计算机应用研究
Application Research of Computers
ISSN 1001-3695,CN 51-1196/TP

《计算机应用研究》网络首发论文

题目：帝国竞争算法求解 CVRP
作者：蔡延光，王世豪，戚远航，王福杰，林卓胜
DOI：10.19734/j.issn.1001-3695.2020.01.0006
收稿日期：2020-01-22
网络首发日期：2020-07-06
引用格式：蔡延光，王世豪，戚远航，王福杰，林卓胜. 帝国竞争算法求解 CVRP[J/OL]. 计算机应用研究. https://doi.org/10.19734/j.issn.1001-3695.2020.01.0006

网络首发时间：2020-07-06 13:57:05
网络首发地址：https://kns.cnki.net/kcms/detail/51.1196.TP.20200706.1050.005.html

帝国竞争算法求解 CVRP *

蔡延光[1]，王世豪[1]，戚远航[2†]，王福杰[3]，林卓胜[4]

(1. 广东工业大学 自动化学院，广州 510006; 2. 电子科技大学中山学院 计算机学院，广东 中山 528402; 3. 东莞理工学院 电子工程与智能化学院，广东 东莞 523808; 4. 五邑大学 智能制造学部，广东 江门 529020)

摘 要：针对带容量约束的车辆路径问题(CVRP)，提出了一种带分裂机制的帝国竞争算法进行求解。首先，结合 CVRP 的特性，采用基于贪婪准则的编解码策略实现算法空间到解空间的转换。其次，提出帝国分裂策略来增强算法的全局搜索能力，并结合 2-Opt 提高算法的局部搜索能力。最后，通过 25 个基准算例的仿真实验表明：所提出的算法能有效求解 CVRP，所有算例的优化误差不超过 1.0%；与已有的帝国竞争算法、粒子群算法、遗传算法、布谷鸟搜索算法相比，所提出算法的求解效率更高。

关键词：车辆路径问题；帝国竞争算法；粒子群算法；遗传算法；2-Opt
中图分类号：TP301 doi: 10.19734/j.issn.1001-3695.2020.01.0006

Imperialist competitive algorithm for solving CVRP

Cai Yanguang[1], Wang Shihao[1], Qi Yuanhang[2†], Wang Fujie[3], Lin Zhuosheng[4]

(1. *School of Automation*, *Guangdong University of Technology*, *Guangzhou Guangdong* 510006, *China*; 2. *School of Computer Science*, *University of Electronic Science & Technology of China*, *Zhongshan Institute*, *Zhongshan Guangdong* 528402, *China*; 3. *School of Electrical Engineering & Intelligentization*, *Dongguan University of Technology*, *Dongguan Guangdong* 523808, *China*; 4. *Faculty of Intelligent Manufacturing*, *Wuyi University*, *Jiangmen Guangdong* 529020, *China*)

Abstract: For the capacitated vehicle routing problem (CVRP) , this paper proposed an imperialist competitive algorithm which integrates a split mechanism to solve the problem. Firstly, combined with the characteristics of CVRP, this algorithm used the encoding and decoding strategies based on greedy criteria to switch from the algorithm space to the solution space. Secondly, this algorithm presented an imperialist splitting mechanism to improve the global search ability of the algorithm, and simultaneously combined with the 2-Opt algorithm to enhance the local search ability. Finally, the results of simulation experiments with 25 benchmark examples indicate that: The proposed algorithm can effectively solve CVRP, and the optimization errors of all examples are less than 1.0%; furthermore, the proposed algorithm presents more efficiently than the other existing imperialist competitive algorithms, particle swarm optimization algorithm, genetic algorithm and cuckoo search algorithm.

Key words: vehicle routing problem; imperialist competitive algorithm; particle swarm optimization; genetic algorithm; 2-Opt

0 引言

车辆路径问题(vehicle routing problem，VRP)[1]是 1956 年由 Dantzig 和 Ramser[2]首次提出，已被证明是 NP-hard 问题，该问题一直是计算机科学与组合优化领域学者们关注和研究的热点问题。带容量约束的车辆路径问题(capacitated vehicle routing problem，CVRP)[3~5]是 VRP 的基本问题，与实际生活中资源配送和路径规划息息相关，主要解决当多配送车辆服务多个客户点，且配送车辆具有容量限制时，如何规划配送车辆的配送路径使得总服务成本最低的问题。因为精确求解算法无法满足性能要求，所以学者们常常使用启发式算法等近似求解算法进行求解。Akhand 等人[6]针对 CVRP 问题比较了多种路线优化方法，发现使用粒子群优化算法进行路线优化，能够提供更好的解决方案。为了进一步提高求解精度，Foysal 等人[7]提出基于双层局部搜索的粒子群优化算法。在双层本地搜索中，本地搜索的一层适用于所有迭代中的整个种群，而另一层仅应用于不同代生成的最佳粒子池。因此，该算法能在较短优化时间内得到较优求解。Stanley 等人[8]设计了二进制编码方法的遗传算法求解 CVRP 问题，为遗传算法求解 CVRP 问题提供了思路。为了减少优化耗时，姜昌华等人[9]针对提出一种结合 2-Opt 局部优化的混合遗传算法，增强算法的局部搜索能力和收敛速度。Jon 等人[10]通过将布谷鸟搜索算法结合 2-Opt 算法和双桥运算，为解决 CVRP 问题提供了新的思路。

帝国竞争算法(imperialist competitive algorithm，ICA)是 2007 年由 Atashpaz-Gargari 等人[11]提出，具有收敛速度快和全局搜索能力强的特点。ICA 已被应用于求解旅行商问题(traveling salesman problem，TSP)[12, 13]。张鑫龙等人[14]提出一种新型 ICA 求解 TSP 问题，帝国同化采用替换重建方式，革命过程结合自适应算子，得到了较好的优化效果；裴小兵

收稿日期：2020-01-22；**修回日期：**2020-04-08 **基金项目：**国家自然科学基金项目(61074147，61901304)；广东省自然科学基金项目(S2011010005059，2019A1515010493，2016A030313018)；广东省教育部产学研结合项目(2012B091000171，2011B090400460)；广东省科技计划项目(2012B050600028，2014B010118004，2016A050502060)；广州市花都区科技计划项目(HD14ZD001)；广州市科技计划项目(201604016055)；广州市天河区科技计划项目(2018CX005)；广东省普通高校青年创新人才项目(2018KQNCX333，2018KQNCX252)；中山市重大科技专项(2017A1024，2017SF0603，2016A1028)；中山市科技计划重点项目(2018B1018)

作者简介：蔡延光(1963-)，男，湖北咸宁人，教授，博导，博士，主要研究方向为网络控制与优化、组合优化、智能优化、智能交通系统；王世豪(1996-)，男，湖南郴州人，硕士研究生，主要研究方向为智能优化、机器学习；戚远航(1993-)，男(通信作者)，广东湛江人，讲师，硕导，博士，主要研究方向为复杂系统建模与优化、智能优化、运输调度优化、布局优化、运筹学(qiyuanhang77@163.com)。

等人[15]为了降低帝国同化的复杂度，设计概率矩阵挖掘可行解中的优秀组合区块，能够对一定规模的 TSP 问题进行高效求解。但是，目前还没有学者应用 ICA 求解 VRP 问题。

因此，本文根据帝国竞争算法思想为框架，提出求解 CVRP 问题的带分裂机制的帝国竞争算法(imperialist competitive algorithm with split mechanism，ICAS)。该算法针对 CVRP 问题设计实数编解码策略，实现算法空间到解空间的转换；提出一种帝国分裂策略，允许殖民地割裂其所属帝国并创建自己的帝国，该策略可以增加算法运行后期帝国数量，保持帝国之间竞争状态，解决算法“早熟”问题；同时引入 2-Opt[12]算法增强算法局部搜索能力和收敛速度。最后，本文使用多个基准算例进行优化实验，实验结果表明 ICAS 求解 CVRP 问题具有较好的求解精度和求解效率。

1　CVRP 问题数学模型

本文所研究问题为带容量约束的车辆路径问题[16, 17]，问题可以描述为：某一配送仓库中心可安排多辆车为多个客户点提供配送服务，当各客户点对于货物有不同需求量，且配送车有最大载货量限制时，如何规划配送方案可以使总配送服务费用最低。本文对于该问题作如下假设：

a)配送仓库中心以及各客户点地理位置信息均为已知，各点之间完全互通，所有配送车辆均从配送仓库中心出发，最后回到配送仓库中心；

b)客户点只接受一次配送服务，即仅接受一台车进行单次服务；

c)所有配送车辆均为同一规格，每台车仅安排一次配送服务；

d)客户点的需求量不超过配送车辆最大载货量，单条线路配送客户总需求量不超过配送车辆最大载货量。

符号定义如下：

N：地图中节点数量，包含 N-1 个客户点与一个配送中心；

C：C=[1,2,...,N-1]，表示一组解空间，数字 1 至 N-1 代表客户点序号，0 代表配送中心；

M：配送中心拥有配送车辆数；

Q_i：各节点的需求量 Q_i(i=0,1,...,N-1)，Q_0 为配送中心需求量，设置为 0；

$Q_{\max}$：配送车辆最大载货量；

d_{ij}：节点 i、j 之间的距离(j=0,1,...,N-1)；

$$x_{ijs}=\begin{cases}1, & \text{如有车辆}s\text{经节点}i\text{驶向节点}j\\ 0, & \text{其他}\end{cases};$$

$$y_{is}=\begin{cases}1, & \text{节点}i\text{的需求由车辆}s\text{满足}\\ 0, & \text{其他}\end{cases};$$

本文只考虑货物配送过程中所有车辆的总路径长度，则问题数学模型为

目标函数：

$$\min D(C)=\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\sum_{s=1}^{M}d_{ij}x_{ijs} \tag{1}$$

约束条件：

$$\sum_{i=0}^{N-1}Q_i y_{is}\le Q_{\max},\ s=1,2,...,M \tag{2}$$

$$\sum_{s=1}^{M}y_{is}=\begin{cases}1, & i=1,2,...,N-1\\ M, & i=0\end{cases} \tag{3}$$

$$\sum_{i=1}^{N-1}x_{ijs}=y_{js},\ j=1,2,...,N-1;s=1,2,...,M \tag{4}$$

$$\sum_{j=1}^{N-1}x_{ijs}=y_{is},\ j=1,2,...,N-1;s=1,2,...,M \tag{5}$$

其中式(2)保证每辆车配送的客户总货物量不超过车辆最大运载货物量；式(3)~(5)则约束了由 M 辆车共同完成配送任务，且保证每个客户点只被一辆车提供配送服务。

2　求解 CVRP 的帝国竞争算法

2.1　传统帝国竞争算法

帝国竞争算法是基于人类社会政治进化现象提出的智能优化算法，算法中的每个国家均表示为一个可能的解空间，在模拟社会政治进化的过程中逐渐优化国家的解空间，找到最优解。算法主要包括帝国初始化、帝国同化、帝国革命、殖民地竞争四个步骤[18]。

a)帝国初始化：通过随机生成的方式产生多个国家，根据国家的强弱分为殖民国家与殖民地，各殖民国家及其所属殖民地构成一个帝国。

b)帝国同化：将每个帝国的殖民国家的解信息传递给其殖民地，殖民地逐步靠近殖民国家，帝国整体解质量得到逐步优化。

c)帝国革命：该阶段模拟历史进程中的革命行为，一定几率下殖民地会主动优化，甚至超越其所属殖民国家成为帝国的殖民国家。

d)殖民地竞争：该阶段是殖民地再分配的过程，帝国之间通过竞争机制争夺殖民地。理想情况下会只存在一个帝国，即只有一个殖民国家，其他国家均沦为殖民地，此时算法停止，殖民国家即代表算法优化的最优解；若仍存在多个国家则转到帝国同化阶段，依次循环更新。

2.2　编解码策略

针对 CVRP 问题，算法中的国家定义为每辆车客户点遍历顺序串联组成的数组 C_i。为了实现帝国竞争算法的搜索空间与 CVRP 问题的解空间之间地转换，本文采用一种基于贪婪准则划分实数的编解码策略：每次从数组第一个客户点开始遍历，当一辆车配送服务至当前客户点，加入下一个点则超载时截断为一辆车的配送服务范围，确定为一条配送路线，然后由下一个点开始安排下一辆车开始配送，直至所有客户点均安排配送车辆。

例如，对于解空间 C={1,3,5,4,6,2,8,7}，客户点对应需求量 Q={10,22,7,10,9,15,9,18}，车辆最大载货量为 40，则解编码后结果为：$route_1$={(1,10),(3,22),(5,7)}，$route_2$={(4,10),(6,9),(2,15),(8,6)}，$route_3$={(7,18)}。最后可以得到解空间 C 代表的配送车辆路径规划结果为：车辆 1 配送路径为“0-1-3-5-0”，车辆 2 配送路径为“0-4-6-2-8-0”，车辆 3 配送路径为“0-7-0”，其中标号“0”表示配送仓库中心。

2.3　算法设计

2.3.1 初始化

算法中每个国家即代表一个解，国家的初始化通过对各客户点随机组合的方式生成，然后根据式(1)计算所有国家配送总路径长度并由小到大升序排列，选择前 N_{imp} 个国家作为初始的殖民国家，其余国家作为殖民地。最后分别计算殖民国家的势力值，势力值的大小与总路径长度有关。第 i 个殖民国家势力值由如下公式计算：

$$p_i=\frac{S.D(C_i)}{\sum_{j=1}^{N_{imp}}S.D(C_j)} \tag{6}$$

$$S.D(C_i)=\frac{1}{D(C_i)-\min\{D(C_j)\}+1} \tag{7}$$

其中 p_i 为第 i 个殖民国家的势力值，$D(C_i)$为第 i 个殖民国家配送服务总路径长度，$\min\{D(C_j)\}$为殖民国家中配送总路径最小值，$S.D(C_i)$为标准化总路径长度。则第 i 个殖民国家分配的殖民地个数 $N.C_i$ 为

$$N.C_i=\text{Round}\{p_i\cdot(N-N_{imp})\} \tag{8}$$

其中 Round 函数表示为四舍五入取整。

2.3.2 帝国更新

a)帝国同化：本文通过随机替换再重建的方式将殖民国家解空间的信息传递给殖民地。例如，已知殖民地解空间为 C_{col}={1,3,5,4,6,2,8,7}，其所属殖民国家解空间为 C_{imp}={4,2,8,1,5,3,6,7}，同化过程中间变量示例如表 1 所示。首先，在殖民国家解空间各客户点的编码位置上，随机生成一个 0~1 范围内的键值，组成概率向量 R。设置传递阈值 α，本文取 0.5，接下来具体步骤为：将概率大于 α 的位置直接放入同化殖民地 C_{new1} 中，其中“*”表示该位置上客户点未知；若“*”位置对应殖民地解空间位置的客户点未出现在 C_{new1} 中，则将其直接放入该“*”位置得到 C_{new2}；剩余“3”号、“8”号客户点的位置，则通过同时随机插入 3 次取配送总路径最小的方式确定，可插入位置包括数组首尾以及已知节点间任意位置。假设随机插入 3 次分别得到 C_{temp1}、C_{temp2} 以及 C_{temp2}，且 $D(C_{temp1}) < D(C_{temp2}) < D(C_{temp3})$，则最终同化殖民地 $C_{new}=C_{temp1}$。

表 1　帝国同化过程中间变量示例

Tab. 1　Example of intermediate variables in the process of imperialist assimilation

变量	变量值
R	{0.5,0.6,0.2,0.8,0.9,0.4,0.7,0.3}
C_{new1}	{4,2,*,1,5,*,6,*}
C_{new2}	{4,2,*,1,5,*,6,7}
C_{temp1}	{4,2,8,1,3,5,6,7}
C_{temp2}	{4,2,8,3,1,5,6,7}
C_{temp3}	{8,4,2,3,1,5,6,7}

b)帝国强化：主要由殖民地革命与殖民国家强化两个阶段组成。殖民地革命阶段具体操作为：计算每个殖民地的革命概率，对于每个殖民地各产生一个 0~1 范围的随机键值，若小于等于其革命概率，则随机选取该殖民地解空间两个位置的客户点编号进行交换。其中革命概率调节公式为

$$p_n = p_0(1-\frac{p_i}{\max\{p_i\}}\cdot\exp(-\frac{t}{t_{\max}})) \tag{9}$$

其中 p_i 为当前殖民地所属殖民国家的势力值，$\max\{p_i\}$ 为所有殖民国家中最大势力值，t 为算法当前运算迭代次数，t_{max} 为算法总迭代次数，p_0=0.3。

殖民国家强化阶段具体操作为：随机选取殖民国家解空间中两个客户点编号，分别随机插入 3 次，利用贪心准则保留插入后配送总路径长度最小的解，得到第一个候选殖民国家；将上一步两个插入位置的中间客户点序列进行翻转，得到第二个候选殖民国家；将上一步翻转序列的部分，通过随机的方式选择两个位置的客户点编号进行交换，得到第三个候选殖民国家。将原始殖民国家与三个候选殖民国家进行比较，选择配送总路径长度最小的国家作为强化后的殖民国家。

c)帝国分裂：本文提出帝国分裂策略解决帝国竞争算法易过早收敛的问题。在帝国同化与帝国强化两个阶段，均有可能出现殖民地优于其所属殖民国家的情况，帝国分裂策略允许殖民地割裂当前帝国、并创建新的帝国，从而增加算法运行后期帝国数量，保持帝国之间竞争状态，促进解的不断优化。

本文通过对当前帝国设置分裂概率，当概率值大于设定阈值时，该殖民地上升为新的殖民国家，将该帝国一定比例的殖民地分配给新殖民国家，成立新帝国。本文分裂阈值为 0.5，分裂比例为 50%，分裂概率计算公式为

$$S.P_i = \beta\cdot\frac{N_i}{N}\cdot\exp(\frac{t}{t_{\max}})\cdot x_t \tag{10}$$

$$x_t=\begin{cases}1, & N_{cur} < N_{imp};\\ 0, & \text{其他}.\end{cases} \tag{11}$$

其中 N_i 为第 i 个帝国拥有的殖民地数量，N_{cur} 为当前帝国数，β 为调节因子，本文取 0.36。

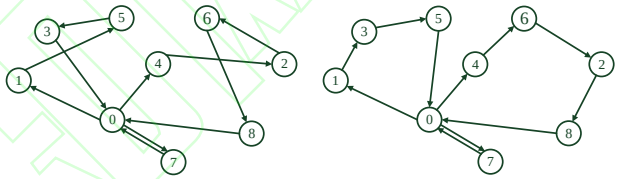
d)帝国竞争：本文对所有帝国进行帝国竞争胜利概率计算，将胜利概率最小的帝国中配送总路径最长的殖民地，分配给胜利概率最大的帝国。帝国竞争胜利概率计算公式为

$$I.P_i = (1-\zeta)\frac{\sum_{j=1}^{N_i} p_{ij}}{N_i} + \zeta\, p_i - \text{Rand}(0,\frac{N_i}{N}) \tag{12}$$

其中 N_i 为第 i 个帝国的殖民地个数；p_i 为第 i 个帝国的殖民国家势力值，p_{ij} 为第 i 个帝国第 j 个殖民地的势力值；ζ 为权系数，本文取 0.1；Rand 表示为根据两个输入参数生成闭区间内随机值。

2.3.3 局部搜索策略

针对 CVRP 问题，本文设计了一种 2-Opt 局部搜索策略[5]。根据编解码策略将解空间拆分为每辆配送车辆的路线，各路线包括配送中心节点独立使用 2-Opt 进行局部搜索，优化每条路线的客户节点遍历顺序，使得每辆车当前配送服务费用最优。局部搜索策略示意图如图 1 所示：图 1(a)为 3.1 节例子代表 3 辆车对 8 个客户节点服务的路线图；使用 2-Opt 局部搜索策略优化后的路线图如图 1(b)所示。明显地，通过 2-Opt 局部搜索可以使本次配送服务费用降低。



(a)优化前路径　　(b)优化后路径

图 1　局部搜索策略示意图

Fig. 1　Local search strategy diagram

2.3.4 算法流程

综上所述，本文提出算法的具体步骤如下所示。算法终止条件为优化迭代次数达到最大设定次数，或者优化过程中出现只有一个帝国存在时。

a) 初始化算法中最大迭代次数 $t_{\max}$、初始国家数量 N_{all}、初始帝国数量 N_{imp}、同化传递阈值 α、、革命概率调节因子 p_0、竞争胜利概率权系数 ζ 等基本参数；

b) 使用随机方法初始化国家种群。

c) 根据各国家配送总成本从小到大排序，选择前 N_{imp} 个国家作为殖民国家，剩余国家作为殖民地，根据式(6)(7)计算殖民国家势力值，根据式(8)分配殖民地组成帝国，然后根据式(10)(11)计算各帝国分裂概率。

d) 每个殖民地生成随机概率向量，使用替换重建方式进行殖民地同化操作，假设帝国 i 同化后殖民地为 a_i，再对 a_i 使用 2-Opt 局部搜索策略。

e) 若 $D(a_i) < D(i)$，则转步骤 f)；否则，转步骤 g)；

f) 若 $S.P_i > P_{sr}$，则 a_i 成为新的殖民国家，将帝国 i 中比例 P_s 的殖民地分配给 a_i，更新 $S.P_i$ 并计算 a_i 分裂概率；否则 a_i 取代 i 成为殖民国家，i 沦为 a_i 的殖民地，更新 $S.P_i$；

g) 根据式(9)计算每个殖民地革命概率 p_n，若产生 0~1 之间随机数小于 p_n，则帝国 i 生成革命殖民地 r_i，转步骤 h)；否则转步骤 j)。

h) 若 $D(r_i) < D(i)$，则转步骤 i)；否则，转步骤 j)；

i) 若 $S.P_i > P_{sr}$，则 r_i 成为新的殖民国家，将帝国 i 中比例 P_s 的殖民地分配给 r_i，更新 $S.P_i$ 并计算 r_i 分裂概率；否则 r_i 取代 i 成为殖民国家，i 沦为 r_i 的殖民地，更新 $S.P_i$；

j) 所有殖民国家进行强化操作，强化操作后分别进行 2-Opt 局部搜索策略；

k) 根据式(12)计算帝国竞争胜利概率值，将胜利概率值最小帝国的配送成本最大殖民地，分配给胜利概率最大帝国；

l) 判断是否满足迭代停止条件，若小于最大迭代次数，或者只有一个帝国存在时，算法停止迭代，输出最优帝国；否则，转步骤 d)。

3　实验与分析

3.1　测试用例与环境

为了验证ICAS求解CVRP的性能，本文选用25个CVRP基准算例进行测试 (http://neo.lcc.uma.es/vrp/vrp-instances/capacitated-vrp-instances/.)。实验环境：CPU为Intel (R) Core (TM) i5-3230M@2.60 GHz，内存为8GB，操作系统为64位Windows 10，仿真软件使用Visual Studio 2013。均采用浮点数运算。实验中的误差由如下公式定义，其中“已知最优解”指测试库提供的最优解。

$$\text{误差}=\frac{\text{最优解}-\text{已知最优解}}{\text{已知最优解}}\% \tag{13}$$

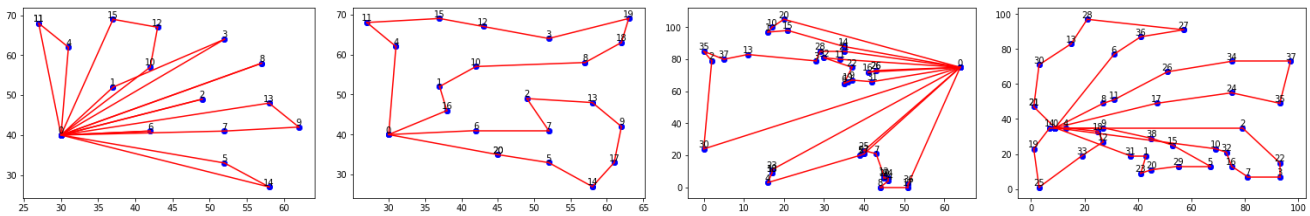
3.2　数据分析

实验 1　本文参数设置最大迭代次数为 200，初始国家数量为 100，初始帝国数量为 15。每个基准算例独立运行 20 次统计得到实验结果，如表 2 所示。其中，BSK 表示已知最优解，*Cost* 表示目标函数值即配送服务所行程总路程，*Gap* 表示误差率，BS 表示算法优化最优解，AVS 表示算法平均优化解，AVT 表示算法平均优化耗时。由表 2 可以看出，ICAS 针对 CVRP 问题具备有效求解精度以及较快的求解速度。在求解精度方面，ICAS 对于使用的 25 个基准算例均能求得最优解或接近最优解，算法优化最优解误差均不超过 1.0%，平均最优解误差均不超过 5.85%，最优解误差取平均值为 0.16%，平均最优解误差取平均值为 1.98%；25 组测试用例中有 15 组找到了最优解，车辆使用数量均达到最优，且该 15 组测试算例中 6 组能稳定求解得到最优解，即平均解误差为 0.0%。在求解速度方面，所有算例均在毫秒级时间完成优化，且均不超过 600 毫秒。算法对部分基准算例优化最优解如图 2 所示，图中标号表示需要进行服务的客户点，“0”标号点为配送服务中心。

表 2　ICAS 求解 CVRP 算例实验结果

Tab. 2　Results of ICAS for CVRP instance

Instance	BSK	BS		AVS		AVT/s
	Cost	*Cost*	*Gap*	*Cost*	*Gap*	
A-n33-k5	661	**661**	**0.0%**	666.25	0.79%	0.24
A-n33-k6	742	743	0.13%	749.1	0.96%	0.15
A-n36-k5	799	**799**	**0.0%**	800.05	0.13%	0.14
A-n37-k6	949	955	0.63%	986.75	3.98%	0.3
A-n38-k5	730	**730**	**0.0%**	747.05	2.34%	0.16
A-n39-k5	822	**822**	**0.0%**	848.0	3.16%	0.26
A-n39-k6	831	835	0.48%	864.5	4.03%	0.19
A-n44-k7	937	**937**	**0.0%**	984.2	5.04%	0.33
B-n31-k5	672	**672**	**0.0%**	683.1	1.65%	0.15
B-n34-k5	788	**788**	**0.0%**	798.35	1.31%	0.19
B-n35-k5	955	956	0.1%	974.3	2.02%	0.17
B-n38-k6	805	**805**	**0.0%**	822.1	2.12%	0.24
B-n41-k6	829	832	0.36%	853.5	2.96%	0.24
B-n43-k6	742	743	0.13%	758.55	2.23%	0.26
B-n45-k5	751	755	0.53%	768.85	2.38%	0.36
B-n52-k7	747	749	0.27%	766.2	2.57%	0.42
P-n16-k8	450	**450**	**0.0%**	**450.0**	**0.0%**	0.1
P-n19-k2	212	**212**	**0.0%**	**212.0**	**0.0%**	0.02
P-n20-k2	216	**216**	**0.0%**	**216.0**	**0.0%**	0.03
P-n21-k2	211	**211**	**0.0%**	**211.0**	**0.0%**	0.04
P-n22-k2	216	**216**	**0.0%**	**216.0**	**0.0%**	0.04
P-n22-k8	590	**590**	**0.0%**	**590.0**	**0.0%**	0.05
P-n23-k8	529	**529**	**0.0%**	531.1	0.4%	0.16
P-n45-k5	510	515	0.98%	539.85	5.85%	0.29
P-n55-k8	588	590	0.34%	620.25	5.48%	0.55



(a)ICAS 优化 P-n16-k8 最优解　(b)ICAS 优化 P-n21-k2 最优解　(c)ICAS 优化 B-n38-k6 最优解　(d)ICAS 优化 A-n39-k5 最优解

图 2　ICAS 优化不同 CVRP 算例的最优解

Fig. 2　Optimal Solution of ICAS for different CVRP instance

实验 2　在实验环境及参数设置相同的情况下，本文选取文献[14]的帝国竞争算法(标记为 ICA-CVRP)、文献[15]的混合帝国竞争算法(标记为 HICA-CVRP)、文献[18]的改进型帝国竞争模型算法(标记为 IICA-CVRP)对 CVRP 进行求解，所得的实验结果与 ICAS 进行比较，如表 3 所示，其中 BSG 表示算法优化最优解误差率。由表 3 可知，在 4 种帝国竞争算法中，IICA-CVRP 的优化耗时最长，总体的求解能力最差。而相对于 ICA-CVRP、HICA-CVRP 两个算法，ICAS 求解 CVRP 的优化耗时更多，但是求解能力却明显更优，尤其是在大规模的算例中。特别地，在 A-n44-k7 中，CAS 的优化耗时多比 ICA-CVRP、HICA-CVRP 多，但是，ICAS 的误差率为 0%，远远优于 ICA-CVRP 的 7.47%和 HICA-CVRP 的 8.22%。

表 3　ICAS 与其他帝国竞争算法实验结果对比

Tab. 3　Comparison of results between ICAS and other imperialist competition algorithms

Instance	ICAS		ICA-CVRP		HICA-CVRP		IICA-CVRP	
	BSG	AVT	BSG	AVT	BSG	AVT	BSG	AVT
A-n33-k5	**0.0%**	0.24	8.93%	0.06	3.78%	0.13	11.04%	0.83
A-n33-k6	0.13%	0.15	2.29%	0.05	1.35%	0.08	4.58%	0.72
A-n36-k5	**0.0%**	0.14	8.89%	0.05	5.01%	0.09	8.64%	0.35
A-n37-k6	0.63%	0.3	7.8%	0.1	3.16%	0.08	9.17%	0.59
A-n38-k5	**0.0%**	0.16	5.21%	0.11	1.92%	0.09	9.18%	0.75
A-n39-k5	**0.0%**	0.26	10.83%	0.06	8.76%	0.1	16.42%	0.53
A-n39-k6	0.48%	0.19	6.5%	0.05	6.74%	0.08	9.51%	0.73
A-n44-k7	**0.0%**	0.33	7.47%	0.07	8.22%	0.12	12.59%	0.89
B-n31-k5	**0.0%**	0.15	2.53%	0.05	1.19%	0.09	2.83%	0.37
B-n34-k5	**0.0%**	0.19	2.79%	0.07	1.27%	0.08	4.57%	0.68
B-n35-k5	0.1%	0.17	4.19%	0.05	2.2%	0.08	3.35%	0.91
B-n38-k6	**0.0%**	0.24	3.35%	0.06	2.61%	0.08	4.97%	1.45
B-n41-k6	0.36%	0.24	5.07%	0.06	4.34%	0.09	7.6%	0.95
B-n43-k6	0.13%	0.26	9.16%	0.06	2.56%	0.09	7.01%	0.97
B-n45-k5	0.53%	0.36	8.66%	0.07	5.33%	0.12	9.19%	0.79
B-n52-k7	0.27%	0.42	8.17%	0.17	3.75%	0.11	6.02%	0.88
P-n16-k8	**0.0%**	0.1	**0.0%**	0.04	**0.0%**	0.06	**0.0%**	0.17
P-n19-k2	**0.0%**	0.02	**0.0%**	0.03	**0.0%**	0.04	**0.0%**	0.21
P-n20-k2	**0.0%**	0.03	0.93%	0.11	0.93%	0.05	1.39%	0.22
P-n21-k2	**0.0%**	0.04	0.95%	0.05	0.95%	0.05	4.27%	0.37
P-n22-k2	**0.0%**	0.04	2.31%	0.04	0.93%	0.06	4.63%	0.39
P-n22-k8	**0.0%**	0.05	**0.0%**	0.04	**0.0%**	0.04	**0.0%**	0.27
P-n23-k8	**0.0%**	0.16	1.13%	0.05	**0.0%**	0.07	2.08%	0.4
P-n45-k5	0.98%	0.29	10.0%	0.08	13.53%	0.14	24.51%	0.69
P-n55-k8	0.34%	0.55	17.01%	0.08	19.05%	0.20	25.51%	0.98

实验 3　为了进一步验证 ICAS 算法性能，本文选取若干个启发式算法与 ICAS 进行对比实验，结果如表 4 所示。其中，CSA[10]为结合 2-Opt 算法和双桥运算的布谷鸟搜索算法，BLS-PSO[7]为基于双层局部搜索的粒子群优化算法，GA+GM[8]为二进制编码方法的遗传算法。

从表 4 可得：同样结合 2-Opt 算法进行改进的情况下，ICAS 在求解能力与优化耗时方面均明显优于 CSA；相较于使用经典遗传算法的 GA+GM，ICAS 同样在求解精度和求解效率上表现优异；相较于 BLS-PSO，虽然某些最优值求解方面 ICAS 不如 BLS-PSO，但在优化耗时方面，ICAS 平均优化耗时能够保持在毫秒级。以上分析结果表明：ICAS 不仅能有效地求解 CVRP 问题，同时相对于其他算法而言在求解效率方面有着较好表现。由此可见，在帝国竞争算法中加入帝国分裂策略与结合 2-Opt 的局部搜索策略，能有效解决帝国竞争算法过早收敛的问题，提高算法全局搜索能力和收敛速度，本文提出的 ICAS 为求解 CVRP 问题提供了较优的思路。

表 4　ICAS 与其他算法实验结果对比

Tab. 4　Comparison of results between ICAS and other algorithms

Instance	ICAS		CSA		BLS-PSO		GA+GM	
	BSG	AVT	BSG	AVT	BSG	AVT	BSG	AVT
A-n33-k5	**0.0%**	0.24	38.28%	0.92	**0.0%**	**0.15**	-	-
A-n33-k6	0.13%	**0.15**	35.44%	1.25	**0.0%**	0.43	-	-
A-n36-k5	**0.0%**	**0.14**	36.67%	0.95	**0.0%**	0.8	-	-
A-n37-k6	0.63%	**0.3**	34.77%	2.71	**0.0%**	0.64	-	-
A-n38-k5	**0.0%**	**0.16**	69.73%	1.79	**0.0%**	0.69	-	-
A-n39-k5	**0.0%**	**0.26**	59.12%	1.48	**0.0%**	1.07	-	-
A-n39-k6	0.48%	**0.19**	55.48%	1.56	**0.0%**	1.15	-	-
A-n44-k7	**0.0%**	**0.33**	-	-	**0.0%**	1.03	-	-
B-n31-k5	**0.0%**	**0.15**	11.01%	0.95	**0.0%**	2.07	-	-
B-n34-k5	**0.0%**	**0.19**	27.79%	1.92	**0.0%**	2.42	-	-

续表 4

Instance	ICAS		CSA		BLS-PSO		GA+GM	
	BSG	AVT	BSG	AVT	BSG	AVT	BSG	AVT
B-n35-k5	0.1%	**0.17**	30.89%	1.12	**0.0%**	1.12	-	-
B-n38-k6	**0.0%**	**0.24**	31.3%	1.01	**0.0%**	3.01	-	-
B-n41-k6	0.36%	**0.24**	57.78%	1.37	**0.0%**	3.93	-	-
B-n43-k6	0.13%	**0.26**	42.32%	1.17	**0.0%**	2.55	-	-
B-n45-k5	0.53%	**0.36**	85.35%	2.72	**0.0%**	2.04	-	-
B-n52-k7	0.27%	**0.42**	72.16%	1.35	**0.0%**	6.18	-	-
P-n16-k8	**0.0%**	**0.1**	0.22%	3.00	**0.0%**	0.11	2.67%	60
P-n19-k2	**0.0%**	**0.02**	-	-	**0.0%**	0.1	20.28%	70
P-n20-k2	**0.0%**	**0.03**	-	-	**0.0%**	0.35	18.06%	80
P-n21-k2	**0.0%**	**0.04**	-	-	**0.0%**	0.32	8.53%	90
P-n22-k2	**0.0%**	**0.04**	-	-	**0.0%**	0.71	24.07%	100
P-n22-k8	**0.0%**	**0.05**	7.12%	41.34	2.2%	0.83	4.75%	100
P-n23-k8	**0.0%**	**0.16**	2.46%	458.47	**0.0%**	1.02	8.51%	110
P-n45-k5	0.98%	**0.29**	67.65%	2.11	**0.0%**	1.45	-	-
P-n55-k8	0.34%	**0.55**	72.45%	1.61	-	-	-	-

4 结束语

本文提出 ICAS 求解 CVRP 问题。算法设计了一种帝国分裂策略用于解决帝国竞争算法过早收敛的问题，使算法运算后期仍然保持帝国间的竞争状态，促进解的不断优化，提高算法全局搜索能力；算法设计了结合 2-Opt 的局部搜索策略，应用于殖民地同化阶段与殖民国家增强阶段，提高算法局部搜索能力和收敛速度。本文通过 25 个基准算例，将 ICAS 与其他启发式算法从求解精度和求解效率两个方面进行对比实验，包括已有的帝国竞争算法、经典的遗传算法和粒子群算法。实验表明：ICAS 能有效求解 CVRP 问题，所有测试用例均求得最优解或接近最优解；保持较好求解精度的同时，能够保持较好的求解效率。在未来的工作中，仍需要对算法参数取值等做进一步研究，还可以尝试把 ICAS 应用于实际物流运输调度问题等更复杂的组合优化问题。

参考文献：

[1] 李卫斌, 董影影, 李小林, 等. 改进蚁群算法在应急 VRP 中的应用及收敛性分析 [J]. 计算机应用研究, 2014, 31 (12): 3557-3559+3567. (Li Weibin, Dong Yingying, Li Xiaolin, *et al.* Application of improved ant colony algorithm in emergency VRP and its convergence analysis [J]. Application Research of Computers, 2014, 31 (12): 3557-3559+3567.)

[2] Dantzig G B, Ramser J H. The truck dispatching problem [J]. Management Science, 1959, 6 (1): 80-91.

[3] Chen Ailing, Yang Genke, Wu Zhiming. Hybrid discrete particle swarm optimization algorithm for capacitated vehicle routing problem [J]. Journal of Zhejiang University SCIENCE A, 2006, 7 (4): 607-614.

[4] 万博, 卢昱, 陈立云, 等. 求解 CVRP 的改进混合蛙跳算法研究 [J]. 计算机应用研究, 2011, 28 (12): 4503-4506. (Wan Bo, Lu Yu, Chen Liyun, *et al.* Study of modified shuffled frog leaping algorithm for solving CVRP [J]. Application Research of Computers, 2011, 28 (12): 4503-4506.)

[5] 蔡延光, 陈厚仁, 戚远航. 变邻域量子烟花算法求解 CVRP [J]. 计算机工程与应用, 2019, 55 (09): 230-236. (Cai Yanguang, Chen Houren, Qi Yuanhang. Variable neighborhood quantum fireworks algorithm for solving CVRP [J]. Computer Engineering and Applications, 2019, 55 (09): 230-236.)

[6] Akhand M A H, Peya Z J, Sultana T, *et al.* Solving capacitated vehicle routing problem using variant sweep and swarm intelligence [J]. Journal of Applied Science and Engineering, 2017, 20 (4): 511-524.

[7] Ahmed A K M F, Sun J U. Bilayer local search enhanced particle swarm optimization for the capacitated vehicle routing problem [J]. Algorithms, 2018, 11 (3): 31-52.

[8] Lima S J D A, Araújo S A D A. New binary encoding scheme in genetic algorithm for solving the capacitated vehicle routing problem [C]// BIOMA 2018: Bioinspired Optimization Methods and Their Applications, Cham: Springer. 2018: 174-184.

[9] 姜昌华, 戴树贵, 胡幼华. 求解车辆路径问题的混合遗传算法 [J]. 计算机集成制造系统, 2007, 13 (10): 2047-2052. (Jiang Changhua, Dai Shugui, Hu Youhua. Hybrid genetic algorithm for capacitated vehicle routing problem [J]. Computer Integrated Manufacturing System, 2007, 13 (10): 2047-2052.)

[10] Santillan J H, Tapucar S, Manliguez C, *et al.* Cuckoo search via Lévy flights for the capacitated vehicle routing problem [J]. Journal of Industrial Engineering International, 2018, 14: 293–304.

[11] Atashpaz-Gargari E, Lucas C. Imperialist competitive algorithm: An algorithm for optimization inspired by imperialistic competition [C]// IEEE Congress on Evolutionary Computation. Singapore: IEEE, 2007: 4661-4667.

[12] 戚远航, 蔡延光, 蔡颢, 等. 旅行商问题的混沌混合离散蝙蝠算法 [J]. 电子学报, 2016, 44 (10): 2543-2547. (Qi Yuanhang, Cai Yanguang, Cai Hao, *et al.* Chaotic hybrid discrete bat algorithm for traveling salesman problem [J]. Acta Electronica Sinica, 2016, 44 (10): 2543-2547.)

[13] 杨进, 郑允, 马良. 改进的猫群算法求解 TSP [J]. 计算机应用研究, 2017, 34 (12): 3607-3610. (Yang Jin, Zheng Yun, Ma Liang. Improved cat swarm optimization for solving traveling salesman problem [J]. Application Research of Computers, 2017, 34 (12): 3607-3610.)

[14] 张鑫龙, 陈秀万, 肖汉, 等. 一种求解旅行商问题的新型帝国竞争算法 [J]. 控制与决策, 2016, 31 (04): 586-592. (Zhang Xinlong, Chen Xiuwan, Xiao Han, *et al.* A new imperialist competitive algorithm for solving TSP problem [J]. Control and Decision, 2016, 31 (04): 586-592.)

[15] 裴小兵, 于秀燕, 王尚磊. 混合帝国竞争算法求解旅行商问题 [J]. 浙江大学学报 (工学版) , 2019, 53 (10): 2003-2012. (Pei Xiaobing, Yu Xiuyan, Wang Shanglei. Solution of traveling salesman problem by hybrid imperialist competitive algorithm [J]. Journal of Zhejiang University (Engineering Science) , 2019, 53 (10): 2003-2012.)

[16] Ewbank H, Wanke P, Hadi-Vencheh A. An unsupervised fuzzy clustering approach to the capacitated vehicle routing problem [J]. Neural Comput

& Applic, 2016, 27: 857-867.

[17] Zhang S Z, Lee C K M. An improved artificial bee colony algorithm for the capacitated vehicle routing problem [C]// 2015 IEEE International Conference on Systems, Man, and Cybernetics (SMC) . Hong Kong: IEEE press, 2015: 2124-2128.

[18] 陈禹，冯翔，虞慧群. 改进型帝国竞争模型算法的研究 [J]. 计算机工程与应用, 2018, 54 (12): 206-213. (Chen Yu, Feng Xiang, Yu Huiqun. Research of improved imperialist competitive algorithm [J]. Computer Engineering and Applications, 2018, 54 (12): 206-213.)