学士学位论文

基于进化算法的车辆路径问题求解初探

学　　号：　**20161003592**

姓　　名：　李瑞

学 科 专 业：　信息安全

指 导 教 师：　龚文引　教授

培 养 单 位：　计算机学院

二〇二〇年四月

中国地质大学（武汉）学士学位论文原创性声明

本人郑重声明：本人所呈交的学士学位论文《基于进化算法的车辆路径问题求解初探》，是本人在指导老师的指导下，在中国地质大学（武汉）攻读学士学位期间独立进行研究工作所取得的成果。论文中除已注明部分外不包含他人已发表或撰写过的研究成果，对论文的完成提供过帮助的有关人员已在文中说明并致以谢意。

本人所呈交的学士学位论文没有违反学术道德和学术规范，没有侵权行为，并愿意承担由此而产生的法律责任和法律后果。

学位论文作者签名：_____

日　期：　　年　　月　　日

摘　要

车辆路径规划的求解在现实生活中有着非常重要的意义，如物流配送，道路洒水等。而基于容量的车辆路径问题（CARP）是一个 NP 难问题。采用穷举搜索无法解决。现在有很多可求解 CARP 的方法如数值构造法，启发式算法等。但它们依赖初值的特点也很明显。1975 年由 Holland 教授提出遗传算法。差分进化算法 1995 年问世。其优秀的全局搜索能力得到了广泛的认可。

本文介绍了一种基于协同进化的算法 RDG-MAENS，它在 MAENS 的基础上加入了协同进化的框架，同时重点讲述了分解策略 route distance grouping（RDG），为了测试其性能同时也介绍了 MAENS 以及 RTS。扩展邻域的文化遗传算法主要在于提出了 merge-split（MS）操作增加了步长，更能收敛到全局最优解。RTS 的关键在于 global repair option(GRO)全局修复操作，以及如何嵌入到 TSA 中。这三种算法各具特点。

本文的主要工作在于将两种基于进化算法的求解算法与经典启发式算法进行了对比，在四个数据集上进行实验，分析其优缺点。

通过实验得出 RDG-MAENS 的精度比 MAENS 更好，RTS 收敛速度更快且在中型数据集的效果比前两者要好。RDG-MAENS 在分解策略存在缺陷，根据分解得到的解容易陷入局部最优。且没有利用不可行解。存在很大改进空间

关键词：车辆路径问题；CARP；RTS；MAENS；RDG-MAENS。

Abstract

The vehicle routing problem is very important in real life, such as logistics distribution, road sprinkling and so on. The volume-based vehicle path problem (CARP) itself is NP hard. Exhaustive search is not an option. There are many ways to solve CARP such as numerical constructs, heuristic algorithms, and so on. But their dependence on initial values is also obvious. Genetic algorithm was proposed by professor Holland in 1975. The differential evolution algorithm came into being in 1995. Its excellent global search ability has been widely recognized.

This paper introduces an algorithm based on co-evolution, RDG-MAENS, which adds the framework of co-evolution on the basis of MAENS, and focuses on the decomposition strategy route distance grouping (RDG). In order to test its performance, it also introduces MAENS and RTS. The cultural genetic algorithm for extending neighborhood mainly proposes merge-split (MS) operation, which increases the step size and converges to the global optimal solution. The key to RTS is the global repair option (GRO) global repair operation and how it is embedded in the TSA. These three algorithms have their own characteristics.

The main work of this paper is to compare the two solving algorithms based on evolutionary algorithm with the classical heuristic algorithm, and carry out experiments on four data sets to analyze their advantages and disadvantages.

The experimental results show that the accuracy of RDG-MAENS is better than that of MAENS, the convergence rate of RTS is faster and the effect in medium data sets is better than the former two. Rdg-maens has some defects in the decomposition strategy, and the solutions obtained from the decomposition are easy to fall into local optimality. And no infeasible solution is used. There is much room for improvement.

Key words: vehicle routing problem; CARP; RTS; MAENS; RDG - MAENS

目 录

第一章 引言

1.1 国内外研究现状

车辆路径规划问题[1]（Vehcile Routing Problem）是一个计算优化问题，即找到一个最优解使得目标函数最小。车辆路径问题首次在 1959 年由 Dantzig 与 Ramser[2] 提出。[2]中描述了一个卡车配送物资的路径优化问题。1981 年 Lenstra and Kan[3] 证明 VRP 是一个 NP 难问题。

CARP 问题在优化调度领域有着非常重要的地位。研究广泛，种类繁多。因为物流配送问题与我们的生活息息相关，覃和毛对于快递配送[4]提出了采用元胞自动机来解决带时间窗的车辆路径问题，赵和范对于紧急物质运输[5]问题构建了车辆路径的模型取得了良好的效果，李对于蔬果配送[6]在车辆路径模型中结合了遗传算法，使得配送的成本大幅降低，从这些研究可以发现，VRP 问题与我们的生活息息相关，很多生活中的问题都可以通过建立 VRP 模型来求解。

影响 VRP 模型的实际可用性的因素有很多，总的来看可以分为两类，一种是静态和动态，一种是确定的和随机的因素。静态的因素在求解的过程中是不变的。而动态的因素却相反。在设计模型时，往往那些不变的因素是常量，或者说是固定的，而变化的是动态的因子，这样的有很多，比如行驶的时间，客户的需求，服务的可能性等。因此 VRP 模型便延伸出许多变种模型。例如带时间窗的车辆路径问题[7]，装货卸货的车辆路径问题[8]，多车场的车辆路径问题[9]，开放式的车辆路径问题[10]，拆分配送的车辆路径问题[11]，基于容量的车辆路径问题[12]。

经典 VRP 问题有很多求解方法，数值型和启发式两种。张使用近似动态规划的方法求解车辆路径问题[13]，余和李提出了结合新插入法的禁忌搜索算法求解车辆路径问题[14]等。启发式有禁忌搜索[15]，李和刘将经典的模拟退火[16]算法用于车辆路径问题取得了良好的效果，人工蚁群，改进蚁群[17]，免疫算法[18]，蝙蝠算法[19]，以及混合算法[20]，遗传算法[21]等。随着进化算法[22]的出现，人们关注能否使用智能优化的方法求解 VRP 问题。

在随后的发展里，人们生活的需求也发生了许多变化，随之而来的是带来了新的问题，这些问题也被研究 VRP 的学者们所关注，因此 VRP 开始演化出各种新颖的模型。如 Pillac 和 Gendreau 对于近年来动态 VRP[23]中出现的变量因素做出了

回顾，其中的影响因素也是越来越多，Choy 和 Lin 认为在车辆运输的过程中应该考虑节能减排的问题，加入环保的考虑因素因此提出绿色 VRP[24]，Vidal 和 Crainic 任务在研究模型时不应该固定其他因素，并只考虑一种影响因素，应该多方位的考虑多因素带来的影响，因此提出了多属性 VRP[25]，Ponce 和 Hui 认为社区垃圾的增多，使得垃圾回收车辆数量增加，而如何规划好行车路线成了生活中的迫切需求，因此提出了回收垃圾的 VRP[26]，Gitae 和 Kim 认为如今城市中道路拥堵的问题急需解决，规划城市中海量车辆的出行路线是非常重要的，于是提出了城市车辆 VRP[27]，Ann 和 Gill 认为如果全程都需要规划好形式路径，这样的模型显得僵硬没有变化，而很多时候，人们的出行是不需要规划的，只是当遇到复杂的路况时，规划操作才会起到重要的作用，增加了其使用效率，提出了时段性计算的 VRP[28]等。

基于容量的车辆路径问题：车辆的行驶过程中，有负载的限额，所有车辆的负载都不能超过其限额 Q，限额 Q 远小于所有用户的需求之和。

基于时间窗的车辆路径问题：车辆在完成任务有时间限制，如果完成任务的时间超过了时间限制，则认为这是一个不可行解，于是会生成新的解。这属于硬时间窗。而当车辆的时间超过规定时间后，增加惩罚因子，则为软时间窗。

开放式的车辆路径问题：车辆没有硬性规定必须会到车场，也可以以其他路径为终止结点。但是其他模型中车辆都是从车场出发并回到车场。

多车场车辆路径问题：常规的模型中只有一个车场，所有车辆从这里出发并回到这里，而多车场问题中，可以同时有多个车场，根据任务就近完成，有并行的特点。

以上介绍了经典的 VRP 模型，下面将简单描述一下进化算法的由来。

遗传算法结合生物进化的知识提出，在生物进化论中达尔文认为，物种总是朝着越来越适应于环境的方向进化，受此启发。在 1975 年 Holland 提出遗传算法（Genetic Algorithms,GA）。受遗传算法启发，1995 年 Storn 和 Price[29-30]提出差分进化算法(Differential Evolution，DE)，用于求解实数优化问题。该算法是一种基于群体的自适应全局优化算法，属于演化算法的一种，具有容易实现、结构简单、收敛快速、鲁棒性强等特点。遗传算法已经有很多学者将其应用到 VRP 问题中，而进化算法作为更加优秀的启发式算法，如何将其用于 VRP 问题成为本文关注的焦点。

1.2 本文结构安排

本文主要工作在于介绍 RTS、MAENS 和 RDG-MAENS 的思想，以及通过实验来测试三个算法的求解能力。本文结构分为几个部分。

1. 介绍车辆路径问题的发展及其研究的意义。
2. 介绍 CARP 的问题模型以及进化算法的概念;
3. 详细介绍 RTS、MAENS 以及 RDG-MAENS 的核心思想和主要算法。
4. 测试 4 种不同的数据集，每一个算法独立运行相同次数，通过比较得到结果的均值，和边界的差距来判断算法求解能力的差异。

根据实验结果得出，RGD-MAENS 的求解能力优于 MAENS，并且这两个算法在小型数据集上的表现优于 RTS，但 RTS 在中型数据集的表现优于 MAENS 和 RDG-MAENS。这三个算法各有优缺点，在中型数据集上取得的效果都有很大的提升空间。

本论文从进化算法出发，探求不同框架以及数据集下进化算法求解求解带弧的基于容量的车辆路径规划问题的差异，并对比分析这几个经典算法适用的场景以及优缺点。是目前为止没人涉及到的。

第二章 车辆路径规划与进化算法

2.1 VRP 问题模型描述

CARP 全称为基于容量的带弧车辆路径问题，在 CARP 中所研究的数据结构是以图为研究对象，描述为$Graph(vertix, edge, arc)$，vertix 是所有服务点对应的点集合，服务点是网络中的结点，edge 是连接点与点之间的边，边是双向的，可以以 (v1, v2)的顺序经过边 edge1 也可以以 (v2, v1)的顺序经过 edge1，arc 是连接点与点之间的弧，称单向的边为弧，即只能以 (v1, v2)或者 (v2, v1)一种顺序经过边 edge1。这里的边和弧对应的是一个城市中街道的双行道和单行道。

道路由许多边构成$Z_E \subseteq E$。而$Z_A \subseteq A$则对应的是弧集上的任务。在图中所有点中会指定一个点为车场 depot，所有车从车场出发回到车场。形成闭合回路。一般多辆车分别行驶不同的道路。所有任务$Z = Z_E \cup Z_A$都有期望值$d(z) > 0$，表示任务的重要程度。执行每个任务 Z 都会有服务开销$\mathrm{sc}(z) > 0$。在行驶过程中经过不需要的边，会有空车行驶开销$\mathrm{dc}(a,b) > 0$。如果点与点不可达则$\mathrm{dc}(a,b) = \infty$。每个任务的开销由两个部分组成$\mathrm{sc}(l_k, l_{k+1}) \times y_k + dc(l_k, l_{k+1}) \times (1 - y_k)$，$\mathrm{y_k} \in \{0,1\}$当到达任务则为服务开销，否则是空车行驶。所有车的负载为 Q，即 CARP 的容量约束。车辆执行的任务所有期望和不能超过 Q，Q 是远小于图中所有期望和。执行任务途中不会出现卸货和装货同时进行的情况。数学描述如下

该模型的目标函数是使得总开销最小,根据顶点与顶点之间边的开销加上空车行驶的服务开销累加和最小。

$$\min \sum_{\mathrm{k=1}}^{\mathrm{K\text{-}1}} (\mathrm{sc}(v_k, v_{k+1}) \times y_k + dc(v_k, v_{k+1}) \times (1 - y_k)) \tag{2.1}$$

任务点之间的边是完成任务点时需要的边一定是来自需要的边和弧表示所有边的顶点都是在点集 V 中选取的

$$l_k \in E_R \cup A_R \quad v_k \in V \quad k = 1,2,...,K \tag{2.2}$$

一辆车完成的任务量要小于车辆的负载

$$\sum_{\mathrm{k}=r_i}^{r_{i+1}} d(v_k, v_{k+1}) \times y_k \le Q \quad i = 0,1,2,...,m-1 \tag{2.3}$$

在 VRP 模型中所有的解的结构都是由 0 和 id 号组成，例如(0,1,2,3,0,4,5,6,0,7,8,9,0)，解的结构以 0 为分隔符表示每一条路径，如(0,1,2,3,0)是

一条路径(0,4,5,6,0)是一条路径 0 表示车场，所有车从车场出发回到车场构成一个闭合回路。

2.2 进化算法

进化算法[22]是以种群为基本单位，种群中有许多个体。问题的解被抽象成这些种群中的个体，每个体即一个解，这些都是随机产生的初始解。如图 2.1 所示。(1)种群产生许多个初始解。(2)在所有个体中随机选择一些个体进行突变，改变其中解的结构。(3)在所有个体中随机选择进行交叉，使得不同解之间的信息交换，产生新的个体。(4)在产生的新个体中选择适应值比当前全局解更优的个体保留下来，作为新的种群。(5)对于新种群重复上述步骤直到解不再更新或者达到迭代条件就停止。得到的近似解在一定的接受范围之内就认为最优解。这样的启发式算法还有很多。都是模拟现实中动物族群的行为，如蚁群算法、蜂群算法、鱼群算法、粒子群算法。特点是在值域内进行全局搜索，不断缩小选择范围进而逼近最优解的过程。体现出了一定的智能性。关键在于其中选择策略的好坏。表 2.1 介绍了差分进化算法的基本过程。

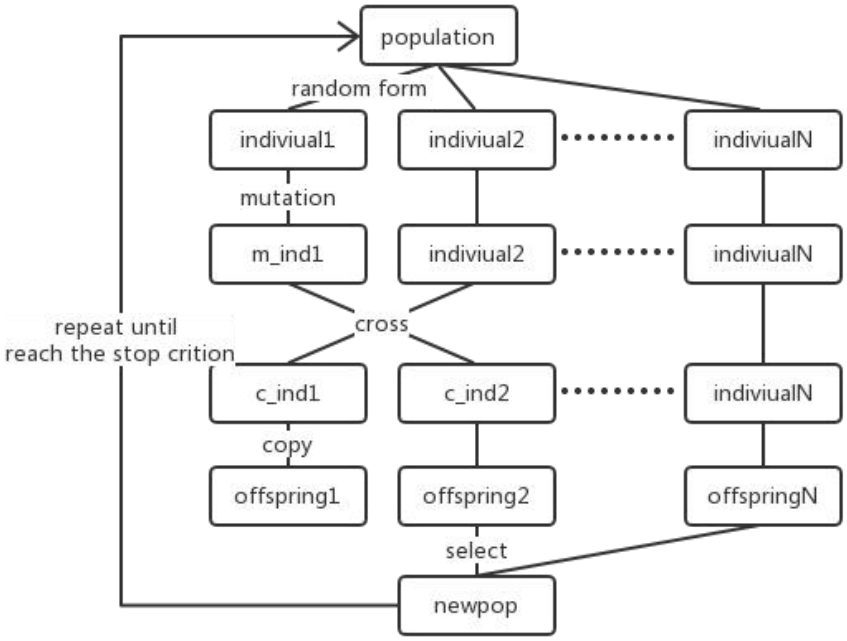

图 2.1 进化算法的基本框架

2.3 扩展邻域的文化遗传算法

2.3.1 文化遗传算法 MA

Memetica Algorithm 又叫文化算法是进化算法结合局部搜索改进而来。该框架许多名称[31]比如 Baldwinian 进化算法、文化算法、基因局部搜索算法等。表 2.1 详细描述其具体过程。

表 2.1 文化算法的详细步骤

步骤 1：产生一个初始种群。

步骤 2：如果没有达到停止迭代的条件则进行步骤 3。

步骤 3：计算种群中的所有个体的适应值。

步骤 4：将种群中所有的个体采用进化算法，产生新的个体。

步骤 5：对于种群中所有的个体以一定的概率 P 采用局部搜索的操作。

步骤 6：根据目标函数值比较新个体与旧个体的适应性，如果新个体的目标函数值更小则替换，反之则不替换。

步骤 7：当前一代已经进化完毕，进入下一代，代数加 1，转步骤 2。

2.3.2 归并拆分算子

前一节介绍的 MA 算法主要思想是，进化种群中所有个体，对于得到的新个体进行局部搜索，在邻域中搜寻更优解。

而局部搜索操作算子非常关键，常用的局部搜索算子有单插 SI，双插 DI，以及交换 Swap，而这些算子由于步长过小，当局部解存在时会陷入其中。如图 2.2 清楚展示了前文所述的情况。

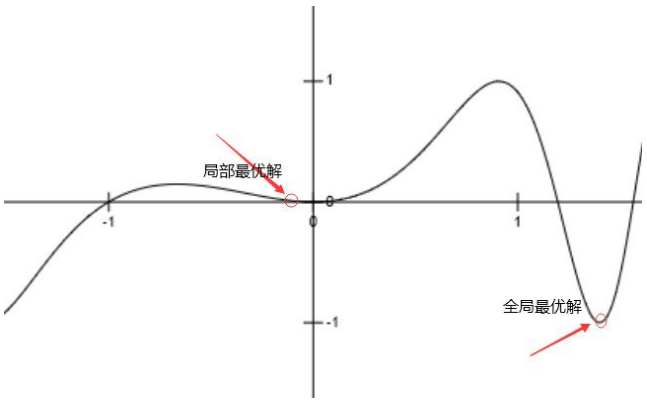

图 2.2 函数图的局部最优解和全局最优解

在[32]中姚鑫提出模拟退火结合局部搜索在扩展邻域搜寻更优解。想到使用大步长的算子增加局部搜索的效率。传统的算子迭代步长小。于是提出了一种新算子，归并拆分 merge-spilt 简称 MS。其主要分为两部分

归并操作：给出初始解，随机选择解中 p 条路径，并将其整合产生一个无序的任务列表，列表包含所选路径上所有任务。

拆分操作：作用在刚才生成的列表。首先采用 path scanning 算法。PS 首先初始化一个空的路径，该操作要找到不满足约束条件的任务，把满足容量约束的任务连接到路径的尾部，其中连接的操作是会选择当前任务最近的任务，根据最短路径来区别。直到所有满足的都完成，剩下的都是不满足的任务，则会在这个序列的尾部连接车场形成一条新的路径。如果有许多路径都满足容量约束且与当前路径的最后一个任务是最近的。就会根据五个准则来判断选择哪一个任务。(1)第一个任务经过所有任务最后到车场的距离是最大的。(2)第一个任务经过所有任务最后到车场的距离是最小的。(3)当前任务的权值 demand 和服务成本 servercost 的比值最大。(4)上一条中提到的比值是最小。(5)如果车辆容量使用没有到达一半则使用(1)否则使用(2)。如果经历这些规则依然碰巧有多选，那么就在这些任务里随机选择一个。此外 PS 操作并不会自动选择使用那一条规则，而是将一条路径扫描五遍，每一条规则都使用一遍，最后会产生五条不同规则下的路径。图 2.3 展示了 MS 操作的过程。

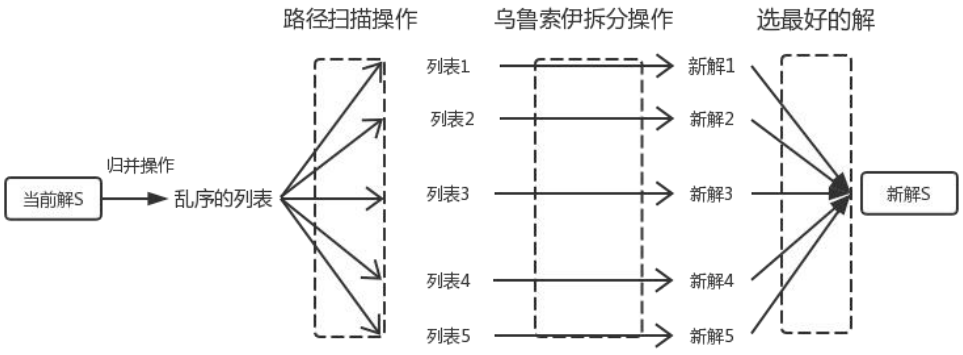

图 2.3 MS 操作过程图

路径扫描操作后，会使用乌鲁索伊拆分操作，作用于五条路，以此来提升解的质量。最后在生成的五个新解里选最好的。乌鲁索伊操作就是寻找一个可行的方法分解这些有序的任务列表，企图通过这样的拆分，组成一个质量更高的新解。

2.3.3 MAENS

Mei[31]在 MA 框架上做了改进。由于 MS 操作十分耗时，所以在进行局部搜索时，先采用单插，双插，交换等小步长算子，当无法提升就停止，从三个解里面选最好的。当做一个局部解。对于得到的解采用 MS 操作进行第二次局部搜索。确保陷入局部最优时可以跳出。表 2.3 详细讲述 MAENS 的过程。

表 2.3 扩展邻域文化算法的详细步骤

步骤 1：输入一个 CARP 的样例，种群大小 p，子种群个数 op，实验个体最大个数 n 以及使用局部搜索的概率 P

步骤 2：初始化一个种群 POP 为空集，设置计数器为 0。

步骤 3：产生一个种群并使计数器加 1

步骤 4：当生成了 n 个实验个体或者连续生成不合格的实验个体时进行步骤 5，否则跳到步骤 3 继续循环。

步骤 5：如果产生的初始化解是初始种群 POP 里的解，则进行步骤 6，否则跳到步骤 2 继续循环。

步骤 6：产生一个暂时中间替代作用的种群 POPt。

步骤 7：随机选择两个不同的解 S1 和 S2 作为父代

步骤 8：采用交叉操作 SBX 让 S1 和 S2 生成新的子代 SX

步骤 9：在（0, 1）之间取一个随机数 r

步骤 10：如果 r 小于局部搜索概率 P 则对于 SX 使用局部搜索操作产生 Sls 并跳到下一步，否则跳到步骤 14.

步骤 11：如果 Sls 和种群 POP 中的元素不重复则插入到 POPt 中并跳到步 14，否则进行下一步

步骤 12：如果 SX 和 POP 中的元素不重复则插入到 POPt 中并跳到步骤 14，否则跳到步骤 15

步骤 13：如果 SX 和 POP 中的元素不重复则插入到 POPt 中并跳到步骤 14

步骤 14：如果当前迭代次数小于子种群个数 op 则继续循环跳到步骤 8 否则进行下一步

步骤 15：将中间种群 POPt 中的解采用锦标赛算法排序

步骤 16：将 POP 赋值为 POPt 中适应值 p 个最好的解。如果没有达到迭代停止条件则跳到步骤 6 继续迭代，否则进返回种群 POP 中最好的解 Sbest

整个流程以种群进化为中心思想，结合局部搜索操作进一步提升了解的精确性。步骤 1-6 是初始化，目的是产生多个可用解为下一步进化做准备，而交叉过程中采用的 SBX 操作名为，基于序列的交叉。其原本是用于 VRP 模型，而 CARP 模型和 VRP 模型相似，于是可用排序并拆解顶点序列，拆成不同路径，如 R1，R2 作为父代拆成(R11, R12)和(R21, R22)将 R12 和 R22 交换产生两个子代(R12, R22)和(R21, R12)。到达了交叉目的，但是交换可能会使某一个任务重复，也可能导致一个任务消失，对于重复的需要删除，对于丢失的需要重新插入到解中，而且需要寻找到不违反容量约束且不增加服务开销的地方，如果有多个位置可插入则随机选择一个。在交叉操作后会进行局部搜索操作，先使用传统算子，当不能找到更优解时，就使用 MS 操作使其跳出局部。

2.4 带修复操作的禁忌搜索

2.4.1 全局修复操作

GRO-TSA[33]虽然不是基于种群的进化算法，但其提出的思想很重要，TSA 作为 CARP 中经典算法，这一节将介绍全局修复操作(global repair option)GRO 以及 TSA，以及其结合算法 GRO-TSA 简称 RTS。

在 CARP 求解最优解的过程中，每产生子代，都会在它的领域中搜寻更优解，如果不满足约束条件会被直接丢弃，甚至没有计算它的开销。这些被丢弃的解虽然违背了容量约束，但是在其中有一些超出约束较小的解，这些解可以通过修复，进而很快地变成可行解或者将违背约束的程度最小化。进而达到解的重复利用，增加利用效率的同时加快了收敛速度。原文因此提出了一种全局修复操作 GRO，以禁忌搜索算法 TSA 为载体。

把不可行解如(0, a, b, 0, b, c, 0, c, a, 0) 每一条路通过用 01 表示，那么可以用这样的 01 序列表示(0, 1, 1, 0, 1, 1, 0, 1, 1)，表示(0, a)不走(a, b)走，以此类推。

而在之前 2.1 的公式(1)-(4)中可以变形写成等式 2.8。

$$\min(\sum(dc(l_k,l_{k+1})+\quad\sum(sc(i,j)-dc(i,j))\times y_k)) \tag{2.8}$$

每个任务的期望 dc 不变，而服务开销与期望的差是固定的，而 yk 在 0,1 之间取值，如果想要整个等式取得最小值，那么应该合理选择 yk 的值，通过(0,1)的赋值，使整个等式最小，问题的关键在于如何对一个任务序列赋予 0,1 值，使其开销达到最小，这就是 GRO 最根本的思想。GRO 目的是对一个不可行解重新赋予 0,1 值，使得其变成一个可行解。而这样的问题可以被抽象成为 0,1 背包问题。背包容

量为 Q，而每个物体就是每一个任务，其重量就是等式后半部的值。

2.4.2 背包算法过程

表 2.4 背包算法的详细步骤

步骤 1：初始化所有 yk=0。所有背包的当前负载 cl(bi)=0，A={1,2,3,...,N},重复步骤 2 到 4 直到 A 为空集。

步骤 2：对于每个 A 中的值 i，定义 $\alpha(i)$ 为解 S 中，在路径 i 上的所有任务集合，再定义 $\beta(k)=\{i\in\alpha(k)\,|\,currentload(bag_i)+server_k\le Q\}$，表示当前路径 k 中满足约束条件的任务集合，选择 $|\beta(k)|$ 模最小，即含有任务数最少的路径 k 插入到当前背包中。如果有多个任务同时最小则选 $server_k$ 最大的那个。通过这样的方式使得插入算法采用最小的步数，并且不会违背容量约束。

步骤 3：定义 bi 表示 $currentload(bag_i)$ 最小的背包，如果多个背包都是最小值则根据 $\sum I_{\alpha(j)}(i)\times s_j$ 最小的那个选择，如果任务 i 在路径 j 的集合中则 $I_{\alpha(j)}(i)=1$ 否则为 0。

步骤 4：插入选择的任务 aj 到背包 b 中 yk 对应为 1，将 j 从 A 中去除。

表 2.4 详细讲述了背包算法的过程，讲述如何将不可行解拆分成许多个任务点，封装到不同的背包中，最后构造出一些可行的不违背约束的路径出来，每个背包中的任务对应的 01 赋值为 1 其他的为 0 这样把选取的边按照解的构造拼凑起来就是一可行的解，或者将其违背容量约束的部分放在后面。通过这一步只是得到了一个初始的解，下一步还需通过禁忌搜索算法进一步提高解的质量。

2.4.3 禁忌搜索算法

TSA 作为经典的启发式算法，其中特色部分是禁忌表，禁忌表相当于当前解的缓存。通过对比禁忌表得到更优解，最后得到最优解。表 2.5 展示禁忌搜索算法的详细步骤

表 2.5 禁忌搜索算法的详细步骤

步骤 1：输入一个解 S0

步骤 2：设置初始解为 S0，最好的解为 Sbest，设置一个存档 A 为空集。

步骤 3：首先使得中间解 tmpS 的函数值 f(tmpS)=∞，这里一般用一个很大的数代替无穷大。并不是真正的无穷大。

步骤 4：对于解 S 采用局部搜索的算子，在其邻域集合 N(S)中的所有变换的

解 SN 采用以下操作。

步骤 5：如果解 SN 不在禁忌表且 f(SN)<f(tmpS)则 tmpS=SN,f(tmpS)=f(SN)并把对于解 S 的变换操作计入到禁忌表中。如果当前解在禁忌表中则选择稍微差一点的另一个解加入到禁忌表中，如果当前没有任何解比当前最优解好则当前最优解设置成为禁忌表中的最优解，该解解除禁忌。

步骤 6：如果得到的中间解 tmpS 的函数值直接为 0，那么直接加入存档。如果没有达到循环的停止条件则跳到步骤 3 继续循环。否则进行下一步

步骤 7：如果 A 为空集则把当前的最优解 Sbest 直接加入，并返回 A

总结整个过程，首先将不可行解重构，抽象成背包问题，得到一个改造后的解，再使用禁忌搜索进一步提升解的质量，将得到的 01 序列的赋值序列使用 Fredierckson 启发式[34]算法进一步的提升解的质量。该算法是基于中国混合邮递员问题，提出了一个在优化求解中新的策略。

2.5 RDG-MAENS

2.5.1 协同进化

RGD-MAENS[41]是在 MAENS 上做了进一步改进，采用分而治之的思想使得 MAENS 有解决大规模的数据集的能力。其中重要的框架就是协同进化。采用类似多模态优化中小生境的技术，将原有的数据集分成多个种群，在每个种群中都使用 MAENS 算法，然后在从中选取出最好的个体作为进化的结果。表 2.6 详细讲述了 CARP 的框架。

表 2.6 大规模 CARP 中的协同进化框架

步骤 1：初始化种群 POP(Z)

步骤 2：计算 POP(Z)中的最优解 S(Z)。

步骤 3：将任务集合 Z 分为 g 组分别是 Z1，Z2...，Zg。

步骤 4：对于每一个分组采取以下操作，将 Zi 中任务对应的个体从 POP(Z)中抽离出来作为子种群 subPOP(Zi)。

步骤 5：计算找出 subPOP(Zi)中的最右解个体 S(Zi)。

步骤 6：输入子种群 subPOP(Zi)和当前子种群的最优解 S(Zi),采用进化算法得到新的最优解 S(Zi)和新的子种群 subPOP(Zi).重复步骤 4-6 直到所有分组都遍历到

步骤 7：将所有子种群整合为一个种群 P(Z)

步骤 8：将每个种群的最优解整合成为一个集合 St(Z)

步骤 9：如果 St(Z)的函数值小于 S(Z)则更新 S(Z)为 St(Z)，直到循环进行到边界，否则跳转到步骤 3 继续循环

步骤 10：返回当前最优解 S(Z)

该框架关键在于如何分解任务，良好的分解策略可以极大地提升程序的效率以及结果的精度，有的算法根据坐标分类，且依据欧拉距离衡量，但这不符合实际情况，中，欧拉距离是直线距离。而直线距离通常无法直接抵达。所以使用 Dijkstra 算法计算最短路径更加合理。而下面介绍的道路距离分组是在最短路径的基础上提出的更精确的距离衡量方式。

2.5.2 道路距离分组

道路距离分组原名 Route Distance Grouping 是新的距离衡量方式。这里提出的任务距离是第一步，如果两个任务之间相似度高，则这两个任务应该放到一个分组里，因为距离更近的任务可以减少空车行驶的成本，达到了优化目的。定义两个任务之间的距离表示如 2.9

$$\Delta task(z_1,z_2)=\sum_{i=1}^{2}\sum_{i=1}^{2}\Delta(v_i(z_z),v_j(z_2))\Big/4 \tag{2.9}$$

一条边有两个顶点 v1 和 v2，分别计算(v1(z1),v1(z2))，(v1(z1),v2(z2))，(v2(z1)，v1(z2))，(v2(z1),v2(z2))，即两条边两组顶点之间的相互距离，两两计算，之间的距离以边的个数为衡量单位而不是任务量 demand，取均值即为任务变化量。

据任务变化量，可进一步定义路径变化量，一条路径由若干任务结点构成如 s=(0,z1,z2,z3,0)，可以根据任务结点间的距离定义算出路径间的距离。

$$\Delta route(s_1,s_2)=\sum_{z1\in s1}\sum_{z2\in s2}\Delta task(z_1,z_2)\Big/|s1|*|s2| \tag{2.10}$$

把路径看做一个向量，任务 z 看做元素，计算路径 s1 上的元素与 s2 上的元素的任务距离，逐个计算，最后累加，除以向量长度即路径上任务的个数的乘积。这样便得到了路径变化量或路径距离。

而得到的所有路径距离必须归一化，便于比较，参考 cos 角

$$\bar{\Delta} route(s_1,s_2)=\frac{\Delta route(s_1,s_2)}{\Delta route(s_1,s_1)}*\frac{\Delta route(s_1,s_2)}{\Delta route(s_2,s_2)} \tag{2.11}$$

归一化得到了所有路径间的相似度。这样便得到了关联矩阵。根据矩阵，可以设置阈值使得相似度高的路径分到一组。而设置了衡量标准之后，如何将这些相似路径分到一起成了个问题。这里设计到机器学习中类似 K-means 的分类方法 k-medoid，和 k-mean 类似，给定分组个数后，自动的找到中心然后将周围个体分到中心周围形成一组。

2.5.3 模糊 k 中心问题

上节末尾提到，有了关系矩阵，要寻找到所有路径的中心，使得剩下的非中心路径分类，从而完成分组，达到分解的目的，将此类问题用数学公式 2.12 表示。

$$\min_{c\subseteq s} J_\alpha(c;s)=\sum_{si\in s\backslash c}\sum_{cj\in c}\psi_\alpha(s_i,c_j)*\bar{\Delta} route(s_i,c_j) \tag{2.12}$$

c 为中心路径，s 为所有路径集合，模糊 k-medoid 问题就是使得非中心路段距离各自中心路段的距离最小即成功得到一个 J。ϕ 为非中心路段 si 和第 j 条中心路段 cj 的成员系数，用于衡量距离不同中心 c 的权重，距离 cj 越近得的比重越大。ϕ 的具体计算公式 2.13。

$$\psi_\alpha(s_i,c_j)=\frac{(1/\bar{\Delta} route(s_i,c_j))^\alpha}{\sum_{k=1}^{g}(1/\bar{\Delta} route(s_i,c_k))^\alpha} \tag{2.13}$$

分子部分是 si 和当前中心路段 cj 的相似度的倒数，变化值越小隔得越近，倒数越大。分母为距离所有中心 c 的倒数的累和，也就是衡量距离哪一条路更近的权重。指数 α 为模糊系数，越大越模糊越清晰，区分度越强。一般设置为 1，5，10 这三个值。

有了这些衡量系数以及目标函数，具体求解模糊 k 中心问题的算法在表 2.7 展示。

表 2.7 中心分块算法 PAM

步骤 1：随机从边的集合中选择一条作为中心路 c。

步骤 2：根据公式计算该路段在集合中的权值 J(c;s)。

步骤 3：初始化中间变量 c_old 和 c_new 为 c，且 J(c_new;s)=J(c;s)。

步骤 4：对于 g 个分组进行循环，当前循环次数为 j。

步骤 5：对于每一个边集 s 中非中心路段 s\c 中剩下的边 si 进行循环。

步骤 6：中间变量 tmpc=c，交换 cj 和 si，计算 J(tmpc;s)系数

步骤 7：如果 J(tmpc;s)<J(c_new;s)则更新 c_new 为 tmpc

步骤 8：直到 s 中的边遍历完，否则跳到步骤 5 继续循环

步骤 9：直到 g 个分组都遍历完否则跳到步骤 4 继续循环

步骤 10：将 c_new 的值赋予 c 如果 c_old=c 则结束并返回 c 的值否则跳到步骤 3 继续循环。

步骤总体分为三大循环，外部循环控制迭代结束，二层循环控制遍历到所有分组，内层循环控制更新迭代产生新的中心 c。通过不断交换，产生新中心，在所有边都当过中心之后则可以算出哪一个边的权重系数最大，则最应该成为中心。当然这样的反复循环计算时间开销很大。

当得到了每个分组的中心后，需将剩余边分到对应组别中。表 2.8 介绍如何将剩余边分组。

表 2.8 对非中心边赋值

步骤 1：初始化分组集合 G 中的元素为{ci}。只有一个元素即中心边

步骤 2：对于每一个非中心的边 $s_i \in s \setminus c$ 进行如下操作。

步骤 3：随机取一个 0-1 之间的随机数 r，设置 ϕ=0。

步骤 4：对于 g 个分组进行如下循环。

步骤 5：累加每一个分组里其他路段与中心边的系数 ϕ=ϕ+ψ (ck;si)。

步骤 6：如果累加的概率值 ϕ>r 则跳出当前循环否则将边 si 归类到分组 Gk 中 Gk=Gk∪si

步骤 7：直到 g 个分组都遍历完否则跳到步骤 4 继续循环

步骤 8：直到 s 中的边遍历完，否则跳到步骤 2 继续循环

步骤 9：返回得到的分组(G1...Gg)

得到路径分组，现在需将路径上的任务分组，用于协同进化中后续计算。

得到了任务分组之后，分解任务才算完成，表 2.9 展示基于路径分组的任务分解

表 2.9 基于路径分组的任务分解

步骤 1：初始化 g 个分组集合 Zk 中的元素为{}。即空集
步骤 2：对于每一个路径分组 Gi 中的边 si 进行如下循环。
步骤 3：对于每一条边 si 上的任务 zj 进行如下循环
步骤 4：将 z 并入到当前分组任务集合 Zk 中。
步骤 5：直到每一条边遍历完，否则跳到步骤 3 继续循环
步骤 6：直到每一个分组 Gi 遍历完否则跳到步骤 2 继续循环
步骤 7：返回任务分组(Zi...Zg)

到此所有任务都已经分组完成，任务分解的目的达成。总结，首先进行根据顶点间的距离的均值，定义了任务距离，根据路径上带有任务的距离的均值，定义了路径间的距离。最后归一化，得到相关系数作为评判标准。得到路径相似度矩阵后，路径上的任务如何分解，关键是找到路径集合的中心，根据中心进行分类。因此定义了路径模糊权值 ϕ，相关性系数越小，说明相关性越大隔得越近那么系数权值就越大。有了每条路在所有路的权重后，便可根据权重与路径相关性系数的乘积综合评判最中心的路径，找到每一个分组的中心 c 后。对于剩余路径需归入到分组中。设置一个随机数作为阈值，只要阈值没有达到则会将当前遍历路径归入到分组中。这样便完成了路径的分组，最后将路径上的任务对应地分解到任务集合中。这样便完成了整个任务分解。具体框架如表 2.10 展示

表 2.10 路径距离分解

步骤 1：输入路径集合 s，分组数 g，模糊系数 α
步骤 2：根据公式计算出路径相关性矩阵 $\bar{\Delta} route(s_{k1}, s_{k2})_{m*m}$。
步骤 3：根据路径相关性矩阵使用 PAM 算法计算出所有中心 c
步骤 4：使用非中心边赋值算法计算出路径分组(G1...Gg)。
步骤 5：根据路径分组得到任务分组(Z1...Zg)
步骤 6：返回任务分组(Zi...Zg)

第三章 对比与分析

3.1 算法思想对比

3.1.1 RTS 与 MAENS

相同之处：

①都是启发式算法，都是通过一个初始解，经过不停地迭代更新，改变解的结构，计算函数适应值，通过适应值的更新替换原解，得到当前最优解，逼近全局最优解。

②都使用了局部搜索这一操作。通过局部搜索在邻域中寻找更优的解。

不同之处：

①搭建的体系不同，RTS 是通过禁忌表，设定一定的禁忌与解禁的规则，配合修复操作 GRO。而 MAENS 是基于种群，种群通过进化选择更优的子代，且结合局部搜索操作进行进一步的优化操作。

②RTS 的局部搜索是经典的单插 SI 双插 DI 以及交换 Swap，而 MAENS 在此基础上进行了一点改进加入了 MS 操作，通过改变当前解的结构，变换产生更大的步长，跳出局部最优解。

3.1.2 RDG-MAENS 与 MAENS、RTS

相同之处：

①都是在定义域随机的产生个体

②都会通过局部搜索产生新的个体

③RDG-MAENS 与 MAENS 都是基于种群的进化迭代，求解最优解。

不同之处：

①RDG-MAENS 是采用分而治之的思想，对于大型数据集，大而化小的想法计算，而 MAENS、RTS 都是只考虑中小型数据集。相当于 RGD-MAENS 中分组执行的操作

②RDG-MAENS 采用了协同进化的框架，比 MAENS、RTS 高一个处理级别。但这两个各有好处。MAENS、RTS 更加适用于在中小型数据集求解，由于

RDG-MAENS 任务分组的开销比较大，只有在数据规模达到一定级别的时候才会展现出它的优势，而适用 RDG-MAENS 去求解中小型数据集时，多余的分解工作的计算开销以及精度，并没有另外两个算法表现优秀。

3.2 测试数据以及统计

3.2.1 测试用例

本次实验采用了两种不同规模的数据集，一共有四个，val，egl，gdb 属于小型数据集，Bullen 属于中型数据集。Gdb 数据集是由 DeArmon[35]制作。一共含有 23 个测试用例从 gdb1 到 gdb23。Val 数据集共包含 34 个测试用例，由 Benavent 制作[36]。该数据集总共是 1-10 十个标号 1、2、3 各自有 A、B、C 三个小分组，点和边的数量一样，LowerBound 越来越大而 4、5 则含有 A、B、C、D 四个小分组，情况如表 3.2。Egl 数据是由 Eglese 根据兰开夏郡冬季磨砂应用制作[37]-[40]而成。总共两个图包含了 24 个测试用例，一个以 e 标记一个以 s 标记。每个图分为四组分别是 E1-E4，一个组包含三小组 A、B、C，小组内的点边弧的数量一致，不同点是边界越来越大。如表 3.3 所示。Egl 是中型数据集。一般来说点的个数在 50 以上是中型。另一个数据集 Beullens[42]是由本人 Beullens 制作。其中包括 C、D、E、F，其中 C 和 E 为一组，D 和 F 为一组，唯一的区别是车容量不同，后者为前者的两倍如表 3.4-3.7。

表 3.1 gdb 测试用例各算法得到的均值以及运行时间

测试集	顶点数 V	边数 E	边界值 LB	RTS		MAENS		RDG-MAENS	
				均值	时间	均值	时间	均值	时间
gdb1	12	22	316	345	0.010	316	7.450	323	0.950
gdb2	12	26	339	365	0.000	339	9.330	339	2.065
gdb3	12	22	275	317	0.009	275	8.450	275	1.098
gdb4	11	19	287	296	0.009	287	6.290	287	0.037
gdb5	13	26	337	442	0.000	377	9.720	377	5.824
gdb6	12	22	298	330	0.000	298	7.310	298	1.886
gdb7	12	22	325	363	0.000	325	7.430	325	0.859
gdb8	27	46	348	462	0.000	348	22.98	350	2.870
gdb9	27	51	303	355	0.000	303	26.96	303	45.15
gdb10	12	25	275	302	0.010	275	8.240	275	2.384
gdb11	22	45	395	444	0.009	395	20.05	395	4.056

gdb12	13	23	458	608	0.000	458	7.260	458	2.657
gdb13	10	28	536	572	0.010	543	9.77	538	9.745
gdb14	7	21	100	118	0.000	100	3.580	100	1.397
gdb15	7	21	58	60	0.009	58	0.671	58	0.786
gdb16	8	28	127	129	0.009	127	2.610	127	0.176
gdb17	8	28	91	91	0.009	91	1.250	91	0.024
gdb18	9	36	164	179	0.010	164	12.05	164	0.781
gdb19	8	11	55	57	0.000	55	1.920	55	0.137
gdb20	11	22	121	132	0.010	121	9.710	121	0.223
gdb21	11	33	156	178	0.000	156	12.08	156	5.556
gdb22	11	44	200	211	0.000	200	7.810	200	0.915
gdb23	11	55	233	249	0.010	235	16.06	233	3.486
Averag	----	----	252.04	287.1	----	254.1	----	254.2	----

表 3.2 val 测试用例各算法得到的均值以及运行时间

测试集	顶点数 V	边数 E	边界值 LB	RTS		MAENS		RDG-MAENS	
				均值	时间	均值	时间	均值	时间
val1A	24	39	173	290	0.009	173	12.90	296	1.111
val1B	24	39	173	193	0.010	173	16.10	173	1.388
val1C	24	39	245	304	0.000	245	16.99	245	5.882
val2A	24	34	227	249	0.000	227	10.74	248	0.113
val2B	24	34	259	284	0.009	259	11.81	260	0.293
val2C	24	34	457	510	0.009	457	13.55	463	3.820
val3A	24	35	81	82	0.009	81	10.84	84	0.331
val3B	24	35	87	92	0.000	87	12.68	87	2.399
val3C	24	35	138	166	0.000	138	18.99	138	0.129
val4A	41	69	400	429	0.010	400	52.20	406	2.796
val4B	41	69	412	487	0.000	412	50.15	412	58.01
val4C	41	69	428	473	0.010	436	52.51	428	46.22
val4D	41	69	526	652	0.009	536	49.30	539	32.05
val5A	34	65	423	469	0.000	423	42.67	423	4.177
val5B	34	65	446	489	0.010	446	42.53	446	2.653
val5C	34	65	473	531	0.010	474	42.98	474	19.45
val5D	34	65	573	715	0.009	585	39.89	585	103.3
val6A	31	50	223	259	0.009	223	23.35	223	4.724
val6B	31	50	233	275	0.009	233	23.77	233	12.64

val6C	31	50	317	381	0.009	317	26.17	317	8.505
val7A	40	66	279	316	0.000	279	41.16	279	8.284
val7B	40	66	283	340	0.009	283	38.66	283	6.275
val7C	40	66	334	400	0.000	334	40.40	340	11.41
val8A	30	63	386	429	0.009	386	37.63	397	1.535
val8B	30	63	395	462	0.009	395	36.82	395	9.146
val8C	30	63	518	593	0.009	527	39.28	521	27.67
val9A	50	92	323	365	0.009	327	106.6	328	28.21
val9B	50	92	326	364	0.010	326	94.81	326	84.87
val9C	50	92	332	376	0.010	332	90.29	333	13.25
val9D	50	92	385	480	0.009	396	88.32	393	37.15
val10A	50	97	428	451	0.010	429	131.0	436	14.61
val10B	50	97	436	485	0.009	438	125.4	438	34.90
val10C	50	97	446	480	0.000	447	110.5	446	47.47
val10D	50	97	525	596	0.000	538	109.6	527	226.0
Averag	----	----	343.82	396.0	----	345.9	----	350.6	----

表 3.3 egl 测试用例各算法得到的均值以及运行时间

测试集	顶点数 V	弧数 R	边数 E	边界值 LB	RTS		MAENS		RDG-MAENS	
					均值	时间	均值	时间	均值	时间
E1-A	77	51	98	3548	3548	0.039	3548	1.830	3548	2.623
E1-B	77	51	98	4498	4525	7.320	4525	31.93	4498	5.280
E1-C	77	51	98	5566	5679	8.240	5613	33.83	5595	8.717
E2-A	77	72	98	5018	5018	15.90	5018	66.24	5018	11.99
E2-B	77	72	98	6305	6335	17.67	6344	66.60	6317	23.13
E2-C	77	72	98	8243	8431	20.38	8340	70.28	8335	28.09
E3-A	77	87	98	5898	5924	25.15	5910	94.29	5898	32.00
E3-B	77	87	98	7704	7854	28.72	7801	95.50	7777	74.73
E3-C	77	87	98	10163	1036	32.90	1033	89.62	1029	185.9
E4-A	77	98	98	6408	6546	35.09	6461	116.9	6461	143.1
E4-B	77	98	98	8884	9112	41.02	9062	116.3	8988	96.19
E4-C	77	98	98	11427	1225	44.42	1163	105.3	1159	113.0
S1-A	140	75	190	5018	5189	20.07	5095	61.77	5018	11.62
S1-B	140	75	190	6384	6515	22.30	6439	69.37	6388	106.4
S1-C	140	75	190	8493	8663	25.91	8518	69.70	8518	166.7
S2-A	140	147	190	9824	1028	109.8	1006	288.7	9875	343.9

S2-B	140	147	190	12968	1370	125.8	1340	246.6	1315	222.5
S2-C	140	147	190	16353	1690	144.0	1659	220.8	1643	141.4
S3-A	140	159	190	10143	1047	135.7	1040	344.3	1025	527.3
S3-B	140	159	190	13616	1407	159.6	1397	275.3	1369	292.6
S3-C	140	159	190	17100	1761	176.3	1750	262.1	1721	417.3
S4-A	140	190	190	12143	1264	225.4	1243	419.9	1227	331.9
S4-B	140	190	190	16093	1661	266.1	1652	378.9	1638	251.8
S4-C	140	190	190	20375	2219	289.2	2101	353.8	2068	528.3
Averag	----	----	----	9673	1200	----	9856	----	9758	----

表 3.4 beullens C 测试用例各算法得到的均值以及运行时间

测试集	顶点数 V	弧数 R	边数 E	边界值 LB	RTS		MAENS		RDG-MAENS	
					均值	时间	均值	时间	均值	时间
C01	69	79	98	1590	2490	0.030	4225	216.3	3230	62.42
C02	48	53	66	1095	2040	0.010	3135	29.44	2520	4.753
C03	46	51	64	875	1650	0.01	2575	31.57	2085	3.221
C04	60	72	84	1285	2170	0.009	3510	56.67	2785	13.39
C05	56	65	79	2410	2895	0.010	5390	44.85	3935	5.566
C06	38	51	55	855	1640	0.010	2550	28.66	2165	3.852
C07	54	52	70	1735	2280	0.000	4075	31.87	3165	12.40
C08	66	63	88	1640	2360	0.010	4100	39.28	3055	2.537
C09	76	97	117	1775	3440	0.009	5305	107.8	4130	11.10
C10	60	55	82	2190	2430	0.010	4730	30.01	3340	48.77
C11	83	94	118	1728	2825	0.021	4715	97.67	3760	30.98
C12	62	72	88	1510	2630	0.010	4240.	56.32	3370	26.88
C13	40	52	60	1050	1845	0.009	2960	28.99	2540	10.76
C14	58	57	79	1620	2350	0.000	4050	32.90	3280	2.613
C15	97	107	140	1765	3080	0.000	4950	133.1	4030	34.49
C16	32	32	42	580	890	0.010	1475	12.34	1280	18.20
C17	43	42	56	1590	1945	0.010	3555	19.00	2620	11.40
C18	93	121	113	2135	3235	0.010	5655	180.3	4165	242.6
C19	62	61	84	1345	1720	0.009	3160	41.71	2410	11.75
C20	45	53	64	665	1455	0.000	2140	30.52	1900	9.555
C21	60	76	84	1705	2245	0.010	3970	63.61	3100	19.91
C22	56	43	76	1070	1175	0.010	2245	20.84	1865	6.018
C23	78	92	109	1620	2395	0.009	4145	96.95	3145	47.19

C24	77	84	115	1330	2040	0.009	3400	75.97	2755	25.38
C25	37	38	50	905	1405	0.009	2310	14.35	1815	6.078
Averag	----	----	----	1442	2185	----	3702	----	2897	----

表 3.5 beullens D 测试用例各算法得到的均值以及运行时间

测试集	顶点数 V	弧数 R	边数 E	边界值 LB	RTS		MAENS		RDG-MAENS	
					均值	时间	均值	时间	均值	时间
D01	69	79	98	725	2490	0.009	3235	72.53	3215	99.57
D02	48	53	66	480	2040	0.009	2520	27.95	2520	15.92
D03	46	51	64	415	1650	0.010	2070	26.91	2056	26.57
D04	60	72	84	615	2170	0.000	2785	57.33	2785	23.91
D05	56	65	79	1040	2895	0.009	3935	42.94	3935	6.497
D06	38	51	55	485	1640	0.000	2125	26.18	2125	6.594
D07	54	52	70	835	2280	0.000	3165	29.40	3115	18.23
D08	66	63	88	685	2360	0.010	3075	36.77	3045	50.44
D09	76	97	117	680	3440	0.010	4120	104.2	4120	8.949
D10	60	55	82	910	2430	0.000	3340	27.32	3410	4.857
D11	83	94	118	930	2825	0.011	3770	109.1	3760	6.424
D12	62	72	88	680	2630	0.010	3310	55.03	3310	19.61
D13	40	52	60	690	1845	0.010	2535	28.37	2535	6.909
D14	58	57	79	920	2350	0.010	3280	34.13	3280	8.248
D15	97	107	140	910	3080	0.010	4000	143.4	4000	52.43
D16	32	32	42	170	890	0.009	1060	10.42	1260	0.324
D17	43	42	56	675	1945	0.010	2620	17.99	2620	1.284
D18	93	121	113	930	3235	0.010	4185	195.9	4165	98.44
D19	62	61	84	650	1720	0.009	2400	40.98	2400	8.276
D20	45	53	64	415	1455	0.010	1870	31.04	1870	7.689
D21	60	76	84	695	2245	0.010	3080	69.28	3090	8.340
D22	56	43	76	690	1175	0.000	1865	19.93	1865	2.888
D23	78	92	109	715	2395	0.010	3200	98.25	3140	87.44
D24	77	84	115	620	2040	0.000	2710	79.11	2710	7.034
D25	37	38	50	410	1405	0.000	1815	14.29	1835	2.540
Averag	----	----	----	678.80	2185	----	2882	----	2886	----

表 3.6 beullens E 测试用例各算法得到的均值以及运行时间

测试集	顶点数 V	弧数 R	边数 E	边界值 LB	RTS		MAENS		RDG-MAENS	
					均值	时间	均值	时间	均值	时间
E01	73	85	105	1855	2975	0.010	4960	80.12	4140	11.07
E02	58	58	81	1580	2380	0.009	4080	33.06	3300	15.70
E03	46	47	61	750	1265	0.009	2025	25.02	1765	1.500
E04	70	77	99	1580	2545	0.010	4255	68.00	3495	124.1
E05	68	61	94	2130	2425	0.009	4765	38.97	3605	4.455
E06	49	43	66	670	1385	0.010	2055	18.65	1875	1.837
E07	73	50	94	1780	2255	0.000	4155	28.63	3335	19.51
E08	74	59	98	2080	2560	0.000	4710	36.63	3855	19.51
E09	93	103	141	2160	3585	43.06	5850	117.1	4810	91.44
E10	56	49	76	1690	1915	0.009	3605	25.34	2925	4.234
E11	80	94	113	1810	2820	0.000	4720	101.6	3895	13.51
E12	74	67	103	1580	2485	0.009	4235	48.44	3475	6.126
E13	49	52	73	1300	2020	0.010	3345	30.05	2855	9.558
E14	53	55	72	1780	2305	0.010	4145	33.42	3375	12.77
E15	85	107	126	1555	2615	0.009	4250	127.8	3560	25.31
E16	60	54	80	1785	1950	0.010	3785	34.92	2815	4.375
E17	38	36	50	1290	1450	0.009	2745	14.97	2055	5.866
E18	78	88	110	1600	2225	0.010	3845	90.12	3075	101.2
E19	77	66	103	1400	1800	0.009	3245	47.22	2535	5.097
E20	56	63	80	950	1835	0.000	2825	45.41	2450	10.63
E21	57	72	82	1700	2025	0.009	3790	61.11	2940	11.69
E22	54	44	73	1155	1285	0.009	2490	21.06	2075	4.541
E23	93	89	130	1395	2280	0.010	3770	90.13	3010	32.71
E24	97	86	142	1695	2235	0.010	4080	92.49	3260	10.23
E25	26	88	35	655	960	0.009	1615	9.557	1525	0.287
Averag	----	----	----	1517	2143	----	3733	----	3040	----

表 3.6 beullens F 测试用例各算法得到的均值以及运行时间

测试集	顶点数 V	弧数 R	边数 E	边界值 LB	RTS		MAENS		RDG-MAENS	
					均值	时间	均值	时间	均值	时间
F01	73	85	105	1065	2975	0.010	4050	90.96	4040	56.45
F02	58	58	81	920	2380	0.009	3300	36.59	3300	3.954
F03	46	47	61	400	1265	0.000	1665	24.93	1655	1.382

F04	70	77	99	930	2545	0.010	3510	72.55	3485	81.97
F05	68	61	94	1180	2425	0.010	3605	42.50	3605	27.48
F06	49	43	66	490	1385	0.009	1875	19.23	1875	2.471
F07	73	50	94	1080	2255	0.000	3335	28.67	3335	8.820
F08	74	59	98	1135	2560	0.000	3705	39.84	3705	3.014
F09	93	103	141	1145	3585	34.90	4810	128.7	4730	97.65
F10	56	49	76	1010	1915	0.010	2925	26.05	2975	1.657
F11	80	94	113	1015	2820	0.000	3865	117.8	3855	99.41
F12	74	67	103	900	2485	0.011	3450	50.74	3395	41.58
F13	49	52	73	835	2020	0.000	2855	29.89	2855	24.82
F14	53	55	72	1025	2305	0.000	3395	34.46	3375	6.583
F15	85	107	126	945	2615	0.010	3560	147.9	3560	57.75
F16	60	54	80	775	1950	0.010	2725	36.21	2725	9.462
F17	38	36	50	605	1450	0.010	2055	13.98	2055	3.668
F18	78	88	110	835	2225	0.000	3110	104.1	3075	30.33
F19	77	66	103	685	1800	0.000	2525	56.86	2525	8.388
F20	56	63	80	610	1835	0.009	2450	47.45	2445	33.70
F21	57	72	82	905	2025	0.009	2930	67.52	2930	17.78
F22	54	44	73	790	1285	0.010	2075	22.12	2075	1.971
F23	93	89	130	705	2280	0.010	3025	98.49	3005	49.75
F24	97	86	142	975	2235	0.000	3250	101.3	3210	72.48
F25	26	88	35	430	960	0.000	1390	8.677	1390	2.110
Averag	----	----	----	855.6	2143	----	3017	----	3007	----

3.2.2 实验结果分析

从表 3.1 可以看出，当运行 gdb 数据集时，三个算法的均值并没有明显的差异，MAENS 和 RDG-MAENS 略好于 RTS。在 gdb1 到 gdb14 时 RTS 的数值劣与后两者且差距明显，从图 3.1 来看也是在 gdb14 以前有着明显的差距。RTS 的图像高出后两者一截。前半部分顶点数大于 12 时 RTS 表现没有后两者好，小于 12 时效果相当。也就是说进化算法的模型在顶点数大于 12 时表现要优于 TSA 算法。但 TSA 的时间开销比后两者小很多。

从表 3.2 可以看出在 val1 组别三种算法效果相当。Val2 组别中，边界值的变大，RTS 的效果不如 MAENS 和 RDG-MAENS。从 val3 组也可以看出相同的结果，边界值小效果相同，边界值变大 RTS 效果不如后两者好。并且观察 val6-val10 中

可以发现大的边界值使得 RTS 不能很好的收敛，且越大效果越不好。从图 3.2 也可以看到。三个为一小组，每组前两个数据相差不大，最后一两个都会以峰值拉开差距。也就是说相同顶点数和边数，边界值越大进化算法模型的效果就比 RTS 好。RTS 的收敛时间从头到尾比后两者快很多。

从表 3.3 可以看到，在顶点数大于 50 达到中型数据集标准时，分两组 77 和 140。三种算法的差距并不明显，RDG-MAENS 略优于另外两者。从图 3.3 上看，三条曲线几乎重叠在一起，只在中间略微红色突出。整体差距并不明显。MAENS 与 RTS 的效果相同，但都微劣于 RDG。

表 3.4 和表 3.5 应该对比起来看，因为 C 和 D 数据集的顶点、边和弧的数量相同，不同的只有边界值或者说是车辆负载。C 的负载为 300，D 为 600。小的负载需要更好的规划算法。从表 3.4 上看，RTS 的效果全程比后两者好，且与边界值的差距小。进化算法模型表现不好，但 RDG 全程比 MAENS 好。但表 3.5 和图 3.5 上 RDG 和 MAENS 的差距几乎没有。原因是富裕的负载让协同进化的优势消失。但与边界值有很大差距。

表 3.6 和 3.7 同样应该对比起来，因为 F 的负载是 600，是 E 的两倍。从表格上看规律与 C 和 D 的分布相同。负载为 300 时 RDG 比 MAENS 好，但都劣于 RTS。负载为 600 时 RDG 与 MAENS 相同且劣于。但三者与边界差距较大。

总结：在小的数据集 gdb 和 val 上，RTS 效果比后两者差，RDG 比 MAENS 好。且边界值越大 RTS 的效果越不好。原因 RTS 的局部搜索步长较小，在边界值大时陷入局部最优无法跳出。而后两者有 MS 操作，使得收敛值更小。在中型数据集 egl 上 MAENS 与 RTS 效果相当，且都比 RDG 略差。因为协同进化的框架细致了搜索空间，使得收敛效果略好于前两者。在 beullens 数据集上，RTS 表现的效果比进化算法好。且进化算法模型数值偏大，效果不好。原因是当解的搜索范围变大，局部解峰值较多时，MS 操作并不能跳出局部解。相反 RTS 的不可行解的利用使得收敛的程度更好。协同进化的框架，细致了解的搜索范围使得效果优于 MAENS，当容量较大时，满足约束的解有很多，协同进化的优势便体现不出。所以协同进化只有在小容量，解搜索困难的空间里才有优势。而 RTS 的 GRO 修复操作增加了种群的多样性。增加了收敛速度，以及收敛的可能性。

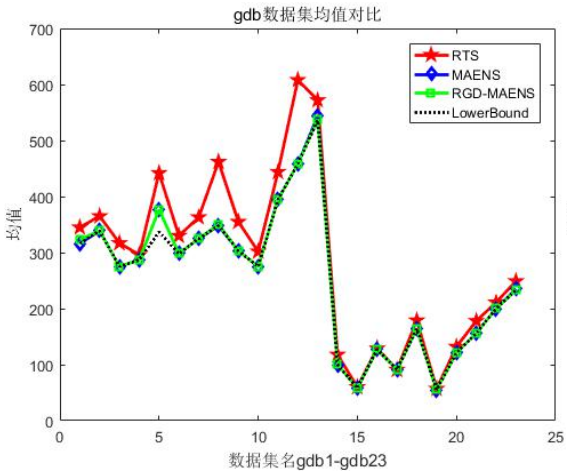

图 3.1 gdb 数据集测试结果

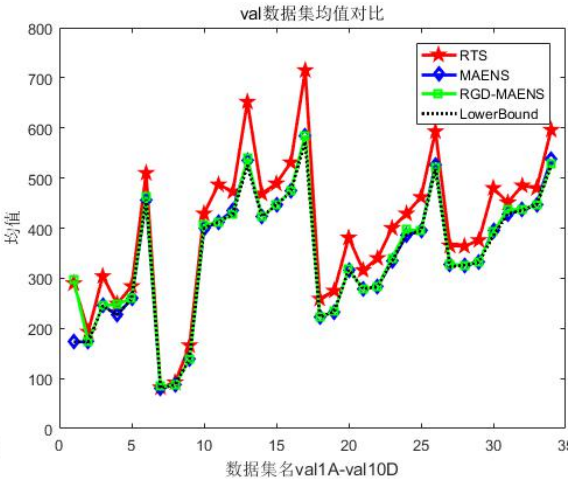

图 3.2 val 数据集测试结果

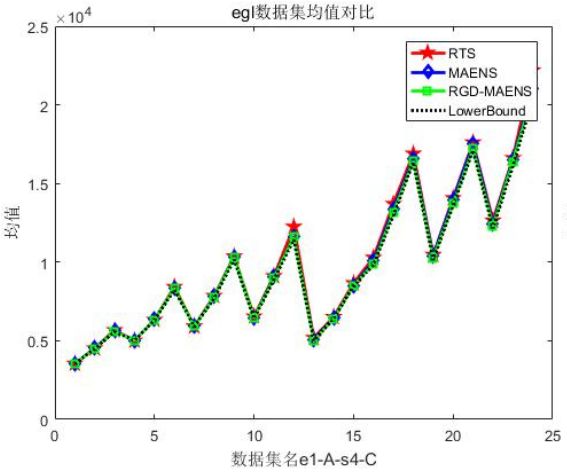

图 3.3 egl 数据集测试结果

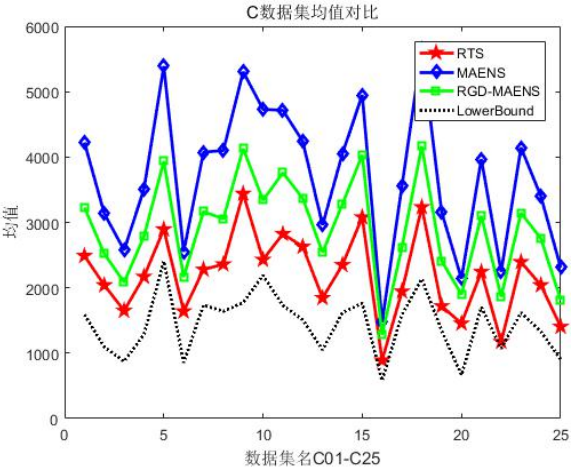

图 3.4 C 数据集测试结果

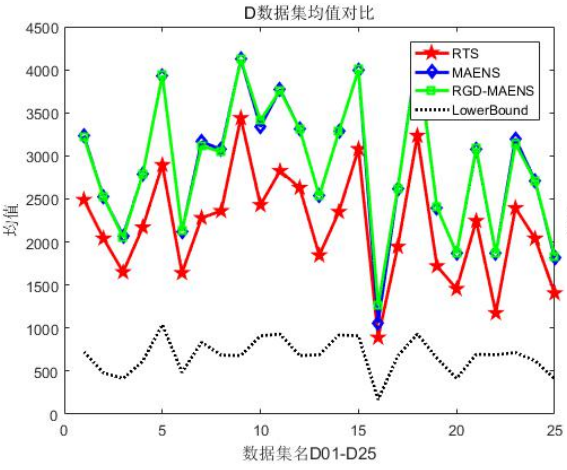

图 3.5 D 数据集测试结果

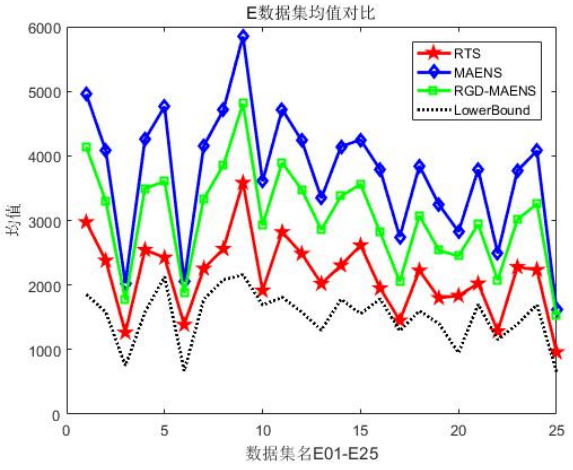

图 3.6 E 数据集测试结果

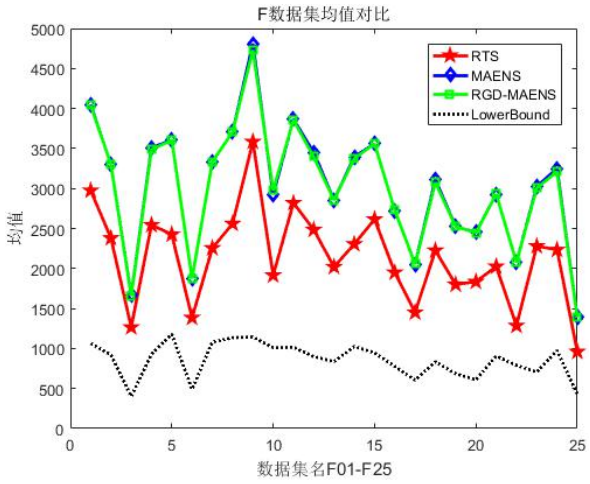

图 3.7　F 数据集测试结果

第四章 总结与展望

4.1 总结

本文共介绍了三种求解 CARP 的算法，其中 RTS 是 TSA 加入 GRO 操作。MAENS 和 RDG-MAENS 都是基于进化算法。前两种经典算法却没有被用于比较其性能。而本文详细的比较了其在不同数据集上性能的差异。分析了其改进与不足，以及不同的适用场景。

总结 RTS 的优点①对于不可行解的利用增加的多样性②收敛速度快操作步骤少，同时 RTS 也有一定的缺点，简单的局部搜索步骤以及禁忌表的结构，导致解的精确程度不高，没有更加细致的局部搜索，导致整体结果欠佳。

总结 MAENS 的优点①速度快于 RDG-MAENS②采用更加细致的局部搜索操作 MS 可以跳出局部解收敛到最优解。缺点①MS 操作仅在小型数据集有良好效果，当数据集变大效果并不好。

总结 RDG-MAENS 的优点①在处理中型数据集的效果优于 MAENS②分而治之的思想加入了协同进化，在不同范围搜索最优解，增加了找到最优解的机会。缺点和 MAENS 相同。

4.2 展望

在运行迭代的过程中，种群进化总是淘汰掉不可行的个体，而 RTS 中的 GRO 操作便很好的使用了这一点，增加了种群的多样性，协同进化的框架虽然好，但是如果能够增加不可行解的利用性，那么收敛速度以及结果都会比现在更好。RDG-MAENS 还有很多改进的地方，其中定义的一些距离以及矩阵运算复杂时间开销大。应该换成更为简单的定义权值形式，且应该有更准确的衡量方式，且在分类路径的时候，K 中心分法并非良好的分类方法，相反其方法简陋，如果换成 K-Means 这样的机器学习算法，进化算法与机器学习算法结合会更好。

参考文献

[1]李军．物流配送车辆优化调度理论与方法[M]．中国物资出版社，2001.

[2]Dantzig G B, Ramser J H. The Truck Dispatching Problem[J]. Management Science, 1959, 6(1):80-91.

[3] Lenstra J K, Kan A H G R. Complexity of vehicle routing and scheduling problems[J]. Networks, 1981, 11(2):221-227.

[4]覃运梅，毛海军，黑秀玲．基于自动快递机的快递配送车辆路径优化研究[J]. 公路交通科技，2015(10):138-144.

[5]赵彤，范厚明，王桂琳，等．带时间窗的应急救助物资配送车辆路径优化模型研究[J]. 物流技术，2010, 029(020):P.63-65,68.

[6]李泽华．带时间窗约束的生鲜产品配送车辆路径优化问题研究[D]．大连海事大学.

[7]蒋波．基于遗传算法的带时间窗车辆路径优化问题研究[D]. 北京交通大学, 2010.

[8]张利城, 吴金卓, 何荣. 基于循环取货模式的车辆路径优化研究[J]. 森林工程, 2013, 029(004):86-89,99.

[9]王铁君，邬开俊．带时间窗的多车场车辆路径优化的粒子群算法[J]．计算机工程与应用，2012, 48(27):27-30.

[10]於世为, 郭海湘, 诸克军. 基于 GA-TS 的开放式车辆路径优化算法及应用[J]. 系统管理学报, 2012(02):123-128+133.

[11]汪婷婷, 倪郁东, 何文玲. 需求可拆分车辆路径问题的蜂群优化算法[J]. 合肥工业大学学报:自然科学版, 2014(37):1018.

[12]何小年, 谢小良. 带装载量约束的物流配送车辆路径优化研究[J]. 计算机工程与应用, 2009, 045(034):236-238.

[13]张晨. 基于近似动态规划的随机车辆路径问题研究[D]. 清华大学, 2012.

[14]余明殊, 李建斌, 雷东. 装卸一体化的车辆路径问题及基于插入法的新禁忌算法[J]. 中国管理科学, 2010, 18(2):89-95.

[15]李松, 李瑞彩, 刘兴. 基于改进禁忌搜索算法的车辆路径优化[J]. 铁道运输与经济, 2008, 030(005):91-94.

[16]裴小兵, 贾定芳. 基于模拟退火算法的城市物流多目标配送车辆路径优化研究[J]. 数学的实践与认识, 2016, v.46(02):107-115.

[17]陈迎欣. 基于改进蚁群算法的车辆路径优化问题研究[J]. 计算机应用研究, 2012(06):37-40.

[18]亓霞, 陈森发, 黄鹄, 等. 基于免疫算法的物流配送车辆路径优化问题研究[J]. 土木工程学报, 2005, 36(7):43-46.

[19]马祥丽, 张惠珍, 马良. 蝙蝠算法在物流配送车辆路径优化问题中的应用[J]. 数学的实践与认识, 2015, 045(024):80-86.

[20]王素欣, 高利, 崔小光. 多集散点车辆路径优化的混合算法[J]. 北京理工大学学报, 2007, 027(002):130-134.

[21]孔志周, 官东. 基于改进遗传算法的车辆路径优化研究[J]. 统计与决策, 2007, 000(016):163-165.

[22]宁桂英. 差分进化算法及其应用研究[D]. 广西民族大学, 2008

[23] Pillac V, Gendreau M, Guéret, et al. A review of dynamic vehicle routing problems[J]. European Journal of Operational Research, 2013, 225(1):1–11.

[24] Canhong Lin, K.L.Choy, G.T.S.Ho, et al. Survey of Green Vehicle Routing Problem: Past and future trends[J]. Expert Systems with Application, 2014, 41(4pt.1): 118-1138

[25] Vidal T, Crainic T G, Gendreau M, et al. Heuristics for multi-attribute vehicle routing problems:A survey and synthesis[J]. European Journal of Operational Research, 2013, 231(1):1-21.

[26]Han, Hui, Ponce-Cueto, et al. Waste collection vehicle routing problem:Literature review[J]. Promet-traffic & transportation, 2015, 27(4):345-358.

[27]Kim, Gitae,Ong, Yew-Soon, et al. City Vehicle Routing Problem (City VRP):A Review[J]. Intelligent Transportation Systems, IEEE Transactions on, 2015, 16(4):16 54-1666.

[28]Ann Melissa Campbell, Jill Hardin Wilson. Forty Years of Periodic Vehicle Routing[J]. Networks, 2014, 63(1):2-15

[29] Storn R, Price K. Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces[J]. Journal of Global Optimization, 1997, 11(4):341-359.

[30]Storn R, Price K. Differential Evolution:A Simple and Efficient Adaptive Scheme for Global Optimization Over Continuous Spaces[J]. Journal of Global Optimization, 1995, 23(1).

[31]Tang K, Mei Y, Yao X. Memetic Algorithm With Extended Neighborhood Search for Capacitated Arc Routing Problems[J]. IEEE Transactions on Evolutionary Computation, 2009, 13(5):1151-1166.

[32] Yao X. Simulated annealing with extended neighbourhood[J]. International Journal of Computer Mathematics, 1991, 40(3-4):169-189.

[33]Mei Y, Tang K, Yao X. A Global Repair Operator for Capacitated Arc Routing Problem[J]. IEEE Transactions on Systems, Man, and Cybernetics,Part B: CyberNetic S, 2009, 39(3):723–734.

[34]Frederickson G N. Approximation Algorithms for Some Postman Problems[J]. Journal of the Acm, 1979, 26(3):538-554.

[35] DeArmon J S. A comparison of heuristics for the capacitated Chinese postman problems[D]. Univ. Maryland, College Park, MD, 1981.

[36] Benavent E, Campos V, Corberan A, et al. The Capacitated Arc Routing Problem: Lower bounds[J]. Networks, 1992, 22(7):669-690.

[37] Eglese R W. Routing winter gritting vehicles[J]. Discrete Applied Mathematics, 1994, 48(3):231–244.

[38] Osman I H, Kelly J P. Meta-Heuristics || A Tabu Search based Heuristic for Arc Routing with a Capacity Constraint and Time Deadline[J]. 1996, 10.1007/978-1-461 3-1361-8(Chapter 38):633-649.

[39] Eglese R W. An Interactive Algorithm for Vehicle Routeing for Winter - Gritting[J]. Journal of the Operational Research Society, 1996, 47(2):217-228.

[40]Richard Eglese. A deterministic tabu search algorithm for the capacitated arc routing problem[M]. Elsevier Science Ltd. 2008.

[41] Mei Y, Li X , Yao X. Cooperative Coevolution With Route Distance Grouping for Large-Scale Capacitated Arc Routing Problems[J]. IEEE transactions on evolutionary computation, 2014, 18(3):435-449.

[42] Beullens P, Muyldermans L, Cattrysse D, et al. A guided local search heuristic for the capacitated arc routing problem[J]. European Journal of Operational Research, 2003, 147(3):629-643.

致谢

毕业设计是对于毕业生实验能力的检测包括阅读文献能力，查阅文献能力，代码编写能力，代码调试能力，实验设计能力。我有幸选择龚文引教授作为我的指导老师，龚老师无论在毕设开始，中间过程，以及收尾都给了我很大的帮助。

感谢龚老师的耐心指导，龚老师给予的文献搜索，文献下载方法给予了很大帮助。同时还给予了大方向上的指导以及细节上的把控，并且紧张且舒适的毕设氛围收益良多。

感谢廖作文、李水佳、甄慧翔、张子佳、明飞博士，何为、吴建业、王开、郑小操、杨茜、胡真祯、戴晓虎硕士。在算法数据结构的设计，数据集收集给予的支持。