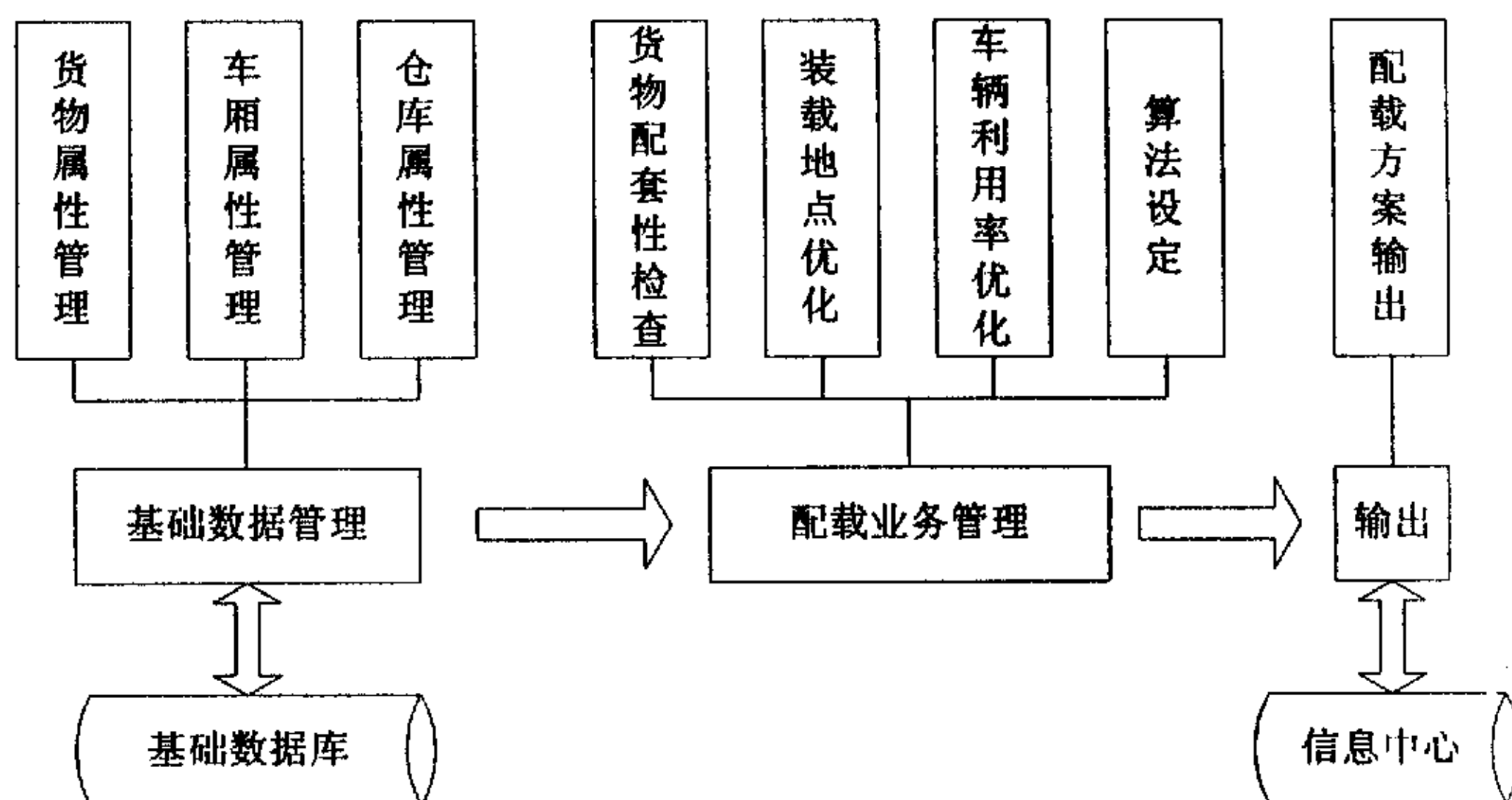


图 4-1 家电行业智能化配车系统整体结构图

4.3.2 系统数据流程

数据流程图用来描述整个系统的数据流向，从输入数据经过系统各模块加工处理之后转化成相应的输出数据。

对于内嵌式系统输入数据分别是：货物数据、车辆数据、订单数据、提货地点数据；系统加工模块是：订单优化、提货地点优化、装载效率优化、图形绘制、货物清单统计，数据接口；系统经过加工后的输出数据是：装载方案，各车装载图，货物装载清单。整个数据流从上倒下，从左到右，其图示如下：

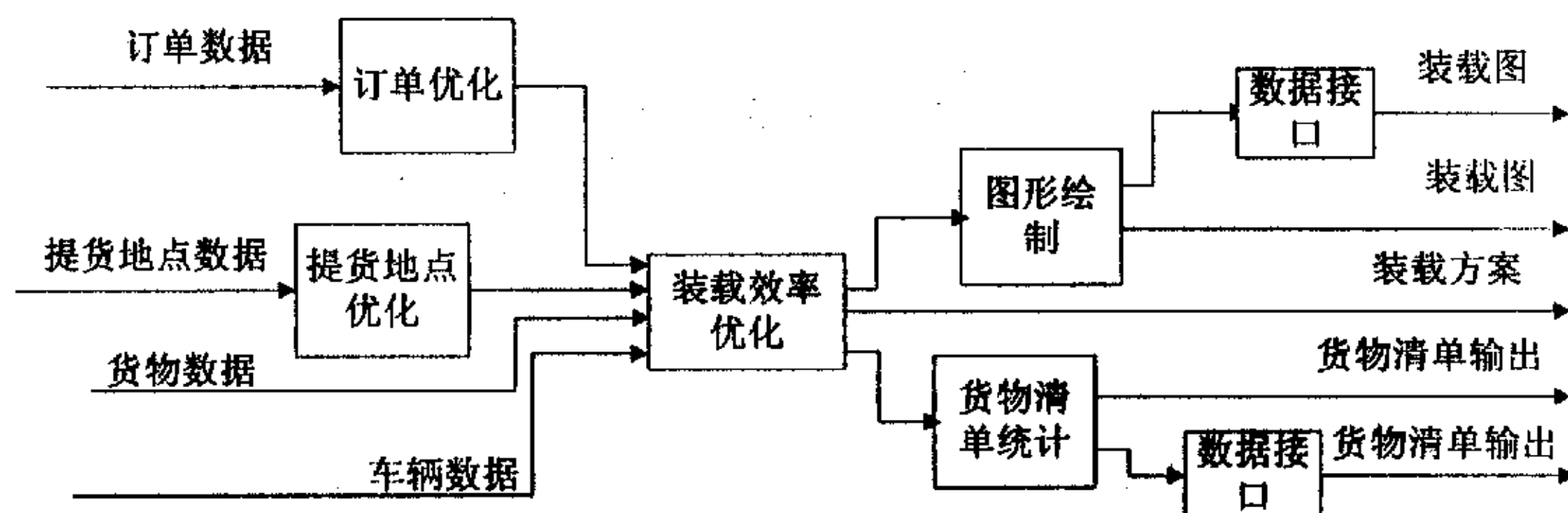


图 4-2 内嵌式系统数据流程图

对于对独立式系统输入数据分别是：货物数据、车辆数据、订单数据、提货地点数据；系统加工模块是：订单优化、提货地点优化、装载效率优化、图形绘制、货物清单统计；系统经过加工后的输出数据是：装载方案，各车装载图，货物装载清单。整个数据流从上倒下，从左到右，其图示如下：

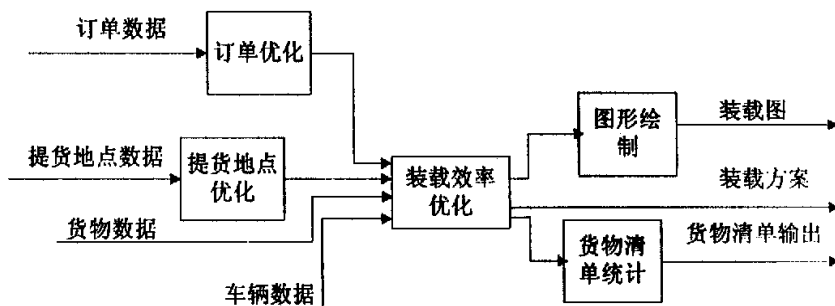


图 4-3 独立式系统数据流程图

4.3.3 系统逻辑结构与物理结构

系统使用 Delphi 语言开发而成的，根据系统的运行方式，逻辑结构可灵活地采用 2 层或 3 层模式。

内嵌式运行时，根据所嵌入业务管理系统的结构，系统可以采用 2 层模式，也可以采用 3 层模式。3 层模式时，3 层分别为：业务处理层、数据接口层、数据操纵层。其图示如下：

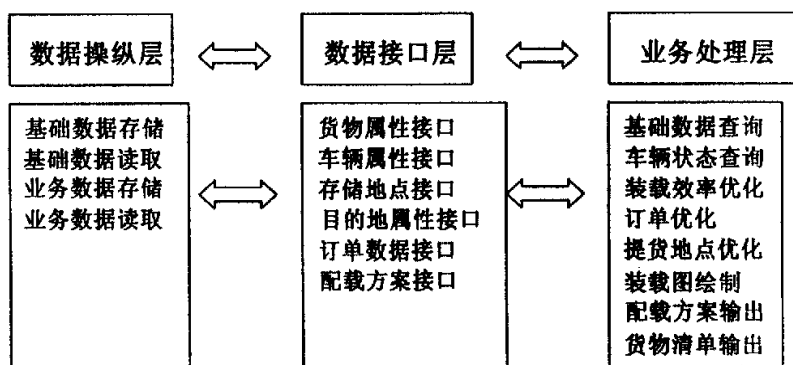


图 4-4 内嵌式系统逻辑结构图

系统物理结构为 3 层模式，3 层结构分别是：客户端，应用服务器和数据库服务器。其中客户端就是终端用户，一般的 PC 机即可；应用服务器实质上相当于主机，所有业务逻辑处理都在应用服务器上运行；数据库服务器与应用服务器可以是同一台计算机（PC 服务器），实现对数据库的操作与管理。其图示结构：

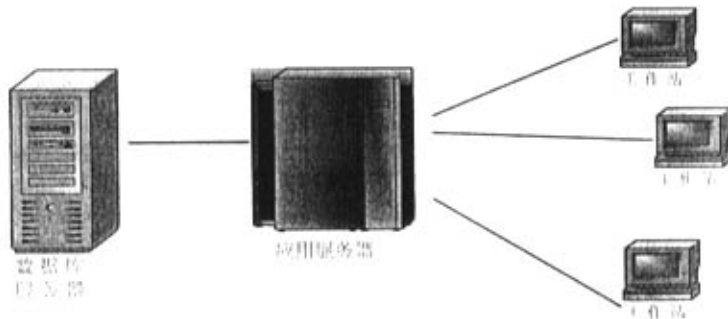


图 4-5 内嵌式系统物理结构图

独立式运行时，系统一般采用传统的 C/S 模式。客户端处理业务，服务器端为数据维护。系统可以采用 2 层模式分别为：业务处理层、数据操纵层。其图示如下：

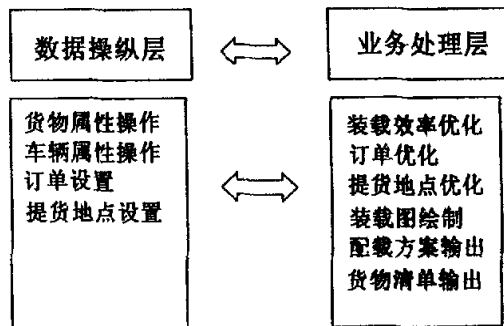


图 4-6 独立式系统逻辑结构图

4.4 核心算法的实现

4.4.1 遗传算法设计

1) 遗传算法的改进

传统的遗传算法存在对各种实际问题的搜索空间的大小变化适应能力较差，计算量大，容易出现早熟，陷入局部极优等不足现象。为此本系统在传统遗传算法的基础上作了以下的改进：

- 选择操作采用跨世代精英选择策略即上世代的种群与通过新的交叉方法产生的个体群混合起来，从中按一定概率选择较优的个体，从而保持了健

壮性和遗传的多样性。

➤ 进化过程中动态调整交叉概率 p_c 和变异概率 p_m ，其公式描述如下：

$$p_c = \begin{cases} p_{c1} & f' < f_{avg} \\ p_{c1} - (p_{c1} - p_{c2}) * (f' - f_{avg}) / (f_{max} - f_{avg}) & f' \geq f_{avg} \end{cases} \quad (4-1)$$

$$p_m = \begin{cases} p_{m1}, & f < f_{avg} \\ p_{m1} - (p_{m1} - p_{m2}) * (f_{max} - f) / (f_{max} - f_{avg}), & f \geq f_{avg} \end{cases} \quad (4-2)$$

其中： f_{avg} 表示每代群体的平均适应度值， f_{max} 表示群体中最大的适应度值， f' 表示要交叉的两个个体中较大的适应度值； f 表示要变异个体的适应度值； p_{m1} ， p_{c1} 分别表示最大变异概率和最大交叉概率， p_{m2} ， p_{c2} 分别表示最小变异概率和最小交叉概率。

2) 遗传算法的参数设定

根据多次运行的收敛情况和解的质量，设定参数如下：

变异概率： $p_{m1}=0.1$ ， $p_{m2}=0.001$ ；

交叉概率： $p_{c1}=0.9$ ， $p_{c2}=0.6$ ；

初始种群大小： $M=20$ ；

进化终止代数： $T=100$ ；

3) 遗传操作的设定

➤ 选择操作

本系统采用类似于博彩游戏中的轮盘赌的轮盘赌选择法和跨世代精英选择策略。轮盘赌选择法的基本思想是：首先计算个体的相对适应度值

$p_i = f_i / \sum f_i$ ，再根据选择概率选择个体，实现方法是生成一个随机数 $\gamma \in [0,1]$ ，若 $p_0 + p_1 + \dots + p_{i-1} < \gamma \leq p_0 + p_1 + \dots + p_i$ ，则第 i 个个体被选择到下一代。可见，个体的适应度值越大被选择到下一代的机会也越多。

跨世代精英选择策略基本思想是：上世代种群与通过交叉方法产生的新的个体混合起来，从中按一定的概率选择较优的个体。利用这一策

略,即使当交叉操作产生较劣个体偏多时,由于原种群大多数个体残留,不会引起个体的评价价值降低,该策略强调优良个体的保留。

➤ 交叉操作

对于系统这种类似于 TSP 问题的以符号编码的基因串,采用 1985 年 Goldberg 等针对 TSP 提出的部分匹配交叉操作。部分匹配交叉 (Partially Matched Crossover),这种交叉操作的主要思想是:随机选取两个交叉点,以便确定一个匹配段,根据两个父个体中两个交叉点之间的中间段给出的映射关系生成两个子个体。

例如:对下面的两个父个体的表示,随即地选择两个交叉点“|”。

P_1 : (1 2 3 | 4 5 6 7 | 8 9)

P_2 : (4 5 2 | 1 8 7 6 | 9 3)

首先,两个交叉点之间的中间段交换,得到:

O_1 : (x x x | 1 8 7 6 | x x)

O_2 : (x x x | 4 5 6 7 | x x)

其中 x 表示暂未定义码,得到中间段的映射关系,有:

$1 \leftrightarrow 4, 8 \leftrightarrow 5, 7 \leftrightarrow 6, 6 \leftrightarrow 7$

然后,对子个体 1,子个体 2 种的 x 部分,分别保留从其父个体中继承未选定码 2, 3, 9, 得到:

O_1 : (x 2 3 | 1 8 7 6 | x 9)

O_2 : (x x 2 | 4 5 6 7 | 9 3)

最后,根据中间段的映射关系,对于上面子个体 1 的第一个 x,使用最初父码 1,由 $1 \leftrightarrow 4$ 交换得到第一个 x 为 4,类似的子个体 1 的第二个 x,使用最初父码 8,由 $8 \leftrightarrow 5$ 交换得到第二个 x 为 5。如果映射关系中存在传递关系,即被选交换有多个码,则选择从前未确定的一个码作为交换。类似的进行操作,最终得到的子个体为:

O_1 : (4 2 3 | 1 8 7 6 | 5 9)

O_2 : (1 8 2 | 4 5 6 7 | 9 3)

➤ 变异操作

对于变异操作,采用逆位遗传算子,对父个体随机选择两个变异点,

两点间的基因按相反顺序重新排列。举例说明如下：

原个体：(1 2 3 4 5 6 7 8 9)

随机选择两点：(1 2 | 3 4 5 6 | 7 8 9)

倒置后的个体：(1 2 | 6 5 4 3 | 7 8 9)

4) 编码

编码就是把一个问题的可行解从其解空间转换到遗传算法所能处理的搜索空间的操作，它是 GA 应用成功与否的关键。一般来说，遗传算法对解的编码方案大多采用二进制编码形式，但是对于装箱这类排序问题显然是行不通的，一是物品的三维坐标用二进制表示编码太长，将会导致搜索空间急剧扩大，算法性能降低，二是坐标值的调整变化仅描述了待布局物本身位置的改变，可能会使待布局物之间出现交叉的现象，而处理交叉问题计算量特别大。因此系统采用 1985 年 Grefenstette 等针对 TSP 提出的基于顺序表示的遗传基因编码方式。对于本文来说就是把待装物品的编号按排放顺序排的串作为一个解的编码，即

$$P = \{p_1, p_2, \dots, p_n\}$$

其中 n 表示待装物品的个数； p_i 为整数，其值代表物品的编号。

5) 解码

能否简单、快速地将解的基因型表示转化为表现型，从而快速求出有关评价参数，是遗传算法能否有效应用的关键。对于本文来说解码就是按字符串 P 的顺序实际的装一遍箱。

解码算法描述如下：布局空间采用三叉树数据结构表示，原始布局空间是树的根节点，将第一个没有打过标记的物品应用布局规则定位于当前布局空间后部的左下角。并将该物体打上标记。这样，原布局空间的剩余空间可分为 3 个子空间，分别对应于三叉树根结点的左、中、右 3 个子结点。对这 3 个布局空间依次重复上述过程，直至没有待布局物体满足要求或集装箱没有可利用空间时停止。

6) 适应度函数设计

遗传算法在进行搜索中基本不利用外部信息，仅以适应度函数为依据，利用种群中每个个体的适应度值进行搜索。因此适应度函数的选取至关重要，将直接影响到遗传算法的收敛速度以及能否找到最优解。一般而言，适应度函数是由目标函数变换而成的。对一个解的好坏用适应度函数值的大小来评价，适应度的值越大，解的质量越好。

根据惩罚函数法和处理多目标优化的加权系数法，单个箱子的适宜度函数为：

$$f_j = k_{1j} \times spaceVol_j + k_{2j} \times weight_j + k_{3j} \times barycenter_j$$

因此遗传算法的适宜度函数为：

$$f = \sum_{j=1}^n f_j k_j$$

其中 $spaceVol_j$ 为第 j 个箱子的空间利用率函数：

$$spaceVol_j = \left(\sum_{i=1}^m BV_{ji} \right) / CV_j$$

$weight_j$ 为第 j 个箱子的重量考察函数：

$$weight_j = \begin{cases} \left(\sum_{i=1}^m BW_{ji} \right) / CW_j & \left(\sum_{i=1}^m BW_{ji} \right) \leq CW_j \\ 0 & \left(\sum_{i=1}^m BW_{ji} \right) > CW_j \end{cases}$$

$barycenter_j$ 为第 j 个箱子的重心考察函数：

$$barycenter_j = \left(CH_j \times 1.5 - \left(\sum_{i=1}^m BW_{ji} \times BWP_{ji} \right) / \sum_{i=1}^m BW_{ji} \right) / CH$$

其中 BV_{ji} 表示装在第 i 个车辆中的第 j 物品的体积， CV_i 表示第 i 个车辆的体积。 k_{1j} ， k_{2j} ， k_{3j} 为权重。

4.4.2 布局规则

1) 定序规则

用定序规则来确定布局物体放入布局空间的先后顺序，将影响最终布局

结果的质量。布局时，各布局块在布局者心目中的重要程度不同。重要的优先放入，不甚重要的后放入。其主要定序规则有：1) 按布局块底面积递减定序；2) 按布局块最长边递减定序；3) 按布局块体积递减定序。其中以 3) 最为常用，且综合考虑 1)，2)。故本文采用 3) 为定序规则

2) 定位规则

矩形物体布局定位规则主要有：

- (1) 占角策略，即布局块首先摆放在布局空间的某一角。
- (2) 顺放策略，从一个角开始，沿布局空间的某一边顺序摆放。
- (3) 先沿布局空间的四边摆放，然后填中心。
- (4) 先由上往下放并向左看齐，然后由下往上放并向右看齐。

应用领域不同，布局的定位规则也不同。经分析发现，所有的定位规则都是先占角，在不能占角的情况下占边，最后填中心，即“金角银边草肚皮”。本文采用占角优先策略作为定位规则，每次将布局块定于当前布局空间的后部左下角。

4.4.3 空间分割

系统采用三空间分割的方法，详细介绍如下：当一个货物在摆放入一个集装箱后，该集装箱被分割成三个空间（除自己占用的空间），分别为左空间，右空间，上空间。同理每个子空间在充填过程中，被摆放入货物后，同样被继续分割为三个空间，而原空间消失。为了保证装载的稳定性，装载顺序为左空间——>右空间——>上空间。分割如图所示：

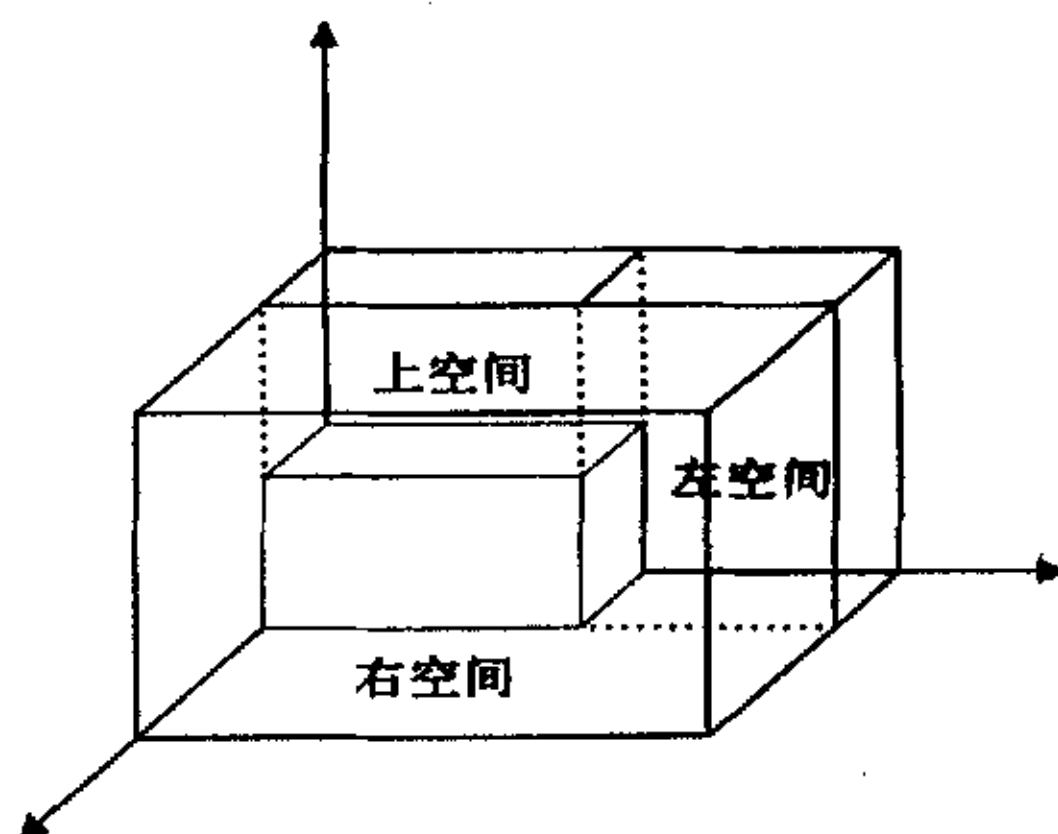


图 4-7 三维空间分割图

4.4.4 遗传算法与启发式算法集成

遗传算法与启发式算法集成是实现系统功能的基础，同时也是系统的难点所在。本系统实现两种算法动态集成应用的总体设计思想是：将算法可分为以下 3 个主要模块：(1) 遗传操作模块；(2) 启发式装载模块；(3) 群体评价模块。模块之间的通讯通过染色体数据库进行。算法的流程图示如下：

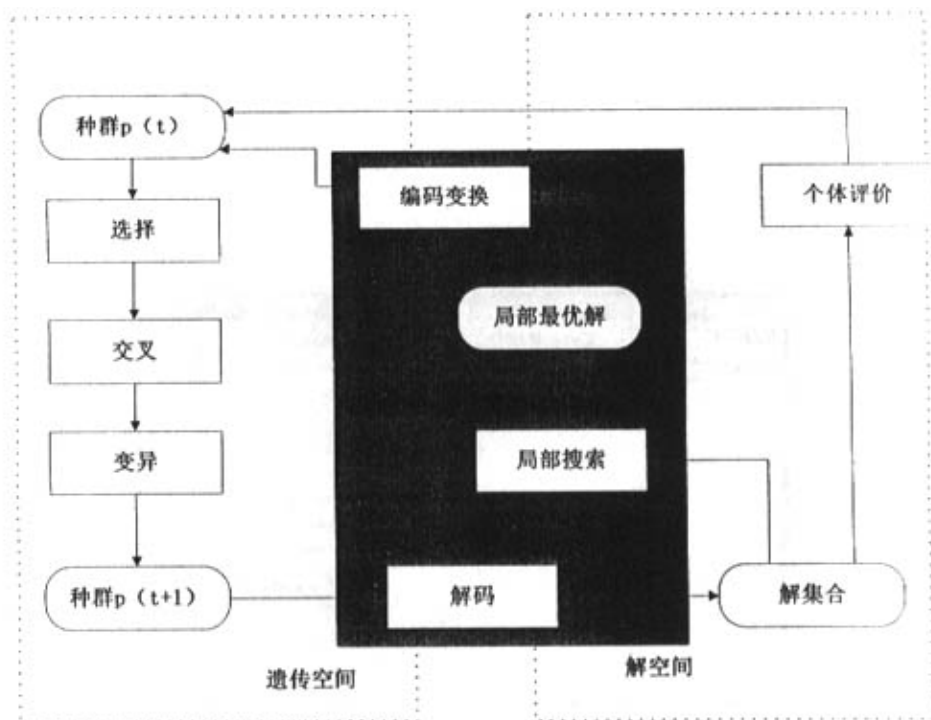


图 4-8 混合遗传算法图

4.5 系统实例演示

1. 启用该系统时，默认界面就是商品维护界面，如图 4-9 所示。本界面可以维护货物的数量、单位、长、宽、高、编码、体积、优先级和侧放规格属性。内嵌式运行方式还可以对货物的顶置层数、是否配套装载等属性进行维护。在内嵌式运行方式中，点击图 4-9 上的增加按钮，将会弹出图 4-10 所示的维护界面，可以对货物的装载进行更强的限制。



图 4-9 商品维护界面

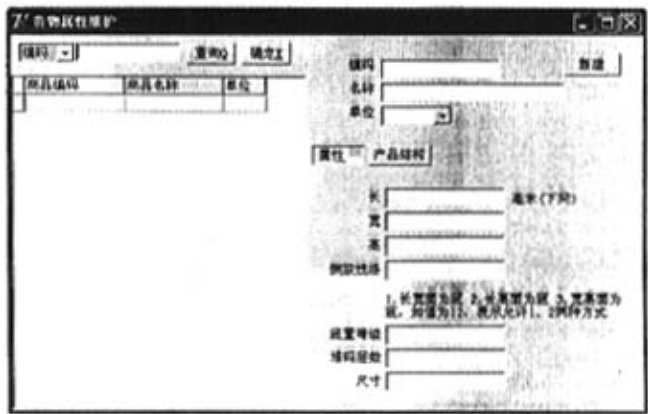


图 4-10 内嵌式系统商品维护界面

2. 点击图 4-9 上的箱体维护按钮，界面将转向如图 4-11 所示的箱体维护界面。本界面可以维护车辆的长、宽、高、编码、优先级。



图 4-11 箱体维护界面

3. 点击图 4-9 上的电子配载按钮，界面将转向如图 4-12 所示的优化计算界面。系统根据输入货物的数量、属性、装载要求、货物微调尺寸以及车辆的数量、规格、车辆微调尺寸，系统能够自动分析，自动计算出一种最佳的装配方案。并且每辆车的空间利用率会随即显示出来。系统根据计算的结果，用三维立体图形模拟车厢及货物，显示出装配后的效果。并且可以用逐步显示或从里到外逐层显示的方式，显示车辆装载的流程与效果，对多车辆同时装载时各车辆装载图单独显示，以方便理解。绘图时采用车辆货物、不同种货物不同种颜色，同种货物同种颜色的策略，增加理解的容易度。

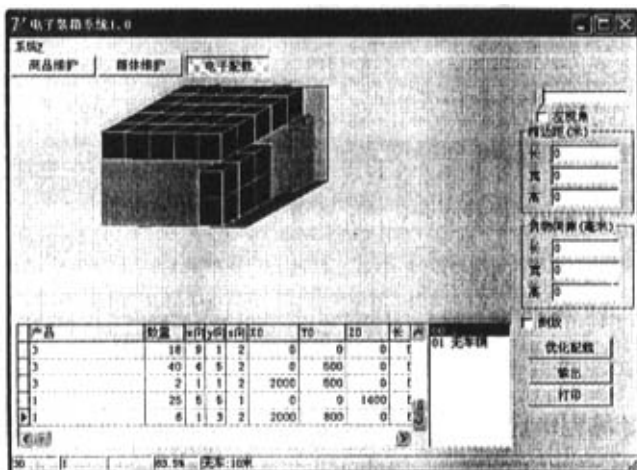


图 4-12 优化计算界面

4. 系统还可以多角度观察装载图。装车计算完后，滑动电子配载界面上的旋转滑杆，则坐标原点不动，旋转三维装载图，如图 4-13 所示；如果选中该界面上的左视角，再滑动电子配载界面上的旋转滑杆，则即可以变换视角，又旋转三维装载图，从而实现多角度观察装载图功能。

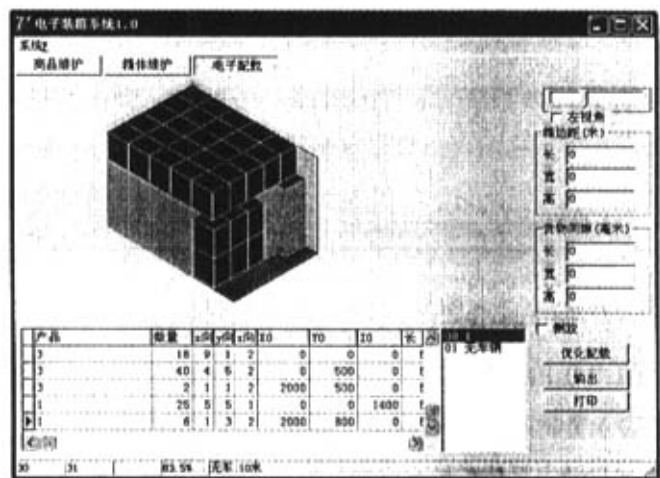


图 4-13 旋转三维装在图

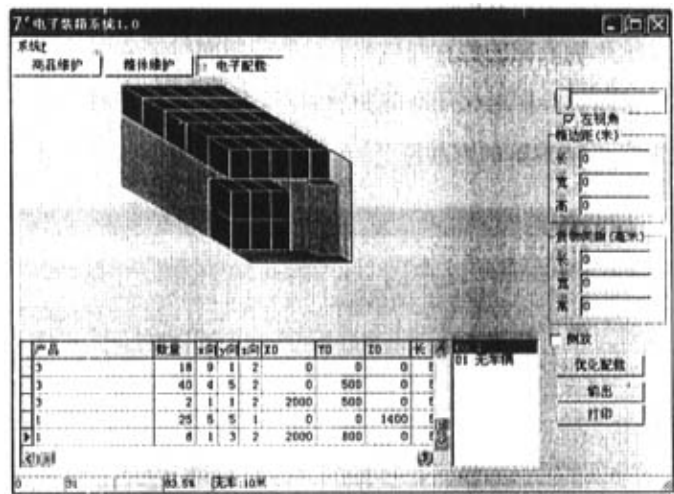


图 4-13 视角变换三维装在图

5. 装车计算完毕后，点击图 4-12 优化计算界面上的输出按钮，货物装载清单栏中的货物就会输出到 excel 表格中，内容包括：车厢号、货物号、货物数量，

如图 4-14 所示。

箱号	产品	数量
1	叉车	1
2	叉车	2
3	叉车	1
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		
15		
16		
17		
18		
19		
20		
21		
22		
23		
24		
25		
26		
27		
28		
29		
30		

图 4-14 装载清单输出到 Excel 表格

6. 装车计算完毕后, 点击图 4-12 优化计算界面上的打印按钮, 货物装载图就会被保存起来。然后可以将装载图打印出来或传回原信息系统。

4.6 系统部分运行结果及分析

1. 在货物装载过程中, 究竟什么影响容积利用率? 如何提高容积利用率? 从现场了解到: 有经验的装货员一般只能模糊地回答, 供装货物越多, 容积利用率越高。因此本文采用国际标准的 20 英尺集装箱, 尺寸为 $2.352 \times 2.388 \times 5.899$, 单位是米为布局空间, 进行大量实验。按待布局货物的类型数, 布局货物总体积分别取不同值, 计算 30 次容积利用率, 取其平均值得到表 5-1 (体积比为待装货物的总体积与集装箱的体积之比), 从大量的实验数据中可以得出以下结论:

(1) 表 5-1 中的实验数据的其平均空间利用率为 83.2%, 而目前同规格的集装箱内部空间利用率平均仅为 75%, 因而可以证明本算法的有效性。

(2) 从表 5-1 中的实验数据可以看出, 供装货物体积、供装货物类型数 (货物几何尺寸完全相同, 即定义为同类型货物) 和待装货物的体积分布对空间利用率有很大影响。只装一种货物的利用率偏低且不稳定, 甚至会出现极大的浪费, 平均利用率为 64%, 而随着待装货物种类的增多, 体积比的增大, 集装箱平均利

利用率呈现逐渐增加的趋势。但是,也不总是这样,有时会出现一些反复情况。从表中不难看出会有如下情况发生:相同体积比,类型数少的利用率反而高如斜体数据所示;相同类型数,体积比高的利用率反而低,如下划线数据所示。表中之所会出现这种反复情况是因为待装货物的体积分布极其数量对集装箱的利用率有很大影响,试验表明如果多种物品的体积都比较大而又相差不大,则利用率通常会较小。这是因为装完较大的货物后,空间分割所得到的空间可能不足以装下体积次之的货物,造成空间浪费。如果待装货物的体积相差比较大,而大的物品的种类和数量都不是很多,体积小的货物数量较多,则利用率通常较大。这是因为装完较大的货物后,小体积的货物可以填充空间分割所得到的空间有利于充分利用空间。当然如果货物种类很多,而体积都较小,数量又少的话,因空间分割过于破碎而导致利用率低下。要提高集装箱的利用率,大体应该在待装货物种类 4 种以上,体积比在 3 以上。

类型 体积比	1	2	3	4	5	7	9	>10
1	62.3	68.2	72.5	78.1	79.1	81.4	81.7	82.2
2	64.2	73.5	75.1	<i>81.8</i>	<u>82.0</u>	82.8	83.5	83.7
3	66.7	74.2	76.2	<i>81.4</i>	82.6	83.9	84.8	85.1
4	70.1	74.6	80.8	<u>83.3</u>	<u>83.1</u>	85.2	85.6	86.9
5	74.8	77.9	82.7	84.7	85.1	87.7	88.2	88.8
7	74.2	78.4	84.3	85.5	87.9	88.6	89.1	90.7
9	78.1	80.0	86.1	86.8	90.5	90.5	91.3	93.4
15	80.3	82.1	86.4	88.6	91.7	92.8	93.5	93.6
>15	81.7	83.7	88.6	89.9	91.1	93.6	93.7	93.9

表 5-1 体积比-类型实验数据

2. 分别用 2.4 节给出的构造启发式装箱算法和本章给出的混合遗传算法进行大量实验,按待布局货物的类型数、体积比和物品的总数量,计算 30 次容积利用率,取其平均值得到表 5-2。从表 5-2 的数据可以看出:混合遗传算法的空间利用率比较高且布入的物品较多,但是耗时长;启发式算法空间利用率比较低布入的物品也较少,但是耗时非常短。这是因为启发式方法较多的依赖于对问题构造和性质的认识和经验,而应用遗传算法求解问题时,在编码方案、适应度函数及遗传算子确定后,算法将利用进化过程中获得的信息自行组织搜索,不需要其他的先决条件或辅助信息。

物品的 种类	体积 比	物品 的总 数量	遗传算 法所耗 时间	启发式 算法所 耗时间	遗传算 法利用 率	启发式 算法利 用率	遗传算法 布入物品 的个数	启发式算 法布入物 品的个数
2	1.1	40	126 秒	1.34 秒	76.21%	69.8%	28	22
5	1.1	40	125 秒	1.31 秒	85.76%	75.14%	27	23
10	1.1	30	113 秒	1.09 秒	88.92%	82.65%	21	17
15	1.1	30	112 秒	1.11 秒	91.01%	85.22%	20	17
20	1.1	30	113 秒	1.12 秒	93.81%	89.35%	21	18
30	1.1	30	114 秒	1.14 秒	94.22%	91.82%	23	20

表 5-2 实验结果比较

5 第五章 结论

本文在前人的研究成果基础上,分别对装箱问题方法中的启发式算法、遗传算法的主要问题进行理论探讨和方法研究,提出一些有效的改进方法。并且协同的应用这些技术构建了一个人机一体化的家电行业智能化配车系统。通过对装箱问题方法的研究,获得了以下有益的结论:

1. 在启发式算法方面:启发式算法具有很高的解题成功率及稳定性,基于空间布局的构造启发式装箱算法,简单易行,装箱快。
2. 在遗传算法方面:进化过程中动态调整参数在保持群体多样性的同时,保证算法的收敛性。有引导的进化的遗传启发式算法能有效地解决特定领域的复杂工程优化问题。
3. 在算法集成方面:我们综合应用了遗传算法、启发式算法等多种优化方法,构建了一个货物智能装载系统。实验结果表明,该系统可以大大提高空间利用率,降低运输成本,提高企业物流竞争力。

此技术填补了国内家电行业利用计算机进行智能辅助装载的空白,打破了传统的装载模式,促进了家电行业货物装载信息化、数字化和智能化的建设发展。已经在海尔物流运行一年多,并已通过青岛市专家科技成果鉴定,定位级别为国际先进。本文论述的多种装箱问题算法集成方法与系统,也为其他行业的智能辅助装载提供了很好的借鉴,在整个运输行业及相关行业具有较为广阔的应用前景和推广空间。

由于理论水平有限和专业知识浅疏,本文初步研究了装箱问题中的常见方法,应用到实际工业问题取得一定成果。今后还需要改进用于解码的启发式算法以及提出更有效的遗传交叉、变异算子,采用新的遗传机制如免疫遗传算法,提高算法的收敛性;还应研究其他更有效的装箱方法,应用到此类问题中,进一步完善系统。

参考文献

- [1] 越民义. 组合优化导论. 浙江科学技术出版社, 2000
- [2] 张立昂译. 计算机和难解性—NP 完全性理论导论. 北京: 科学出版社, 1990
- [3] R Morabito, S Morales. A simple and effective cursive procedure for the manufacturer's pallet loading problem. *Journal of the Operational Research Society*, 1998, 49: 819-828
- [4] H Gehring, K Menschner, M Meyer. A computer-based heuristic for packing pooled shipment containers. *Europe Journal of Operational Research*. 1990, 44: 277-288
- [5] Johnson, D. S. Near-Optimal bin packing algorithms. Technical Report, MAC TR-109 (1973), Project MAC, MIT, Cambridge, Mass
- [6] Fuh-Hwa, Fliu, C-Jhsiao. A three-dimensional pallet loading method for single-size boxes. *Journal of the Operational Re-search Society*, 1997, 48: 726-735
- [7] Kenndy M P, Chua L O. Unifying the tank and Hopfield linear programming circuit and the canonical nonlinear programming circuit of chua and lin. *IEEE Transactions on Circuits and Systems*, 1987, CAS-34 (2): 210-214
- [8] Kantorovich LV. Mathematical methods of organizing and planning production[J]. *Management Science*, 1960, 6:366-422
- [9] Rudolph G., Convergence analysis of canonical genetic algorithms. [J]*IEEE Trans. On Neural networks*, 1994, 5(1): 96-101
- [10] Dowslang K A. An exact algorithm for the pallet loading problem. *European Journal of Operational Research*, 1987, 31:78-84
- [11] Ann B Ngoi Kok, Chua C K. A knowledge-based system for strip layout design. *Computers in Industry*, 1994, 25:31-44
- [12] Csirik J, Johnson D S. Bounded Space On-line Bin Packing: Best is Better than First. *The Proceedings of 2nd Annual ACM-SIAM Symposium on Discrete Algorithms*. 1991, 309-319
- [13] Cheng K.H. and Li K. heuristic algorithms for on-line packing in three dimensions. *Journal of Algorithms*, 1992, 13: 589-605
- [14] Cheng K.H. and Li K. On three-dimensional packing. *SIAM Journal on Computing*. 1990, 19: 847-867
- [15] Grove E. Online bin packing with lookahead. In: *Proceedings of the Annual ACM SIAM Symposium on Discrete Algorithms*. ACM, San Francisco, CA, 1994, Jan.: 430

- [16] Change E., Kankanhalli M. S and Wang W. Multidimensional on-line bin-packing: An algorithm and its average-case analysis. Institute of Systems Science. Information Processing Letters, 1993, 48: 121-125
- [17] Bengtsson B. Packing rectangular pieces-a heuristic approach. The Computer Journal, 1982, 25: 353-357
- [18] Biles W. E. and Ikonen I. A genetic algorithm with 3-dimensional chromosome for packing non-convex parts for selective laser sintering. In: Proceedings of the Nordic Workshop on Genetic Algorithms and their Applications, NWGA, 1997, 83-96
- [19] Biles W. E., Ikonen I. T., Kumar A., Lewis J. E. and Ragade R. K. Using distributed genetic algorithm in three-dimensional bin-packing for rapid prototyping machines. Proceedings of SPIE, Boston, 1998, MA: 45-53
- [20] Fleisher M. Simulated annealing: Past, present, and future. In: proceedings of the 1995 Winter Simulation Conference, Arlington, VA, 1995, Dec.:155-161
- [21] Garey M. R., Johnson D. S. Computer and Intractability: A Guide to the Theory of NP-Completeness [M]. New York: Freeman, 1979
- [22] Hansen P. Tabu search for multiobjective optimization: MOTS[R]. Denmark: Technical University of Denmark, 1997
- [23] 戴佐, 查建中. 三维实体布局的八叉树语言及优化算法[J]. 计算机辅助设计与图形学学报, 1996, 8(1):61~67
- [24] Johnson D. S., Near-Optional bin packing algorithms. Technical Report, MAC TR-109(1973), Project MAC, MIT, Cambridge, Mass
- [25] 李荣琦, 越民义. $FFD(L) \leq \frac{11}{9}OPT(L) + \frac{7}{9}$. 科学通报, 1997(6): 1137-1140
- [26] Coffman E. G. Jr. Garey M. R., Johnson D. S. Approximation Algorithms for Bin Packing A Updated Survey, in Algorithm Design for Computer System Design, G. Ausiello, M. Lucertini and P. Sealfini(eds.)Spring, Verlag, N. Y. 1984, 49-106
- [27] Keqin Li, kam Hoi Cheng. A heuristic algorithms for On Line Packing in Three Dimensions. Journal of Algorithms, 1992, (13) : 819~828
- [28] J. Csirik and A. Van Vliet. An on-line algorithm for multidimensional bin packing. Operation Research letter, 1993, 13: 149-158
- [29] 段国林, 查建中, 徐安平, 张满囤. 启发式算法及其在工程中的应用. 机械设计, 2000, 6: No. 6
- [30] 吴一帆, 李思昆. 基于神经网络的自动布局算法. 计算机工程与科学, 2000, 22(2):77-86
- [31] 吴一帆, 赵平, 李思昆. 基于神经网络的并行布局算法. 中南工业大学学报, 2000, 31

- (3):275-278
- [32] 滕弘飞, 高轩, 郁志诚. 中国象棋棋谱在装填布局中的工程应用. 大连理工大学学报, 1995, 35(4), 497-492
- [33] 赵素芬, 几何布局问题及其应用——装箱问题研究进展, 呼和浩特科技, 1993(4):22-24
- [34] Yang Chuan-min, Chen Shao-wei. Discontinuous optimization on the containerization of cubic. Package Engineering, 1996, 17(2): 6-9
- [35] Mannchen K. Solution methods for two and three dimensional packing problems [Ph. D. Thesis] Kar Lsruhe, Germany: Kar Lsruhe University, 1989
- [36] Gehring H. A computer-based heuristic for packing pooled shipment containers. European Journal of Operational Research, 1990, 44(2):277-288
- [37] Eberhard Bischof, Michael D Marriott. A comparative evaluation of heuristics for container loading. European Journal of Operational Research, 1990, 44:267-276
- [38] 王金敏, 陈东祥, 查建中. 关于约束底盘装载问题的一种启发式方法. 软件学报, 1996, 7(10): 616-620
- [39] 何大勇, 鄂明成, 查建中, 王春喜, 姜义东. 基于空间分解的集装箱布局启发式算法及布局空间利用率规律. 计算机辅助设计与图形学学报, 2000, 5:376-370
- [40] 沈灏. 底为正方形的三维装箱问题. 浙江大学学报(理学版) 1997, 7:44-51
- [41] D E Goldberg. Genetic algorithm in search, optimization and machine learning. MA Addison Wesley, 1989(21)
- [42] 席裕庚, 柴天佑, 恽为民. 遗传算法综述. 控制理论与应用, 1996, 13(6): 696-708
(Caponetto R. et al. A Survey Trans Inst. MC, 1993, 15(3):143~156)
- [43] [日]玄光男、程润传, 遗传算法与工程设计, 北京:科学出版社, 2000
- [44] 周明、孙树栋, 遗传算法原理及应用, 北京:国防工业出版社, 1999
- [45] Holland J. H. Adaptation Natural and Artificial Systems. Ann Arbor: University of Michigan Press, 1975
- [46] Khoo L. P., Lee S. G., Yin X. F. Prototype genetic algorithm-enhanced multi-objective scheduler for manufacturing systems. International Journal of Advanced Manufacturing Technology, 2000, 16(2): 131-138
- [47] 吴庆洪, 张纪会, 徐心和. 具有变异特性的蚁群算法. 计算机研究与发展, 1996, 36(10): 1240-1245
- [48] 胡国四, 韩生廉. 遗传算法适值函数定义方法的研究. 控制与决策, 1999, 14(6): 694-697
- [49] Mak K. L., Wong Y. S., Wang X. X. Adaptive genetic algorithm for manufacturing

- cell formation. International Journal of Advanced Manufacturing Technology, 2000, 16(7): 491-497
- [50] Enrique Alba, Jose M Troya. Influence of the Migration Policy in Parallel Distributed Gas with Structured and Panmictic Populations. Applied Intelligence, 2000, (12): 163-181
- [51] 吴浩扬. 基于模拟退火机制的多种群并行遗传算法. 软件学报, 2000, 11(3): 416-420
- [52] 曹先彬, 刘克胜, 王煦法. 基于免疫遗传算法的装箱问题求解. 2000 21 (4): 361-363.
- [53] Rudolph G., Convergence analysis of canonical genetic algorithms. [J] IEEE Trans. On Neural networks, 1994, 5(1): 96-101
- [54] 方平, 李娟. 求解装箱问题的遗传算法. 南昌航空工业学院学报, 1998, 2: 21-24
- [55] 卜雷, 尹传忠, 蒲云. 零担货物序贯装箱优化问题的遗传模拟退火算法 西南交通大学学报. 2002, 10: 531-535
- [56] 何人勇, 查建中, 姜义东. 遗传算法求解复杂集装箱装载问题方法研究. 软件学报 2001, 12(9): 1380-1386

致 谢

本文的研究工作是在丁香乾教授精心指导和亲切关怀下完成的。在攻读硕士学位三年学习期间，作者深深受益于导师的关心、爱护和诲人不倦的教导。丁香乾教授渊博的专业知识、严谨的治学态度以及勤奋的敬业精神给作者留下了深刻的印象，这将是激励作者在今后的人生旅途上不断进取，勇于探索的宝贵精神财富。在本文完成之际，谨向为本人倾注了大量心血的导师表示最诚挚的感谢和深深的敬意！

在课题的研究过程中，海大新星计算机工程中心的各位老师及工程师在论文写作过程中提出了不少很好的建议，特别是魏旭工程师，在此表示诚恳的感谢！

感谢家人对作者的支持与理解，特别要感谢父母为我的求学生涯所作出的无私奉献。感谢所有帮助和支持过的各位老师、同学、朋友。

作者攻读硕士期间完成论文

1. 韩运实、张晓丽、魏旭、丁香乾, 海尔物流电子化配车, 中国储运, 2003, 第四期, P44
2. 韩运实、丁香乾、石硕、谢康、魏旭, 仿真电子化配车系统, 科技信息: 科技在线, 2003, 第五期, P47
3. 丁香乾、韩运实、张晓丽, 多约束条件下的一种启发式集装箱装箱算法, 第十三届中国神经网络学术会议论文集, 2003 年 11 月, P453
4. 丁香乾、韩运实、张晓丽, 自适应遗传算法解决集装箱装载问题的方法探讨, 已通过中国海洋大学学报审核, 将在近期正式发表。
5. 张晓丽、韩运实等, 双星集团: 基于 J2EE 技术的分销电子商务平台, 中国储运, 2002, 第五期, P40
6. 张晓丽、韩运实、刑贞信、丁香乾, 条码技术与现代网络库存, 青岛大学学报自然科学版增刊, 2003, P46

作者攻读硕士期间开发系统

1. 以主要身份参加设计、开发家电行业智能化配车系统。该系统已通过青岛市科技成果鉴定, 鉴定级别为: 国际先进。目前正在申请青岛市科技进步奖。
2. 2002 年参与潍坊柴油机厂售后服务系统的研发。
3. 2002 年进行教学实践, 教授夜大学生的 pascal 程序设计
4. 2003 年开始参与海尔住宅设施系统三期的研发。
5. 2003 年下半年参与 863 课题: 服装纺织业 ASP 平台开发及应用。主要负责手机短信上报接收系统的研发。