分类号：0246 密　级：公 开
U D C ：____ 单位代码：10424

学 位 论 文

基于遗传算法的设施选址问题算法研究

李岳佳

申请学位级别：硕士学位　专业名称：运筹学与控制论
指导教师姓名：赵 茂 先　职　称：教　授

山 东 科 技 大 学

二零一二年四月

论文题目:

基于遗传算法的设施选址问题算法研究

作者姓名: 李岳佳　　入学时间: 2009 年 9 月

专业名称: 运筹学与控制论　　研究方向: 运筹与管理科学

指导教师: 赵茂先　　职　　称: 教　　授

论文提交日期: 2012 年 4 月

论文答辩日期: 2012 年 6 月

授予学位日期:　　年　月

STUDY ON THE ALGORITHMS FOR THE FACILITY LOCATION PROBLEM BASED ON GENETIC ALGORITHM

A Dissertation submitted in fulfillment of the requirements of the degree of

MASTER OF SCIENCE

from

Shandong University of Science and Technology

by

LiYuejia

Supervisor：Professor Zhao Maoxian

College of Information Science and Engineering

May 2012

声　　明

本人呈交给山东科技大学的这篇硕士学位论文，除了所列参考文献和世所公认的文献外，全部是本人在导师指导下的研究成果。该论文资料尚没有呈交于其它任何学术机关作鉴定。

硕士生签名：

日　　期：

AFFIRMATION

I declare that this dissertation， submitted in fulfillment of the requirements for the award of Master of Science in Shandong University of Science and Technology， is wholly my own work unless referenced of acknowledge. The document has not been submitted for qualification at any other academic institute.

Signature：
Date：

摘 要

本文讨论的主要内容是设施选址问题。论文首先简单介绍了设施选址问题在现代社会生活中的作用和发展进程，然后介绍了一些经典的设施选址问题和求解算法。常见的设施选址问题包括 *p*-中位问题、*p*-中心问题、经典的无容量限制设施选址问题等。多数设施选址问题属于 NP-Hard 问题，目前已有很多方法来求解这类问题，主要分为精确算法与启发式算法。常见的精确算法有分支定界法、拉格朗日松弛算法等；启发式算法主要有遗传算法、粒子群优化算法、禁忌搜索算法等。随后给出了本文的研究意义和主要工作。

论文主要研究了两类设施选址问题，一类是经典的无容量限制设施选址问题（Uncapacitated Facility Location Problem，简称 UFLP）；另一类为无容量限制的可靠性设施选址问题（Reliability for the Uncapacitated Facility Location Problem，简称 RUFLP）。针对上述两类设施选址问题，本文分别给出了基于遗传算法的求解方法，因此论文第二部分对传统的遗传算法进行了介绍。

针对 UFLP，提出了一种改进的遗传算法，该算法较之传统的遗传算法有如下改进：（1）提出了新的自适应交叉概率，并与均匀交叉相结合，使交叉算子不仅更有针对性，而且具有自适应性，从而提高进化效率；（2）在已有的自适应变异概率设计方法下进行改进，设计出一种更合理的自适应变异算子。最后通过数值试验说明本文提出的遗传算法在求解大规模 UFLP 时具有很好的效果。

随着设施选址问题的不断发展，新的选址模型也不断涌现，尤其是可靠性设施选址问题的提出具有非常重要的意义。国外对可靠性设施选址问题的研究模型大多基于 *p*-中位问题而提出的，其假设条件比较严格，模型比较复杂。本文提出的无容量限制的可靠性设施选址模型放宽了假设条件，降低了问题的复杂度。针对 RUFLP，提出了基于遗传算法的分阶段近似算法，并证明了该算法对求解 RUFLP 是可行的。

最后，对论文进行了总结，并对设施选址问题以后的研究进行了展望。

关键词：设施选址问题；无容量限制设施选址问题；启发式算法；遗传算法；自适应；可靠性；近似算法

Abstract

The main content discussed in this paper is the facility location problem. In this thesis, we first briefly introduce the role of the facility location problem in the modern society and its development process, and then introduce some classical facility location problems and solving algorithms. Common facility location problems include *p-median* problem, *p-center* problem, the classic uncapacitated facility location problem, and so on. The majority of the facility location problems are NP-Hard problems. There are many methods to solve this kind of problem, which can be mainly divided into exact algorithms and heuristic algorithms. Common exact algorithms include branch and bound method, Lagrange relaxation algorithm, etc. Heuristic algorithms mainly include genetic algorithm, particle swarm optimization algorithm, tabu search algorithm, and so on. Subsequent we presented the significance study in this thesis and the main work.

This thesis mainly studied two kinds of facility location problems. One kind of them is classic Uncapacitated Facility Location Problem (UFLP). Another kind is Reliability for the Uncapacitated Facility Location Problem (RUFLP). In view of the above two types of facility location problem, this paper presented solving method based on genetic algorithm respectively, so the second part of the thesis introduced traditional genetic algorithm.

For the UFLP, we proposed an improved genetic algorithm. Compared to the traditional genetic algorithm the algorithm has the following differences: The first, proposed a new adaptive crossover probability, and combined it with uniform crossover, thus make the crossover operator is not only more targeted but also adaptive, so as to improved the efficiency of evolution. The second, improved the existing adaptive mutation probability, and developed a more reasonable adaptive mutation operator. Finally, through the numerical test proof that the genetic algorithm is put forward in this paper have a very good effect in solving large-scale UFLP.

Along with the development of the facility location problem, the new location models also emerging, especially the come up with the reliability of the facility location problem has a very important significance. The study of reliability facility location model in foreign most based on the *p-median* problem, but the assumptions are strict and the models are complex. The Reliability for the Uncapacitated Facility Location model put forward in this paper has relaxed the assumptions, and reduced the complexity of the problem. Aiming at the RUFLP, we put forward an approximation algorithm of staging thought based on genetic algorithm for the RUFLP, and then we proved that the algorithm is feasible for solving the RUFLP.

At last, we concluded the full text, and Looking to the future study on the facility location problem.

KeyWords: Facility location problem; Uncapacitated Facility Location Problem; Heuristic algorithm; Genetic algorithm; Adaptive; Reliability; Approximate algorithm

目 录

Contents

1 绪 论

1.1 引 言

选址问题是运筹学中的经典问题之一，在生产、生活、物流、甚至军事中都有着非常广泛的应用，其研究内容涵盖了城市、产业带、经济技术开发区、机场、水利设施、人类居住区、销售网点以及配送中心等方面的区位决策。设施选址是众多选址问题的一个重要研究领域，所研究的设施主要指与生产、商业流通及人类生活有关的用地规模相对较小的具体网点、场所，如工厂、仓库、消防站、变电站、污水处理中心，加油(气)站等。设施选址问题的研究方法主要依靠运筹学、拓扑学、管理学等，涉及经济、政治、社会、管理、心理及工程地质等多门学科，这是设施选址与其他选址问题的重要区别。

随着经济、社会的发展，城市规划与设施建设越来越受到人们的重视。设施作为个体与社会的桥梁，扮演着非常重要的角色。从个体角度来看，它影响到人们是否能够享受服务以及服务质量的好坏；从社会角度来说，它对城市产业布局、经济发展、环境保护等方面都有非常重要的战略意义。在理论上，设施选址问题研究的是在规划区域内如何选定一个或多个设施的地理位置，使得既定目标最优。设施选址问题是一个古老而又现代的问题，其广泛存在于生产、生活、物流、军事中，如仓库、超市、快餐店、加油站、物流中心、医院、电厂、消防站、垃圾处理中心、核电站等选址。在实际应用中，选址决策需要选址理论合理解释其决策行为，为其奠定坚实的理论基础。科学合理的设施选址可以有效地节约资源、降低成本、确保提供优质的服务、优化网络结构和空间布局、提高经济效益和社会效益。

设施选址问题是一个古老的问题，古代的选址决策往往以经验、制度甚至迷信思想为依据，缺乏科学性。1909 年，德国学者 Weber[1]研究了怎样在平面上确定一个仓库的位置，使得该仓库与多个顾客之间的距离和最小的问题，并将该问题称为 *Weber* 问题，由此正式开始了选址理论的研究，标志着设施选址问题进入到科学研究的时代。设施选址问题在其发展的百年历程中，各时期研究侧重点各有不同。按时间可分为三个阶段：（1）零散研究阶段（1909 年-20 世纪 60 年代），该阶段侧重研究生产生活中各种实际问题，内容零散。最早研究设施选址问题的是 Weber（1909 年）；随后 Hotelling（1929 年）研究了两个竞争供应商在一条直线上的区位选择问题，并建立了选址模型；Smithies（1941

年）和 Stevens（1961 年）对 Hotelling 提出的问题进行了更深入的研究；经济学家 Isard（1956 年）从土地利用、投入产出角度对工业区位的选择进行了分析；20 世纪 50 年代，越来越多的研究者侧重于设施选址的实际应用研究，包括产品销售网点的分布于设计、消防设施选址、垃圾处理中心选址、电话网络程控交换设备选址、铁路货运编组站选址等。（2）系统研究阶段（1964 年-20 世纪 80 年代）。Hakimi 于 1964 年发表的关于网络多设施选址的论文，标志着设施选址问题系统研究阶段的开始，随后，选址问题被引入到更广泛的领域，包括生产中心选址、变电站选址、交通枢纽选址等，采用的研究方法主要是运筹学、拓扑学。（3）不确定性选址问题研究阶段（20 世纪 80 年代至今）。进入 20 世纪 80 年代，随着市场变化的加剧，实际生产、生活中运输时间、需求量、需求空间分布、设施建造费用等变量的不确定性加强，以前静态、确定性选址模型与算法已经不能适应选址问题研究的发展，因此不确定性选址问题成为研究的热点与焦点，其中以随机选址问题为代表。Church（1983 年）、Mirchandani（1985 年）、Louveaux（1986 年）等在研究不确定性中值问题时均假定运输时间与需求量为随机变量，Berman 与 Odoni（1982 年）、Berman 与 Leblanc（1984 年）将运输费用或时间假定为不确定系统变量来研究随机交通网络问题。

1.2 设施选址问题的分类

设施选址问题的类型多种多样，按决策变量的特征，可将其分为两类：连续设施选址问题与离散设施选址问题。连续设施选址问题的决策变量在某一连续空间内取值，如 *Weber* 问题[1]；离散设施选址问题的决策变量是离散的，只能在有限的点集合中取值，本文主要介绍几类经典的离散选址问题。离散设施选址问题的模型[2]一般为整数 0-1 规划、混合整数规划，主要包括 p-中位问题、p-中心问题、无容量限制的设施选址问题（Ucapacitated Facility Location Problem，简称 UFLP）、有容量限制的设施选址问题（Cpacitated Facility Location Problem，简称 CFLP）等。

在下面各种设施选址问题中，用 I 表示备选设施组成的集合，J 表示需求点组成的集合，变量 x_{ij} 和 $y_i(i \in I,\ j \in J)$ 的定义如下：

$$y_i = \begin{cases} 1, & \text{开放设施}i \\ 0, & \text{否则} \end{cases}, \qquad x_{ij} = \begin{cases} 1, & \text{由设施}i\text{来满足需求点}j \\ 0, & \text{否则} \end{cases}。$$

1.2.1 *p*-中位问题

p-中位问题最早由 Hakimi[3,4]提出，问题假设所有的备选设施已经建造完成，并且每个设施都没有容量限制，如何在备选设施集合 I 中选择 p 个设施作为开放的中心设施，使得所有需求点与离其最近的开放设施的加权距离和最小。令 h_j 表示需求点 j 的需求量，d_{ij} 表示需求点 j 到设施 i 的距离，模型如下：

$$\min \quad \sum_{i\in I}\sum_{j\in J} h_j d_{ij} x_{ij} \tag{2.1.1}$$

$$s.t. \quad \sum_{i\in I} x_{ij} = 1, \qquad \forall j \in J \tag{2.1.2}$$

$$x_{ij} \le y_i, \qquad \forall i \in I,\ j \in J \tag{2.1.3}$$

$$\sum_{i\in I} y_i = p \tag{2.1.4}$$

$$x_{ij}, y_i \in \{0,1\}, \qquad \forall i \in I, j \in J \tag{2.1.5}$$

目标函数(2.1.1)是求各个需求点到离它们最近的设施的加权距离和的最小值;等式约束(2.1.2)保证每个需求点都能得到满足且仅由一个中心设施来满足；不等式约束(2.1.3)表示只有开放的中心设施才能服务于需求点；等式约束(2.1.4)限制开放的中心设施个数为 p 个；(2.1.5)表明模型的变量为 0-1 变量。

1.2.2 *p*-中心问题

p-中心问题[3,4]是探讨如何在网络中选择 p 个服务站，使得任意一个需求点到距离该需求点最近的服务站的最大距离最小的问题。其模型如下：

$$\min \quad r \tag{2.1.6}$$

$$s.t. \quad \sum_{i\in I} h_j d_{ij} x_{ij} \le r, \qquad \forall j \in J \tag{2.1.7}$$

$$\sum_{i\in I} x_{ij} = 1, \qquad \forall j \in J \tag{2.1.8}$$

$$x_{ij} \le y_i, \qquad \forall i \in I,\ j \in J \tag{2.1.9}$$

$$\sum_{i\in I} y_i = p \tag{2.1.10}$$

$$x_{ij}, y_i \in \{0,1\}, \qquad \forall i \in I, j \in J \tag{2.1.11}$$

目标函数(2.1.6)是求最大距离最小；不等式约束(2.1.7)保证r是最大距离；等式约束(2.1.8)保证每个需求点都能得到满足且仅由一个中心设施来满足；不等式约束(2.1.9)表示只有开放的中心设施才能服务于需求点；等式约束(2.1.10)限制开放的中心设施个数为p个；(2.1.11)表明模型的变量为 0-1 变量。

1.2.3 UFLP

如果在p-中位问题中，假设设施未修建好，则需要考虑设施的建造费用，且不限制开放设施的个数，那么该问题就是 UFLP。设每个需求点j的需求量为h_j，从设施i到需求点j的单位运输费用为c_{ij}，每个设施i的修建费用为f_i，并且每个设施可以满足的需求点个数没有限制，即没有容量限制，定义x_{ij}、y_i分别为分配决策变量、选择决策变量，则 UFLP 的模型如下：

$$\min \quad \sum_{i=1}^{m}\sum_{j=1}^{n} h_j c_{ij} x_{ij} + \sum_{i=1}^{m} f_i y_i \tag{2.1.12}$$

$$s.t. \quad \sum_{i=1}^{m} x_{ij} = 1, \qquad \forall j \in J \tag{2.1.13}$$

$$x_{ij} \le y_i, \qquad \forall i \in I, j \in J \tag{2.1.14}$$

$$x_{ij}, y_i, \in \{0,1\}, \qquad \forall i \in I, j \in J \tag{2.1.15}$$

目标函数(2.1.12)是求设施的建造费用与各个需求点到离它们最近的设施的运输费用之和的最小值；等式约束(2.1.13)保证每个需求点都能得到满足且仅由一个设施来满足；不等式约束(2.1.14)表示需求点只能由选择建造的设施服务；(2.1.15)表明模型的变量为 0-1 变量。

1.2.4 CFLP

UFLP 假设每个设施没有容量限制，当考虑每个设施的容量限制时，就是 CFLP[5]。假设每个设施i的容量限制为s_i，则 CFLP 的模型如下：

$$\min \quad \sum_{i=1}^{m}\sum_{j=1}^{n} h_j c_{ij} x_{ij} + \sum_{i=1}^{m} f_i y_i \tag{2.1.16}$$

$$s.t. \quad \sum_{i=1}^{m} x_{ij} = 1, \qquad \forall j \in J \tag{2.1.17}$$

$$\sum_{j=1}^{n} h_j x_{ij} \le s_i y_i, \qquad \forall i \in I \tag{2.1.18}$$

$$x_{ij} \le y_i, \qquad \forall i \in I, j \in J \tag{2.1.19}$$

$$x_{ij}, y_i, \in \{0,1\}, \qquad \forall i \in I, j \in J \tag{2.1.20}$$

目标函数(2.1.16)是求设施的建造费用与各个需求点到离它们最近的设施的运输费用之和的最小值；等式约束(2.1.17) 保证每个需求点都能得到满足且仅由一个设施来满足；不等式约束(2.1.18)保证每个设施服务需求点的总量不超过其最大容量限制；不等式约束(2.1.19)表示需求点只能由选择建造的设施服务；等式约束(2.1.20)表明模型的变量为 0-1 变量。

目前国内外研究的离散设施选址问题除上述几类外，还有覆盖问题[6,7]、动态选址问题[8,9,10]、竞争选址问题[11]、多产品选址和多级选址问题[12]等。

1.3 设施选址问题的求解算法

求解设施选址问题的算法可以分为两大类，一类是精确算法，常用的有分支定界算法、拉格朗日松弛法等；另一类是启发式算法，常用的有遗传算法、粒子群优化算法、禁忌搜索法、模拟退火算法等。

1.3.1 分支定界法

分支定界法是 Lawler[13]等在 20 世纪 60 年代提出的，该算法的思想不仅适用于整数线性规划及混合整数线性规划，也适用于组合优化问题。分枝定界法在分析一个组合优化问题的所有可行解的过程中，采用了必要的限制条件，目的是排除可行域中大量非最优解区域，从而使算法能有效求解目标问题。

1.3.2 拉格朗日松弛算法

拉格朗日松弛算法[14]的基本原理是将复杂的、造成问题难以求解的约束吸收到目标函数中，使原问题变得较易求解。一些组合优化问题在原有的约束条件下很难得到求解，但减少某些约束条件后，原问题就变得较容易求解。正是因为此类问题的存在及拉格朗日松弛算法的特点，使得很多学者热衷于研究用朗格朗日松弛技术解决组合优化问题。

1.3.3 遗传算法

遗传算法（Genetic Algorithm，简称 GA）起源于对生物系统的计算机模型研究，其基本思想是模拟生物进化循环，即初始群体经过竞争，优胜劣汰得到一个新的种群，种群里的个体进行婚配产生子群，子群在经过变异后得到新的群体，由此进入循环。最早意识到自然遗传算法可以转化人工智能算法的是 Holland，1962 年 Holland 提出了利用群体进化模拟适应性系统的思想，引进群体、适应值、选择、交叉、变异等基本概念；1967 年，Holland 的学生 Bagley[15]在其博士论文中首次提出“遗传算法”一词；1975 年，John Holland 出版了专著《Adaptation in Natural and Artificial Systems》，提出了对遗传算法的理论发展极为重要的模式理论。Holland 教授所提出的 GA 为简单遗传算法。

1.3.4 粒子群优化算法

Eberhart[16]等从鸟群捕食的模型中得到启示，提出了粒子群优化 (Particle Swarm optimization，简称PSO) 算法，并用于解决优化问题。在PSO算法中，每个优化问题的解称为“粒子”，所有的粒子都有一个由目标函数决定的适应值(*fitnessvalue*)、一个决定他们飞翔方向和距离的速度。PSO算法首先给出一群随机粒子(随机解)，然后每个粒子通过跟踪两个“极值”来更新自己，一个是粒子本身所找到的最优解，这个解叫做个体极值*pBest*，另一个极值是整个种群目前找到的最优解，这个极值是全局极值*gBest*。所有粒子追随当前的最优粒子在解 空间中进行搜索，通过迭代找到最优解。

1.3.5 禁忌搜索算法

禁忌搜索（Tabu Search，简称TS）的思想最早由Glover[17,18]提出。TS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

方向作为试探，选择使目标函数值变化最多的方向。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，对已经进行过的优化过程进行记录和选择，指导下一步的搜索方向，通过建立Tabu表来实现。为了找到“全局最优解”，不应执着于某一个特定的区域。禁忌搜索对于找到的一部分局部最优解，有意识地避开它（但不是完全隔绝），从而获得更多的搜索区间。

1.3.6 模拟退火算法

模拟退火算法(Simulated Annealing Algorithm，简称SAA)来源于固体退火原理，将固体加温至充分高，再让其徐徐冷却，加温时固体内部粒子随 温度升高变为无序状，内能增大，而徐徐冷却时粒子渐趋有序，在某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。SAA最早由Kirkpatrick [19]等应用于组合优化领域，它是基于Mente-Carlo迭代求解策略的一种随机寻优算法。其出发点是基于物理中固体物质的退火过程与一般组合优化问题之间的相似性，SAA从某一较高初温出发，伴随温度参数的不断下降，结合概率突跳特性在解空间中随机寻找目标函数的全局最优解，即能概率性地跳出局部最优解并最终趋于全局最优。

除上述六种常用算法外，求解设施选址问题的精确算法还有 weisefeld 算法、重心法、割平面法、动态规划法等；常用的启发式算法还有贪婪算法、层次分析法、蚁群优化算法等。通常对不同的模型其求解算法也不同，求解小规模 UFLP 比较适用的有分枝定界法、贪婪算法等传统算法；适用于求解大规模 UFLP 的算法有 GA、PSO 算法、TS 算法、SAA、蚁群优化算法等启发式算法。

1.4 本文的研究意义

UFLP 是经典的设施选址问题，在组合优化问题研究领域占有重要地位，是国内外学者研究的热点问题。国外关于 UFLP 的研究已经取得了很多成果，但国内对此问题的研究还处在一个初级阶段，因此研究 UFLP 及其求解算法是非常有意义的。

从复杂性角度看，UFLP 是 NP-Hard 问题[20]，不能在多项式时间内求得其最优解。已有的精确算法只适合求解小规模的 UFLP，为解决这类问题，学者们提出能在合理时间范围内求解 NP-Hard 问题的近似算法与启发式算法。启发式算法在搜索最优解的过程中，依据对所求问题结构与性质的认识，使搜索向着预期方向进行，其中遗传算法的提

出具有非常重要的意义。遗传算法是基于生物的自然选择与遗传进化机制提出的一种全局优化算法，与传统的启发式算法相比，遗传算法的主要优点是遗传算子简单易操作且具有群体搜索性质。遗传算子仅通过适应度值指标进行操作，从而降低了一般启发式算法在搜索过程中对人机交互的依赖；群体搜索性质使遗传算法可以突破局部搜索的限制，从而实现整个解空间的分布式信息搜索。正是因为遗传算法的上述特点使得其在求解设施选址问题方面有非常广泛的应用，但现有的求解 UFLP 的遗传算法很少有自适应的，因此本文针对 UFLP 提出了一种改进的自适应遗传算法。对遗传算法的改进主要针对交叉算子和变异算子，分别进行了自适应设计，并将均匀交叉加入到自适应交叉算子中，构成自适应均匀交叉。首先从理论上分析了本文的自适应遗传算法的优越性，然后通过数值试验，证明了该自适应遗传算法对求解大规模的 UFLP 很有效。

经典的设施选址问题是通过选择设施位置与需求点到设施的连接分配关系以达到设施开放费用与连接费用之和最小的一类问题。然而在现实世界中，开放的设施会因为自然灾害、工人罢工、恐怖袭击或者所有权的转移等原因而出现故障。由于设施出现故障使得一部分需求点可能需要连接到距离它们较远、有较高的连接费用的设施，或者由于未得到供应而遭受巨额损失，这就导致了总费用的增加。因此，可靠的设施选址成为一个亟待解决的问题。可靠性设施选址问题指：为了保证每个需求点在设施出现故障时的需求仍能得到满足，提高整个系统的可靠性，将每个需求点分配给多个不同的设施。目前关于可靠性设施选址问题的研究刚起步，研究成果主要见于国外。国外对可靠性设施选址问题的研究模型大多基于 p-中位问题而提出的，其模型比较复杂，假设条件比较严格。本文的研究放宽了假设条件，提出了基于 UFLP 的可靠性设施选址问题，将其称为无容量限制的可靠性设施选址问题，并给出了相应的模型。针对无容量限制的可靠性设施选址问题，提出了基于遗传算法的分阶段近似算法，该算法的基本思想是把原问题划分为多个不同的阶段，每个阶段对应一个子问题，再对每个子问题分别进行处理。每处理完一个子问题，每个需求点未连接的设施个数将会减少一个，并且每一阶段的需求点与设施的分配关系也随之确定，直到所有子问题都得到处理，就完成了设施的选择与需求点的分配，即原问题得到求解。本文首先从理论上证明了该算法的可行性，随后给出了一个小规模算例，进一步说明了该算法对求解无容量限制的可靠性设施选址问题是有效的。

1.5 本文的主要工作

本文分为五章。第一章为绪论，首先简单介绍了设施选址问题的研究内容和发展历程。第二节是设施选址问题的分类，按模型变量取值空间分为连续型设施选址问题与离散型设施选址问题，并介绍了几类主要的离散选址问题，包括 *p*-中位问题、*p*-中心问题、UFLP、CFLP。第三节是设施选址问题的求解算法，将其分为两大类，一类是精确算法，常用的有分支定界算法、拉格朗日松弛法；另一类是启发式算法，常用的有遗传算法、粒子群优化算法、禁忌搜索法、模拟退火算法。第四节是本文的研究意义，第五节是论文的主要工作。

第二章主要介绍了传统的遗传算法。包括遗传算法的定义，相关遗传术语的简单说明，并给出了遗传算法的特点、一般步骤；详细描述了遗传算法的实现技术，包括编码方法、适应度函数、选择算子、交叉算子、变异算子等；简单介绍了遗传算法的应用领域，主要包括函数优化、组合优化、生产调度等；最后给出了遗传算法的国内外研究现状。

第三章主要研究基于遗传算法的 UFLP 求解方法。第一节介绍了用 GA 求解 UFLP 的研究现状，由于国内关于这方面的文献较少，所以主要介绍了国外的研究现状。第二节给出了求解 UFLP 的自适应遗传算法，详细描述了编码机制、适应度函数的选取、染色体的表示、初始群体的生成、选择算子、交叉算子、变异算子。本文求解 UFLP 的 GA 采用的是 0-1 编码；适应度函数使用了简单适应度函数的改进方法，该适应度函数更适用于本文设计的遗传算子；初始群体的生成法与常用的方法有很大区别，采用的是具有方向性的初始群体生成法，该方法的优点是生成的初始解质量较高，更易于算法的进化搜索；选择算子采用的是锦标赛选择法，该方法的优点是选择概率容易控制，适合求解大规模的选址问题；交叉与变异算子都运用自适应思想将其设计成进化代数的函数，能随着算法的进化不断进行调整，使算法具有自我调整的性能，更符合生物进化自我调整的特征。第三节是数值试验，通过分析比较本文的自适应 GA 与其他几种常用求解 UFLP 的算法，说明了本文的自适应 GA 更适合求解大规模的 UFLP。第四节对该章的主要研究内容及成果进行了简单总结。

第四章研究的内容为可靠性设施选址问题。第一节简单介绍了可靠性设施选址问题及其研究现状。第二节提出了一种新的基于无容量限制的可靠性设施选址模型，该模型与已有的可靠性设施选址模型不同，建设所有设施都以常数概率出现故障，并将每个需

求点分配给多个设施，以提高整个供求系统的可靠性。第三节是基于遗传算法的分阶段近似算法，首先对相关变量及参数进行了描述，然后给出了算法的基本思想，该算法的主要原理是将原模型分解为多个阶段，然后分阶段处理，所有阶段都处理完毕后，原问题也得到了求解。第四节是算法的分析论证，通过引进容错性设施选址问题中的分层算法证明思想，证明了基于遗传算法的分阶段近似算法对求解 RUFLP 是可行的。为了进一步说明算法的可行性，第五节给出了一个小规模的算例，验证了所给出的算法对求解 RUFLP 是有效的。第六节对本章内容进行了简单总结。

第五章是对论文的总结与展望，通过总结提出了进一步研究的工作。

2 遗传算法

2.1 遗传算法简介

2.1.1 遗传算法定义

GA 是模拟生物进化过程的一类启发式算法，是自然遗传学与计算机科学相互结合、相互渗透而形成的计算方法。在应用 GA 进行求解时，从代表问题解集的种群开始，种群由经过基因编码的一定数目的个体组成，每个个体对应实际问题的一个解。染色体作为遗传过程的主要载体，其内部表现是某种基因组合，它决定了个体形状的外部表现，因此，算法的一开始需要进行编码工作。编码工作完成后，根据一定的规则产生初始种群，然后按照适者生存和优胜劣汰的原理，逐代进行演化，产生更好的近似解。在进化的每一代，根据个体的适应度大小通过选择算子选择个体，通过交叉算子、变异算子进行交叉和变异，产生一系列新的个体组成新种群。进化过程使种群产生比前一代个体更适应环境的新个体，末代种群中的最优个体经过解码，即为原问题的近似最优解。

2.1.2 遗传算法术语说明

GA 是由生物进化论和遗传学原理产生的搜索算法，在运用时会涉及以下生物遗传术语：

1．染色体(*Chromosome*)：又叫个体(*Individuals*)，一定数量的个体组成种群(*Population*)，种群中个体的数量叫做种群规模。

2．基因(*Gene*)：是组成染色体的元素，用于表示个体的特征。

3．基因位置(*Gene Position*))：在算法中表示一个基因在字符串中的位置，简称基因位。

4．适应度(*Fitness*，简记为 *F*)：表示个体对环境的适应能力，为了体现个体的适应性，引入与所求问题目标函数相关且能对每个个体进行度量的函数，将其称为适应度函数。

5．选择算子：主要作用是按照一定的规则在初始群体中选择一定数量的个体，将其保存于“交叉池”中，等待交叉操作。

6．交叉算子：对交叉池中的个体进行配对，形成一对一对的双亲，按照一定的规则，产生子代个体。

7．变异算子：对交叉算子产生的子代个体，按照一定的规则进行变异操作，生成新的个体。

8．表 2-1 描述了遗传算法中的生物遗传概念。

表 2-1 遗传算法中的生物遗传概念
Table 2-1 Biological genetic concept in GA

生物遗传概念	遗传算法中的作用
适者生存	算法停止时，最优目标值的解有最大的可能性被留住
个体	解
染色体	解的编码（字符串，向量等）
基因	解中每一分量的特征（如各分量的值）
适应性	适应函数值
群体	选定的一组解（其中解的个数为群体的规模）
种群	根据适应函数值选取的一组解
交配	通过交配原则产生一组新解的过程
变异	编码的某一分量发生变化的过程

2.1.3 遗传算法的特点

GA 是解决搜索问题的一种通用算法，由于其特有的编码操作及选择、交叉、变异算子，使得其具有以下特点：

1．GA 在搜索前，先将问题的初始解进行编码，形成字符串集，然后从该字符串集开始搜索，因此搜索范围大，利于全局寻优。

2．GA在搜索时很少使用其它 辅助信息，通常仅通过适应度值来评价个体，在此基础上进行遗传操作。适应函数虽然与目标函数相关，但遗传操作不要求其连续可微，对其定义域也没有要求，此特点大大扩展了GA的应用范围。

3．GA利用进化过程获得的信息自行组织搜索时，适应度大的个体具有较高的生存概率，并获得更适应 环境的基因结构，因此使得GA具有自组织和自学习性。

4．GA 在求解很多组合优化问题时，不需要对问题有非常深入的了解，也不需要有

很强的技巧。GA 对问题的决策变量进行编码后，按照一定的规则进行计算，其过程并不复杂，且能够较快收敛到一个满意解。

5．GA 具有较好的兼容性，能够与其他启发式算法较好地混合使用，如可以运用其他算法求得初始解，然后再运行 GA，或者将其他算法插入 GA 操作的过程中等。

2.1.4 遗传算法的一般步骤

GA 的基本步骤如下：

Step 1：选择问题的一个编码，设置与所求问题相关的参数；

Step 2：生成一个有 N 个染色体的初始群体 $POP(1)$，$t:=1$；

Step 3：确定适应度函数 F，对群体 $POP(t)$ 中的每一个染色体 $pop_i(t)$ 计算它的适应度函数值 F_i；

Step 4：若停止规则满足，则算法停止；否则，计算概率

$$p_i = \frac{F_i}{\sum_{j=1}^{N} F_j}, \quad i = 1,2,\cdots,N$$

并以上述概率分布从 $POP(t)$ 中随机选择一些染色体构成一个种群

$$SPOP(t+1) = \{ pop_j(t) \mid j = 1,2,\cdots,N \}$$

（注：$SPOP(t+1)$ 集合中可能会重复选到 $POP(t)$ 中的一个元素）；

Step 4：通过交配，得到一个有 N 个染色体的 $CPOP(t+1)$；

Step 5：以一个较小的概率 p_m，使得染色体的一个基因发生变异，形成 $MPOP(t+1)$，$t:=t+1$，一个新的群体 $POP(t)=MPOP(t)$，返回 Step 2。

2.2 遗传算法的实现技术

2.2.1 编码

因为 GA 不能直接处理解空间中的数据，所以必须把解空间中的解转化成遗传进化

里的染色体，这个转化过程成为编码。编码是 GA 中的基础工作之一，在进行编码时必须考虑以下编码规则：完备性、健全性、非冗余性。在对实际问题进行编码时必须将编码、交叉、变异结合起来考虑。常用的编码方式有 0-1 编码、自然数编码、实数编码、符号编码、混合编码、矩阵编码等。因为 UFLP 的决策变量为 0-1 变量，故适合的编码方法为 0-1 编码。

2.2.2 编码长度

编码长度是指组成一个染色体的基因个数，常与所求问题的决策变量个数相关。0-1 编码长度与精度要求和变量个数有关；实数编码长度则与实际问题的自变量位数相等；符号编码长度则由具体问题的特征决定；也有的编码长度会随进化的过程进行调整，但并不常用。对于 UFLP，因其常采用的是 0-1 编码，故其编码长度常与备选设施的个数相等。

2.2.3 种群规模

根据模式理论[21]，若种群规模为 N，则 GA 能从这 N 个个体中生成和检测 $O(N^3)$ 个模式，并在此基础上不断形成和优化积木块，直到找到最优解。显然 N 越大，遗传操作处理的模式就越多，生成有意义的积木块并逐渐进化为最优解的机会就越高。也就是说，种群规模越大，种群中个体的多样性越高，算法陷入局部最优解的危险就越小。但若 N 太大，在进化过程中评价适应度的次数会大大增加，计算量也就大大增加，从而影响算法的效率。另一方面，若种群规模太小，则会使 GA 的搜索空间分布范围太过局限，从而有可能使算法早熟收敛到局部最优解。实际应用中群体规模 N 一般在 20~100 之间取值。

2.2.4 算法终止条件

常用的有如下几种：

1. 给定一个最大的遗传代数 $T_{\max}$，一般取 100~1000，算法迭代次数达到 $T_{\max}$ 时停止。

2. 记录每一代种群最大适应度值，当该值不再发生变化或变化很小时，算法终止。

3. 记录每一代目标函数最优值，当连续 M 代都保持不变或变化很小时，算法停止。

2.2.5 适应度函数的选取

GA 在进化过程中基本不使用外部信息，仅以适应度函数为依据，评价个体的优劣。适应度函数的选择直接影响到 GA 的性能，尤其适应度值是选择操作的依据，在设计时，必须结合 GA 采用的选择算子及其他遗传操作进行综合考虑。最简单的适应度函数为目标函数，将其称为简单适应度函数，但其适用性比较局限，因此对特定的问题需对其进行改进，针对最小化问题（最大化问题的改进方法类似），常有如下两种种适应度函数的选取：

1．改进的简单适应度函数，令适应度函数为

$$F(x)=\begin{cases} C-f(x), & \text{若} f(x)<C \\ 0, & \text{否则} \end{cases}$$

其中 C 可以是一个输入值或是理论上的最大值。

2．非线性加速适应度函数，令适应度函数为：

$$F(x)=\begin{cases} \dfrac{1}{f(x)-f_{\min}}, & \text{若} f(x)<f_{\min} \\ W>0, & \text{若} f(x)=f_{\min} \end{cases}$$

其中 W 为充分小的整数， $f_{\min}$ 为当前代的最优目标函数值。

对 UFLP 最简单的方法是令适应度函数为简单适应度函数或者改进的简单适应度函数，但也要根据具体问题进行相应的调整。

2.2.6 初始群体的产生

GA 中初始群体的产生有如下几种方法：

1．完全随机生成法。该方法的优点是简单，缺点是产生的初始解或者远离最优解，使迭代效果不佳，或者离最优解太近，使算法容易陷入局部最优解。

2．根据所求问题的性质，了解最优解所占空间在整个解空间中的分布范围，然后在此分布范围内设定初始群体。

3．先随机生成一定数目的个体，然后从中挑出最好的个体加入到初始群体中，不断重复这一过程，直到初始群体中个体数达到了种群规模。

4．均匀种群法。该方法能一定程度上克服完全随机法的缺点，但会使操作上更复

杂，可能会付出更多的处理时间。

针对 UFLP 可以设计更符合具体问题的初始群体生成法。

2.2.7 选择算子

选择算子是指从当前群体中选择适应度高的个体，淘汰适应度低的个体，生成配对池（*mating pool*），是一种优胜劣汰的过程。选择算子主要包括：

1．按适应度值比例选择，是指首先计算每个个体的适应度值，然后计算此适应度值在所有个体适应度值总和中所占的比例，依此比例分布选择个体。

2．基于适应度值大小顺序的选择，将群体中的个体按适应度值由大到小排成一个序列，将事先确定的序列概率分配给每个个体，依此概率分布选择个体。

3．锦标赛选择[22-25]，指从当前群体中随机选择一定数量的个体，选择其中适应值最大的个体放入配对池中，反复执行这一过程，直到达到种群规模为此。

4．精英选择，是指当下一代种群中的最优个体劣于当前种群中最优个体时，将当前种群中的最优个体直接复制到下一代，替代下一代种群中最差的个体。

UFLP 常用的选择算子有轮盘赌选择、最优选择、锦标赛选择。

2.2.8 交叉算子

GA 的交叉算子是模仿生物交配繁殖的基因重组过程，其作用是将亲代的优良基因遗传给下一代，并生成具有复杂基因结构的新个体。实现基因结构重组的交叉算子的设计一般与所求解的具体问题有关，任何交叉算子需要满足一定的准则，即交叉算子要保证上一代种群中优秀个体的特征能在下一代的新个体中尽可能得到遗传。此外，交叉算子的设计需要和编码方式进行协调，交叉算子通过交叉概率 p_c 来进行操作。p_c 的选取可分为两种方式，一种是采用确定的常数值，一般取 $0.4 \sim 0.99$，该值设定后，在进化过程中保持不变；另一种方法是将其设置为自适应的。适用于 0-1 编码的有单点交叉、两点交叉、多点交叉、均匀交叉等。

2.2.9 变异算子

变异操作模拟了生物进化过程中染色体某基因发生突变的过程，通过变异改变染色

体的结构和物理性状，从而生成新的个体。在 GA 中，变异算子通过变异概率 p_m 将个体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。变异概率 p_m 的选择需要根据具体的问题来确定由于变异概率比较小，在操作过程中一些个体可能根本不发生变异，从而造成大量计算资源的浪费，因此，在 GA 的具体应用中，可以实施一些变通操作。变异概率 p_m 也有两类设定方法，一种是取经验值 $0.0001 \sim 0.1$，一旦取定，在整个进化过程中保持不变；另一种是将其设置为自适应的，即能随着进化过程，不断调整。常用的的变异算子有基本位变异、反转变异、二元变异、均匀变异、高斯变异、非均匀变异等，适用于 0-1 编码的有基本位变异、反转变异、二元变异等。

2.3 遗传算法的应用领域

1．函数优化：函数优化是 GA 的经典应用领域，也是对 GA 进行性能测试评价的常用算例。对于一些非线性、多目标的函数优化问题，用其他优化方法较难求解，而 GA 可以有效地求得较好的结果。

2．组合优化：GA 是寻求组合优化问题满意解的最佳工具之一，实践证明，GA 对于组合优化问题中的 NP-Hard 问题非常有效，如 GA 在选址问题中的应用。

3．自动控制：GA 已经在自动控制领域中得到了很好的应用，如基于 GA 的模糊控制器的优化设计、基于 GA 的参数辨识、基于 GA 的模糊控制规则的学习、利用 GA 进行人工神经网络的结构优化设计和权值学习等。

4．生产调度：生产调度问题的数学模型在很多情况下难以求解，即使经过一些简化之后可以进行求解，也会因为简化得太多而使求得的结果与实际相差太远，然而 GA 能够有效地求解复杂调度问题。

5．机器人学：GA 的起源来自于对人工自适应系统的研究，而机器人是一类复杂的难以精确建模的人工系统，因此机器人学自然成为 GA 的一个重要应用领域。

6．图像处理：图像处理是计算机视觉中的一个重要研究领域，在图像处理过程中，扫描、特征提取、图像分割等不可避免地存在一些误差，这些误差会影响图像处理的效果。如何使这些误差最小是图像处理中的重要要求，GA 在图像处理中误差的优化计算方面有很好的应用。

7．人工生命：自组织能力和自学习能力是人工生命的两大重要特征，人工生命与

GA 有着密切的关系，基于 GA 的进化模型是研究人工生命现象的重要理论基础。

8．机器学习：基于 GA 的机器学习，在很多领域中得到了应用。如基于 GA 的机器学习可用来调整人工神经网络的连接权，也可以用于人工神经网络的网络结构优化设计。

2.4 遗传算法的研究现状

进入 90 年代以来，GA 迎来了兴盛发展的时期，尤其是应用研究显得格外活跃。随着 GA 的研究进展，出现了一些新的应用领域：一是基于 GA 的机器学习，这一新的研究课题把 GA 从离散空间的优化搜索算法扩展到具有独特生成功能的机器学习中，突破了人工智能中知识获取与知识优化的瓶颈。二是 GA 正逐步与神经网络、混沌理论、模糊推理等其它智能算法相互渗透结合，这对开拓 21 世纪新的智能计算方法具有非常重要的意义。三是 GA 的并行处理应用研究十分活跃，对 GA 本身的发展及新一代智能计算机体系结构的研究都有重要的意义。四是 GA 与人工生命的结合，其中生物的进化、免疫、自适应等现象是人工生命的重要研究对象，GA 在这方面发挥了重要的功能。五是 GA 与进化规划、进化策略等进化计算理论的结合，目前这三者的比较研究和结合研究日益成为启发式算法研究的热点。

1991 年Whitey提出了基于领域交叉的交叉算子（Adjacency based crossover），该算子特别针对自然数序号编码，并将其应用到了 旅行商问题中。Ackley等提出了随机迭代遗传爬山法（Stochastic Iterated Genetic Hill-climbing，SIGH），采用了一种复杂的概率选择机制，实验结果表明，SIGH比单点交叉、均匀交叉的神经GA更优越。Bersini与Seront将GA与单一方法结合起来，形成单一操作的多亲交叉算子，该算子根据两个亲代及一个额外的个体产生新个体。国内也有不少学者对GA的交叉算子进行改进，2002 年戴晓明等应用多种群遗传并行进化思想，对不同种群使用不同的遗传策略进行搜索，并利用种群间的迁移算子进行遗传信息的交流，该方法能在一定程度上克服经典GA易收敛到局部最优值解的缺点。2004 年赵宏立等针对简单GA求解较大规模组合优化问题搜索效率不高的缺点，提出了一种基因块编码的并行GA（Building-block Coded Parallel GA，BCPGA），该方法以粗粒度并行GA为基本框架，在染色体群体中识别出可能的基因块，然后用基因块作为新的基因单位对染色体重新编码，产生长度较短的染色体，并以这些染色体作为进化的下一代个体。2005 年江雷等针对并行GA求解旅行商问题，用弹性策略来维持种群的多样性, 使算法跳出局部最优解，向全局最优解方向进化。

3 求解无容量限制的设施选址问题的遗传算法

3.1 用 GA 求解 UFLP 的研究现状

UFLP 的研究成果最早见于二十世纪六七十年代运筹学领域的文献[26-35]中。初始阶段，学者们试图用各种直观启发式方法得到问题的最优解，如贪婪算法[36]、交换搜索法，然而实际表明这些求解方法不能保证总是可以得到问题的最优解。随后，一些学者从概率分析、最坏情况分析、组合数学、经验启发式方法等方面对该问题进行了研究。从复杂性角度来看，UFLP 是非常难解的 NP-Hard 问题[20]，虽然有很多精确算法[37]能够求解该问题，但不能在多项式时间内找到问题的最优解，而且精确算法对大规模的 NP-Hard 问题并不适用，由此引发了计算机科学领域的众多算法研究者的兴趣。自 20 世纪六十年代以来，运用各种技术设计出 UFLP 的多种求解算法，其中以近似算法研究成果最为突出，如领域搜索算法[38]。Shmoys[39]等给出了 UFLP 的第一个常数近似度的近似最优解，该常数近似度为 1.50；Guha 和 Kuller[40]证明了 UFLP 的下界的存在性，并给出其下界为 1.463。

然而随着问题规模的不断扩大，一些非常有用的近似算法不再有效，一些学者开始研究启发式算法，用于求解大规模的 UFLP，其中以 GA 的研究成果较为显著。最早将 GA 应用到设施选址问题中的是 Hosage 和 Goodchild，他们在文献[41]中首次将 GA 用于求解 p-中位问题，并且得到了很好的结果。这一创新激起了学者的热情，更多的学者开始将 GA 应用到不同类型的选址问题中。Kratica 等人在文献[42,43]中也提出了基于 GA 的求解 UFLP 的算法；Jaramillo[44]等人讨论了基于启发式算法的 GA 求解 UFLP 等问题；最近日本学者 Tohyama 等人在文献[45]中设计了一种新的 GA 操作算子，数值试验表明他们的算法在求解大规模的 UFLP 问题上有很大的优势。

3.2 求解 UFLP 的自适应遗传算法

虽然遗传算法在诸多领域都有深入的应用，但在 UFLP 中的应用还不完善，尤其在国内，这方面的文章较少，且尚无系统的研究。因此继续深入研究这方面的内容具有重要的理论意义和应用价值。本文提出了一种改进的遗传算法来求解 UFLP，通过数值试

验说明该算法求解大规模的 UFLP 具有一定的优越性。

用遗传算法求解 UFLP 最重要的是三个操作算子的设计，文献[42-44]介绍了遗传算法如何求解 UFLP。本文的遗传算法采用了具有方向性的初始群体生成法，对交叉概率与变异概率进行了自适应设计，提出一种新的自适应交叉与变异概率，并将自适应交叉概率与均匀交叉相结合。通过数值试验，证明本文提出的求解 UFLP 的遗传算法不仅能寻找到令人满意的最优解，而且运行速度较快。

3.2.1 编码

本文的 GA 采用二进制编码，二进编制编码对于 UFLP 的优点在于简单容易操作。如在有四个备选设施的设施集{1,2,3,4}选址问题中，（1001）表示选择设施 1 和 4，（1111）表示选择所有设施。编码的长度 L 与具体的 UFLP 相关，一般 $L=m$，m 是备选设施的个数。

3.2.2 染色体的表示

令 $Y_u^t=(y_{u1}^t,y_{u2}^t,\cdots,y_{um}^t)$ （$u=1,2,...,N$）表示第 t 代的一个染色体，其中 $y_{ui}^t=\begin{cases}1, & \text{在备选点}i\text{修建设施}\\ 0, & \text{否则}\end{cases}$，$y_{ui}^t$、$x_{uij}^t$ 分别表示第 t 代时决策变量与分配变量的值，根据就近分配原则确定 x_{uij}^t 的值。则当算法运行至最大进化代数 $T_{\max}$ 时，令 $y_i=y_{ui}^{T_{\max}}$ 即为 UFLP 中选择变量 y_i 的值，再通过就近分配原则，确定 x_{ij} 的值。

3.2.3 初始群体的产生

在实际中，设施建造费用往往比连接费用高很多，因此，在解决实际选址问题时，要对修建费用与连接费用进行平衡。一方面，当设施建造费用比连接费用高很多时，可以通过修建尽可能少的设施来降低整体费用；另一方面，当设施建造费用与连接费用的比不是很大时，可以通过减少连接费用来降低整体费用。也就是说， UFLP 的最优解对应的染色体中值为 1 的基因的大致情况可以通过设施建造费用与设施到需求点的连接费用比值来估计。因此，在产生初始解时，就不是完全随机的，而是带有方向性，这种具

有方向性的初始种群产生方法使得对最优解的搜索更有效。为了说明此方法，下面引进一些参数：令v表示设施的平均修建费用与平均连接费用的比，则有$v=\dfrac{\frac{1}{m}\sum_{i=1}^{m}f_i}{\frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}h_jc_{ij}}$（其中$m$、$n$、$h_j$、$c_{ij}$、$f_i$是 UFLP 中的相关已知量，分别为备选设施个数、需求点个数、需求点j的需求量、从设施i到需求点j的单位运送费用、设施i的建造费用，为方便起见把h_jc_{ij}称为设施i到需求点j的连接费用），利用参数v就可以产生较好的初始解。虽然此方法有利于算法加速搜索最优解，但由于太过集中，可能导致初始解过于靠近最优解而使算法陷入局部最优。为了防止这种情况的发生，采取如下措施：

令$\sigma_1,\sigma_2\in[0,1]$，且$\sigma_1<\sigma_2$，$\sigma_1$与$\sigma_2$的经验值分别为 0.07、0.3；$\mu$是$v$的一个参考值，常取$\mu=1.5$。分两种情况处理：当$v\geq\mu$时，表明设施修建费用相比连接费用高很多，则生成个体基因中 1 较少的初始解，相对减少建造费用，从而降低整体费用；反之，设施修建费用不比连接费用高出很多，则生成个体基因中 1 较多的初始解，相对减少连接费用，从而降低整体费用。由此可知这种分类的标准是设施建造费用是否比连接费用高出很多，下面给出通过v产生有方向性且不易陷入局部最优的初始种群的具体步骤。

具有方向性的初始种群生成算子 *IO*：

Step1：给定参数$0\leq\sigma_1<\sigma_2\leq1$，$\mu=1.5$令$POP(1)$表示初始群体，计算

$$v=\frac{\frac{1}{m}\sum_{i=1}^{m}f_i}{\frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}h_jc_{ij}}$$，置$u:=1$；

Step2：对每个$i\in I$，随机产生实数$r_{ui}\in[0,1]$；

Step3：如果$v\geq\mu$，则转（1），否则转（2）；

（1）若$r_{ui}\leq\sigma_1$，则令$y_{ui}^1=1$，否则令$y_{ui}^1=0$；

（2）若$r_{ui}\leq\sigma_2$，则令$y_{ui}^1=1$，否则令$y_{ui}^1=0$；

Step4：生成初始个体$Y_u^1=(y_{u1}^1,y_{u2}^1,\cdots,y_{um}^1)$，若$u<N$，则令$u:=u+1$，转 Step2；否

则，算法停止，生成初始群体 $POP(1)=\{Y_u^1 \mid u=1,2,...,N\}$。

3.2.4 适应度函数

在应用GA算法求解UFLP时，最简单的方法是选取目标函数作为适应度函数，本文采用的适应度函数为 $F(Y_u^t)=C-\sum_{i=1}^{m} f_i y_{ui}^t - \sum_{i=1}^{m}\sum_{j=1}^{n} h_j c_{ij} x_{uij}^t$，其中 h_j, c_{ij} 是具体UFLP的已知常数，即当 $y_{ui}^t=1$ 时，表示建造设施 i； x_{ij} 是UFLP模型中的分配决策变量，其取值为0或1，其具体的取值由选择决策变量决定。由于设施没有容量限制，当 y_i 确定后，根据就近分配原则将需求点分配给离其最近的开放的设施，这样 x_{ij} 的取值就确定了，不需要为其设计染色体； C 是根据所有的 $y_i=1$，且将每个需求点分配给离其最远的设施求得的 x_{ij} 计算的目标函数值；

3.2.5 选择算子

本文采取的是锦标赛选择算子[22-26]，其基本思想是先从种群 $POP(t)$ 中随机选择 T 个个体（这里的 T 称为锦标赛规模，常取 $T=2$），比较这 T 个个体的适应度值，选择适应度值最大的个体作为新种群 $SPOP(t)$ 中的个体，这样重复 N 次便产生了新种群 $SPOP(t)$。锦标赛选择算子的特点主要有：对个体适应度值的正负没有要求；程序简单易行、收敛速度较快；有较大的概率保证最优个体被选择，最差个体被淘汰。对于求解 UFLP，锦标赛选择算子是一种较好的选择方法。

锦标赛选择算子 *SO*：

Step1：置 $s:=1$；

Step2：从 $POP(t)$ 随机选出 T 个个体，比较这 T 个个体的适应度值，选择适应度值最大的个体，记为 Y_s^t；

Step3：若 $s<N$，则令 $s:=s+1$，返回 Step2；否则转 Step4；

Step4：令 $Y_u^t := Y_s^t$，则选择算子产生的新种群可表示为 $SPOP(t)=\{Y_1^t, Y_2^t, ..., Y_N^t\}$。

3.2.6 交叉算子与变异算子

通过交叉操作，子代可以保留父代的优秀特征，故交叉算子主要负责局部搜索。而变异算子主要作用是保持种群的多样性，承担全局搜索的任务。传统的交叉、变异算子采用的是常数交叉、变异概率，然而，生物进化是一个自适应的过程，在整个进化过程中，个体可以通过自身的调节来适应进化的每个阶段。在进化的初始阶段，个体进化速度会很快，而在后期，随着进化的成熟，个体的进化速度会降低。GA 正是对生物遗传进化的模拟，因此也符合生物自适应的进化理论，在设计算子时，也要考虑自适应性。近年来，国外关于自适应遗传算法的研究文献[46-50]越来越多，但是关于求解 UFLP 的自适应遗传算法的文章还比较少，尤其是在国内。

鉴于此，本文研究自适应的遗传算法是有实际意义的。对于自适应的遗传算法有多种设计策略，可以将交叉概率、变异概率、适应函数等单独设计成自适应的，也可以将它们组合起来设计成自适应的。Jaramillo 通过数值试验证明了 p_c 越大局部搜索能力越强，可以加速种群的进化；p_m 越小，越能保护适应性好的个体。所以在算法的初始阶段将交叉概率 p_c 设置得大一些，将变异概率 p_m 设置得小一些，这样可以提高算法的收敛速度，保留较优的解。在进化后期，如果种群中个体趋于相似，则可能陷入了局部最优解，此时可以减小 p_c，同时增加 p_m，增加 GA 探索新解的能力，从而增强 GA 的全局搜索能力。总而言之，自适应遗传算法能够在进化速度与解的质量之间进行权衡，做出最好的选择。

本文将交叉概率与变异概率均设计成与进化代数 t 有关的函数，自适应交叉、变异概率函数分别为：

$$p_c^t=\begin{cases} p_{c0}, & \text{若}\dfrac{F_{avg}(t)}{F_{\max}(t)}>\alpha,\text{且}\dfrac{F_{\min}(t)}{F_{\max}(t)}\ge\dfrac{p_{c1}}{p_{c2}} \\ p_{c2}-\dfrac{p_{c2}-p_{c1}}{1-F_{\min}(t)/F_{\max}(t)}, & \text{若}\dfrac{F_{avg}(t)}{F_{\max}(t)}>\alpha,\text{且}\beta<\dfrac{F_{\min}(t)}{F_{\max}(t)}<\dfrac{p_{c1}}{p_{c2}} \\ p_{c2}, & \text{其他} \end{cases} \tag{3.3.1}$$

$$p_m^t=\begin{cases} p_{m2}, & 若\dfrac{F_{avg}(t)}{F_{\max}(t)}>\alpha,且\dfrac{F_{\min}(t)}{F_{\max}(t)}\geq\dfrac{p_{m1}}{p_{m2}} \\ p_{m2}-\dfrac{p_{m2}-p_{m1}}{1-F_{\min}(t)/F_{\max}(t)}, & 若\dfrac{F_{avg}(t)}{F_{\max}(t)}>\alpha,且\beta<\dfrac{F_{\min}(t)}{F_{\max}(t)}<\dfrac{p_{m1}}{p_{m2}} \\ p_{m0}, & 其他 \end{cases} \tag{3.3.2}$$

其中，p_c^t、p_m^t分别表示第t次迭代时的交叉与变异概率，p_{c0}、p_{c1}、p_{c2}、p_{m0}、p_{m1}、p_{m2}、α、β是给定的常数参数，满足$0<p_{c0}<p_{c1}<p_{c2}$，$0<p_{m0}<p_{m1}<p_{m2}$，通常取$p_{c1}=0.6$，$p_{c2}=0.9$，$p_{m1}=0.001$，$p_{m2}=0.1$，$\alpha=0.78$，$\beta=0.0005$；在(3.3.1)式中，$F_{avg}(t)$、$F_{\max}(t)$、$F_{\min}(t)$分别表示第t代种群$SPOP(t)$中个体的平均适应度值、最大适应度值、最小适应度值；在(3.3.2)式中，$F_{avg}(t)$、$F_{\max}(t)$、$F_{\min}(t)$分别表示第t代种群$CPOP(t)$中个体的平均适应度值、最大适应度值、最小适应度值。$\dfrac{F_{avg}(t)}{F_{\max}(t)}$反映了种群内部个体适应度的分布情况，两者越接近，表明该代个体越相似；$\dfrac{F_{\min}(t)}{F_{\max}(t)}$反映了整个种群的近似程度，两者越接近，表明 GA 可能陷入了局部最优解，失去了寻求新解的能力。当$\dfrac{F_{avg}(t)}{F_{\max}(t)}>\alpha,且\dfrac{F_{\min}(t)}{F_{\max}(t)}\geq\dfrac{p_{c1}}{p_{c2}}$时为进化后期，种群近似程度最大，应该选择小的交叉概率与大的变异概率；当$\dfrac{F_{avg}(t)}{F_{\max}(t)}>\alpha,且\beta<\dfrac{F_{\min}(t)}{F_{\max}(t)}<\dfrac{p_{c1}}{p_{c2}}$时为进化中后期，种群近似程度较大，应该选择较小的交叉概率与较大的变异概率；其他情况表明为进化初期，种群近似程度最小，应该选择较大的交叉概率与较小的变异概率。通过(3.3.1)与(3.3.2)式的自适应交叉与变异设计，可以使交叉与变异概率根据种群的集中程度自适应地变化。

本文的自适应思想与刘姝廷等人在文献[51]中提出的自适应思想虽有相同之处，但进行了较大改进，使得我们的自适应算子更符合实际情况。文献[51]提出的自适应交叉与变异概率函数分别为：

$$p_c=\begin{cases} p_{c2}-\dfrac{p_{c2}-p_{c1}}{1-F_{\min}(t)/F_{\max}(t)}, & \dfrac{F_{avg}(t)}{F_{\max}(t)}>\alpha,且\dfrac{F_{\min}(t)}{F_{\max}(t)}>\beta \\ p_{c2}, & 其他 \end{cases} \tag{3.3.3}$$

$$p_m=\begin{cases} p_{m2}-\dfrac{p_{m2}-p_{m1}}{1-F_{\min}(t)/F_{\max}(t)}, & \dfrac{F_{avg}(t)}{F_{\max}(t)}>\alpha,\text{且}\dfrac{F_{\min}(t)}{F_{\max}(t)}>\beta \\ p_{m2}, & \text{其他} \end{cases} \tag{3.3.4}$$

刘姝廷等人通过数值试验证明了他们的自适应变异算子对一般情况都是有效的，且比传统遗传算法在收敛性能和搜索能力上都有很大的提高。但是对于某些函数优化、UFLP等，会出现 $p_c, p_m < 0$ 的情况，这与 $p_c, p_m \in [0,1]$ 矛盾，下面通过一个简单函数优化例子来说明此问题。

函数优化问题：max $\quad f(x)=x^2-8x$，其中 $x\in[0,15]$，且 x 为整数。

设种群规模为 8，采用 0-1 编码，编码长度为 4，适应函数为 $F(x_i)=(x-4)^2$，相关常数参数按照上述取法设定。假设交叉操作前种群 $SPOP(t)$ 中的个体为：（1001），（1001），（1001），（1001），（1001），（1010），（1010），（1010），此时种群个体平均适应度值、最大适应度值、最小适应度值分别为：$F_{avg}=\dfrac{233}{8}$ 、$F_{\max}=36$、$F_{\min}=25$，根据公式（3.3.3）求得 $p_c\approx$ -0.08 <0，与 $p_c\in[0,1]$ 矛盾，因此该自适应交叉算子此时不可行。假设变异操作之前种群 $CPOP(t)$ 中个体为：（1100），（1100），（1100），（1100），（1100），（1101），（1101），（1101），此时种群个体平均适应度值、最大适应度值、最小适应度值分别为：$F_{avg}=\dfrac{563}{8}$ 、$F_{\max}=81$、$F_{\min}=64$，根据公式（3.3.4）求得 $p_m\approx$ -0.37 <0，与 $p_m\in[0,1]$ 矛盾，因此该自适应变异算子此时不可行。而通过本文的自适应交叉、变异概率函数(3.3.1)、(3.3.2)计算的交叉与变异概率不会出现小于 0 的情况。

本文将(3.3.1)式的自适应交叉概率与均匀交叉法[52]相结合，从而使交叉算子更有效。均匀交叉的思想为：对每一对配对的个体，预先随机产生一个由 0、1 两种字符组成的与所求问题编码长度相同的模板，然后通过模板相应基因位的值决定配对的个体相应基因位是否交叉，当模板基因位值为 1 时，交换两个配对个体相应基因，否则就不交换。模板中 1 与 0 的个数根据均与交叉概率来确定，一般取均与交叉概率为 0.3，若所求 UFLP 的备选设施个数为 m，则模板中 1 的个数为$\left\lceil \dfrac{3m}{10} \right\rceil$，0 的个数为 $m-\left\lceil \dfrac{3m}{10} \right\rceil$。下面通过一个简单的例子来说明此交叉操作。

设问题的种群规模为 6，编码长度为 8，种群中 6 个个体分别为

$pop_1, pop_2, pop_3, pop_4, pop_5, pop_6$，随机配成 3 对，假设结果为 $pop_1 \otimes pop_4$，$pop_2 \otimes pop_5$，$pop_3 \otimes pop_6$，$\otimes$ 表示交叉。用 $TP_i, i=1,2,3$ 表示三对待交叉个体相应的模板，均匀交叉操作结果如下：

$$\text{第一对：} \left.\begin{matrix} pop_1: 11001011 \\ pop_4: 00110100 \end{matrix}\right\} TP_1: 10001100 \to \left\{\begin{matrix} pop_1': 01000111 \\ pop_4': 10111000 \end{matrix}\right.$$

$$\text{第二对：} \left.\begin{matrix} pop_2: 10011101 \\ pop_5: 01100101 \end{matrix}\right\} TP_2: 00100110 \to \left\{\begin{matrix} pop_2': 10111101 \\ pop_5': 01000101 \end{matrix}\right.$$

$$\text{第三对：} \left.\begin{matrix} pop_3: 10111011 \\ pop_6: 01100101 \end{matrix}\right\} TP_3: 10101011 \to \left\{\begin{matrix} pop_3': 00110001 \\ pop_6': 11101111 \end{matrix}\right.$$

单点交叉的结果如下：

$$\text{第一对：} \left.\begin{matrix} pop_1: 110|01011 \\ pop_4: 001|10100 \end{matrix}\right\} \to \left\{\begin{matrix} pop_1': 11010100 \\ pop_4': 00101011 \end{matrix}\right.$$

$$\text{第二对：} \left.\begin{matrix} pop_2: 10011|101 \\ pop_5: 01100|101 \end{matrix}\right\} \to \left\{\begin{matrix} pop_2': 10011101 \\ pop_5': 01100101 \end{matrix}\right.$$

$$\text{第三对：} \left.\begin{matrix} pop_3: 1011101|1 \\ pop_6: 0110010|1 \end{matrix}\right\} \to \left\{\begin{matrix} pop_3': 10111011 \\ pop_6': 01100101 \end{matrix}\right.$$

通过上述结果的比较，我们发现单点交叉第二、三对都是无效的，因为单点交叉只选择一个交叉点，虽然能较好保持父代的特征，但很容易产生超级个体，从而陷入局部最优。相比于单点交叉，均匀交叉既考虑了待交叉种群中个体适应度值大小，又突破了单点交叉的交叉局限性，此外均匀交叉的程序语言更容易实现。

自适应均匀交叉算子 *CO*：

Step1：计算种群 $SPOP(t)=\{Y_1^t, Y_2^t, ..., Y_N^t\}$ 中每个个体 Y_u^t 的适应度值 $F(Y_u^t)(u=1,2,...,N)$；

Step2：令 $F_{\max}(t)=\max\{F(Y_u^t) \mid u=1,2,...,N\}$，$F_{\min}(t)=\min\{F(Y_u^t) \mid u=1,2,...,N\}$，$F_{avg}(t)=\sum_{u=1}^{N} F(Y_u^t) \Big/ N$；

Step3：计算

$$p_c^t=\begin{cases} p_{c0}, & 若\dfrac{F_{avg}(t)}{F_{\max}(t)}>\alpha,且\dfrac{F_{\min}(t)}{F_{\max}(t)}\geq\dfrac{p_{c1}}{p_{c2}} \\ p_{c2}-\dfrac{p_{c2}-p_{c1}}{1-F_{\min}(t)/F_{\max}(t)}, & 若\dfrac{F_{avg}(t)}{F_{\max}(t)}>\alpha,且\beta<\dfrac{F_{\min}(t)}{F_{\max}(t)}<\dfrac{p_{c1}}{p_{c2}}; \\ p_{c2}, & 其他 \end{cases}$$

Step4：将种群 $SPOP(t)=\{Y_1^t,Y_2^t,...,Y_N^t\}$ 中的个体随机配成 $\dfrac{N}{2}$ 对待交叉的父代，第 k 对父代记为（Y_k^t,Y_{N-k+1}^t），置 $k:=1$；

Step5：生成[0,1]之间的均匀随机数 p_k^t；

Step6：若 $p_k^t<p_c^t$，则生成由 $\left\lceil\dfrac{3m}{10}\right\rceil$ 个1、$m-\left\lceil\dfrac{3m}{10}\right\rceil$ 个0组成的交叉模板 $TP_k^t=（t_{k1},t_{k2},...,t_{km}）$，其中 $t_{kl}=0或1(l=1,2,\cdots,m)$。若 $t_{ki}=1$，则令 $\bar{y}_{ki}^t:=y_{N-k+1\,i}^t$、$\bar{y}_{N-k+1\,i}^t:=y_{ki}^t$；否则令 $\bar{y}_{ki}^t:=y_{ki}^t$、$\bar{y}_{N-k+1\,i}^t:=y_{N-k+1\,i}^t$，第 k 对父代产生的两个子代分别记为 $\overline{Y}_k^t=(\bar{y}_{k1}^t,\bar{y}_{k2}^t,...,\bar{y}_{km}^t)$、$\overline{Y}_{N-k+1}^t=(\bar{y}_{N-k+1\,1}^t,\bar{y}_{N-k+1\,2}^t,...,\bar{y}_{N-k+1\,m}^t)$；

Step7：若 $k<\dfrac{N}{2}$，则令 $k:=k+1$ 转（5），否则令 $Y_u^t:=\overline{Y}_u^t(u=1,2,...,N)$，交叉算子产生的新种群记为 $CPOP(t)=\{Y_1^t,Y_2^t,...,Y_N^t\}$。

自适应变异算子 *MO*：

Step1：计算种群 $CPOP(t)=\{Y_1^t,Y_2^t,...,Y_N^t\}$ 中每个个体 Y_u^t 的适应度值 $F(Y_u^t)(u=1,2,...,N)$；

Step2：令 $F_{\max}(t)=\max\{F(Y_u^t)\,|\,u=1,2,...,N\}$，$F_{\min}(t)=\min\{F(Y_u^t)\,|\,u=1,2,...,N\}$，$F_{avg}(t)=\sum_{u=1}^{N}F(Y_u^t)\Big/N$；

Step3：计算

$$p_m^t=\begin{cases} p_{m2}, & \text{若}\dfrac{F_{avg}(t)}{F_{\max}(t)}>\alpha,\text{且}\dfrac{F_{\min}(t)}{F_{\max}(t)}\ge\dfrac{p_{m2}}{p_{m1}} \\ p_{m2}-\dfrac{p_{m2}-p_{m1}}{1-F_{\min}(t)/F_{\max}(t)}, & \text{若}\dfrac{F_{avg}(t)}{F_{\max}(t)}>\alpha,\text{且}\beta<\dfrac{F_{\min}(t)}{F_{\max}(t)}<\dfrac{p_{m1}}{p_{m2}} \\ p_{m0}, & \text{其他} \end{cases};$$

Step4：置$u:=1$；

Step5：生成[0,1]之间的均匀随机数p_u^t，若$p_u^t<p_m^t$，则随机选择个体$Y_u^t=(y_{u1}^t,y_{u2}^t,...,y_{um}^t)$的一个基因位$y_{ui}^t$，令$y_{ui}^t:=1-y_{ui}^t$，否则令$y_{ui}^t:=y_{ui}^t$；

Step6：若$u<N$，则令$u:=u+1$，转Step5，否则令$Z_u^t:=(y_{u1}^t,y_{u2}^t,...,y_{um}^t)\ (u=1,2,...,N)$，变异算子产生的新种群记为$MPOP(t)=\{Y_u^t\mid u=1,2,...,N\}$，变异操作结束。

3.2.7 求解 UFLP 的 GA

通过上述对GA相关实现技术的描述与设计，求解UFLP的GA步骤如下：

Step 1：设置参数μ，$0\le\sigma_1\le\sigma_2\le1$，$0<p_{c0}<p_{c1}<p_{c2}$，$0<p_{m0}<p_{m1}<p_{m2}$，$\alpha$，$\beta$，锦标赛规模$T$，种群规模$N$（取偶数），最大进化代数$T_{\max}$。

Step2：计算$v=\dfrac{\frac{1}{m}\sum_{i=1}^{m}f_i}{\frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}h_jc_{ij}}$，生成初始群体：

（1）令$POP(1)$表示初始群体，置$u:=1$；

（2）对每个$i\in I$，随机产生实数$r_{ui}\in[0,1]$；

（3）若$v\ge\mu$，则转（i），否则转（ii）；

（i）若$r_{ui}\le\sigma_1$，则令$y_{ui}^1=1$，否则令$y_{ui}^1=0$；

（ii）若$r_{ui}\le\sigma_2$，则令$y_{ui}^1=1$，否则令$y_{ui}^1=0$；

（4）生成初始个体$Y_u^1=(y_{u1}^1,y_{u2}^1,\cdots,y_{um}^1)$，若$u<N$，则令$u:=u+1$，返回（2）；否

则算法停止，生成初始群体 $POP(1)=\{Y_u^1 \mid u=1,2,...,N\}$。

Step3：对群体 $POP(t)$ 中的每个个体 Y_u^t 计算其适应度值，即 $F(Y_u^t)=\sum_{i\in I}\sum_{j\in J}h_j c_{ij}x_{uij}^t+\sum_{i\in I}f_i y_{ui}^t$，记 $H^t=\{i\in I \mid y_{ui}^t=1\}$，令 $d_t=|H^t|$，对每个需求点 j 将 $\{c_{ij} \mid y_i^t=1\}$ 中的元素按从小到大的顺序排列，记为 $c_{i_{j1}j}\le c_{i_{j2}j}\le ...\le c_{i_{jd_t}j}$，其中 $i_{jl}\in H^t(l=1,2,...,d_t)$，令 $x_{uij}^t=\begin{cases}1,若i=i_{j1}\\0,否则\end{cases}$，计算 $F(Y_u^t)$。

Step4：若 $t<T_{\max}$，则从 $POP(T_{\max})$ 中选出适应度值最大的个体记为 $Y_{u^*}^{T_{\max}}=(\mathrm{y}_{u^*1}^{T_{\max}},\mathrm{y}_{u^*2}^{T_{\max}},...,\mathrm{y}_{u^*m}^{T_{\max}})\ (u^*\in\{1,2,...,N\})$，令 $y_i:=y_{u^*i}^{T_{\max}},x_{ij}:=x_{u^*ij}^{T_{\max}}\ (i=1,2,...,m)$，输出 x_{ij}、y_i；否则转 Step5。

Step5：执行选择算子 SO：

（1）置 $s:=1$；

（2）从 $POP(t)$ 随机选出 T 个个体，比较这 T 个个体的适应度值，选择适应度值最大的个体，记为 Y_s^t；

（3）若 $s<N$，则令 $s:=s+1$，返回（2）；否则转（4）；

（4）令 $Y_u^t:=Y_s^t$，选择算子产生的新种群可表示为 $SPOP(t)=\{Y_1^t,Y_2^t,...,Y_N^t\}$。

Step6：执行交叉算子 CO：

（1）计算种群 $SPOP(t)=\{Y_1^t,Y_2^t,...,Y_N^t\}$ 中每个个体 Y_u^t 的适应度值 $F(Y_u^t)(u=1,2,...,N)$；

（2）令 $F_{\max}(t)=\max\{F(Y_u^t) \mid u=1,2,...,N\}$，$F_{\min}(t)=\min\{F(Y_u^t) \mid u=1,2,...,N\}$，$F_{avg}(t)=\sum_{u=1}^{N}F(Y_u^t)\Big/N$；

（3） 计算

$$p_c^t=\begin{cases} p_{c0}, & \text{若}\dfrac{F_{avg}(t)}{F_{\max}(t)}>\alpha,\text{且}\dfrac{F_{\min}(t)}{F_{\max}(t)}\geq\dfrac{p_{c1}}{p_{c2}} \\ p_{c2}-\dfrac{p_{c2}-p_{c1}}{1-F_{\min}(t)/F_{\max}(t)}, & \text{若}\dfrac{F_{avg}(t)}{F_{\max}(t)}>\alpha,\text{且}\beta<\dfrac{F_{\min}(t)}{F_{\max}(t)}<\dfrac{p_{c1}}{p_{c2}} \\ p_{c2}, & \text{其他} \end{cases};$$

（4）将种群$SPOP(t)=\{Y_1^t,Y_2^t,...,Y_N^t\}$中的个体随机配成$\dfrac{N}{2}$对待交叉的父代，第$k$对父代记为$Y_k^t,Y_{N-k+1}^t$，置$k:=1$；

（5）生成[0,1]之间的均匀随机数p_k^t；

（6）若$p_k^t<p_c^t$，则生成由$\left\lceil\dfrac{3m}{10}\right\rceil$个1、$m-\left\lceil\dfrac{3m}{10}\right\rceil$个0组成的交叉模板$TP_k^t=$（$t_{k1},t_{k2},...,t_{km}$）, 其中$t_{kl}=0$或$1,l=1,2,...,m$, 若$t_{ki}=1$, 则令$\overline{y}_{ki}^t:=y_{N-k+1\,i}^t$、$\overline{y}_{N-k+1\,i}^t:=y_{ki}^t$；否则令$\overline{y}_{ki}^t:=y_{ki}^t$、$\overline{y}_{N-k+1\,i}^t:=y_{N-k+1\,i}^t$，第$k$对父代产生的两个子代分别记为$\overline{Y}_k^t=(\overline{y}_{k1}^t,\overline{y}_{k2}^t,...,\overline{y}_{km}^t)$、$\overline{Y}_{N-k+1}^t=(\overline{y}_{N-k+1\,1}^t,\overline{y}_{N-k+1\,2}^t,...,\overline{y}_{N-k+1\,m}^t)$；

（7）若$k<\dfrac{N}{2}$，令$k:=k+1$，返回（5）；否则令$Y_u^t:=\overline{Y}_u^t(u=1,2,...,N)$，交叉算子产生的新种群记为$CPOP(t)=\{Y_1^t,Y_2^t,...,Y_N^t\}$。

Step7：执行变异算子MO：

（1）计算种群$CPOP(t)=\{Y_1^t,Y_2^t,...,Y_N^t\}$中每个个体$Y_u^t$的适应度值$F(Y_u^t)(u=1,2,...,N)$；

（2）令$F_{\max}(t)=\max\{F(Y_u^t)\mid u=1,2,...,N\}$，$F_{\min}(t)=\min\{F(Y_u^t)\mid u=1,2,...,N\}$，$F_{avg}(t)=\sum_{u=1}^{N}F(Y_u^t)\Big/N$；

（3）计算

$$p_m^t=\begin{cases} p_{m2}, & 若\dfrac{F_{avg}(t)}{F_{\max}(t)}>\alpha,且\dfrac{F_{\min}(t)}{F_{\max}(t)}\ge\dfrac{p_{m1}}{p_{m2}} \\ p_{m2}-\dfrac{p_{m2}-p_{m1}}{1-F_{\min}(t)/F_{\max}(t)}, & 若\dfrac{F_{avg}(t)}{F_{\max}(t)}>\alpha,且\beta<\dfrac{F_{\min}(t)}{F_{\max}(t)}<\dfrac{p_{m1}}{p_{m2}} \\ p_{m0}, & 其他 \end{cases};$$

（4）置$u:=1$；

（5）生成[0,1]之间的均匀随机数p_u^t，若$p_u^t<p_m^t$，则随机选择个体$Y_u^t=(y_{u1}^t,y_{u2}^t,...,y_{um}^t)$的一个基因位$y_{ui}^t$，令$y_{ui}^t:=1-y_{ui}^t$，否则令$y_{ui}^t:=y_{ui}^t$；

（6）若$u<N$，则令$u:=u+1$，返回（5）；否则令$Y_u^t:=(y_{u1}^t,y_{u2}^t,...,y_{um}^t)\ (u=1,2,...,N)$，变异算子产生的新种群记为$MPOP(t)=\{Y_u^t\,|\,u=1,2,...,N\}$，令$t:=t+1$，且令新的群体为$POP(t)=MPOP(t)$，返回 Step3。

3.3 数值试验

为叙述方便，将本文提出的遗传算法称为自适应遗传算法（Adaptive Genetic Algorithm,简称 AGA）。本文的数值试验在“Windows XP professional computer (Pentium R, 2.79GHZ) with Matlab 7.10”平台上实现的，相关数据来源于“OR-Library”[53]。

表 3-1 列示了 AGA 与局部搜索算法[54]（Local Search Algorithm，简称 LSA）、禁忌搜索算法[55]（Tabu Search Algorithm，简称 TSA）的比较。表 3-1 的第一列是具体的 UFLP，包括问题的规模、已知的最优目标函数值；第二列是具体的算法；第三列是求得最优解的成功率，该值越大对应的算法越能求得最优解；平均值为目标函数值的平均，该值越小即越接近最优目标函数值，说明对应的算法越好；最大值表示在所有迭代中出现的目标函数的最大值，该值越接近最优目标函数值对应的算法越好。通过表 3-1 可以看到，除了 “*Cap* B”外 LSA 不能求得表中所列任何 UFLP 的最优解，是这三种算法中最差的一种； TSA 能求得所列大部分 UFLP 的最优解，并在成功率、平均值、最大值方面都有不错的表现；与 LSA、TSA 相比，除了“*Cap* C”外，AGA 能以 100%的成功率求解所列 UFLP，虽然对表中大部分 UFLP 比其他算法需要花更多的时间，但在实际应用中 3 分钟以下的时间并不算多。

表 3-1 AGA 与 LSA、VA、TSA 的比较

Table 3-1 Comparative results of AGA, LSA, VA, and TSA

UFLP [$m\times n$] 最优目标值	算法	成功率（%）	平均时间（*s*）	平均值	最大值
*Cap*72 [16×50] 977799.4	LSA	0	0.00	979,099.6	979,099.6
	TSA	100	0.00	977,799.4	977,799.4
	AGA	100	0.03	977,799.4	977,799.4
*Cap*73 [16×50] 1,010,641.5	LSA	0	0.00	1,011,067.7	1,011,067.7
	TSA	100	0.00	1010641.5	1010641.5
	AGA	100	0.04	1010641.5	1010641.5
*Cap*101 [25×50] 796,648.4	LSA	0	0.00	797,582.3	797,582.3
	TSA	80	0.01	796,820.5	797,508.7
	AGA	100	0.11	796,648.4	796,648.4
*Cap*103 [25×50] 893,782.1	LSA	100	0.01	893,782.1	893,782.1
	TSA	94	0.02	893,795.7	894,008.1
	AGA	100	0.10	893,782.1	893,782.1
*Cap*104 [25×50] 928,941.8	LSA	0	0.00	930,026.6	930,026.6
	TSA	100	0.00	928,941.8	928,941.8
	AGA	100	0.02	928,941.8	928,941.8
*Cap*131 [50×50] 793,439.6	LSA	100	0.01	793,439.6	793,439.6
	TSA	84	0.03	793,577.2	794,299.9
	AGA	100	3. 81	793,439.6	793,439.6
*Cap*133 [50×50] 893,076.7	LSA	0	0.01	895,292.1	895,292.1
	TSA	96	0.03	893,104.9	893,782.1
	AGA	100	0.41	893,076.7	893,076.7
*Cap*134 [50×50] 928,941.8	LSA	0	0.01	935,422.7	935,422.7
	TSA	100	0.01	928,941.8	928,941.8
	AGA	100	0.07	928,941.8	928,941.8
Cap B [100×1000] 12,979,071.6	LSA	50	0.55	13,041,143.9	—
	TSA	53	0.39	13,022,893.3	13,214,718.1
	AGA	100	107.63	12,979,071.6	12,979,071.6
Cap C [100×1000] 11,505,594.3	LSA	0	0.48	11,534,161.4	—
	TSA	68	0.81	11,514,330.7	11,672,443.3
	AGA	96	175.13	11,505,687.9	11,511,743.8

表 3-2 AGA 与 GA、SAA 的比较结果

Table 3-2 Comparative results of AGA, GA, and SAA

问题	平均相对误差（%）			平均计算时间（s）		
UFLP [$m \times n$]	SGA	ESA	AGA	SGA	ESA	AGA
*Cap*71 [16×50]	0.000	0.000	0.000	0.287	0.041	0.043
*Cap*72 [16×50]	0.000	0.000	0.000	0.322	0.028	0.029
*Cap*73 [16×50]	0.033	0.000	0.000	0.773	0.031	0.037
*Cap*74 [16×50]	0.000	0.000	0.000	0.200	0.018	0.022
*Cap*101 [25×50]	0.020	0.000	0.000	0.801	0.256	0.107
*Cap*102 [25×50]	0.000	0.000	0.000	0.896	0.098	0.069
*Cap*103 [25×50]	0.015	0.000	0.000	1.371	0.119	0.093
*Cap*104 [25×50]	0.000	0.000	0.000	0.514	0.026	0.018
*Cap*131 [50×50]	0.065	0.008	0.000	6.663	2.506	3. 814
*Cap*132 [50×50]	0.000	0.000	0.000	5.274	0.446	0.425
*Cap*133 [50×50]	0.037	0.002	0.000	7.189	0.443	0.412
*Cap*134 [50×50]	0.000	0.000	0.000	2.573	0.079	0.067
Cap A [100×1000]	0.000	0.000	0.000	184.422	17.930	17.048
Cap B [100×1000]	0.172	0.070	0.000	510.445	91.937	107.625
Cap C [100×1000]	0.131	0.119	0.001	591.516	131.345	175.132

表 3-2 列示了 AGA 与遗传算法[44]（Genetic Algorithm 简记 GA），模拟退火算法[56]（Simulated Annealing Algorithm，简称 SAA）的比较。平均相对误差为 $(\sum_{t=1}^{T}(f_t - f_0) \Big/ f_0 T) \times 100\%$，其中 t 、$T_{\max}$ 、 f_0 、 f_t 分别表示第 t 次循环、最大的循环次数、最优目标函数值、第 t 次循环的目标函数值。平均相对误差越小、平均计算时间越少，

对应的算法求解 UFLP 的性能越好。通过表 3-2 可以知道，SAA 在求解 UFLP 时不管是求解的质量还是时间上都比 GA 好，但求解“*Cap*131、*Cap*133、*Cap* B、*Cap* C”时存在误差，而 AGA 只有在求解“*Cap* C”时存在误差。从算法所需时间上看，对 SAA 与 AGA 都能百分之百求得最优解的 UFLP，AGA 所需的时间更少；对 SAA 与 AGA 都不能百分之百求得最优解的 UFLP，AGA 的相对误差更小；但对 SAA 不能而 AGA 能百分之百求得最优解的 UFLP，AGA 所需的时间较多。

通过上述比较，本文的 AGA 在求解小规模与中等规模的 UFLP 时，不管是解的质量还是时间上，都具有优越性；在求解大规模的 UFLP 时，尽管所需的时间有所增加，但它能获得令人满意的解，并且能精确求解其他算法不能精确求解的某些大规模 UFLP。在实际应用中人们会选择在合理的时间内能求得更接近精确最优解的算法，而降低对时间的要求，且在上述数值试验中，本文提出的 AGA 求解 UFLP 所需时间都在 3 分钟以下，该时间在实际应用中并不算多，因此本文提出的 AGA 在求解大规模 UFLP 时具有一定的优越性。

3.4 本章小结

本章主要介绍了求解无容量限制的设施选址问题的遗传算法，详细描述了编码机制、适应度函数的选取、染色体的表示、初始群体的生成、选择算子、交叉算子、变异算子。在遗传算法的实现技术中，采用了 0-1 编码，改进方法的简单适应度函数，该适应度函数更适用于本文设计的遗传算子。初始群体的生成法与常用的方法有很大区别，采用的是具有方向性的初始群体生成法，该方法的优点是生成的初始解质量较高，更易于算法的进化搜索；选择算子采用的是锦标赛选择法，该方法的优点是选择概率容易控制，适合求解大规模的选址问题；交叉与变异算子都运用自适应思想将其设计成进化代数的函数，能随着算法的进化不断进行调整，使算法具有自我调整的性能，更符合生物进化自我调整的特征。给出了求解 UFLP 改进的的自适应遗传算法的具体步骤，分析了该算法的优越性，并通过数值试验进一步证明了该算法求解大规模无容量限制设施选址问题具有较好的性能。

4 可靠性设施选址问题及其求解算法

经典的设施选址问题是通过选择设施位置与需求点到设施的连接分配关系以达到设施开放费用与连接费用之和最小的一类问题。然而在现实世界中，开放的设施会因为自然灾害、工人罢工、恐怖袭击或者所有权的转移等原因而出现故障。由于设施出现故障使得一部分需求点可能需要连接到距离它们较远、有较高连接费用的设施上，或者由于未得到供应而遭受巨额损失，这就导致了总费用的增加。因此，可靠的设施选址成为一个亟待解决的问题。

4.1 可靠性设施选址问题研究现状

随着可靠性理念在网络安全中的盛行，设施选址问题中的可靠性研究也引起了越来越多的学者的关注。目前关于可靠性设施选址问题的研究主要见于国外，最早是由 Snyder 和 Daskin[57]于 2005 年提出的，他们的模型是基于 p-中位问题与 UFLP 提出的，在他们的模型中，将每个需求点连接到多个设施，且将所有的设施分为三类：一类是会以固定概率出现故障的设施；一类是不会出现故障的设施；还有一类是备用设施，即当需求点连接的所有设施都出现故障时，只能选择备用设施。因为备用设施的修建费用非常高，所以一般只有一个备用设施。文[57]提出了拉格朗日松弛算法来求解这类问题，并通过 *trade-off* 曲线来描述模型是如何以费用的最小增加达到设施供应系统可靠性的最大化。随后，Lim[58] 等人于 2009 年提出了新的可靠性设施选址模型，在他们的模型中将设施分为会出现故障与不会出现故障两类，且设施出现故障的概率是相互独立的，每个需求点只需连接到两个设施：一个是会出现故障的设施，一个是不会出现故障的设施。虽然模型有所改变，但仍然使用了拉格朗日松弛方法。Cui [59]等于 2010 年提出的可靠性设施选址模型中，设施出现故障的概率互不相同，且每个需求点连接到多个设施。他们提出了拉格朗日松弛方法与连续近似法相结合的新方法，该方法比单纯的拉格朗日松弛方法有所改进，能解决较大规模的问题，但是他们的模型不如前述两个具有代表性。

4.2 无容量限制的可靠性设施选址问题

本文研究的可靠性设施选址问题是在UFLP基础上提出的，即无容量限制的可靠性设

施选址问题(Reliability for the Uncapacitated Facility Location Problem，简称RUFLP）。RUFLP描述如下：

记I表示备选设施集合，J表示需求点集合，$|I|=m$，$|J|=n$。f_i表示设施$i\in I$的开放费用（或称为修建费用），每个需求点$j\in J$的需求量为h_j，从设施i到需求点j每单位需求量的连接费用（或称为运输费用）为c_{ij}，且c_{ij}满足非负性、对称性、三角不等式。令$\omega_{ij}=h_jc_{ij}$，表示设施i到需求点j的连接费用，集合I中的设施以相同的概率q出现故障。为了保证每个需求点在设施出现故障时的需求仍能得到满足，从而提高整个系统的可靠性，需将每个需求点分配给K个不同的设施，这里的K是满足$2\le K<m$的正整数，可根据具体情况来选取。记每个需求点连接到的K个不同的设施分别对应不同的K层，所有需求点首先由它们的第一层设施来满足，如果第一层设施出现故障，则由第二层设施来提供需求，以此类推，只有当需求点j的前$r-1$层设施都出现故障时，第r层的设施才为其提供需求。RUFLP就是在备选设施集合I中寻找一个子集$F\subseteq I$，开放F中的设施，然后将J中的需求点按层次分配给F中的设施，且每个需求点j的每一层都由不同的设施提供需求，使得需求点与设施的期望连接费用与设施的开放费用之和达到最小。

分配决策变量x_{ijr}、开放决策变量y_i分别定义如下：若需求点j的第r层设施为i，则$x_{ijr}=1$，否则$x_{ijr}=0$；若开放设施i，则$y_i=1$，否则令$y_i=0$。RUFLP的目标函数为：

$$\sum_{i\in I}\sum_{j\in J}\sum_{r=1}^{K}h_jc_{ij}q^{r-1}(1-q)x_{ijr}+\sum_{i\in I}f_iy_i=\sum_{i\in I}\sum_{j\in J}(1-q)\omega_{ij}\sum_{r=1}^{K}q^{r-1}x_{ijr}+\sum_{i\in I}f_iy_i\text{ 。}$$

根据以上假设与定义，RUFLP的数学模型如下：

(LP1)

$$\min\ \sum_{i\in I}\sum_{j\in J}(1-q)\omega_{ij}\sum_{r=1}^{K}q^{r-1}x_{ijr}+\sum_{i\in I}f_iy_i \tag{4.2.1}$$

$$s.t.\ \sum_{i\in I}x_{ijr}=1,\qquad \forall j\in J, r=1,2,\cdots,K \tag{4.2.2}$$

$$\sum_{r=1}^{K}x_{ijr}\le y_i,\qquad \forall i\in I, j\in J \tag{4.2.3}$$

$$x_{ijr},y_i\in\{0,1\},\qquad \forall i\in I, j\in J, r=1,2,\cdots,K \tag{4.2.4}$$

(4.2.1)——目标函数，表示需求点与设施的期望连接费用与设施开放费用之和；

(4.2.2)——保证每个需求点 j 的每一层 r 有且仅有一个设施；

(4.2.3)——保证每个需求点 j 的每一层 r 连接到互不相同的开放的设施；

(4.2.4)——分配决策变量 x_{ijr} 和开放决策变量 y_i 为 0-1 变量。

RUFLP 模型是一个 0-1 整数线性规划，与此相关的算法都可以作为该模型求解算法的参考。本文的算法参考了 Jain 和 Vazirani[60]提出的关于容错性设施选址问题的分阶段近似算法，通过第四部分的分析论证，我们的算法能得到 RUFLP 的近似度为 H_K 的近似最优解。

4.3 基于遗传算法的分阶段近似算法

算法的基本思想：把 RUFLP 划分为 K 个不同的阶段，每个阶段对应一个子问题，再对每个子问题分别进行处理。每处理完一个子问题，每个需求点未连接的设施将会减少一个，并且每一阶段的需求点与设施的分配关系也随之确定，直到 K 个子问题都得到处理，就完成了设施的选择与需求点 K 层的分配，即原问题得到求解。但是需求点与设施现有的层次连接分配可能并不是最优的，所以需要重新分配需求点与设施的层次连接关系，但最后开放的设施集合不会发生改变。

在问题的第 r 阶段开始时，设前 $r-1$ 个阶段已经开放的设施集合为 F_{r-1}，X_{r-1} 表示第 $r-1$ 个阶段值为 1 的分配决策变量集合；对每个需求点 j，令 $F_{r-1}(j)$ 表示前 $r-1$ 个阶段已经连接到需求点 j 的设施集合，$W_{r-1}(j)$ 表示需求点 j 前 $r-1$ 层的连接费用集合；令第 r 阶段新开放的设施集合为 N_r。在第 r 阶段，对任意的需求点 j 可能被分配给 $F_{r-1} \setminus F_{r-1}(j)$ 中的设施，也可能被分配给 N_r 中的设施。对于前一种情形，只需考虑需求点与设施的连接费用，而无需考虑设施的开放费用；对于后一种情形，既需要考虑连接费用又需要考虑新开放设施的开放费用。第 r 阶段处理完后开放的设施集为 $F_r = F_{r-1} \cup N_r$；值为 1 的分配决策变量集合为 $X_1 \cup X_2 \cup \cdots \cup X_r$，需求点 j 的第 r 层连接费用集合 $W_r(j) = W_{r-1}(j) \cup \{\omega_{ij} \mid x_{ijr} = 1, i \in F_r \setminus F_{r-1}(j)\}$。所有阶段都处理完后，开放的实施集合为 F_K，值为 1 的分配决策变量集合为 $X_1 \cup X_2 \cup \cdots \cup X_K$，需求点 j 在第 K 层的连接费用集合

为$W_K(j)$。第r阶段对应的子问题的目标函数表示该阶段新开放设施的开放费用与所有需求点第r层与开放设施的期望连接费用之和。

根据以上假设与定义，RUFLP 的第r阶段即第r个子问题的模型如下：

(LP2)

$$\begin{aligned} \min \quad & \sum_{i\in I}\sum_{j\in J}(1-q)\omega_{ijr}q^{r-1}x_{ijr}+\sum_{i\in I\setminus F_{r-1}} f_i y_i \\ s.t. \quad & \sum_{i\in I\setminus F_{r-1}(j)} x_{ijr}=1, \quad \forall j\in J \\ & x_{ijr}\le y_i, \quad \forall j\in J, i\in I\setminus F_{r-1} \\ & x_{ijr}, y_i\in\{0,1\}, \quad \forall i\in I, j\in J \end{aligned}$$

其中ω_{ijr}的定义如下：当$i\in F_{r-1}(j)$时，令$\omega_{ijr}=+\infty$，否则令$\omega_{ijr}=\omega_{ij}$。经过这样的处理能保证每个开放的设施只连接到每个需求点一次。因此，当r固定时，上述模型就是经典的 UFLP 模型，可以采用能求解 UFLP 的常用方法来求解上述第r个子问题，如遗传算法[44] 、分支定界算法[61]、带惩罚的近似算法 [62]、禁忌搜索法[63]等。本文采用文[64]中的遗传算法求解上述第r个子问题。

基于遗传算法的分阶段近似算法如下：

Step1： 给定正整数$K(2\le K<m)$，置$r:=1$，$F_0=\phi$，$X_0=\phi$，对每个$j\in J$，置$F_0(j)=\phi$，$W_0(j)=\phi$。

Step2：（1）若$i\in F_{r-1}(j)$，令$\omega_{ijr}=+\infty$，否则令$\omega_{ijr}=\omega_{ij}$；

（2）对第r个子问题运行遗传算法，得到第r阶段新开放的设施集合

$$N_r=\{i\in I\setminus F_{r-1} \mid y_i=1\}，且有 F_r=F_{r-1}\cup N_r，$$

以及第r层值为 1 的分配决策变量集合

$$X_r=\{x_{ijr} \mid x_{ijr}=1, \forall j\in J, i\in F_r\setminus F_{r-1}(j)\}；$$

（3）对每个$j\in J$，求

$$F_r(j)=F_{r-1}(j)\cup\{i\in F_r\setminus F_{r-1}(j) \mid x_{ijr}=1\}；$$

$$W_r(j)=W_{r-1}(j)\cup\{\omega_{ij} \mid x_{ijr}=1, i\in F_r\setminus F_{r-1}(j)\}。$$

（4）若$r<K$，则令$r=r+1$，返回（1）；否则转 Step3。

Step3：对每个 $j \in J$，将 $W_K(j)$ 中的元素按从小到大的顺序进行排序，记为 $\omega_{i_{j_1} j} \le \omega_{i_{j_2} j} \le \ldots \le \omega_{i_{j_K} j}$，其中 $i_{j_r} \in F_K(j)\,(r = 1, 2, \cdots, K)$。令 $x_{i_{j_r} jr} = 1$，其中 $i_{j_r} \in F_K(j)\,(r = 1, 2, \cdots, K)$。

Step4：则最终开放的设施集合为 F_K，需求点 j 的第 r 层连接到开放的设施 i_{j_r} 上，其中 $i_{j_r} \in F_K(j)\,(r = 1, 2, \cdots, K)$。

4.4 算法的分析论证

（LP1）的线性松弛问题表示为：

(LP3)

$$
\begin{aligned}
\min \quad & \sum_{i\in I}\sum_{j\in J}(1-q)\omega_{ij}\sum_{r=1}^{K} q^{r-1}x_{ijr} + \sum_{i\in I} f_i y_i \\
s.t. \quad & \sum_{i\in I} x_{ijr} = 1, \quad \forall j \in J, r = 1, 2, \cdots, K \\
& y_i - \sum_{r=1}^{K} x_{ijr} \ge 0, \quad \forall i \in I, j \in J \\
& -y_i \ge -1, \quad \forall i \in I \\
& x_{ijr}, y_i \ge 0, \quad \forall i \in I, j \in J, r = 1, 2, \cdots, K
\end{aligned}
$$

(LP3)的对偶规划为：

(LP4)

$$
\begin{aligned}
\max \quad & \sum_{j\in J}\sum_{r=1}^{K}\alpha_{jr} - \sum_{i\in I}\beta_i \\
s.t. \quad & \alpha_{jr} - \gamma_{ij} \le (1-q)q^{r-1}\omega_{ij}, \quad \forall i \in I, j \in J, r = 1, 2, \cdots, K \\
& \sum_{j\in J}\gamma_{ij} - \beta_i \le f_i, \quad \forall i \in I \\
& \alpha_{jr}, \beta_i, \gamma_{ij} \ge 0, \quad \forall i \in I, j \in J, r = 1, 2, \cdots, K
\end{aligned}
$$

(LP2)的线性松弛问题为：

(LP5)

$$
\begin{aligned}
\min \quad & \sum_{i\in I}\sum_{j\in J}(1-q)\omega_{ijr}q^{r-1}x_{ijr} + \sum_{i\in I\setminus F_{r-1}} f_i y_i \\
s.t. \quad & \sum_{i\in I\setminus F_{r-1}(j)} x_{ijr} = 1, \quad \forall j \in J \\
& y_i - x_{ijr} \ge 0, \quad \forall i \in I \setminus F_{r-1}, j \in J \\
& x_{ijr}, y_i \ge 0, \quad \forall i \in I, j \in J
\end{aligned}
$$

相应的对偶规划为：

(LP6)

$$\max \quad \sum_{j\in J}\alpha_{jr}$$

$$s.t. \quad \alpha_{jr}-\gamma_{ij}\le(1-q)q^{r-1}\omega_{ij}, \quad \forall j\in J, i\in I\setminus F_{r-1}$$

$$\alpha_{jr}\le(1-q)q^{r-1}\omega_{ij}, \quad \forall j\in J, i\in F_{r-1}$$

$$\sum_{j\in J}\gamma_{ij}\le f_i, \quad \forall i\in I\setminus F_{r-1}$$

$$\alpha_{jr}\ge 0, \gamma_{ij}\ge 0, \quad \forall i\in I, j\in J$$

令(LP1) 的最优目标函数值为OPT，(LP3)的最优目标函数值为OPT_S，(LP5)的最优目标函数值为OPT_r。则有如下定理：

定理 4.1 $OPT_r\le OPT_S/(K+1-r)$。

证：由线性规划对偶定理，(LP6)也有最优目标函数值OPT_r，设（α，β）为(LP6)的可行解，通过把该可行解扩展为具有目标函数值（$K+1-r$）$\cdot OPT_r$的(LP4)的可行解来证明该定理。首先进行如下赋值：

（1）$\forall j\in J, r=1,\cdots,K, \alpha_{jr}=\alpha_j$；（2）$\forall j\in J, i\in I\setminus F_{r-1}, \gamma_{ij}=\gamma_{ij}$；

（3）$\forall j\in J, i\in F_{r-1}(j), \gamma_{ij}=\alpha_j$；（4）$\forall i\in F_{r-1}, \beta_i=\sum_{j\in J}\gamma_{ij}$。

令(α,γ,β)为扩展后的解，容易验证其为(LP4)的可行解，(LP4)的目标函数值为：

$$\begin{aligned}\sum_{j\in J}\sum_{r=1}^{K}\alpha_{jr}-\sum_{i\in I}\beta_i&=\sum_{j\in J}K\alpha_j-\sum_{i\in I}\sum_{j\in J}\gamma_{ij}=K\sum_{j\in J}\alpha_j-\sum_{j\in J}\sum_{i\in F_{r-1}(j)}\gamma_{ij}\\&=K\sum_{j\in J}\alpha_j-\sum_{j\in J}\sum_{i\in F_{r-1}(j)}\alpha_j \quad =(K+1-r)\sum_{j\in J}\alpha_j\\&=(K+1-r)\sum_{j\in J}\alpha_{jr}=(K+1-r)\cdot OPT_r\end{aligned}$$

结合对偶定理，有

$$OPT_r=(\sum_{j\in J}\sum_{r=1}^{K}\alpha_{jr}-\sum_{i\in I}\beta_i)/(K+1-r)\le OPT_S/(K+1-r)$$ 证毕。

定理 4.2 $\sum_{r=1}^{K}OPT_r\le H_K\cdot OPT_S$，其中$H_K=1+1/2+1/3+\cdots+1/K$。

证：由定理 4.1，有 $OPT_r \le OPT_S/(K+1-r)$，故

$$\sum_{r=1}^{K} OPT_r \le (1+1/2+1/3+\cdots+1/K)\cdot OPT_S = H_K \cdot OPT_S$$ 证毕。

推论 4.1　算法经过 K 个阶段得到的每个子问题的最优目标函数值之和不会超过 $H_K \cdot OPT$，即 $\sum_{r=1}^{K} OPT_r \le H_K \cdot OPT$。

证：因为最小化整数线性规划的最优目标函数值大于等于其松弛线性规划的最优目标函数值，即有 $OPT_S \le OPT$，故有 $\sum_{r=1}^{K} OPT_r \le H_K \cdot OPT$。　证毕。

由上述定理及推论，分阶段近似算法能得到原问题近似度为 H_K 的近似最优解。

4.5 算例

例 4.1　数据见表 4-1

表 4-1 一个算例

Table 4-1 an example

需求点 j \ ω_{ij} \ 设施 i	1	2	3	4	5
1	2	1	2	4	3
2	3	3	1	2	4
3	1	2	4	3	1
4	4	1	3	3	2
5	3	4	2	1	3
6	2	4	3	1	4
开放费用	$f_1=5$	$f_2=2$	$f_3=5$	$f_4=3$	$f_5=4$

由表 4-1 知，该问题有 6 个需求点，可供选择开放的设施有 5 个，即 $|J|=6, |I|=5$。设 $q=0.2$，$K=3$。算法求解过程如下：

（1）当 $r=1$ 时，$F_0=\phi$，$X_0=\phi$，对每个 $j \in J$，$F_0(j)=\phi$，$W_0(j)=\phi$。（2）对第

一个子问题运行遗传算法，得到第一阶段新开放的设施集合

$N_1=\{2,4\}$； $F_1=\{2,4\}$， $X_1=\{x_{211},x_{421},x_{231},x_{241},x_{451},x_{461}\}$；

$F_1(1)=\{2\}$， $F_1(2)=\{4\}$， $F_1(3)=\{2\}$， $F_1(4)=\{2\}$， $F_1(5)=\{4\}$， $F_1(6)=\{4\}$；

$W_1(1)=\{\omega_{21}\}$， $W_1(2)=\{\omega_{42}\}$， $W_1(3)=\{\omega_{23}\}$， $W_1(4)=\{\omega_{24}\}$， $W_1(5)=\{\omega_{45}\}$， $W_1(6)=\{\omega_{46}\}$。

（3）因为 $r<3$，令 $r=2$。

（4）对第二个子问题运行遗传算法，得到第二阶段新开放的设施集合 $N_2=\phi$；

$F_2=F_1\cup N_2=\{2,4\}$， $X_2=\{x_{412},x_{222},x_{432},x_{442},x_{252},x_{262}\}$，

$F_2(1)=F_2(2)=F_2(3)=F_2(4)=F_2(5)=F_2(6)=\{2,4\}$；

$W_2(1)=W_1(1)\cup\{\omega_{41}\}=\{\omega_{21},\ \omega_{41}\}$， $W_2(2)=W_1(2)\cup\{\omega_{22}\}=\{\omega_{42},\ \omega_{22}\}$，

$W_2(3)=W_1(3)\cup\{\omega_{43}\}=\{\omega_{23},\ \omega_{43}\}$， $W_2(4)=W_1(4)\cup\{\omega_{44}\}=\{\omega_{24},\ \omega_{44}\}$，

$W_2(5)=W_1(5)\cup\{\omega_{25}\}=\{\omega_{45},\ \omega_{25}\}$， $W_2(6)=W_1(6)\cup\{\omega_{26}\}=\{\omega_{46},\ \omega_{26}\}$。

（5）因为 $r<3$，令 $r=3$。

（6）对第三个子问题运行遗传算法，得到第三阶段新开放的设施集合 $N_3=\{5\}$；

$F_3=F_2\cup N_3=\{2,4,5\}$， $X_3=\{x_{513},x_{523},x_{533},x_{543},x_{553},x_{563}\}$；

$F_3(1)=F_3(2)=F_3(3)=F_3(4)=F_3(5)=F_3(6)=\{2,4,5\}$；

$W_3(1)=W_2(1)\cup\{\omega_{51}\}=\{\omega_{21},\ \omega_{41},\ \omega_{51}\}$，

$W_3(2)=W_2(2)\cup\{\omega_{52}\}=\{\omega_{42},\ \omega_{22},\ \omega_{52}\}$，

$W_3(3)=W_2(3)\cup\{\omega_{53}\}=\{\omega_{23},\ \omega_{43},\ \omega_{53}\}$，

$W_3(4)=W_2(4)\cup\{\omega_{54}\}=\{\omega_{24},\ \omega_{44},\ \omega_{54}\}$，

$W_3(5)=W_2(5)\cup\{\omega_{55}\}=\{\omega_{45},\ \omega_{25},\ \omega_{55}\}$，

$W_3(6)=W_2(6)\cup\{\omega_{56}\}=\{\omega_{46},\ \omega_{26},\ \omega_{56}\}$。

（7）因为 $r=3$，问题的三个阶段都处理完了，最终开放的设施集合为 $F_3=\{2,4,5\}$，6 个需求点与 $F_3=\{2,4,5\}$ 中的设施层次连接关系为：

$$x_{211}=x_{412}=x_{513}=1，\ x_{421}=x_{222}=x_{523}=1，\ x_{231}=x_{432}=x_{533}=1，$$

$$x_{241}=x_{442}=x_{543}=1，\ x_{451}=x_{252}=x_{553}=1，\ x_{461}=x_{262}=x_{563}=1，$$

此时目标函数值为 19.304。

（8）下面进入 6 个需求点与开放的设施集合 $F_3=\{2,4,5\}$ 连接层次的重新分配阶段。分别对 $W_3(1)$、$W_3(2)$、$W_3(3)$、$W_3(4)$、$W_3(5)$、$W_3(6)$ 中的元素按从小到大的顺序进行排列，分别为：

$$\omega_{21}<\omega_{51}<\omega_{41}，\ \omega_{42}<\omega_{22}<\omega_{52}，\ \omega_{53}<\omega_{23}<\omega_{43}，$$

$$\omega_{24}<\omega_{54}<\omega_{44}，\ \omega_{45}<\omega_{55}<\omega_{25}，\ \omega_{46}<\omega_{26}\le\omega_{56}。$$

因此得

$$x_{211}=x_{512}=x_{413}=1，\ x_{421}=x_{222}=x_{523}=1，\ x_{531}=x_{232}=x_{433}=1，$$

$$x_{241}=x_{542}=x_{443}=1，\ x_{451}=x_{552}=x_{253}=1，\ x_{461}=x_{262}=x_{563}=1。$$

（9）则原问题的最优解可表示为：开放的设施集合为 $F_3=\{2,4,5\}$；需求点 1 的第一、二、三层分别连接到设施 2、5、4，需求点 2 的第一、二、三层分别连接到设施 4、2、5，需求点 3 的第一、二、三层分别连接到设施 5、2、4，需求点 4 的第一、二、三层分别连接到设施 2、5、4，需求点 5 的第一、二、三层分别连接到设施 4、5、2，需求点 6 的第一、二、三层分别连接到设施 4、2、5；此时的目标函数值为 18.024，相对于重新分配前的目标函数值更小。

4.6 本章小结

本章主要介绍了一类新的设施选址问题，即可靠性设施选址问题，首先简单介绍可靠性设施选址，并给出了国外的其研究现状。鉴于目前的研究情况，本文提出了一种新的可靠性设施选址问题，即无容量限制的可靠性设施选址问题，该问题的模型与已有的可靠性设施选址模型都不同，所有设施都以常数概率出现故障，并将每个需求点分配给多个设施，以提高整个供求系统的可靠性。针对该问题，提出了基于遗传算法的分阶段近似算法，该算法的主要原理是将原模型分解为多个阶段，然后分阶段处理，所有阶段都处理完毕后，原问题也得到了求解。通过引进容错性设施选址问题中的分层算法证明

思想，证明了基于遗传算法的分阶段近似算法对求解 RUFLP 是可行的。为了进一步说明算法的可行性，给出了一个小规模的算例，进一步说明了基于遗传算法的分阶段近似算法对求解 RUFLP 是有效的。

5 总结与展望

论文介绍了设施选址问题在现代社会生活中的作用、发展进程以及一些经典的设施选址问题和求解算法，并对传统的遗传算法进行了阐述。论文主要研究了两类设施选址问题，一类是经典的无容量限制设施选址问题（Uncapacitated Facility Location Problem，简称 UFLP）；另一类为无容量限制的可靠性设施选址问题（Reliability for the Uncapacitated Facility Location Problem，简称 RUFLP）。

针对 UFLP，提出了一种改进的遗传算法，该算法较之传统的遗传算法有如下区别：采用了具有方向性的初始群体生成法，该方法的优点是生成的初始解质量较高，更易于算法的进化搜索；选择算子采用的是锦标赛选择法，该方法的优点是选择概率容易控制，适合求解大规模的选址问题；交叉与变异算子都运用自适应思想将其设计成进化代数t的函数，能随着算法的进化不断进行调整，使算法具有自我调整的性能，更符合生物进化自我调整的特征。通过数值试验将本文的自适应 GA 与其他几种常用求解 UFLP 的算法进行比较分析，说明了本文的自适应 GA 更适合求解大规模的 UFLP。

论文研究的另一内容是可靠性设施选址问题，提出了一种无容量限制的可靠性设施选址问题（即 UFLP）。该模型与已有的可靠性设施选址模型不同，建设所有设施都以常数概率出现故障，并将每个需求点分配给多个设施，以提高整个供求系统的可靠性。针对 RUFLP 提出了基于 GA 的分阶段近似算法，该算法的主要原理是将原模型分解为多个阶段，然后分阶段处理，所有阶段都处理完毕后，原问题也得到了求解。通过引进容错性设施选址问题中的分层算法证明思想，证明了该算法能得到问题的近似度为$H_K(H_K=1+1/2+\cdots+1/K)$的最优解，并通过一个小规模的算例，验证了所给算法对求解 RUFLP 是有效的。

本文的创新点：

针对经典的无容量限制的设施选址问题，提出了一种改进的自适应遗传算法，该算法与已有的遗传算法有如下区别：（1）不再选择目标函数为适应度函数，而是令与所求问题有关的较大常数与目标函数的差为适应度函数，该方法的优点是个体的适应度值都是正数，这为后续遗传操作的设计提供了更广泛的选择。（2）对交叉概率p_c的自适应设计进行了改进，将其设计成与进化代数t相关的函数，使其随着算法迭代过程不断更新，

满足不同进化阶段对交叉算子的需求；并将自适应交叉概率与均匀交叉法相结合，组成自适应均匀交叉算子，该交叉算子不仅更有针对性，而且具有自适应性，从而提高进化效率。（3）对变异概率 p_m 进行了自适应设计，其设计方法与前述自适应交叉概率相似，因此两者可以很好地结合，更好地发挥自适应作用。

针对可靠性设施选址问题提出了一种新的模型，该模型以无容量限制的设施选址问题为基础，称之为无容量限制的可靠性设施选址模型。在无容量限制的可靠性设施选址模型中，假设所有设施都会以固定概率出现故障，并使每个需求点都有多个设施来满足需求。针对无容量限制的可靠性设施选址模型提出了基于遗传算法的分阶段近似算法，该算法将原问题分为多个阶段依次处理，每个阶段对应一个 UFLP 子问题，所有子问题都处理完毕后，再进行一个特殊处理便得到原问题的近似最优解。通过理论及算例证明了该算法对求解 RUFLP 是可行的，并且具有常数近似度，相对于已有的求解可靠性设施选址问题的算法不能得到常数近似度的最优解是一个进步。在求解子问题时使用了遗传算法，由于 GA 在求解大规模 UFLP 时具有较好的，因此基于遗传算法的分阶段近似算法可以克服拉格朗日松弛方法不能求解大规模可靠性设施选址问题的困难。

设施选址问题在很多领域都有着广泛的应用，多年来一直备受运筹学和计算科学领域研究者的重视，是国内外优化领域的热点课题，尤其可靠性设施选址问题的研究才刚起步，模型的种类和求解算法还很少，因此对此类问题的研究还有很大空间。对于以后的研究内容可以考虑以下几个方面：（1）应用自适应遗传算法求解 CFLP、p-中位、p-中心等问题；（2）将遗传算法与粒子群优化算法相结合，并将其应用到 UFLP 中；（3）可以增加设施的容量限制，建立带容量限制的可靠性设施选址模型；（4）将本文提出的分阶段思想应用到多级选址等其它的选址模型中；（5）随着网络时代的到来与发展，现实生活中的选址问题不仅规模大，而且各个设施之间的联系更加紧密，其中一个设施出现故障，有可能会影响整个网络，因此研究大规模的可靠性设施选址问题及其求解算法具有很大的现实意义。

致 谢

首先感谢我的导师赵茂先教授，本文是在导师的悉心指导和热情关怀下完成的。导师给我提供了大量的相关文献和良好建议，从论文的选题，撰写到最后的定稿，都得到了他的精心指导。赵老师严谨的治学态度和认真负责的工作精神使我受益终生。在此向老师表示衷心的感谢！

衷心感谢贺国平教授、周长银副教授、王永丽老师、段华老师、韩丛英老师，感谢他们在学术上对我的指导和生活上对我的关心！

衷心感谢运筹所的李翼、付三平、陈海阳、于静、段玉涛、李婷贤、李明强等兄弟姐妹们，在学习上与他们的讨论和交流，使我受益颇多，同样在生活中他们也给予了我莫大的支持和帮助！

衷心感谢我的父母和亲人，感谢他们对我的支持和理解，支撑着我完成学业！

最后再次感谢所有帮助过我，关心过我的老师、同学和朋友！

参考文献

1. Weber A. Theory of the Location of Industries [M]. Chicago: The University of Chicago Press, 1909.
2. Drezner Z, Hamacher H W. Facility Location Applications and Theory [M]. New York: Springer, 2002.
3. Hakimi S. Optimum location of switching center and the absolute center and medians of a graph [J]. Operations Research, 1964, 12:450-459.
4. Hakimi S. Optimum location of switching center in a communications network and some related graph theoretic problem [J]. Operations Research, 1965, 13:462-475.
5. Balinski M L. Integer Programming: methods, uses, computation [J]. Mangement Science, 1965, 12:253-313.
6. Church R L, ReVelle C S. The maximal covering location problem [J]. Papers of the Regional Science Association, 1974, 32:101-118.
7. Toregas C, Swain R, Revelle C, Bergeman L. The location of emergency service facilities [J]. Operations Research, 1971, 19:1363-1373.
8. Manne A S. Capacity expansion and probabilistic growth [J]. Econometrica, 1961, 29:632-649.
9. Ballou R H. Dynamic warehouse location analysis [J]. Journal of Marketing Research, 1968:271-276.
10. Drezner Z, Wesolowsky G O. Facility location when demand is time dependent [J]. Naval Research Logistics, 1991, 38:763-777.
11. Drezner T. Optimal continuous location of a Retail facility, facility attractiveness, and market share: an interactive model [J]. Journal of Rrtail, 1994, 70:49-64.
12. Klose A, Drexl A. Facility location models for distribution system design [J]. European Journal of Operational Research, 2005, 162:4-29.
13. Lawler E L, Wood D E. Branch-and-Bound Methods: A Survey [J]. Operation Research. 1966, 14:699-719.
14. 邢文训，谢金星．现代优化计算方法（第二版）[M]．清华大学出版社, 2001, 210-242.

15. Bagley J D. The behavior of adaptive systems which employ genetic and correlation algorithms [J]. 1967.
16. Eberhart R C, Kennedy J. A New Optimizer Using Particle Swarm Theory. In Proceedings of the Sixth International Symposium on Micro Machine and Human Science, Nagoya Japan, 1995, 39-43.
17. Glover F. Tabu search: partI [J]. ORSA Journal on Computing, 1989, 1(3):190-206.
18. Glover F. Tabu search: partII [J]. ORSA Journal on Computing, 1990, 2(1):4-32.
19. Kirkpatrick S, Gelatt C D, Vecchi M P. Optimization by simulated annealing [J]. Science, 1983, 671-680.
20. 加里，约翰逊．计算机和难解性-NP 完全性理论导引[M]．北京：科学出版社, 1987.
21. Holland J. Adaptation in Natural and Artificial Systems [M]. Michigan: Ann Arbor MI University of Michigan Press, 1975.
22. Blickle T, Thiele L. A Mathematical Analysis of Tournament Selection [J]. Genetic Algorithms: Proceedings of the 6th International Conference (ICGA95), Morgan Kaufmann, San Francisco, CA, 1995.
23. Miller B L, Goldberg D E. Genetic Algorithms, Tournament Selection, and the Effects of Noise [J]. Department of General Engineering, 1995.
24. Beasley J E, Chu P C. A genetic algorithm for the set covering problem [J]. European Journal of Operational Research, 1996, 94:392-404.
25. 杨平，郑金华．遗传选择算子的比较与研究[J]．计算机工程与应用，2007, 43(15):59-62.
26. Kuehn A A, Hamburger M J. A heuristic program for locating warehouses [J]. Management Science, 1963, 9:643-666.
27. Manne A S. Plant location under economies-of-scale-decentralization and computation [J]. Management Science, 1964, 11:213-235.
28. Stollsteimer J F. A working model for plent number and location [J]. Journal of Farm Economics, 1963, 45:631-645.
29. Chvatal V. A greedy heuristic for the set covering problem [J]. Mathematics of Operations Research, 1979, 4:233-235.
30. Edmonds J. Minimum partition of a matroid into independent sets [J]. Journal of Research,

1965, 869:67-72.

31. Feldman E, Lehrer F A, Ray T L. Warehouse locations under continuous economies of scale [J]. Management Science, 1966, 12:670-684.
32. Graham R L. Bounds for certain multiprocessing anomalies [J]. Bell System Technical Journal, 1966, 45:1563:1581.
33. Kaufman L, Eede M V, Hansen P. A plant and warehouse location Problem [J]. Operations Research Quarterly, 1977, 28:547-557.
34. Nauss R M. An improved algorithm for the capacitated facility Location problem [J]. Journal of Operational Research Society, 1978, 29:1195-1202.
35. Shapley L S. On balanced sets and cores [J]. Naval Research Logistics Quarterly, 1967, 14:453-460.
36. Mahdian M, Markakis E, Saberi A, Vazirani V. A Greedy Facility Location Algorithm Analyzed Using Dual Fitting [J]. Lecture Notes in Computer Science, 2001, 2129:127-137.
37. Woeginger G J. Exact Algorithms for NP-Hard Problems: A Survey [J]. Lecture Notes in Computer Science, 2003, 2570:185-207.
38. Ghosh D. Neighborhood search heuristic for the uncapacitated facility location problem [J]. European Journal of Operational Research, 2003, 150(1):150-162.
39. Shmoys D B, Tardos E, Aardal K I. Approximation Algorithms for Facility Location Problems [M]. Proceedings of the 29th ACM Symposium on Theory of Computing. 1997:265-274.
40. Guha S, Kuller S. Greedy Strikes Back: Improved Facility Location Algorithms [J]. Journal of Algorithms, 1999, 31:228-248.
41. Hosage C M, Goodchild M F. Discrete space location-allocation solutions for genetic algorithms [J]. Annals of Operations Research, 1986, 6:35-46.
42. Kratica J, Tosic D, Filipovic V. Solving the uncapacitated warehouse location problem by SGA with Add-Heuristic. Proceeding XV ECPU International Conference on Materia Handling and Warehousing, 1998.
43. Kratica J, Tosi B D, Filipovi B V, Ljubic I. Solving the simple plant location problem by genetic algorithm [J].RAIRO Operations, 2001, 35:127-142.

44. Jaramillo J H, Bhadury J, Batta R. On the use of genetic algorithms to solve location problems [J]. Computers & Operations Research 2002, 29:761-779.
45. Tohyama H, Matsueda J. A Genetic Algorithm for the Uncapacitated Facility Location Problem [J]. Electronics and Communications in Japan, 2011, 94:47-54.
46. DeJong K A. Adaptive system design: a genetic approach [J]. Systems, Man and Cybernitics, IEEE Transactions on, 1980, 9:566-574.
47. Srinivas M, Patnaik L M. Adaptive probabilities of crossover and mutation in genetic algorithms [J]. Systems, Man and Cybernitics, IEEE Transactions on, 1994, 24:656-667.
48. Hinterding R, Michalewicz Z, Peachey T C. Self-adaptive genetic algorithm for numeric functions [J]. Lecture Notes in Computer Science, 1996, 1141:420-429.
49. Smith J E, Fogarty T C. Operator and parameter adaptation in genetic algorithms [J]. 1997, 1:81-87.
50. Serpell M, Smith J E. Self-Adaptation of Mutation Operator and Probability for Permutation Representations in Genetic Algorithms [J]. 2010, 18(3):491-514.
51. 刘姝廷，金太东，王连生. 一种改进的自适应遗传算法[J]. 江西理工大学学报, 2010.
52. Michael K A. Generalized crossover operation for genetic algorithms [J]. Complex Systems, 1995, 9:177-191.
53. Beasley J E. OR-Library, http://people.brunel.ac.uk/~mastjjb/job/info.html.
54. Hoefer M. Performance of heuristic and approximation algorithms for the uncapacitated facility location problem. Forschungsbericht Research Report, MIPI-1-2002-1-005, 2002.
55. Michel L, Hentenryck P V. A simple tabu search for warehouse location [J]. European Journal of Operational Research, 2004, 157:576–591.
56. Aydin M E, Fogarty T C. A distributed evolutionary simulated annealing algorithm for combinatorial optimization problems [J]. Journal of Heuristics, 2004, 10(3):269–292.
57. Snyder L V, Daskin M S. Reliability models for facility location: The expected failure cost case [J]. Transport Science, 2005, 39(3):400-416.
58. Lim M, Daskin M S, Bassamboo A, Chopra S. A facility reliability problem: formulation, properties, and algorithm [J]. Naval Research Logistics, 2009, 57(1):58-70.
59. Cui T, Ouyang Y, Shen Z J M. Reliable facility location design under the risk of disruptions [J]. Working Paper, University of California at Berkeley, 2010.

60. Jain K, Vazirani V V. An approximation algorithm for the fault tolerant metric facility location problem [J]. Algorithmica, 2004, 38:433-439.

61. Tcha D, Lee B. A branch-and-bound algorithm for the multi-level uncapacitated facility location problem [J]. European Journal of Operational Research, 1984, 18:35-43.

62. Xu G, Xu J H. An improved approximation algorithm for uncapacitated facility location problem with penalties [J]. Journal of Combinatorial Optimization, 2009, 17:424-436.

63. Sun M H. Solving the uncapacitated facility location problem using tabu search [J]. Computer & Operations Research, 2006, 33:2563-2589.

64. Kratica J, Filipovic V, Sesum V. Solving the uncapacitated warehouse location problem using a simple genetic algorithm [J]. Proceeding XIV International Conference on Materia Handling and Warehousing, 1996.

攻读硕士期间主要成果

1. 李岳佳. 改进的自适应遗传算法[J]. 山东科技大学学报（自然科学版增刊）（已录用），2012.
2. 李翼，赵茂先，李岳佳. 无容量限制设施选址问题的分支定界算法[J]. 山东理工大学学报（自然科学版）, 2012, 26(1):70-73.
3. 李岳佳. 可靠性设施选址问题的求解算法，已投 数学的实践与认识