

求解多车型车辆路径问题的变邻域搜索算法

陈 萍^{1,2}, 黄厚宽³, 董兴业³

(1.南开大学泰达学院物流管理系, 天津 300457; 2. 南开大学现代物流研究中心, 天津 300071;

3.北京交通大学计算机与信息技术学院, 北京 100044)



摘 要: 多车型车辆路径问题 (FSMVRP) 是车辆类型不同的车辆路径问题 (VRP), 基于变邻域搜索, 提出一种启发式算法 VNS_FSM 求解 FSMVRP。该算法设计了实现变邻域搜索中的“抖动”和局部优化过程的邻域结构组合, 同时, 还提出了一种新的车型调整策略。在通用的基准测试问题的实验验证了 VNS_FSM 算法的有效性, 并给出问题 G_07-G_12 的正确解。实验结果表明, VNS_FSM 算法能够获得大多数测试问题的已知最好解。与已有算法相比, VNS_FSM 算法性能相当或更优。

关键词: 多车型车辆路径问题; 启发式算法; 变邻域搜索; 抖动

中图分类号: TP391.9

文献标识码: A

文章编号: 1004-731X (2011) 09-1945-06

Variable Neighborhood Search Algorithm
for Fleet Size and Mixed Vehicle Routing ProblemCHEN Ping^{1,2}, HUANG Hou-kuan³, DONG Xing-ye³

(1. Department of Logistics Management, TEDA College, Nankai University, Tianjin 300457, China. 2. Modern Logistics Research Center, Nankai University, Tianjin 300071, China; 3. School of Computer and Information Technology, Beijing Jiaotong University, Beijing 100044, China)

Abstract: The fleet size and mixed vehicle routing problem (FSMVRP) is a variant of the Vehicle Routing Problem (VRP), in which the vehicles are heterogeneous. Based on the variable neighborhood search, a heuristic algorithm VNS_FSM was proposed for solving the FSMVRP. In the VNS_FSM, the neighborhood structure combinations were designed to implement the shaking and local search procedure in the VNS. Additionally, a new vehicle type adjustment method was presented. Computational experiments were carried out on benchmark problems to validate the effectiveness of the proposed VND_FSM. The correct solutions to problems G_07 to G_12 were given. Experimental results show that the proposed heuristic VNS_FSM can obtain the best known solutions to most benchmark problems. The VNS_FSM performs quite competitively or even better when compared with other state-of-the-art heuristics.

Key words: fleet size and mixed vehicle routing problem; heuristic; variable neighborhood search; shaking

引 言

在基本的车辆路径问题 (Vehicle Routing Problem, VRP) 中, 一般假设所有车辆的类型相同。然而在现实生活中, 企业面临的实际问题却常常需要不同类型的车辆来共同完成配送任务。这时, 配送车辆的调度和路径问题就需要抽象为多车型车辆路径问题 (Fleet Size and Mixed Vehicle Routing Problem, FSMVRP)。FSMVRP 讨论的是选择哪些类型的车辆、每种类型车辆的数量以及如何安排这些车辆的路径, 使得车队在完成配送任务的前提下总成本最低。如果所有车辆

的类型相同, 并且车辆的固定成本可忽略不计, FSMVRP 就还原成基本的 VRP, 因此, 可将基本的 VRP 视为 FSMVRP 的一种特例。由于基本的 VRP 是 NP 完全问题^[1], 因此, FSMVRP 也是 NP 完全问题。另外, 由于 FSMVRP 中的车辆类型不同, 因此, FSMVRP 比基本 VRP 的求解更加困难和复杂。

FSMVRP 的研究还处于初步阶段, 目前尚没有求解该问题的精确算法。一些研究者提出的算法能够确定解的下界, 其中, Choi 和 Tcha^[2]的算法给出的结果较好。另外一些研究者通过调整求解基本 VRP 的传统启发式方法来解决 FSMVRP, 其中, Renaud 和 Boctor^[3]的“扫描法”产生的解的质量较高。元启发式算法 (metaheuristic) 是求解 VRP 的重要方法, 目前只有禁忌搜索 (Tabu Search, TS) 被应用于 FSMVRP。Gendreau 等人^[4]提出的禁忌搜索算法比较复杂,

收稿日期: 2009-04-29

修回日期: 2009-05-16

基金项目: 中央高校基本科研业务费专项资金资助 (NKZXB10104)

作者简介: 陈萍 (1981-), 女, 山东东明人, 博士, 讲师, 研究方向为物流管理与系统优化等; 黄厚宽 (1940-), 男, 四川遂宁人, 教授, 博导, 研究方向为人工智能、机器学习等; 董兴业 (1974-), 男, 河南漯河人, 博士, 讲师, 研究方向为调度理论和算法、组合优化。

http://www.china-simulation.com

• 1945 •

其中用到了 GENIUS 过程^[5]以及 Rochat 和 Taillard^[6]提出的“自适应记忆过程”；Wassan 和 Osman^[7]的禁忌搜索算法通过使用不同的邻域选择机制、禁忌限制以及禁忌长度机制有多种不同的变形。国内的叶志坚等人^[8]提出了一种大旅程法和禁忌搜索法相结合的混合启发式算法。最近，Brandão^[9]提出了一种确定性的禁忌搜索算法。

基于变邻域搜索算法，本文提出一种求解 FSMVRP 的启发式算法 VNS_FSM。变邻域搜索最重要的是“抖动”过程和局部优化过程的设计，本文设计了实现这两个过程的邻域结构组合；另外还提出一种新的车型调整策略。通过 28 个基准测试问题上的实验，验证 VNS_FSM 算法的有效性。

本文后面部分的内容组织如下：第 1 节给出 FSMVRP 的问题描述；第 2 节详细描述提出的 VNS_FSM 算法；第 3 节给出实验结果并与其它算法进行比较分析；最后一节总结全文。

1 问题描述

FSMVRP 可以描述如下：假设 $G = (V, A)$ 为一完全连通图， $V = \{0, 1, \dots, n\}$ 是节点集合， $A = \{(i, j) | i, j \in V\}$ 是弧集合，其中节点 0 表示配送中心，节点 i ($i \neq 0$) 表示客户，每个客户都有一个非负需求 q_i 。每一条弧 $(i, j) \in A$ 都对应一个距离 d_{ij} ($d_{ii} = 0, \forall i \in V$)。有一个由 K 种类型的车辆组成的车队，假设每种车型的车辆数目不限。对类型 k ($k=1, \dots, K$) 的车辆，其容量约束用 Q_k 表示，固定成本和可变成本系数分别由 F_k 和 c_k 表示。不失一般性，假设 $Q_1 < Q_2 < \dots < Q_K$ 且 $F_1 < F_2 < \dots < F_K$ 。对于弧 $(i, j) \in A$ 而言，其可变成本不但与距离 d_{ij} 有关，还依赖于车辆类型，根据式 $c_{ij} = c_k \cdot d_{ij}$ 计算得到。FSMVRP 问题的求解目标是找到一组总运输成本最小的车队组成及每辆车的行驶路线方案，满足以下条件：

- (1) 每个客户都只能由一辆车访问一次；
- (2) 各类型车辆的固定成本与容量约束各不相同，每种车型的数目无限制，但每车限派一次；
- (3) 所有车辆都从配送中心出发，在不超过车辆容量的情况下，依序服务客户，最后返回配送中心；
- (4) 车辆的总运输成本包含固定成本和可变成本。

2 FSMVRP 的变邻域搜索算法 VNS_FSM

变邻域搜索(Variable Neighborhood Search, VNS)是一种结构简单但功能强大的元启发式算法^[10]，其核心思想是在局部搜索过程中有规律地改变邻域以增强算法的搜索能力。VNS 的基本过程如下：从任意初始解开始，通过“抖动”过程在第一个邻域中随机产生邻域解，并对该邻域解进行局部优化直到产生局部最优解；若该局部最优解优于当前发现的最好解，则继续在该邻域中产生随机解；否则，转向下一个邻域（一般情况下比前一个邻域大）重复上述过程。算法结束

条件一般为最大迭代次数、解最大连续无改进次数或最长 CPU 时间。

2.1 初始解

VNS_VNS 算法利用两阶段构造算法产生 FSMVRP 初始可行解：首先，利用 GENI 法^[5]生成一条包含所有节点的环路 tour；然后，利用“路径划分过程”将该环路划分为若干条满足可行性条件的 FSMVRP 路径，形成完整初始解。“路径划分过程”如下：从配送中心 0 出发，选择一辆容量约束不小于 tour 中第一个客户需求的最小车型的车型，服务当前客户。不断将 tour 中的客户依次添加至当前的车辆路径上，并从 tour 中删除该客户，重复该过程，直到不能继续添加为止，此时得到一条 FSMVRP 路径；若 tour 中仍有客户存在，则重复上述过程构造一条新的路径；否则，结束路径划分过程。

2.2 抖动过程

抖动过程是变邻域搜索算法设计的关键。VNS_FSM 算法中使用以下两种类型的邻域结构来实现抖动：插入(insert)和交换(cross)。插入是指将某段连续的节点从当前路径移到另一条路径；交换是指从两条不同的路径上分别选取一段连续的节点，并将二者位置互换。图 1 和 2 分别给出了“插入”和“交换”过程的示意图。由于“插入”和“交换”的次数直接影响解结构的改变程度，因此，根据使用的邻域结构类型和执行次数的不同，VNS_FSM 算法利用 4 种方式实现“抖动”，分别用 insertI、insertII、crossI 和 crossII 表示，其中，insertI 和 insertII 分别表示执行一次插入和两次插入，crossI 和 crossII 分别表示执行一次交换和两次交换。另外，节点段中包含的节点个数以随机的方式在 $[1, 3]$ 的范围内产生。

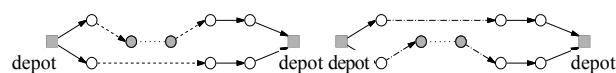


图 1 插入示意图

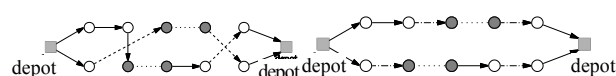


图 2 交换示意图

为扩大解的搜索范围，VNS_FSM 算法在搜索过程中允许接受不可行解。VNS_FSM 算法中利用函数 $f(s) = \sum_{k=1}^r (F_k + c_k \cdot r_k + P \cdot l_k)$ 评价解的质量，其中， F_k 是服务路径 k 的车辆的固定成本， c_k 是该车型的可变成本系数， r_k 是路径 k 的路径长度， $l_k = \max\{0, L_k - Q_k\}$ 是车辆载重超出容量约束的部分，其中 L_k 是车辆的实际载重， P 是惩罚项系数。在 VNS_FSM 算法中， P 的值随搜索过程在 $[P_L, P_U]$ 之间动态变化，并设 P 初值为 P_U ，使得搜索过程在开始时偏向接受可行解。

当利用插入或交换操作产生邻域解时, 车辆服务的客户发生改变, 导致当前车型的车型可能不再满足客户需求, 因此需要对车辆进行车型调整。本文提出一种新的车型调整策略: 若新路径可行, 则逐次使车型减 1, 直至最小车型。根据式(1)比较使用各类型车辆时对应的代价, 并选择最小代价对应车型的车型, 服务当前路径; 若新路径不可行, 则逐次使车型增 1, 直至最大车型。同样根据式(1)比较使用各类型车辆时对应的代价, 并选择最小代价对应车型的车型, 服务当前路径。

$$cost_k = F_k + c_k \cdot r_k + P \cdot l_k \quad (1)$$

2.3 局部优化

当产生新的邻域解后, 首先对参与变换的两条路径分别进行 2-opt 优化; 然后, 利用 2-opt*对整个解进行进一步优化。图 3 和 4 分别给出了 2-opt 和 2-opt*优化的示意图, 其中, 2-opt 优化是指删除某条路径上的两条不相邻的弧, 并添加两条新的弧, 从而生成一条新的代价更小的路径; 2-opt*优化是指分别删除两条不同路径上的弧, 重新组合生成两条总代价更小的路径。在算法 VNS_FSM 中, 考查任意一对路径上的任意两段弧, 若能改进当前解, 则执行 2-opt*操作, 并对生成的两条新的路径分别进行 2-opt 优化。为减少时间代价, 提高 2-opt*搜索效率, 算法中定义了客户 i 的 δ -邻域 $N_\delta(i)$, 即除客户本身 i 和配送中心之外, 与该客户距离最短的 δ 个客户节点。在进行 2-opt*优化时, 算法 VNS_FSM 只考虑满足 $v_i \in N_\delta(v_{j+1})$ 且 $v_j \in N_\delta(v_{i+1})$ 的情形。

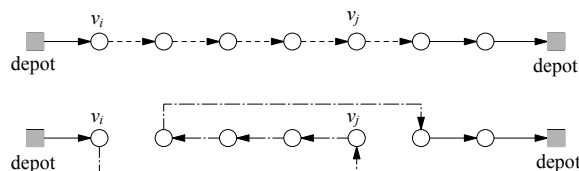


图 3 2-opt 优化示意图

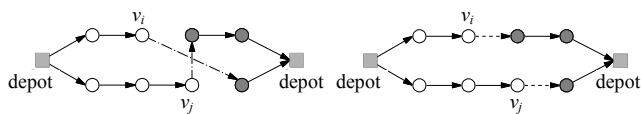


图 4 2-opt* 优化示意图

2.4 解的接受准则

解的接受准则确定接受哪个解进入下次迭代。若 s^* 优于当前解 s , 则用 s^* 替代 s 进入下次迭代; 否则, 利用模拟退火算法^[11]中的解接受准则选择进入下次迭代的解。模拟退火接受准则能够以一定概率接受较差解, 从而避免过早陷入局部最优。模拟退火接受准则如式(2)所示, 其中, q_0 是区间[0,1)上的随机数; 温度 T 初始为 T_0 , 每隔 n_T 代按照 $T_{n+1} = \beta \cdot T_n$ 进行更新, β 是可设定的降温系数。

$$SA(s^*, s) = \begin{cases} s^*, & \text{若 } q_0 > \exp\left(\frac{f(s^*) - f(s)}{T}\right) \\ s, & \text{若 } q_0 \leq \exp\left(\frac{f(s^*) - f(s)}{T}\right) \end{cases} \quad (2)$$

2.5 VNS_FSM 算法详细步骤

步骤 1: 利用 2.1 节中方法生成初始解 s_0 ; 确定邻域结构集合 $N_k(k = 1, 2, \dots, k_{max})$ 和算法结束条件; 令当前解 $s = s_0$, 令历史最优解 $s_b = s_0$ 。

步骤 2: 令 $k=1$;

步骤 3: 在 s 的第 k 个邻域中随机产生一个解 s' , 并对发生改变的两条路径分别进行 2-opt 优化;

步骤 4: 对 s' 进行 2-opt*优化, 产生局部最优解 s'^* ;

步骤 5: 若 s'^* 优于 s , 则用 s'^* 代替 s , 令 $k=1$; 否则, 根据模拟退火接受准则, 若接受 s'^* 代替 s , 则令 $k=1$; 否则, 令 $k=k+1$ 。

步骤 6: 若 s'^* 优于 s_b , 则令 $s_b = s'^*$ 。

步骤 7: 若 $k \leq k_{max}$, 返回步骤 3; 否则, 转步骤 8。

步骤 8: 若不满足算法结束条件, 转步骤 2; 否则, 算法停止, 输出 s_b 作为最终解。

3 实验仿真与分析

为测试算法 VNS_FSM 的性能, 本文利用基准测试问题进行仿真实验。本节首先说明实验测试问题及参数设置, 然后给出算法 VNS_FSM 的实验结果及与其它算法的比较分析结果。

3.1 实验测试问题

用于测试算法性能的 FSMVRP 的基准测试问题包括两组, 分别是 Golden 等人提出的 20 个问题 G_01 至 G_20^[12] 和 Taillard 提出的 8 个问题 T_13 至 T_20^[13]。问题 G_01 至 G_20 中车辆的可变成本系数都是 1.0; 问题 T_13 至 T_20 中, 车辆固定成本可忽略不计, 运输成本只包含可变成本。具体数据文件可从 Heterogeneous Vehicle Routing Problem 网站^[14]上下载。

3.2 参数设置

算法 VNS_FSM 用 C++实现, 运行环境为 Pentium IV 2.93GHz 的处理器, 1G 内存。当连续 n_t 代当前最好解没有得到改进时, 算法结束。根据 Gendreau 等人^[5]的研究结果, 当 p -邻域的 p 值取 5 时, 能够在求解质量和计算时间之间取得较好的平衡。因此, 本文实验设 p 值为 5。模拟退火接受准则中的相关参数设置如下: 令初始温度 $T_0=10$, 每隔 $n_T = n/10$ 代后更新温度 $T_{n+1} = 0.9 \cdot T_n$ 。算法结束条件 $n_t = 500$; δ -邻域中的参数 $\delta = \min\{n/2, 20\}$ 。

3.3 实验结果与比较

本节将报告算法 VNS_FSM 的实验结果, 并与 Choi 和

Tcha 的算法^[2]、Renaud 和 Bactor 的算法^[3]、Gendreau 等人的算法^[4]、Wassan 和 Osman 的算法^[7]、叶志坚等人的算法^[8]、Brandão 的算法^[9]及 Taillard 的算法^[13]进行比较。由于有的算法给出了不同的算法版本(如 Wassan 和 Osman 的算法^[7]、Renaud 和 Bactor 的算法^[3])，有的算法运行多次，因此，本文只比较各算法获得的最好解。

表 1 中给出了各算法在 Golden 等人提出的 20 个问题中的 G_01-G_02 及 G_07-G_12 上得到的最好结果，其中列 n 表示问题规模，列 Best 是已知最好解，其余各列给出了各算法结果(列 cost)及时间代价(列 CPU，秒)。从表中结果可以看出，问题 G_01-G_02 规模较小，各算法获得的结果相同，且花费的时间代价很小。对于问题 G_07-G_12，研究中发现之前的文献中报告的数据存在问题，这一点已与 Heterogeneous Vehicle Routing Problem 专题网站^[14]的负责人和文献[9]的作者 Brandão 讨论并得到认同，故表 1 中不再列出这些数据，只给出本文算法获得的结果，具体的问题解详见附录。

表 2 给出了各算法在 Golden 等人给出的其余 12 个问题，即 G_03-G_06 及 G_13-G_20 上得到的最好结果，其中，行 avg. 是算法在所有测试问题上的平均解。本文使用相对偏差百分比(Relative Percentage Deviation, RPD)表示解偏离已知最好解的程度，计算如下： $RPD = (F_A - F_{Best}) / F_{Best} \times 100$ ，其中 F_A 表示算法 A 得到的问题解的目标值， F_{Best} 表示已知最好解的目标值。行 ARPD 表示各算法在所有测试问题上的平均 RPD，行 Best 中给出了各算法获得已知最好解的问题个数。从表 2 可以看出，对于 Golden 等人的 12 个问题，本文算法 VNS_FSM 可以得到 6 个问题的已知最好解，即 G_03-G_06、G_13-G_14，ARPD 是 0.22。Taillard^[13]的算法获到 5 个问题的已知最好解，ARPD 是 0.23，与本文算法相当。Gendreau 等人^[4]的算法、Wassan 和 Osman^[7]的算法都找到了 6 个问题的已知最好解。Gendreau 等人^[4]的算法、Wassan 和 Osman^[7]的算法的 ARPD 分别是 0.30 和 0.29，略高于本文算法。Renaud 和 Bactor 的算法^[3]仅获得 1 个问题的已知最好解，叶志坚等人^[8]的算法没有找到任何问题的已知最好

解，这两个算法的 ARPD 也都高于其它算法，性能表现略差。Choi 和 Tcha^[2]、Brandão^[9]的算法都找到了 9 个问题的已知最好解，ARPD 分别是 0.05 和 0.03，略优于本文算法。但是，在除问题 G_20 外的其它问题上，三个算法的结果差异都很小，ARPD 分别是 0.07、0.03 和 0.09。

表 3 给出了各算法在 Taillard 给出的 8 个问题 T_13-T_20 上得到的最好结果，各行和各列的含义同表 2。表 3 中的结果表明，对于 Taillard 的 8 个问题，本文算法 VNS_FSM 获得了 5 个问题的已知最好解，ARPD 是 0.21。Taillard^[13]的算法仅得到问题 T_14 的已知最好解，ARPD 是 0.93。Choi 和 Tcha^[2]的算法也仅仅找到 1 个问题的已知最好解，但 ARPD 为 0.3，性能远远优于 Taillard^[13]的算法。Gendreau 等人^[4]的算法及 Wassan 和 Osman^[7]的算法都找到了 4 个问题的已知最好解，ARPD 分别是 0.21 和 0.47。Brandão^[9]的算法找到 6 个问题的已知最好解，ARPD 是 0.15，整体性能表现最好。除问题 T_20 外，本文算法和 Brandão^[9]的算法的求解结果完全相同。

3.4 算法时间的比较

由于各算法运行环境不同，因此不能直接比较各算法的 CPU 时间。根据 Dongarra 的 Linpack 基准测试^[15]，可以通过 Mflops (Million Floating Point/Second，每秒百万个浮点操作)标准得到不同微机的大致相对速度，然后粗略地比较各算法时间。表 4 中给出了文献[2,4,7,9,13]及本文实验中使用的各微机的机型主频、运算速度(Mflops)以及转换为 Sun Sparc workstations 50 MHz 上 CPU 时间的转换因子(factor)。Taillard^[13]的算法、Gendreau 等人^[4]的算法及 Wassan 和 Osman^[7]的算法运行环境相同。由于文献[3,8]中没给出计算时间，故不在表中列出 Renaud 和 Bactor^[3]及叶志坚等人^[8]的算法。列 Golden 和 Taillard 分别给出了转换后各算法求解 Golden 问题和 Taillard 问题平均时间。表 4 中的结果表明，文献[13,4,7]中算法的时间花费相对较少，本文算法 VNS_FSM 与 Brandão^[9]的算法转换后的时间花费相当，都远远少于 Choi 和 Tcha^[2]的算法。

表 1 Golden 等人的 8 个问题实验结果

问题	n	Best	Wassan 和 Osman		Renaud 和 Bactor		Brandão		叶志坚等		VNS_FSM	
			cost	CPU	cost	CPU	cost	CPU	cost	CPU	cost	CPU
G_01	12	602	602	17	602	0	602	7	605	--	602	0
G_02	12	722	722	16	722	0	722	6	722	--	722	0
G_07	30	--	--	--	--	--	--	--	--	--	6291	3
G_08	30	--	--	--	--	--	--	--	--	--	2005	4
G_09	30	--	--	--	--	--	--	--	--	--	1937	4
G_10	30	--	--	--	--	--	--	--	--	--	2049	4
G_11	30	--	--	--	--	--	--	--	--	--	4109	3
G_12	30	--	--	--	--	--	--	--	--	--	3493	4
avg.	--	--	--	--	--	--	--	--	--	--	2651	3

表 2 Golden 等人的 12 个问题的实验结果

问题	n	Best	Taillard		Gendreau 等		Wassan 和 Osman		Choi 和 Tcha	
			cost	CPU	cost	CPU	cost	CPU	cost	CPU
G_03	20	961.03	961.03		961.03	164	961.03	88	961.03	0
G_04	20	6437.33	6437.33		6437.33	253	6437.33	80	6437.33	1
G_05	20	1007.05	1008.59		1007.05	164	1007.05	52	1007.05	1
G_06	20	6516.47	6516.47		6516.47	309	6516.47	88	6516.47	0
G_13	50	2406.36	2413.78	470	2408.41	724	2422.10	2084	2406.36	10
G_14	50	9119.03	9119.03	570	9119.03	1033	9119.86	1660	9119.03	51
G_15	50	2586.37	2586.37	334	2586.37	901	2586.37	2349	2586.37	10
G_16	50	2720.43	2741.50	349	2741.50	815	2730.10	689	2720.43	11
G_17	75	1734.53	1747.24	2072	1749.50	1022	1755.10	1874	1744.83	207
G_18	75	2369.65	2373.63	2744	2381.43	691	2385.52	2261	2371.49	70
G_19	100	8659.74	8661.81	12528	8675.16	1687	8659.74	8570	8664.29	1179
G_20	100	4039.49	4063.18	2117	4086.76	1421	4061.64	2692	4039.49	264
avg.		4046.46	4052.50	2648	4055.84	765	4053.52	1874	4047.85	150
ARPD (%)			0.23		0.30		0.29		0.05	
BestNum			5		6		6		9	

问题	n	Best	Renaud 和 Bactor		Brandão		叶志坚等		VNS_FSM	
			cost	CPU	cost	CPU	cost	CPU	cost	CPU
G_03	20	961.03	963.61	--	961.03	21	965.97	--	961.03	0
G_04	20	6437.33	6437.33	--	6437.33	22	6491.12	--	6437.33	0
G_05	20	1007.05	1007.96	--	1007.05	20	1013.00	--	1007.05	0
G_06	20	6516.47	6537.74	--	6516.47	25	6569.14	--	6516.47	0
G_13	50	2406.36	2406.43	--	2406.36	145	2437.00	--	2406.36	14
G_14	50	9119.03	9122.01	--	9119.03	220	9159.40	--	9119.03	13
G_15	50	2586.37	2618.03	--	2586.37	110	2628.86	--	2586.84	13
G_16	50	2720.43	2761.96	--	2728.14	111	--	--	2721.96	12
G_17	75	1734.53	1757.21	--	1734.53	322	--	--	1745.32	53
G_18	75	2369.65	2413.39	--	2369.65	267	--	--	2373.74	36
G_19	100	8659.74	8687.31	--	8661.81	438	--	--	8667.31	173
G_20	100	4039.49	4094.54	--	4042.59	601	--	--	4107.47	95
avg.		4046.46	4067.29		4047.53	192	4180.64		4054.16	34
ARPD (%)			0.69		0.03		0.87		0.22	
BestNum			1		9		0		6	

表 3 Taillard 的 8 个问题的实验结果

问题	n	Best	Taillard		Gendreau 等		Wassan 和 Osman		Choi 和 Tcha		Brandão		VNS_FSM	
			cost	CPU	cost	CPU	cost	CPU	cost	CPU	cost	CPU	cost	CPU
T_13	50	1491.86	1494.58	473	1491.86	626	1499.69	140	1493.36	4	1491.86	101	1491.86	8
T_14	50	603.21	603.21	575	603.21	669	608.57	575	603.21	37	603.21	135	603.21	15
T_15	50	999.82	1007.35	335	999.82	736	999.82	289	1001.33	6	999.82	137	999.82	10
T_16	50	1131.00	1144.39	350	1136.63	852	1131.00	246	1135.62	6	1131.00	95	1131.00	12
T_17	75	1031.00	1044.93	2245	1031.00	1453	1047.74	2156	1038.60	103	1038.60	312	1038.60	58
T_18	75	1800.80	1831.24	2876	1801.40	1487	1814.11	2469	1801.40	81	1800.80	269	1800.80	62
T_19	100	1100.56	1110.96	5833	1105.44	1681	1100.56	2553	1106.62	299	1105.44	839	1105.44	109
T_20	100	1530.16	1550.36	3402	1541.18	1706	1530.16	2910	1536.10	112	1530.16	469	1537.53	96
avg.		1211.05	1223.38	2011	1213.82	1151	1216.46	1417	1214.53	81	1212.61	295	1213.53	46
ARPD (%)			0.93		0.21		0.47		0.30		0.15		0.21	
BestNum			1		4		4		1		6		5	

表 4 各算法所用机器性能比较

	CPU	Mflops/s	factor	Golden	Taillard	
Taillar(1999)				2648	2011	
Gendreau 等（1999）	Sun Sparc workstations	50 MHz	10	1	765	1151
Wassan 和 Osman（2002）				1874	1417	
Choi 和 Tcha(2007)	Pentium IV 2.6GHz	1190	119	17850	9639	
Brandão（2009）	Pentium M 1.4GHz	250	25	4800	7375	
VNS FSM	Pentium IV 2.93GHz	1573	157	5338	7222	

4 结论

基于变邻域搜索, 本文提出了一个求解多车型车辆路径问题的启发式算法 VNS_FSM。该算法设计了实现“抖动和局部搜索过程的邻域结构组成, 还提出了一种基于惩罚不可行解的新的车型调整策略。通过定义客户节点的 δ -邻域限定局部搜索中的检查范围, 从而避免了不必要的检查, 提高了搜索效率。在通用的基准测试问题上的实验结果表明, 本文算法 VNS_FSM 得到了大多数测试问题的已知最好解, 并给出了问题 G_07 至 G_12 的正确解, 从而验证了算法的有效性。在其它 Golden 基准测试问题上的实验结果表明, 在与已知最好解的平均相对偏差百分比(ARPD)方面, VNS_FSM 算法优于 Taillard^[13]、Gendreau 等人^[4]、Renaud 和 Bactor 等人^[3]的算法; 与 Choi 和 Tcha^[2]、Brandão^[9]的算法性能相当, 但计算时间远小于 Choi 和 Tcha^[2]的算法; 在 8 个 T_13 至 T_20 基准测试问题上的实验结果表明, VNS_FSM 算法和 Brandão^[9]的算法得到相同的求解结果, 求解质量优于 Taillard^[13]、Choi 和 Tcha^[2]的算法。综上可知, 本文算法 VNS_FSM 是当前求解多车型车辆路径问题的最好算法之一。

参考文献:

- [1] Garey M R, Johnson D S. Computers and intractability: A Guide to the theory of NP-completeness [M]. San Francisco, Ca, USA: W. H. Freeman and Company, 1979.
- [2] Choi E, Tcha D. A column generation approach to the heterogeneous fleet vehicle routing problem [J]. Computers and Operations Research (S0305-0548), 2007, 34(7): 2080-2095.
- [3] Renaud J, Bactor F. A sweep-based algorithm for the fleet size and mix vehicle routing problem [J]. European Journal of Operational Research (S0377-2217), 2002, 140(3): 618-628.
- [4] Gendreau M, Laporte G, Musaraganyi C, Taillard E. A tabu search heuristic for the heterogeneous fleet vehicle routing problem [J]. Computers and Operations Research (S0305-0548), 1999, 26(12): 1153-1173.
- [5] Gendreau M, Hertz A, Laporte G. New insertion and postoptimization procedures for the traveling salesman problem [J]. Operations Research (S0030-364X), 1992, 40(6): 1086-1094.
- [6] Rochat Y, Taillard E. Probabilistic diversification and intensification in local search for vehicle routing [J]. Journal of Heuristics (S1381-1231), 1995, 1(1): 147-167.
- [7] Wassan N, Osman, I. Tabu search variants for the mix fleet vehicle routing problem [J]. Journal of the Operations Research Society (S0160-5682), 2002, 53(7): 768-782.
- [8] 叶志坚, 叶怀珍, 周道平, 等. 多车型车辆路径问题的算法[J]. 公路交通科技, 2005, 22: 147-151.
- [9] Brandão J. A deterministic tabu search algorithm for the fleet size and mix vehicle routing problem [J]. European Journal of Operational Research (S0377-2217), 2009, 195(3): 716-728.

- [10] Hansen P, Mladenović N. Variable neighborhood search: Principles and applications [J]. European Journal of Operational Research (S0377-2217), 2001, 130: 346-354.
- [11] 王凌, 等. 智能优化算法及其应用[M]. 北京: 清华大学出版社, 2001: 17-35.
- [12] Golden B, Assad A, Levy L, et al. The fleet size and mix vehicle routing problem [J]. Computers and Operations Research (S0305-0548), 1984, 11(1): 49-66.
- [13] Taillard E. A heuristic column generation method for the heterogeneous fleet VRP [J]. RAIRO-Operations Research (S0399-0559), 1999, 33(1): 1-14.
- [14] The Heterogeneous VRP Web. <http://apice54.ingce.unibo.it/hvrp>.
- [15] Dongarra J. Performance of various computers using standard linear equations software [R]// Report CS-89-85. USA: University of Tennessee, 2008.

附录:

问题	路径	车型
G_07	0-2-0	1
	0-19-8-17-0	2
	0-12-14-27-26-10-0	4
	0-20-30-0	3
	0-21-8-25-0	2
	0-3-4-29-1-0	3
	0-24-28-0	2
	0-22-6-5-11-16-15-9-7-13-23-0	2
G_08	0-30-21-17-0	3
	0-22-24-3-4-28-29-6-5-11-16-15-9-7-13-23-1-2-0	4
	0-20-25-8-18-19-10-26-27-14-12-0	4
G_09	0-2-0	1
	0-24-3-4-28-29-6-5-11-16-15-9-7-13-23-0	5
	0-18-19-10-26-27-14-12-22-0	5
	0-17-21-30-8-25-20-1-0	5
G_10	0-12-14-27-26-10-0	4
	0-22-3-28-4-0	3
	0-2-1-21-30-0	4
	0-17-23-13-7-9-15-16-11-5-6-29-24-0	4
	0-20-25-8-19-18-0	4
G_11	0-19-18-0	2
	0-29-28-4-0	3
	0-12-10-8-0	2
	0-21-30-25-0	3
	0-17-1-2-0	2
	0-24-3-6-5-16-15-9-7-13-11-22-0	2
	0-20-26-27-14-23-0	3
G_12	0-8-30-21-17-0	5
	0-3-28-29-7-15-16-11-5-6-4-0	6
	0-20-25-10-19-18-0	5
	0-24-22-23-0	2
	0-2-1-0	2
	0-13-9-14-27-26-12-0	5