多目标动态车辆路径问题建模及优化

周慧 周良 丁秋林
（南京航空航天大学计算机科学与技术学院 南京 210016）

摘要 针对物流配送中动态车辆路径优化问题，综合考虑动态需求、路网影响、车辆共享、时间窗以及客户满意度，建立了多目标动态数学规划模型。该模型能更好地描述现代物流配送问题。同时，提出一种两阶段求解策略，第一阶段采用多目标粒子群优化算法获取帕累托最优解，采用改进的粒子群优化算法提升粒子群搜索性能，采用自适应网格技术保持解的分布性；第二阶段对客户的动态需求采用贪婪插入和变邻域搜索进行实时路径调整。实验表明，该算法在解问题中有更好的探寻能力和快速收敛到全局最优，满足动态路径优化实时性要求。

关键词 物流配送，车辆路径问题，混合粒子群优化算法，模拟退火，Pareto 最优解

 Modeling and Optimization for Multi-objective Dynamic Vehicle Routing Problem

ZHOU Hui ZHOU Liang DING Qiulin
(College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China)

Abstract For the dynamic vehicle routing problem in logistics distribution, this paper built a multi-objective and dynamic mathematical programming model synthesizing dynamic demands, the effects on the road network, vehicle sharing, time window and customer satisfaction. This model can describe modern logistics distribution better. Meanwhile, the paper put forward a two-phase solving strategy for it. In the first phase, multi-objective hybrid particle swarm optimization is adopted to get preliminary Pareto solutions. The algorithm uses the modified updating strategy of particle states and simulated annealing operation to improve the searching performance of particles, and uses adaptive grid technique to maintain the dispersion of solutions. In the next phase, greedy insertion and variable neighborhood search are applied to adjust routes according to the changes in demand. The experimental results show that the two-phase algorithm has better exploring ability in solution space, and it can also converge to the global optimum rapidly, and satisfy the real-time requirement.

Keywords Logistics distribution, Vehicle routing problem, Hybrid particle swarm optimization, Simulated annealing, Pareto solution

1 引言

车辆路径问题（Vehicle Routing Problem, VRP）通常可以描述为：对一系列装货点和（或）卸货点组织适当的行车路线，使车辆有序地通过，在满足一定的约束条件（如货物需求量、发送量、交发货时间、车辆容量等限制）下，达到一定的目标（如路程最短、费用最少、时间最少、使用车辆数最少等）


Rong Wei 等[1]研究了有载重限制的同时取送货的 VRP，提出了基于多社会结构的改进粒子群优化算法，将单个粒子的解码分为先 m 维维客点后 2n 维的车辆路径方向两步进行。戚铭尧等[2]设计了一种客户间时空距离的目标表达方式，利用遗传算法对客户点进行时空聚类，并将聚类结果应用于路径调整中，有效地减小了搜索范围，能更快地得到问题的更好解。

现代企业在进行物流配送时不再只是追求单一目标的优化，比如在考虑成本优化的同时，也希望达到客户满意度的最佳。本文对现代物流业中涉及的优化目标进行综合，充分考虑车辆数、车辆行驶距离、成本、旅行速度、路径结构、旅行时间、客户满意度、客户时空范围，结合车辆共享，建立多目标车辆路径问题（Multi-objective Dynamic Vehicle Routing Problem, MODVRP）模型，以便最大化工企业利益，进而针对动态需求提出一种两阶段求解策略。在预优化阶段采用多目标混合粒子群优化算法（Multi-objective Hybrid Particle Swarm Optimization, MOHPSO）获取预优化阶段 Pareto 最优解，解在粒子群状态更新策略中自适应调整惯性权重因子和学习因子，结合模拟退火（Simulated Annealing, SA）操作，有效解决粒子群算法易陷入局部最优和难以进行全局搜索的问题，并采用自适应网络技术保证了 Pareto 最优性。在实时路径调整阶段，采用贪婪插入（Greedy Insertion, GI）和变邻域搜索（Variable Neighbourhood Search, VNS）构造 4 种邻域结构，完成后续配送路径的快速改造以满足实时性要求。最后通过仿真实验验证了模型的可解性和算法的高效性、实时性。

2 MODVRP 描述及建模

2.1 问题描述

MODVRP 描述为：有 M 个配送中心，每个配送中心有载重不同的车辆 k，负责给 N 个客户进行货物配送。每个客户有一个固定的服务时间窗，且有车辆用完时，可考虑从第三方机构租赁车辆来服务客户，租赁车辆需先到达配送中心装载货物，然后才能进行配送；且在配送中心车辆用完时，可考虑从第三方机构租车来服务客户，租赁车辆需先到达配送中心装载货物，然后才能进行配送。路面影响，即每段路对车辆的行驶速度造成的影响，会对总的时间约束和客户服务时间造成影响。在这些约束下寻求最优的配送线路。

现代物流配送过程中影响企业决策的优化目标大致分为以下 4 个方面：(1) 总的行驶路径。因为达到行驶路径最佳也会影响达到成本、油耗、时间的最小，所以以往大部分企业都是以该目标为优化因子进行路径优化的。(2) 旅行时间。部分文献将旅行时间加权合并到优化目标中，尤其在优化 VRPTW 时，旅行时间的优化可以最大化企业收益；而在有路况影响时，总的行驶路径与总的旅行时间不成比例，而现实配送中企业希望在达到总的行驶路径最佳的同时也能有较短的旅行时间，因此旅行时间也单独作为一个优化目标。(3) 客户满意度。该目标是近年来企业共同追求的目标，它影响着企业的生存发展。(4) 成本。成本是企业生存的基础，大部分企业在达不到其他目标优化的情况下，也希望能最小化成本。成本一般分为固定成本和变动成本，其中固定成本包括运输费用、发车费用和租赁费用等，而变动成本包括的范围很广，目前的优化文献中并没有过多考虑。车辆的油耗、司机在不熟悉的道路上行驶时所花费的费用以及车辆到达客户点等待的成本以及车辆晚到客户点所支付的延迟成本等，都是变动成本。

综上所述，本文在寻求最优车辆配送路径的同时，优化成本、旅行时间和客户满意度 3 个目标函数。

2.2 MODVRP 模型

问题编码：(1) 车场：配送中心 m = N + 1, N + 2, …, N + M，每个配送中心有 k 辆车，每辆车的额定载重量为 Qi(k=1, 2, …, M)，每辆车的发车成本为 Fj。其中，二方机构 h = N + M + 1, N + M + 2, …, N + M + H，每个机构有 k 辆车，每辆车的额定载重量为 Qi(k=1, 2, …, M)，每辆车的租赁成本为 Zj。配送中心和第三方机构统称车场。(2) 客户：i = 1, 2, …, N，需求量为 qi，每个客户的服务时间窗为 [Ei, Li]，开始服务时间为 ti，客户对开始服务时间为 t 的满意度用梯形模糊函数来量化表示：

$$\mu_s(t) = \begin{cases} 1, & 0 \leq t < E_i \\ e^{-(t - E_i)}, & E_i \leq t < L_i \\ 0, & t \geq L_i \end{cases}$$

在客户 i 处等待的时间为 Wi(t)，服务时间为 Se = qi，延时的时间为 Pa(t)，等待和延时的单位成本分别为 cw 和 c_p。其中，(3) 道路：客户 j 与客户 i 之间的距离为 dij，每段路行驶的速度为 vij，车辆行驶速度的平均影响系数为 $\beta_j$，道路的旅行时间的快速改造以满足实时性要求。最后通过仿真验证了模型的可解性和算法的高效性、实时性。

定义决策变量：

$$x_{ij}^m = \begin{cases} 1, & \text{车场 m 的车 k 从客户 i 到 j} \\ 0, & \text{其他} \end{cases}$$

本文的 MODVRP 模型为：

$$f = (f_1, f_2, f_3)$$

$$f_1 = \sum_{i=1}^{N} \sum_{j=1}^{N} d_{ij} x_{ij}^m + \sum_{i=1}^{N} \sum_{m=1}^{M} F_j x_{ij}^m + \sum_{h=1}^{H} R_h (t_h)$$

$$f_2 = \sum_{i=1}^{N} \sum_{j=1}^{N} (m,k) x_{ij}^m + \sum_{i=1}^{N} (w_i(t_i) + Se)$$

$$f_3 = \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{k=1}^{R} \sum_{h=1}^{H} (1 - \frac{1}{N} x_{ij}^m)$$

$$\sum_{m=1}^{M} x_{ij}^m = 1, j \in \{1, 2, \ldots, N\}$$

$$\sum_{i=1}^{N} x_{ij}^m = 1, i \in \{1, 2, \ldots, N\}$$

$$\sum_{i=1}^{N} x_{ij}^m = 0, h, r \in \{N + M + 1, \ldots, N + M + H\}$$

$$\sum_{j=1}^{N} x_{ij}^m = 1, i \in \{1, 2, \ldots, N\}$$

$$\sum_{j=1}^{N} x_{ij}^m = 0, h, r \in \{N + M + 1, \ldots, N + M + H\}$$

$$\sum_{i=1}^{N} x_{ij}^m \leq L, \forall m,k$$

$$\sum_{i=1}^{N} x_{ij}^m \leq S, \forall m,k, i,j \in \mathbb{R}$$
\[ t_j = \begin{cases} t_j + S_i + t_d(m,k), & t_j \geq E_i \\ E_i + S_i + t_d(m,k), & t_j < E_i \end{cases} \]  (12)

\[ w_i(t_j) = \max(0, E_i - t_j) \]  (13)

\[ p_i(t_j) = \max(0, t_j - L_i) \]  (14)

\[ B_j = P(t_j, \leq \delta_i < t_j) = \int_{t_j}^{\delta_i} r_d(x) \, dx \]  (15)

\[ r_d(x) = \frac{d_i}{\sqrt{2\pi} \sigma} \exp\left(-\frac{(x - \mu)^2}{2\sigma^2}\right), 0 < x \leq 24 \]  (16)

\[ t_{v0}(m,k) = \frac{d_i}{\sqrt{2\pi} \sigma} \]  (17)

式(2)～式(4)为目标函数。式(2)是成本目标，包括5个部分：行驶路径长度、发车成本、租车成本、等待成本和延误成本；式(3)是旅行时间目标，包括车辆行驶时间、服务时间和等待时间。式(4)是客户满意度目标；式(5)是车辆限制；式(6)是车辆数限制；式(7)和(8)保证每个客户被服务一次；式(9)约束租赁车辆不可直接服务客户且服务完客户不可直接回第三方机构；式(10)是最大旅行距离约束；式(11)消除子回路；式(12)～式(14)是与客户时间有关的计算公式，式(12)计算车辆到达时间，式(13)计算车辆等待时间，式(14)计算客户延迟时间；式(15)～式(17)是道路情况影响车速及旅行时间的计算公式，式(15)计算道路(i,j)的复杂情况影响系数，式(16)是路段(i,j)影响车速的时间函数，式(17)计算车辆在路段(i,j)行驶的时间。

3 两阶段求解策略

MODVRP是VRP的一种，也是NP-hard难题，因此采用启发式算法进行求解。MODVRP的动态性主要体现在客户需求的变动上，一般处理策略为重新优化策略或局部优化策略。由于对实时动态信息进行重新优化需要大量时间，无法满足路径规划的实时性约束，而局部优化策略可以满足路径优化的目的，因此本文选择后者作为实时优化阶段的策略。本文针对MODVRP模型设计的两阶段求解策略总体流程如图1所示。

预优化阶段根据相关参数和客户初始需求量建立MODVRP模型，采用MOHPSO产生预优化路径，此时路径是完整配送路径，可以分发给各车辆开始配送。对于客户的变化需求，本文采取时间片法，将配送周期按时间轴划分为适当的时间片，时间片结束时，对收集到的客户需求变动信息进行实时路径优化。

3.1 预优化阶段

MODVRP模型同时优化3个目标，采用Pareto最优解进行多目标优化，本文设计MOHPSO算法来求解预优化阶段的Pareto最优解，并采用自适应网格技术保持解空间的分布性。

3.1.1 MOHPSO算法

多目标混合粒子群优化算法是指在粒子群优化算法中融入模拟退火操作，使之更适应模型的求解。粒子群优化算法(Paráeté Swarm Optimization, PSO)是一种新兴的优化技术，是群智能算法的一种。PSO通过粒子跟随自身最优解和种群最优解来完成任务目标，主要是通过粒子状态更新来进行退火。模拟退火算法(Simulated Annealing, SA)来源于固体退火原理，将固体加热至充分高温，再使其慢慢冷却，在此过程中，根据能态的变化选择最优点。将SA融入到PSO中，既能充分利用SA的全局搜索能力，又能充分利用PSO的局部挖掘能力。本文根据局部挖掘能力，以一定概率接收解差，并保持PSO自身易于陷入局部最优的缺点，能够使算法快速获得全局最优解。

MOHPSO算法流程如图2所示。

粒子群优化算法中，在迭代初期，粒子需要有较强的自我学习能力，这样可以使粒子在全局范围内具有良好的搜索能力；在迭代后期，粒子需要有较强的自我学习能力和种群学习能力，使粒子在全局最优状态聚拢，算法快速收敛。据此，本文改进了粒子状态更新中的各参数的取值方法：

\[ c_1 = c_1^{\max} + \frac{(c_1^{\max} - c_1^{\min}) \times \text{iter}}{\text{iter}_{\max}} \]  (18)

\[ c_2 = c_2^{\max} + \frac{(c_2^{\max} - c_2^{\min}) \times \text{iter}}{\text{iter}_{\max}} \]  (19)

在粒子选取bestP时，利用模拟退火操作得到一个新状态，综合考虑该状态和粒子优化取得的新状态，选取最佳状态作为bestP。通过结合模拟退火操作，以一定概率接收差解，有效地增强了粒子多样性。
为了保证获得的Pareto非劣解集的多样性，通常采用网格的方法将个体空间划分为若干个小网格，这样每个个体就与某个网格相关联，定义网格中个体的数目为挤压系数。网络内的解被视为同一解，通过竞争进行保留。网格太小时，每个网格中包含的个体数多，分布不精确；网格太小时，某些网格中不包含任何个体，技术虽高，但会导致大计算代价，因此网格的大小非常重要。Zhang等利用每一代非支配解集中的解在每一维目标量的差值自适应调整网格的数量；赵燕伟等根据网格密度自适应调整每一维的网格数量。

本文设计一种新的自适应网格算法，利用非劣解集的分布均匀程度来调整每一维的网格数量。非劣解集的分布均匀程度可由解集中相邻解的平均距离来求得，定义目标函数，(z_1, z_2) 的距离为：
\[ d(f(x_1), f(x_2)) = \sqrt{\sum_{i=1}^{m} (f_i(x_1) - f_i(x_2))^2} \]  \( m \) 为解的维数。

则非劣解集的平均距离为：
\[ d_{av} = \frac{\sum_{x_1, x_2 \in N_{ns}} d(f(x_1), f(x_2))}{|N_{ns}|(|N_{ns}|-1)} \]  \( |N_{ns}| \) 为不劣解集的个数。

本文设计的自适应网格算法为：
\[ N_{t+1} = \left[ N, \frac{d_{av}}{d_{av}^{1-t}} \right] \]  \( 0 < t < 1 \)。

式(21)中，\( |N_{ns}| \) 为非劣解集中解的总数，式(22)中为迭代次数，[ ] 表示取整。非劣解集的解分布越集中，式(21)的取值越小，而式(22)的取值越大，此时网格数量的增大使得网格变小，从而每个网格中包含的解个数尽量少，保护了解的解的多样性和解的多样性。

### 3.1.3 算法主要步骤

#### 步骤 1 编码采用整数编码方式[11]。

#### 步骤 2 初始化粒子群 PSO，设置每个粒子 PSO 的初始位置、速度、个体最优解 bestP_i, 全局最优解 bestG。初始化模拟退火温度 T，对粒子 PSO 进行模拟退火操作，获得 bestP', 若 bestP' > bestP，则选其作为 bestP。步骤 7 更新粒子状态，构成新种群。解码并计算新种群中各粒子的目标向量化，即解集 N_{ns}。

#### 步骤 8 利用自适应网格算法更新 N_{ns}，若满足终止条件则停止；否则转步骤 4。

### 3.2 实时优化阶段

在时间片结束时，对客户的需求变动信息进行实时路径重构。首先对预优化阶段已形成的各个子路径过滤跟踪，把服务过的客户点和下一个服务的客户点置为不可改变点，然后对剩余子路径进行操作。对于客户需求的减少或者取消，只需修改路径上的需求配送量。路径重构主要是针对新增的需求点。本文设计了两种操作，贪婪插入、变邻域搜索，来完成实时路径的重优。

G1 操作的主要思想是先不考虑约束，仅考虑距离，对于每一个新增加的客户点，遍历子路径，找到距离最近的点插入后。这样每个子路径都贪婪地包括了能插入的最多的客户。下一步是根据车载量、时间窗约束对路径进行裁剪，形成新的路径同样按照距离最近原则插入车辆，确保其可行性。

在 G1 操作后形成的路径是可行的，但不是最优的。利用 VNS 对其进行快速优化。由 Hansen 和 Mladenović[12] 提出的 VNS 通过搜索过程中系统地改变邻域结构来扩展搜索范围，从而获得局部最优解，再基于此局部最优解重新系统地改变邻域结构，从而找到另一个局部最优解。VNS 的简易、快速和高效是本文选用它作为实时优化阶段手段的重要原因。对于 VNS 搜索，设计了 4 种邻域结构，每种邻域结构都基于使解可行的基础上（满足约束），分别如图 3～图 6 所示。

1. 路径内 2-opt：当两个客户点在同一条路径内，对其进行 2-opt 交换。
2. 路径间点插入：当两个客户点在不同的路径上，将其一个客户点插入到另一客户点之后。
3. 路径间交换：当两个客户点在不同的路径上，将两者位置交换。
4. 路径间片段交换：当两个客户点在不同的路径上，在两条路径上分别选择随机长度的客户片段，将两者整体交换。
步骤 1 令 $t = 0, \omega = 0$。
步骤 2 随机选择两个客户点。
步骤 3 根据两个客户点所在的路径相同或不同分别采取(1)或者(2)一(4)的邻域结构搜索。所在路径不同时，以概率选取(2)一(4)的一种。
步骤 4 若不满足约束，转步骤 2。
步骤 5 若新解 $f^*$ 有改进，则保存新解；否则，若新解的改进量小于原方案 $f$ 加 $\sigma$，也接收新解；否则不接受且令 $\omega = \omega + 1$。
步骤 6 $t = t + 1$，若 $\omega > \beta$，则停止；否则转步骤 2。
其中 $\mu$ 为非改进最大迭代次数，$\nu = 0.2\sigma$ 为偏差值，以接收部分微劣解，使搜索跳出局部最优。VNS 在可行邻域内进行随机变换，在进行一定迭代次数后，将以概率 1 收敛到最优解。

4 实验结果及分析

程序采用 MATLAB 编写，在 Intel(R) Core(TM) i5-3470 CPU 3.2GHz，内存 4.0GB 的 PC 机上运行。由于 MODVRP 尚未有统一的测试实例库，本文分别用设计实例和改进的标准 MDVRP TWP 实例库进行算法分析。

用本文提出的算法求解下述实例：有 4 个配送中心，2 个第三方机构，为 34 个初始客户进行配送；其中 $X$ 为随机生成结果，$\lambda = 0.1, \mu = 12, \sigma = 5$，各车场车辆信息及各客户信息如表 1、表 2 所列。

<table>
<thead>
<tr>
<th>序号</th>
<th>X</th>
<th>Y</th>
<th>需求</th>
<th>E</th>
<th>L</th>
<th>序号</th>
<th>X</th>
<th>Y</th>
<th>需求</th>
<th>E</th>
<th>L</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>36</td>
<td>16</td>
<td>10</td>
<td>7</td>
<td>10</td>
<td>21</td>
<td>30</td>
<td>40</td>
<td>5</td>
<td>9</td>
<td>12</td>
</tr>
<tr>
<td>2</td>
<td>52</td>
<td>41</td>
<td>15</td>
<td>8</td>
<td>11</td>
<td>22</td>
<td>50</td>
<td>30</td>
<td>20</td>
<td>9</td>
<td>15</td>
</tr>
<tr>
<td>3</td>
<td>27</td>
<td>23</td>
<td>3</td>
<td>12</td>
<td>14</td>
<td>23</td>
<td>58</td>
<td>15</td>
<td>30</td>
<td>9</td>
<td>15</td>
</tr>
<tr>
<td>4</td>
<td>17</td>
<td>43</td>
<td>41</td>
<td>10</td>
<td>12</td>
<td>24</td>
<td>37</td>
<td>25</td>
<td>15</td>
<td>7</td>
<td>12</td>
</tr>
<tr>
<td>5</td>
<td>13</td>
<td>13</td>
<td>9</td>
<td>7</td>
<td>12</td>
<td>25</td>
<td>49</td>
<td>47</td>
<td>32</td>
<td>9</td>
<td>18</td>
</tr>
<tr>
<td>6</td>
<td>57</td>
<td>58</td>
<td>28</td>
<td>5</td>
<td>17</td>
<td>26</td>
<td>52</td>
<td>64</td>
<td>19</td>
<td>8</td>
<td>17</td>
</tr>
<tr>
<td>7</td>
<td>27</td>
<td>68</td>
<td>7</td>
<td>9</td>
<td>12</td>
<td>27</td>
<td>20</td>
<td>26</td>
<td>7</td>
<td>11</td>
<td>13</td>
</tr>
<tr>
<td>8</td>
<td>30</td>
<td>48</td>
<td>15</td>
<td>7</td>
<td>11</td>
<td>28</td>
<td>45</td>
<td>31</td>
<td>6</td>
<td>8</td>
<td>14</td>
</tr>
<tr>
<td>9</td>
<td>43</td>
<td>67</td>
<td>14</td>
<td>7</td>
<td>11</td>
<td>29</td>
<td>19</td>
<td>30</td>
<td>25</td>
<td>10</td>
<td>12</td>
</tr>
<tr>
<td>10</td>
<td>58</td>
<td>48</td>
<td>6</td>
<td>9</td>
<td>12</td>
<td>30</td>
<td>23</td>
<td>51</td>
<td>21</td>
<td>7</td>
<td>11</td>
</tr>
<tr>
<td>11</td>
<td>50</td>
<td>16</td>
<td>19</td>
<td>13</td>
<td>16</td>
<td>31</td>
<td>28</td>
<td>7</td>
<td>12</td>
<td>9</td>
<td>14</td>
</tr>
<tr>
<td>12</td>
<td>37</td>
<td>69</td>
<td>9</td>
<td>16</td>
<td>15</td>
<td>32</td>
<td>33</td>
<td>58</td>
<td>9</td>
<td>8</td>
<td>14</td>
</tr>
<tr>
<td>13</td>
<td>5</td>
<td>25</td>
<td>23</td>
<td>9</td>
<td>12</td>
<td>13</td>
<td>42</td>
<td>12</td>
<td>10</td>
<td>7</td>
<td>12</td>
</tr>
<tr>
<td>14</td>
<td>8</td>
<td>52</td>
<td>10</td>
<td>9</td>
<td>16</td>
<td>14</td>
<td>34</td>
<td>9</td>
<td>21</td>
<td>15</td>
<td>9</td>
</tr>
<tr>
<td>15</td>
<td>7</td>
<td>38</td>
<td>28</td>
<td>10</td>
<td>12</td>
<td>15</td>
<td>35</td>
<td>20</td>
<td>60</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>16</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>7</td>
<td>12</td>
<td>16</td>
<td>36</td>
<td>50</td>
<td>58</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>17</td>
<td>5</td>
<td>64</td>
<td>11</td>
<td>8</td>
<td>14</td>
<td>17</td>
<td>37</td>
<td>43</td>
<td>22</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>18</td>
<td>20</td>
<td>52</td>
<td>9</td>
<td>8</td>
<td>12</td>
<td>18</td>
<td>38</td>
<td>7</td>
<td>13</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>19</td>
<td>48</td>
<td>28</td>
<td>18</td>
<td>8</td>
<td>14</td>
<td>19</td>
<td>39</td>
<td>15</td>
<td>35</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>20</td>
<td>39</td>
<td>32</td>
<td>10</td>
<td>7</td>
<td>12</td>
<td>20</td>
<td>40</td>
<td>40</td>
<td>0</td>
<td>0</td>
<td>24</td>
</tr>
</tbody>
</table>

对该实例采用本文提出的两阶段求解策略进行求解，预优化阶段采用 MOHPSO 算法，种群大小为 30，迭代次数为 1000，得到的 Pareto 最优解集为：(5543, 8.35, 8.0 50), (5794, 6.35, 2.0 35), (5560, 4.35, 4.0 40), (5762, 4.44, 6.0 31), (5676, 2.35, 1.0 37), (5587, 6.34, 7.0 42), (5563, 5.34, 3.0 45), (5770, 3.35, 5.0 33)。如图 7 所示。分别选取 3 个目标向量的最优路线表 3 所列。

<table>
<thead>
<tr>
<th>序号</th>
<th>X</th>
<th>Y</th>
<th>需求</th>
<th>E</th>
<th>L</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>35</td>
<td>8</td>
<td>30</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>2</td>
<td>36</td>
<td>16</td>
<td>10</td>
<td>7</td>
<td>10</td>
</tr>
</tbody>
</table>

在实时优化阶段，客户需求变化情况如表 4 所列，其中 4 个原有客户增加需求，10 个新出现客户。采用第二阶段优化算法，其中 VNS 算法中非改进最大迭代次数取 10。实时优化阶段得到的 Pareto 最优解集为：(6008, 3.57, 6.0 57), (6832, 7.52, 0.6 43), (6934, 8.50, 3.0 45), (6792, 4.48, 3.0 49), (6215, 7.50, 9.0 53), (6589, 7.53, 9.0 39), (6872, 3.57, 7.0 35), (6177, 4.55, 7.0 58)，如图 8 所示，最优路线如表 5 所列。

<table>
<thead>
<tr>
<th>序号</th>
<th>X</th>
<th>Y</th>
<th>需求</th>
<th>E</th>
<th>L</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>35</td>
<td>8</td>
<td>30</td>
<td>12</td>
<td>12</td>
</tr>
</tbody>
</table>

图 7 预优化 Pareto 最优解集

<table>
<thead>
<tr>
<th>序号</th>
<th>X</th>
<th>Y</th>
<th>需求</th>
<th>E</th>
<th>L</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>35</td>
<td>8</td>
<td>30</td>
<td>12</td>
<td>12</td>
</tr>
</tbody>
</table>

图 8 重优 化 Pareto 最优解集
为了比较算法性能，在预优化阶段分别采用多目标量化算法（A）与本文提出的 MOHPSO 算法（B）。在标准实例库分别运行两种算法 20 次，结果取平均，得到的运行情况如表 6 所示。可以看出，本文所提算法性能更优。

### 表 6 算法比较

<table>
<thead>
<tr>
<th>实例</th>
<th>C101</th>
<th>R201</th>
<th>RC101</th>
<th>Pr07</th>
</tr>
</thead>
<tbody>
<tr>
<td>解数</td>
<td>9.8</td>
<td>10.5</td>
<td>17.5</td>
<td>17.8</td>
</tr>
<tr>
<td>f1</td>
<td>0.31</td>
<td>0.94</td>
<td>1.77</td>
<td>1.78</td>
</tr>
<tr>
<td>f2</td>
<td>89.7</td>
<td>90.8</td>
<td>347.15</td>
<td>320.87</td>
</tr>
<tr>
<td>f3</td>
<td>63.47</td>
<td>5.94</td>
<td>12.20</td>
<td>11.58</td>
</tr>
<tr>
<td>运行时间</td>
<td>33.23</td>
<td>31.81</td>
<td>114.97</td>
<td>110.25</td>
</tr>
</tbody>
</table>

## 结束语

MODVRP 综合考虑了物流配送过程中多车型、多场景、客户动态需求、路径动态变化、客户时间窗要求和车辆共享等复杂情况，同时通过联合约束和时间窗分配以及客户满意度建立多目标优化函数，更加适应现代物流企业的配送要求，能最大化企业利益。本文首先描述了 MODVRP 问题，并据此建立 MODVRP 模型，然后提出了模型的两阶段求解策略。第一阶段，提出 MOHPSO 算法来求解 Pareto 最优解。对于粒子群优化中易于陷入局部最优的缺点，本文首先动态调整粒子惯性权重和学习因子，使得粒子在初期获得良好的全局搜索能力，在后期能快速收敛到全局最优值；然后将 SA 操作融合到粒子群优化算法中，使算法能搜寻到全局最优值。同时，设计了新的适应性网络参数，利用非解集的分布均匀程度来调整每一轮的网络结构，有效保证解集的分布多样性。

本文采用时间片方式，在时间片结束时进入第二阶段优化。对于新增需求客户需求，采用了 GI 和 VNS 进行路径优化，设计了 4 种邻域结构用于 VNS，使算法能快速有效地收敛到最优解。最后通过仿真实验和与其他算法的比较，表明该算法能够高效地解决 MODVRP 问题，满足调度的实时性要求。随着计算机通信、地理信息系统、全球定位系统等技术的发展，实时道路交通信息获取的便利以及对物流配送的影响，使得根据实时道路交通信息调度配送路线的研究成为后续研究的重点。

## 参考文献