

doi:10.11835/j.issn.1000-582X.2017.03.011

基于 Memetic 算法的两级车辆路径优化

陈立伟^{1,2}, 唐权华³

(1.西南科技大学 计算机科学与技术学院,四川绵阳 621010;2.西南交通大学 信息科学与技术学院,四川 成都 610031;3.江西师范大学 软件学院,江西 南昌 330031)

摘要:针对传统两级车辆路径优化过程,存在的精度不高,收敛时间过长的问题,提出一种基于Q学习理论和差分进化的Memetic算法。首先,对两级车辆路径优化问题进行研究,利用最优分割法获得第一级配送方案,以此确定中转站配送数量,然后求解第二级多配送中心车辆路径问题配送方案,获得两级优化问题的总里程及总配送车辆数量;其次,针对第二级MDVRP配送方案求解,利用Q学习理论和差分进化算法,设计新的Memetic算法,来实现对多配送中心车辆路径问题配送方案的全局优化;最后,通过仿真验证了所提算法的有效性。

关键词:Q学习;差分进化;Memetic算法;两级;车辆路径优化

中图分类号:TP18

文献标志码:A

文章编号:1000-582X(2017)03-095-10

Two-echelon vehicle path optimization based on Memetic algorithm

CHEN Liwei^{1,2}, TANG Quanhua³

(1.College of Computer Science and Technology, Southwest University of Science and Technology, Mianyang 621010, Sichuan, P. R. China; 2.School of Information Science and Technology, Southwest Jiaotong University, Chengdu 610031, P. R. China; 3.School of Software, Jiangxi Normal University, Nanchang 330022, P. R. China)

Abstract: Aiming at the problem of low accuracy and long convergence time of traditional method in solving the two-echelon vehicle routing problem, we proposed a kind of Memetic algorithm based on Q learning theory and differential evolution. Firstly, the two-echelon vehicle routing problem was studied, and the optimum partition method was used to obtain the reasonable distribution plan for SDVRP (split delivery vehicle routing problem) in first stage, and then the total mileage and delivery vehicles were determined for both the two stages. Secondly, according to the distribution scheme of the second level of MDVRP (multi-depot vehicle routing problem), the Memetic algorithm was designed with Q learning theory and differential evolution algorithm, which was used to achieve the global optimization of MDVRP distribution scheme. Finally, through simulation verified the effectiveness of the proposed algorithm.

Keywords: Q learning; differential evolution; Memetic algorithm; two-echelon; vehicle routing optimization

收稿日期:2016-08-23

基金项目:国家支撑计划课题资助项目(2012BAH20F01);西南科技大学博士基金资助项目(16ZX7105);四川省科技厅资助项目(2014GZX0009)。

Supported by National Key Technology Research and Development Program of the Ministry of Science and Technology of China(2012BAH20F01), Doctoral Fund Project of SWVSU(16ZX17105) and Project of Science & Technology Department of Sichuan Province(2014GZX0009).

作者简介:陈立伟(1974-),男,西南科技大学,副教授,主要从事信息处理方向的研究,(E-mail)27736784@qq.com。

随着经济社会发展及人民生活质量提升,物流与人民生活间的联系也越发紧密。在传统物资运输过程中,车辆路径优化问题(vehicle routing problem,VRP),通常假定物资运输过程是单级方向配送,即从仓库中心到客户站点的直接物资配送方式^[1]。但是因为物流长途运输所使用的车辆较大,如果路线规划不佳,极易造成城市交通堵塞、产生较严重的噪音及尾气排放污染,出于考虑城市舒适性居住环境角度,大部分城市都对货运大型车辆实行限行或禁行管制;远距离物资配送一般是由运输能力较大的车辆配送至郊区转运点,然后再利用小型配送车辆将物资发送至城区内不同的需求点,上述配送过程便构成两级配送车辆路径优化问题(Two-echelon vehicle routing problem,2E-VRP)。

2E-VRP与VRP相比,在求解方式和难度上不同,前者是一级与二级配送模式的耦合,应用传统优化模型和算法存在精度差且运算时间长的问题。学者对2E-VRP方向的研究,主要分精确方式和启发优化方式2种,如文献[2]提出两级车辆路径优化问题的商品流模型,利用分支切割方法进行求解,精度高但运行时间稍长;文献[3]对上述分支切割算法进行改进,进一步提高算法性能;文献[4]使用不等式方法强化分支切割法中两级车辆路径优化问题的连续性和松弛性;文献[5]提出开放式车辆路径问题的启发式方法,并利用聚类最小生成树方式实现2E-VRP的启发式求解文献[6]提出改进禁忌搜索算法,对2E-VRP优化问题进行求解,此算法包含2E-VRP问题下界文献[7]提出另一2E-VRP问题求解的禁忌搜索方法,取得理想优化结果;文献[8]设计混合蚁群算法实现2E-VRP优化问题求解,该算法利用禁忌搜索进行局部优化,实现后优化过程改进。精确算法精度高,但运行时间长,不适于大型配送网络。对此,启发式算法应运而生,如文献[9]提出分类规划并行启发式方法,将两级车辆路径优化问题拆解为2个子车辆配送问题,采用规划算法对2个子问题分别求解,虽加快了搜索速度,但因耦合及优化算法性能所限,求解精度仍然不佳。当前,国内对于2E-VRP问题的研究成果相对较少。

研究借鉴文献[9]的拆解方式,一级采用精确的最优割算法,以此作为二级优化的基础,二级采用改进的QDEMA算法进行全局寻优,来处理两级算法的耦合问题,并降低算法运行时间。

1 两级车辆路径问题

2E-VRP问题如图1所示, $S_1 \sim S_3$ 为一级中转站, $c_1 \sim c_9$ 为二级用户点,两级车辆路径采用不同线型区分(一级实线,二级虚线)。

2E-VRP数学模型等价于最小配送路径问题,即最后一辆车完成配送所需总路径最小^[10]

$$f = \min(C_{\max}), \quad (1)$$

式中, C_{\max} 表示总的配送路径。式(2)主要确保每个订单只被处理一次

$$\sum_{k=1}^n Y_{ik} = 1, i = 1, 2, \dots, n, \quad (2)$$

式中, Y_{ik} 表示每个订单处理次数。式(3)则确保当 $\sum_{i=1}^n Y_{ik} = 0, k = 1, 2, \dots, n-1$ 时, $\sum_{i=1}^n Y_{i,k+1} \neq 1$,即没有订单数 k ,就没有订单数 $k+1$ 。

$$\sum_{i=1}^n Y_{i,k+1} \leq H \sum_{i=1}^n Y_{ik}, \quad (3)$$

式(4)用来确保当 $\sum_{i=1}^n Y_{ik} = 0$ 时,当前订单处理次数 $y_k = 0$,

$$y_k \leq \sum_{i=1}^n Y_{ik}, k = 1, \dots, n, \quad (4)$$

式(5)用来确保当 $\sum_{i=1}^n Y_{ik} \geq 1$ 时,当前订单处理次数 $y_k = 1$,

$$\sum_{i=1}^n Y_{ik} \leq H \times y_k, k = 1, \dots, n, \quad (5)$$

式(4)、(5)含义是,当订单 B_k 为有效时, $y_k = 1$;而当订单 B_k 无效时, $y_k = 0$ 。式(6)~(8)给出交货车辆批次限制:

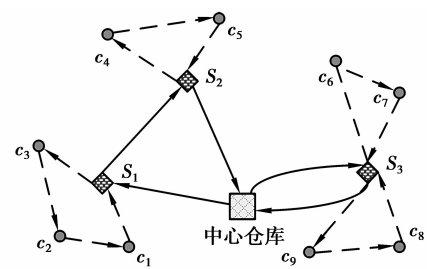


图1 2E-VRP问题

Fig.1 Problem of 2E-VRP

$$\sum_{g=1}^G V_{kg} \leq 1, k=1, \dots, n, \quad (6)$$

$$\sum_{g=1}^G V_{kg} \leq H \times y_k, k=1, \dots, n, \quad (7)$$

$$\sum_{k=1}^n \sum_{g=1}^G V_{kg} = \sum_{k=1}^n y_k, \quad (8)$$

式中, V_{kg} 表示单个车辆运行批次。式(6)限制每个订单只能由一辆货车交付,式(7)表示若该客户暂无订单,则不进行配送,车辆配送批次为0。

式(9)给出车辆容量限制

$$\sum_{i=1}^n e_i Y_{ik} \leq z_g + H(1 - V_{kg}), k=1, \dots, n, g=1, \dots, G, \quad (9)$$

式中, e_i 表示车辆 i 运输容量。式(10)和(11)给出批次 B_k 准备时间

$$r_k \geq \sum_{k'=1}^k \sum_{i=1}^n p_i Y_{ik'} - H(1 - y_k), k=1, \dots, n, \quad (10)$$

$$r_k \leq \sum_{k'=1}^k \sum_{i=1}^n p_i Y_{ik'} - H(1 - y_k), k=1, \dots, n, \quad (11)$$

式中, p_i 表示车辆 i 准备时间。式(12)~(14)给出出发时间 u_k 的限制

$$u_k \leq H \times y_k, k=1, \dots, n, \quad (12)$$

$$\begin{cases} u_1 = r_1 \\ u_k \geq r_k, \end{cases} k=2, \dots, n, \quad (13)$$

$$\begin{cases} u_k' + t_{01}^g + t_{10}^g - H(2 - V_{kg} - V_{k'g}) \leq u_k \leq \\ u_k' + t_{01}^g + t_{10}^g - H(2 - V_{kg} - V_{k'g}), \\ k=2, \dots, n; k'=1, \dots, k-1; g=1, \dots, G, \end{cases} \quad (14)$$

式中, u_k 表示车辆 k 准备时间。式(15)给出订单配送时间 T_k 的限制

$$\begin{cases} T_k \leq H \times y_k, k=1, \dots, n, \\ u_k + t_{01}^g + t_{10}^g - H(1 - V_{kg}) \leq u_k \leq u_k + t_{01}^g + t_{10}^g - H(1 - V_{kg}), \\ k=2, \dots, n; g=1, \dots, G. \end{cases} \quad (15)$$

2 Q 学习理论与 DE 算法

2.1 差分进化算法

差分进化算法(differential evolution algorithm, DE),特点是结构简单、收敛快和精度佳,得到广泛使用。假设 DE 算法有 NP 个种群个体,则第 G 代种群为^[11]

$$P_G = \{X_1(G), X_2(G), \dots, X_{NP}(G)\}, \quad (16)$$

式中, $X_i(G), i \in 1, \dots, NP$ 代表种群个体, P_G 为当前种群。DE 算法流程可通过下列操作实现^[12]。

步骤 1:(初始化)对 $G=0$ 代种群,个体 $X_i(0)$ 可在范围 $[X_{\min}, X_{\max}]$ 中均匀随机初始化

$$\begin{cases} X_{\min} = \{x_{\min-1}, \dots, x_{\min-D}\}, \\ X_{\max} = \{x_{\max-1}, \dots, x_{\max-D}\}, \end{cases} \quad (17)$$

式中, D 为 DE 种群维度。那么,在 $G=0$ 代个体 i 的第 j 个元素可由下式初始化

$$x_{ij}(0) = x_{\min-j} + \text{rand}_{ij}(0,1) \times (x_{\max-j} - x_{\min-j}), \quad (18)$$

式中, $\text{rand}_{ij}(0,1)$ 为 $[0,1]$ 区间的均匀分布函数,并在 $[0,1]$ 区间内随机初始化交叉概率因子 Cr 。

步骤 2:(变异)标准 DE 变异是随机选取 2 个种群个体 ($X_{\text{rand-1}}(G), X_{\text{rand-2}}(G)$),通过与目标个体 $X_i(G)$ 进行向量叠加产生新个体 $V_i(G)$

$$V_i(G) = X_i(G) + F_1(X_{\text{best}}(G) - X_i(G)) + F_2(X_{\text{rand}-1}(G) - X_{\text{rand}-2}(G)), \quad (19)$$

式中, F 为比例因子, $F \in [0, 2]$ 。选取相对简单的变异方式如式(4)。

步骤 3:(交叉)通常有 2 种类型的交叉方式,分别为二项式和指数形式

二项式交叉:通过供体向量 $V_i(G)$ 和目标向量 $X_i(G)$ 进行交叉产生新个体 $U_i(G)$

$$u_{ij}(G) = \begin{cases} v_{ij}(G), & \text{if } \text{rand}_{ij} \leq Cr \text{ or } j = j_{\text{rand}}, \\ x_{ij}(G), & \text{otherwise,} \end{cases} \quad (20)$$

式中, rand_{ij} 和 j_{rand} 表示随机数。

指数交叉:在区间 $[1, D]$ 中随机选取整数 n 作为目标向量 $X_i(G)$ 的起点,表示其与供体向量 $V_i(G)$ 进行元素交换的开始。同样在区间 $[1, D]$ 中随机选取另一个整数 L , 作为供体向量贡献给新个体向量的元素个数,则指数交叉形式如下

$$u_{ij}(G) = \begin{cases} v_{ij}(G), & \text{for } j = \langle n \rangle_D, \dots, \langle n + L - 1 \rangle_D, \\ x_{ij}(G), & \text{otherwise,} \end{cases} \quad (21)$$

式中, $\langle \cdot \rangle_D$ 表示模量为 D 的模函数。

步骤 4:(选择)对于给定目标函数 $f(x)$ 的最小化问题,DE 算法精英选择方式可表述如下

$$X_i(G+1) = \begin{cases} U_i(G), & \text{if } f(U_i(G)) \leq f(X_i(G)), \\ X_i(G), & \text{if } f(U_i(G)) > f(X_i(G)). \end{cases} \quad (22)$$

2.2 Q 学习理论

Q 学习是一种强化学习方法^[13],通过一定操作引起环境状态变化,并赋予该操作相应奖励(或惩罚),从而使操作向着明确的方向进行。在实际中,在状态 s 对未来状态 s' 的奖励进行预测是很难的。Q 学习则仅考虑未来状态 s' 的最佳动作奖励回报。

令 $S = \{s_1, \dots, s_n\}$ 为给定环境下智能体状态集合; $A = \{a_1, \dots, a_n\}$ 为智能体在状态 $s_i \in S$ 可供选择的动作集合; $r(s_i, a_j)$ 表示智能体在状态 s_i 选择动作 a_j 的立即奖励回报; $\delta(s_i, a_j)$ 为智能体在状态 s_i 选择动作 a_j 的下一状态 s_k 的过渡函数; γ 为惩罚未来奖励延迟的贴现因子, $\gamma \in [0, 1]$; $Q(s_i, a_j)$ 为智能体在状态 s_i 选择动作 a_j 的总奖励回报。则 $Q(s, a)$ 可表述为^[14]

$$Q(s, a) = r(s, a) + \gamma \bar{V} Q(\delta(s, a)) = r(s, a) + \gamma \max_{a'} Q(\delta(s, a), a'), \quad (23)$$

式中, \bar{V} 为智能体在状态 s 获得的总奖励。下面给出 Q 学习的微分改进形式

$$Q(s, a) \leftarrow (1 - \alpha) Q(s, a) + \alpha \times (r(s, a) + \gamma \max_{a'} Q(\delta(s, a), a')). \quad (24)$$

上式主要是使 $Q(s, a)$ 在动作 a 指向 $\delta(s, a)$ 时, Q-value 递增,确保下步动作奖励 $r(s, a)$ 要大于 $Q(s, a)$, 保证进化向更优方向进行。当 $\alpha = 0$ 时停止学习,当 $\alpha = 1$ 时表示智能体只考虑最新信息。贴现因子 γ 决定未来信息重要性, $\gamma = 0$ 表示算法只注重当前奖励,而 $\gamma = 1$ 时表示算法注重长期高额回报。

3 基于 QDEMA 算法的 2E-VRP 问题优化

3.1 一级 SVRP 最优割算法

如前所述,这里采用最优切割方式进行初始种群构建,然后利用 QDEMA 算法进行 2E-VRP 问题精细优化。

2E-VRP 可看作一级 SDVRP 和二级 MDVRP 耦合问题,可从二级到一级顺序解耦。首先采用最优切割法得到一级 SDVRP 问题较合理配送计划,以此确定中转站数量,并作为 QDEMA 算法初始解。然后求解二级 MDVRP 配送方案得到总里程及总车辆数。而 QDEMA 只优化二级客户配送方案,此做法有助于提高算法效率。最优切割法流程如图 2^[15]。

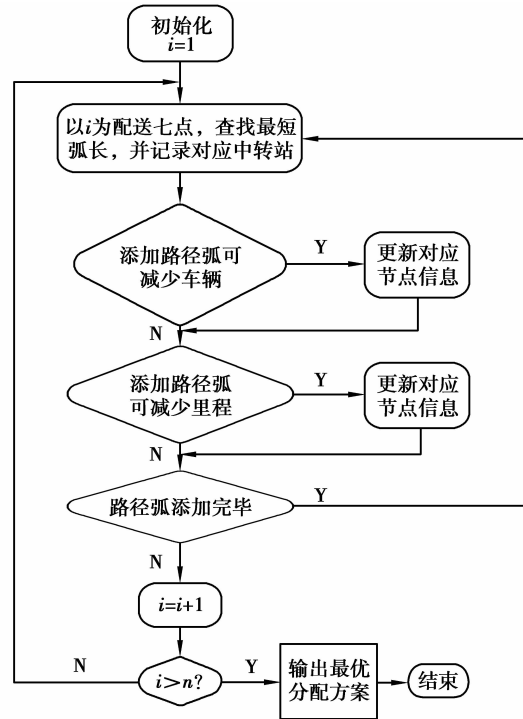


图 2 最优切割算法

Fig.2 Algorithm of optimal cutting

3.2 二级 MVRP 配送优化

QDEMA 算法利用差分进化算法实现全局寻优,利用差分 Q 学习(differential Q learning, DQL)算法实现局部深度探索。QDEMA 算法伪代码见表 1,步骤如下:

步骤 1:(初始化)在初始搜索范围中初始化大小 NP,维数 D 的种群。选取方式同式(3),Q-table 初始化为较小数值,如果 Q-value 能够达到最大 100,则赋予相应的 Q-table 值为 1;

步骤 2:(参数自适应)Q-table 奖励、惩罚措施主要用来选取算法合适的比例因子 F,选择 $F = F_j$ 的概率可由下式计算

$$P(F_j) = \frac{Q(s_i, 10F_j)}{\sum_{l=1}^{10} Q(s_i, 10F_l)} \quad (25)$$

为保持每一行 Q-value 的自适应性,在区间[0,1]中随机产生数值 r,然后选取 F_j ,满足

$$\begin{aligned} & \sum_{m=1}^{j-1} P(F = F_m) < r \leq \sum_{m=1}^j P(F = F_m) \\ \Rightarrow & \frac{\sum_{m=1}^{j-1} Q(s_i, 10F_m)}{\sum_{l=1}^{10} Q(s_i, 10F_l)} < 1 \leq \frac{\sum_{m=1}^j Q(s_i, 10F_m)}{\sum_{l=1}^{10} Q(s_i, 10F_l)} \end{aligned} \quad (26)$$

步骤 3:(DE 操作)利用 DE 算法进行个体排序和状态分配,令 f_i 为个体 i 最新目标值,对 f_i 归一化 $\frac{f_i}{\sum_{j=1}^{NP} f_j}$,并降序排序,可得等级为 r 的排序表,其状态为 s_r ,对 $r=1: NP$ 重复执行上述操作。

步骤 4:(Q-table 更新)若状态为 s_i 的个体在执行操作 F_j 后状态变为 s_k ,其目标适应值增加,那么利用正奖励公式更新 $Q(s_i, 10F_j)$,形如

$$Q(s_i, 10F_j) = (1 - \alpha)Q(s_i, 10F_j) + \alpha(\text{reward}(s_i, 10F_j) + \gamma \max_{F'}(s_k, 10F')), \quad (27)$$

否则, $Q(s_i, 10F_j)$ 利用负奖励回报 $-K$ 更新, $\text{reward}(s_i, 10F_j)$ 表示奖励反馈值。

步骤 5:(判断收敛)重复步骤 2~4,直到满足条件:达到迭代终止代数或满足收敛精度。

表 1 QDEMA 算法伪代码
Table 1 QDEMA algorithm pseudo-code

```

1. // 算法初始化
2. 设置迭代数  $t=0$ , 初始化  $NP$  种群  $P_0 = \{X_1(0), \dots, X_{NP}(0)\}$ 。设置  $\alpha=0.25, \gamma=0.8$ , 计算  $X_i(0)$  的适应值  $f(X_i(0))$ , 并进行排序。  $R(0) = [r_1(0), \dots, r_{NP}(0)]$ ,  $r_i(0)$  为在  $t$  代等级为  $i$  的目标向量。初始化  $Q(r_i(0), j) = 1$ 。
3. while 不满足终止条件 do
    令  $[\text{reward}(r_i(t), j)] = 0, r_i = [1, \dots, NP], j = 1:10$ ;
    for  $i = 1:NP$  do
4. // 轮盘赌选择
        在区间  $[0.1, 1]$  中随机选取比例因子  $F_{ri}$ , 则选取概率为:  $P(F = F_j) = \frac{Q(r_i(t), j)}{\sum_{l=1}^{10} Q(r_i(t), l)}$ ;
5. // 变异操作
        对第  $i$  个目标向量  $X_i(t)$  执行变异操作:
         $V_i(t) = X_i(t) + F(\mathbf{X}_{\text{best}}(t) - X_i(t)) + F(\mathbf{X}_{\text{rand1}}(t) - \mathbf{X}_{\text{rand2}}(t))$ 
6. // 交叉操作
        采用式 (20) 或式 (21) 对目标向量  $X_i(t)$  和变异后向量  $V_i(t)$  执行交叉操作产生  $U_i(t)$ ;
7. // 选择操作
        iff  $(U_i(t)) < f(\mathbf{X}_i(t))$ 
            reward( $r_i(t), 10F_{ri}$ ) =  $f(\mathbf{X}_i(t) - U_i(t))$ ;  $\mathbf{X}_i(t+1) = U_i(t)$ ;
            iff  $(U_i(t)) < f(\mathbf{X}_{\text{best}}(t))$ 
                 $\mathbf{X}_{\text{best}}(t) = U_i(t)$ ; 并更新  $\mathbf{X}_{\text{best}}(t)$ ;
            endif
        else
            reward( $r_i(t), 10F_{ri}$ ) =  $-K$ ;  $\mathbf{X}_i(t+1) = \mathbf{X}_i(t)$ ;
        endif
        更新  $\mathbf{X}_{\text{best}}(t)$ ;
    endfor
8. 对种群降序排序:  $\mathbf{R}(t+1) = [r_1(t+1), \dots, r_{NP}(t+1)]$ ;
9. // 更新 Q-table
    for  $i = 1:NP$ 
        for  $F_j = 0.1:1.0$ 
            if reward( $r_i(t), 10F_j$ )  $\neq 0$ 
                 $Q(s_i, 10F_j) = (1 - \alpha)Q(s_i, 10F_j) + \alpha(\text{reward}(s_i, 10F_j) + \gamma \max_{F'}(s_k, 10F'))$ ;
            else
                 $Q(s_i, 10F_j) = Q(s_i, 10F_j)$ ;
            end if
        end for
    end for
     $t = t + 1$ ;
10. end while

```

4 仿真实验与分析

4.1 QDEMA 算法实验

测试函数: f_1 为 Dejong, f_2 为 Griewank。 $f_1 \sim f_2$ 函数全局最优值为 0。函数形式如下:

$$f_1 = \frac{1}{4\,000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos(x_i/\sqrt{i}) + 1, [-60, 60],$$

$$f_2 = \frac{\sin^2(\sqrt{x_1^2 + x_2^2}) - 0.5}{(1 + 0.001(x_1^2 + x_2^2))^2} - 0.5, [-100, 100].$$

对比算法选用改进自适应空间差分进化算法 (SACPMDE)^[16]、自适应交叉概率因子的差分进化算法 (ACPFDE)^[18]、2 种经典的数值改进算法 DERL 和 DELR^[19]。算法参数设置方式选取 $D=30$, 种群大小 $NP=200$, 迭代终止数 8 000。

ACPFDE 算法的交叉概率因子取值范围设为 $CR \in [0.3, 0.9]$, SACPMDE 算法和 ASMDE 算法的参数设置同文献[16]相关设置, DERL 和 DELR 算法参数设置同文献[17]相关设置, 上述 4 种算法的交叉概率因子选取 $CR=0.9$, 仿真精度选取 $VTR=10^{-6}$ 。对比算法的仿真对比结果如图 3(a)~3(b)所示。

图 3(a)~图 3(b)分别给出对比算法在 Dejong 和 Griewank 测试函数上的仿真对比结果。图 3(a)显示出几种算法呈 2 种不同收敛态势, 一是 QDEMA 和 SACPMDE 算法未出现早熟收敛, 但 SACPMDE 算法要明显比 QDEMA 算法收敛速度慢。二是其余几种算法均出现不同程度的早熟收敛现象。图 3(b)显示出几种算法均未出现早熟收敛现象, 但 QDEMA 算法的收敛速度要明显快于对比算法。通过在测试函数上的仿真一定程度上验证了算法有效性。

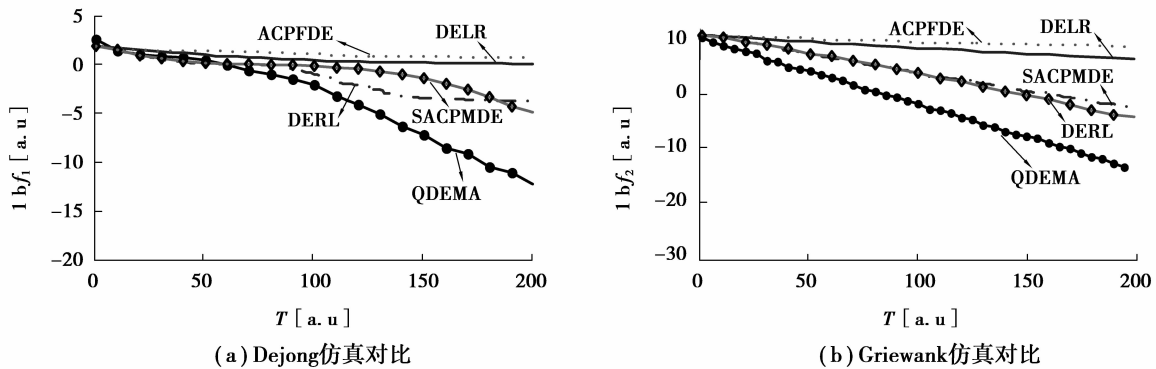


图 3 测试函数仿真对比

Fig.3 Test function of simulation comparison

4.2 算例实验

这里借鉴相关文献做法, 选取标准 VRP 算例: En33-k4、En22-k4、En51-k4, 并以这 3 种算例为基础, 通过设定对应的车辆配送约束, 演化获得本文实验算例, 该算例基于应用背景特点进行设计, 具有一定应用研究价值。

实验设置: Intel(R) Xeon X5550 2.60 GHz 处理器, 4 GB 内存, Win7 系统, 仿真软件采用 matlab2012b。参数设置: 设置差分进化种群规模 $NP=50$, 交叉概率因子 $P_c=0.8$, 终止迭代次数 $T_{max}=1\,000$, 算法最大运行时间 $t_{max}=60\text{ s}$ 。

仿真对比指标: 算法运行 10 次获得最佳配送路径值 (best), 平均算法运行时间 (time)。对比算法采用 Branch and Cut^[20] 算法和 Multi-start^[21] 算法。仿真结果如表 2 及图 4 所示。

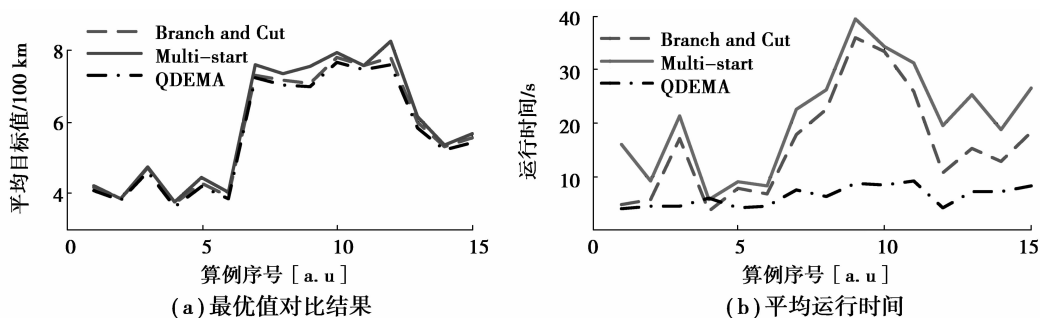


图 4 算法对比结果

Fig.4 Results of algorithm comparison

表 2 仿真对比结果
Table 2 Simulation results

算例	Branch and Cut		Multi-start		QDEMA	
	Best	Time	Best	Time	Best	Time
E-n22-k4-s6-17	417	4.94	421	16.12	427	4.21
E-n22-k4-s8-14	384	5.76	384	9.23	379	4.58
E-n22-k4-s9-20	481	4.89	472	4.77	458	4.65
E-n22-k4-s10-14	371	3.64	375	5.87	382	6.07
E-n22-k4-s11-12	427	7.87	444	9.15	418	4.26
E-n22-k4-s12-16	392	6.82	403	8.34	384	4.67
E-n33-k4-s1-9	730	17.92	757	22.64	721	7.64
E-n33-k4-s2-13	714	22.52	733	26.14	742	6.46
E-n33-k4-s3-17	707	35.8	754	39.28	698	8.84
E-n33-k4-s4-5	778	33.25	792	34.18	765	8.51
E-n33-k4-s7-25	756	25.87	756	31.24	745	9.30
E-n33-k4-s14-22	779	10.75	824	19.57	758	4.37
E-n51-k5-s2-17	597	15.27	614	25.31	581	7.36
E-n51-k5-s4-46	530	12.87	533	18.74	541	7.26
E-n51-k5-s6-12	554	18.34	564	26.47	545	8.40

表 2 给出 3 种对比算法在不同算例上的仿真数据,表中 *Best* 量单位是 km, *Time* 量单位是 s。图 4(a)~图 4(b)分别给出 3 种算法的平均配送路径值和运行时间对比图。图 4(a)可看出在最优配送路径值方面,虽相差不大,但 QDEMA 算法要优于 Branch and Cut 和 Multi-start 2 个对比算法,而 Multi-start 算法要始终优于 Branch and Cut 算法。在图 4(b)所示运行时间上,Multi-start 算法所用时间最长,Branch and Cut 算法的运行所需时间其次,用时最短的为 QDEMA 算法。说明 Branch and Cut 和 Multi-start 2 个算法相比,Multi-start 是在牺牲计算时间的同时提高了算法收敛精度,而 QDEMA 算法不但收敛精度要高于 2 种算法,且在运行时间上也要优于对比算法,且算法的平均运行时间较平稳。极个别情况,如算例 4,由于该算例较简单,Branch and Cut 和 Multi-start 2 个算法的运行时间不高,甚至接近或比 QDEMA 算法运行时间短,主要是算例过于简单,而 QDEMA 算法仍采用种群寻优方式,耗时较长。综上所述,QDEMA 算法适合大规模 2E-VRP 问题,而实际中 2E-VRP 问题规模都是非常大的。

下面以 E-n22-k4-s9-20 算例为例,给出 QDEMA、Branch and Cut 和 Multi-start 算法的配送路径图,如图 5(a)~图 5(b)所示。其中,图 5(a)为原始的节点距离图,图中给出站点间的配送距离,单位 km。图 5(b)分别给出上述 3 种对比算法的配送路径图。从表 2 仿真对比结果看,QDEMA 算法的最佳配送路径总距离为 458 km,而 Branch and Cut 算法配送距离为 481 km,Multi-start 算法为 472 km,QDEMA 算法在配送距离上要优于对比算法。在算法运行时间上,QDEMA 算法要优于 Branch and Cut 和 Multi-start 算法。图 5(b)给出不同算法的配送方案,图中可看出 QDEMA 算法略区别于对比算法。

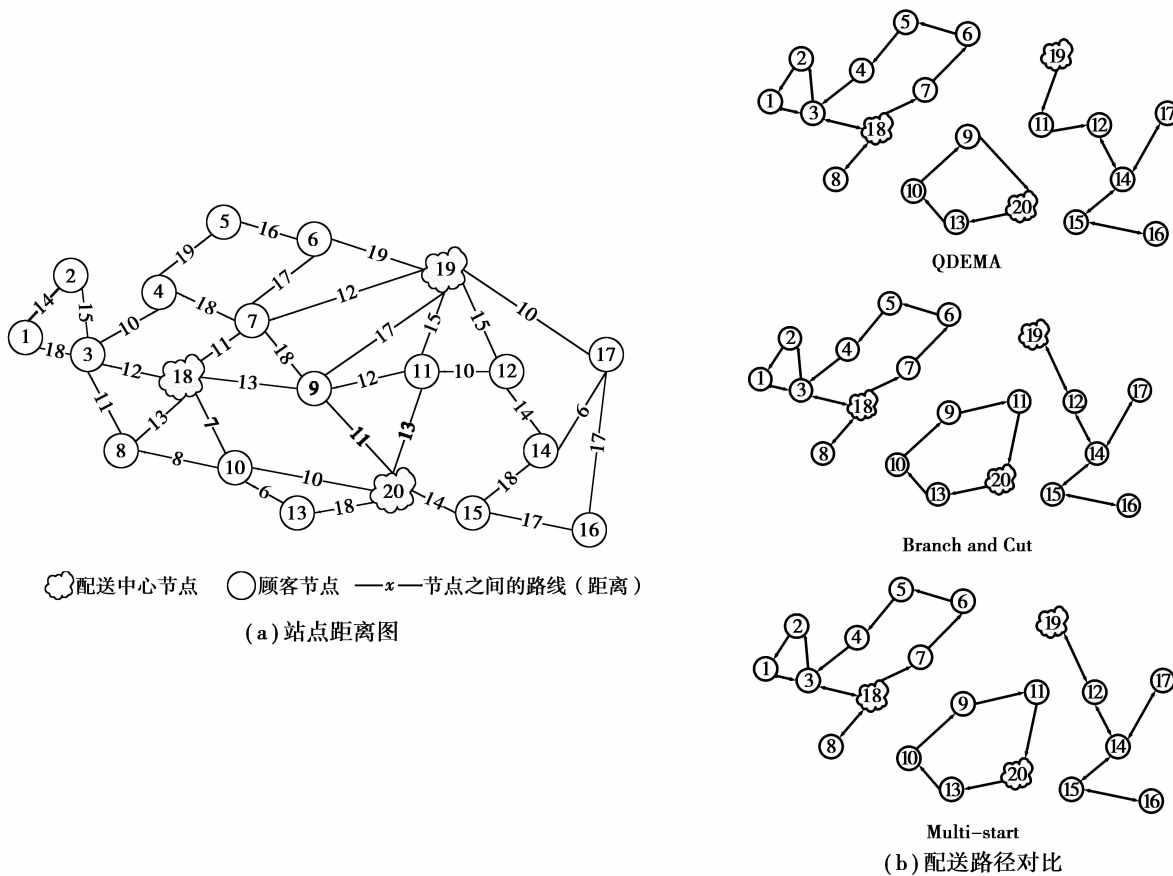


图 5 配送方案路线图

Fig.5 Roadmap of distribution scheme

5 结 语

根据两级车辆路径优化问题特性,将该问题分解为使用最优分割法算法精确分配的一级中转站优化,以及利用 Q 学习理论和差分进化算法设计了新的 Memetic 算法进行车辆路径优化的二级客户端优化。实现了两级车辆路径优化问题的有效解耦,仿真结果显示,所提算法相对于对比算法能够更有效解决两级车辆路径优化问题。

参考文献:

[1] Xia Y M, Cheng B. A vehicle routing problem based on intelligent batteries transfer management for the EV network[J]. China Communications,2014,11(5):160-170.

[2] Crainic T G, Mancini S, Perboli G. Multi-start heuristics for the two-echelon vehicle routing problem[J]. Lecture Notes in Computer Science,2011,32(5):179-190.

[3] Perboli G, Tadei R, Vigo D. The two-echelon capacitated vehicle routing problem: Models and math-based heuristics[J]. Transportation Science,2011,45(3):364-380.

[4] Jepsen M, Spoorendonk S, Ropke S. A branch-and-cut algorithm for the symmetric two-echelon capacitated vehicle routing problem[J]. Transportation Science,2013,47(1):23-37.

[5] Saeiklis D, Powell S. A heuristic method for the open vehicle routing problem[J]. Journal of the Operational Research Society,2000,51(5):564-573.

[6] Brandao J. A tabu search heuristic algorithm for open vehicle problem routing problem[J]. European Journal of Operational Research,2004,157(3):552-564.

- [7] 符草. 带装载能力约束的开放式车辆路径问题及其禁忌搜索算法研究[J]. 系统工程理论与实践, 2004, 24(3): 123-128.
FU Cao. The capacitated open vehicle routing problem and its tabu search algorithm[J]. Systems Engineering-theory & Practice, 2004, 24(3): 123-128. (in Chinese)
- [8] 李湘勇, 田澎. 开放式车辆路径问题的蚁群优化算法[J]. 系统工程理论与实践, 2008, 28(6): 81-93.
LI Xiangyong, TIAN Peng. Research on ant colony optimization algorithm for the open vehicle routing problem[J]. System Engineering-theory & Practice, 2008, 28(6): 81-93. (in Chinese)
- [9] Perboli G, Tadei R. New families of valid inequalities for the two-echelon vehicle routing problem[J]. Electronic Notes in Discrete Mathematics, 2010, 36(6): 639-646.
- [10] Crainic T G, Perboli G, Mancini S, et al. Two-echelon vehicle routing problem: a satellite location analysis[J]. Procedia Social and Behavioral Science, 2010, 2(3): 5944-5955.
- [11] 吴亮红, 王耀南, 袁小芳. 基于快速自适应差分进化算法的电力系统经济负荷分配[J]. 控制与决策, 2013, 28(4): 557-562.
WU Lianghong, WANG Yaonan, YUAN Xiaofang. Fast self-adaptive differential evolution algorithm for power economic load dispatch[J]. Control and Decision, 2013, 28(4): 557-562. (in Chinese)
- [12] Gao Z Q, Pan Z B, Gao J H. A new highly efficient differential evolution scheme and its application to waveform inversion[J]. IEEE Geoscience and Remote Sensing Letters, 2014, 11(10): 1702-1706.
- [13] Konar A, Chakraborty I G, Singh S J. A deterministic improved q-Learning for path planning of a mobile robot[J]. IEEE Transactions on Systems Man and Cybernetics Systems, 2013, 43(5): 1141-1153.
- [14] 于俊, 刘全, 傅启明. 基于优先级扫描 Dyna 结构的贝叶斯 Q 学习方法[J]. 通信学报, 2013, 34(11): 129-138.
YU Jun, LIU Quan, FU Qiming. Bayesian Q learning method with dyna architecture and prioritized sweeping[J]. Journal of Communications, 2013, 34(11): 129-138. (in Chinese)
- [15] Prins C. A simple and effective evolutionary algorithm for the vehicle routing problem[J]. Computers & Operations Research, 2004, 31(12): 1985-2002.
- [16] 姚峰, 杨卫东, 张明. 改进自适应变空间差分进化算法[J]. 控制理论与应用, 2010, 27(1): 32-38.
YAO Feng, YANG Weidong, ZHANG Ming. Improved space-adaptive-based differential evolution algorithm[J]. Control Theory & Applications, 2010, 27(1): 32-38. (in Chinese)
- [17] Hager T, Bruderl G, Lermer T. Current dependence of electro-optical parameters in green and blue (AlIn)GaIn laser diodes[J]. Applied Physics Letters, 2012, 101(17): 1109-1116.
- [18] 杨卫东, 姚峰, 张明. 基于自适应交叉概率因子的差分进化算法及其应用[J]. 信息与控制, 2010, 39(2): 187-193.
YANG Weidong, YAO Feng, ZHANG Ming. Differential evolution algorithm based on adaptive crossover probability factor and its application[J]. Information and Control, 2010, 39(2): 187-193. (in Chinese)
- [19] Kaelo P, Ali M M. A numerical study of some modified differential evolution algorithms[J]. European Journal of Operational Research, 2005, 169(3): 1176-1184.
- [20] Avella P, Boccia M, Vasilyev I. A branch-and-cut algorithm for the multilevel generalized assignment problem[J]. IEEE Access, 2013, 25(1): 475-479.
- [21] Oliveira H, Vasconcelos G C, Alvarenga G B. A multi-start simulated annealing algorithm for the vehicle routing problem with time windows[J]. Symposium on Neural Networks, 2012, 21(3): 137-142.