

求解带容量约束车辆路径问题的混合变邻域 生物共栖搜索算法

李 阳, 范厚明[†]

(大连海事大学 战略管理与系统规划研究所, 辽宁 大连 116026)

摘 要: 针对带容量约束的车辆路径问题, 提出一种混合变邻域生物共栖搜索算法. 设计基于客户点优先序列及车辆参考点模拟信息的有序编码, 该编码方案使生物共栖搜索算法可以参与 CVRP 的离散优化; 为了提高算法的全局搜索能力, 根据有序编码特点构造 3 种共栖搜索算子, 扩大搜索空间; 同时, 结合变邻域搜索算法设计客户点重置、交换和 2-OPT 三种局部搜索策略, 以提高解方案质量. 算例验证分析表明, 所提算法能够有效地解决容量约束车辆路径问题, 求解质量优于所对比算法, 具有可靠的全局稳定性.

关键词: 交通工程; 容量约束车辆路径问题; 生物共栖搜索算法; 变邻域搜索算法; 有序编码

中图分类号: TP301.6

文献标志码: A

Hybrid variable neighborhood symbiotic organisms search for capacitated vehicle routing problem

LI Yang, FAN Hou-ming[†]

(Institute of Strategy Management and System Planning, Dalian Maritime University, Dalian 116026, China)

Abstract: In order to solve the capacitated vehicle routing problem(CVRP), a hybrid variable neighborhood symbiotic organisms search(HVNSOS) algorithm is proposed. A new encoding scheme called ordered encoding is designed, which connects customers' priority sequence with vehicle reference points. Based on the ordered encoding, the SOS algorithm can be applied to discrete optimization problems(e.g. CVRP). For the sake of enhancing the global searching ability of the algorithm, three symbiotic operators are constructed according to the features of the encoding scheme. The symbiotic operators can also expand the search space of the CVRP. To improve the quality of solutions, the algorithm combines the variable neighborhood search algorithm to design three new local search operators of relocate, exchange and 2-OPT. The comparison and analysis of the results show that the proposed algorithm can solve the CVRP effectively, and it is superior in the performance of computing efficiency and global stability to the algorithms compared.

Keywords: traffic engineering; capacitated vehicle routing problem; symbiotic organisms search algorithm; variable neighborhood search algorithm; ordered encoding

0 引 言

车辆路径问题(VRP)是组合优化和运筹学领域的经典问题,带容量约束车辆路径问题(CVRP)作为VRP的扩展,在工程优化和现实生活中具有极为广泛的应用背景,因此如何有效地解决CVRP具有很高的实际应用和理论研究价值.

随着群智能优化算法的不断发展,运用启发式算法对CVRP问题进行求解成为了目前重要的研究方

向,如何更高效、稳定地解决CVRP仍是当前诸多学者们的研究重点.文献[1]设计了一种基于量子旋转门和灾变操作的量子进化算法,通过2-OPT再优化线路内次序;文献[2]提出了混合遗传算法,采用了新的双层染色体编码方案;文献[3]通过改进粒子更新惯性权值对粒子群算法进行了强化.然而,上述启发式算法的单独运用并不能很好地解决CVRP,在迭代过程中会出现陷入局部最优、求解不稳定等问题,因此

收稿日期: 2017-03-28; 修回日期: 2017-06-21.

基金项目: 国家自然科学基金项目(61473053, 70801007); 辽宁省社会科学规划基金重点项目(L16AGL004); 辽宁省教育厅科学技术研究一般项目(L2014046); 大连市科学技术计划项目(2010A16GX084).

责任编辑: 林崇.

作者简介: 李阳(1990—), 男, 博士, 从事物流系统优化设计和智能优化算法的研究; 范厚明(1962—), 男, 教授, 博士生导师, 从事交通运输系统规划与设计、战略管理与系统规划等研究.

[†]通讯作者. E-mail: fhm468@163.com

许多学者对此作了针对性的改进.文献[4]通过引入大领域搜索扩大种群多样性,构建了两阶段的混合变邻域算法;文献[5]将大规模邻域搜索与蚁群算法相结合,对解方案优化作了改进;文献[6]提出了用于布谷鸟算法加强局部搜索的混合智能水滴算法;文献[7]设计了混合变邻域的大邻域搜索算法;文献[8]将变邻域算法用于解局部改进,提出了混合分散搜索算法.

生物共栖搜索算法(SOS)是由Cheng等^[9]提出的一种元启发式算法,目前多被应用于频率控制、生产调度等连续数值优化和实际工程优化问题中,与遗传算法、蚁群算法等经典启发式算法相比,SOS不需要特定的算法参数,鲁棒性强.通过设计基于客户点优先序列及车辆参考点模拟信息的有序编码,本文运用结合变邻域搜索(VNS)的混合变邻域生物共栖搜索算法(HVNSOS)求解CVRP,基于领域半径减少策略的VNS能够有效跳出局部最优,算例验证及分析表明HVNSOS优于所对比的算法,求解稳定性高.

1 问题描述及模型建立

1.1 问题描述

本文所研究的是带有容量约束的车辆路径问题(CVRP),在CVRP中有一个配送中心、多个客户点和多台配送车辆,求解该问题的目标是在确保满足客户点需求的基础上合理安排车辆线路和客户点服务次序,使总配送费用最低.对于该问题,本文作如下假设:

- 1) 配送中心、客户点地理位置信息已知,配送网络中所有节点完全相互连通,配送线路起点和终点均为配送中心;
- 2) 客户点仅接受一台配送车辆进行单次服务,最终所有客户点的需求均应被满足;
- 3) 配送车辆均为同一型号,最大载重量、性能等信息已知,每台车仅能安排一条路线;
- 4) 配送中心服务能力满足客户点要求,可同时派出多辆车(不超过可用车辆总数)进行服务,客户点的最大需求量不超过车辆载重.

在CVRP中,假设有完备有向图 $G = (V, E)$.其中:所有点集合 $V = \{0\} \cup V_o$,0为配送中心,客户点集合 $V_o = \{1, 2, \dots, n\}$;边集合 $E = \{(i, j) | i, j \in V\}$, c_{ij} 为节点 i 与 j 之间的配送成本,仅与两点间距离相关;可用配送车辆集合 $K = \{1, 2, \dots, \varphi\}$,车辆的最大载重量为 Q ; d_i 为客户 i 的需求量;决策变量 x_{ijk} 表示车辆 k 是否直接从点 i 到达点 j ,是为1,否为0; y_{ik} 表示客户 $i(i \in V_o)$ 是否由车辆 k 服务,是为1,否

为0.

1.2 模型构建

所构建的CVRP模型如下:

$$\min \sum_{i \in V} \sum_{j \in V} \sum_{k \in K} c_{ij} x_{ijk}. \quad (1)$$

$$\text{s.t.} \sum_{i \in V} \sum_{j \in V_o} x_{ijk} \cdot d_j \leq Q, \forall k \in K; \quad (2)$$

$$\sum_{i \in V} x_{ijk} = \sum_{i \in V} x_{jik} = 1, \forall j \in V_o, \forall k \in K; \quad (3)$$

$$x_{ijk} = 0, \forall i = j, \forall i, j \in V, \forall k \in K; \quad (4)$$

$$\sum_{j \in V_o} x_{ojk} = \sum_{j \in V_o} x_{jok} \leq 1, \forall k \in K; \quad (5)$$

$$\sum_{k \in K} y_{ik} = 1, \forall i \in V_o; \quad (6)$$

$$\sum_{j \in V} x_{ijk} = y_{ik}, \forall i \in V_o, \forall k \in K; \quad (7)$$

$$\sum_{i \in V} x_{ijk} = y_{jk}, \forall j \in V_o, \forall k \in K; \quad (8)$$

$$\sum_{i \in S} \sum_{j \in S} x_{ijk} \leq |S| - 1, \forall S \subseteq V_o, \forall k \in K; \quad (9)$$

$$x_{ijk} \in \{0, 1\}, y_{ik} \in \{0, 1\}, \forall i, j \in V, \forall k \in K. \quad (10)$$

式(1)为目标函数,表示总成本最小化,其中成本因素仅考虑节点间路径费用;式(2)为配送车辆容量约束,保证车辆服务路径上客户点总需求不大于其最大载重量 Q ;式(3)保证每个客户均有且仅有一条车辆路径对其进行服务,此外还表示车辆进出平衡约束,车辆到达客户点后必须离开;式(4)表示相同节点之间无路径连通;式(5)保证当车辆被启用时仅有一条服务路径,且其始末点均为配送中心;式(6)保证每个客户必被某辆车服务;式(7)和(8)将决策变量 x_{ijk} 与 y_{ik} 联系起来,保证客户点被车辆服务时一定有路径与其连接;式(9)为标准支路消除约束;式(10)为决策变量属性.

2 混合变邻域生物共栖搜索算法设计

本文所设计的混合变邻域生物共栖搜索算法是对变邻域搜索(VNS)和生物共栖搜索算法(SOS)的集成,VNS和SOS都是基于种群优化的元启发式算法,搜索范围广,随机性强,是求解复杂组合优化问题的优秀算法^[10]. SOS结构简单,特定算法参数少,可实施大规模种群搜索;VNS中包含多种邻域变换策略,能够进行针对性的局部搜索,适合与各种启发式算法结合进行优化求解;混合SOS与VNS的启发式算法不仅具有参数设置简洁、局部搜索多样性强等特点,还能够使搜索过程实现由分散到收敛集聚的循

环,避免陷入局部最优,提高解方案质量.在求解本文构建的CVRP模型时,HVNSOS首先通过SOS对解方案进行第一阶段寻优,随后在该方案基础上进行大规模的邻域结构变换,深入搜索种群个体空间,搜索效

果好.HVNSOS中生物共栖和变邻域算子多样性丰富,能够通过有规律地改变种群结构,增强全局搜索能力,求得CVRP的近似最优解.算法流程如图1所示.

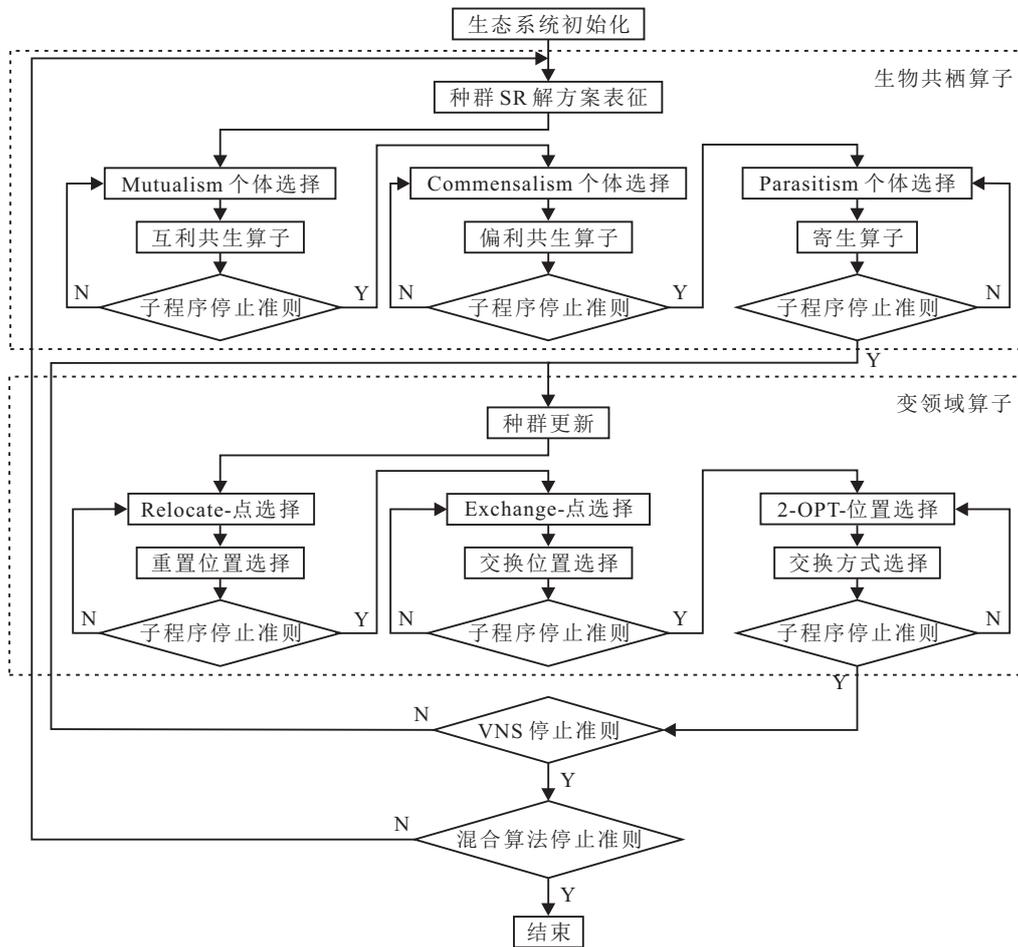


图1 HVNSOS流程图

2.1 生物共栖搜索算法

共栖是自然界不同物种间通过交互作用方式来共同生存繁衍的一种生态现象,生物共栖搜索算法对实际生态系统中物种共栖策略进行了模拟,通过仿真自然共生关系来寻找更适合生存的物种,SOS中设计了3种自然界中常见的共栖策略:互利共生(Mutualism)、偏利共生(Commensalism)、寄生(Parasitism).与GA、ACO等启发式算法类似,SOS也采用种群进化的方式对问题解空间进行大规模搜索,通过算法迭代寻找问题最优解,但SOS并不产生子代,靠不同个体间的多种交互作用产生新种群,且SOS仅设计了最大进化次数及种群规模两个参数来进行算法优化,并不需要特定的计算参数,这相对于GA、DE、PSO等启发式算法较多的(至少3个)参数影响而言,更加简洁,算法运行稳定,鲁棒性强,避免了诸参数间由于灵敏度失调所引发的调试风险^[8].在

SOS进行迭代时,首先依据有序编码方式随机生成由解方案组成的初始生态系统,随后种群中个体依次参与互利共生、偏利共生和寄生算子计算,满足共栖条件的个体将被新个体所替代,当共栖策略执行结束后,算法继续迭代直到满足其终止条件,图1中生物共栖算子描述了部分SOS流程.

2.2 初始生态系统生成

为了增强HVNSOS中初始生态系统种群的多样性,提高算法性能,本文设计了SR₁、SR₂两种算法分别生成解方案来共同组成初始种群,生态系统中SR₁解方案较多,SR₂解方案旨在增加生态系统多样性.

2.2.1 SR₁解方案编码

SOS是一种种群优化算法,最初是为了连续优化问题而设计的,在求解CVRP时并不适用,为了使其可以参与离散问题优化,采取有序编码方式将其原种群方案中的连续变量表征为离散变量,以便进行

CVRP解方案编码(文中以SR₁代表此方案)^[11]. 以6个客户点和2辆配送车辆的SR₁方案编码为例,该解方案编码时首先生成6 + 2 × 2 = 10维度的数据序列(该数据序列为向量数据,记为X),X的前6位为客户点维度,后4位为车辆维度(每辆车在序列中对应2个位). 客户点维度数据为0-1之间的随机数,由于Logistic映射能够产生一系列具有非相关性、随机性、非周期性及确定可再生的混沌序列数据^[12],本文采用经典的Logistic映射来随机生成客户点维度混沌数据. 经典Logistic的映射数学表达式如下:

$$x_{n+1} = \mu x_n(1 - x_n), \mu \in (0, 4], x \in (0, 1), \quad (11)$$

其中μ为控制参数. μ直接影响混沌数据生成时点的分布情况,当μ = 4时,该Logistic映射能生成满映

射,点分布相对较为均匀,且能够避免不动点、聚集区和空白区的出现;此外,当μ = 4时,映射 $x_{n+1} = \mu x_n(1 - x_n)$ 的最大值为μ/4,最小值为μ²/4 - μ³/16,即最大值取1,最小值取0,满足生成0-1之间随机数的要求. 因此,本文在算法中设计了μ = 4的Logistic映射来生成客户点维度随机数.

SR₁方案编码中,车辆维度数据对应为车辆地理位置参考点信息,该数据是模拟笛卡尔直角坐标系内的地理位置坐标,即给定每辆车一个具体的定位坐标. 地理位置受算例中客户点和配送中心x、y坐标范围约束,具体在6个客户点和2辆配送车辆的CVRP方案SR₁编码X中,第7、8位为两车辆的x坐标模拟值,第9、10位依次为两车辆的y坐标模拟值. CVRP解方案编码流程如图2所示.

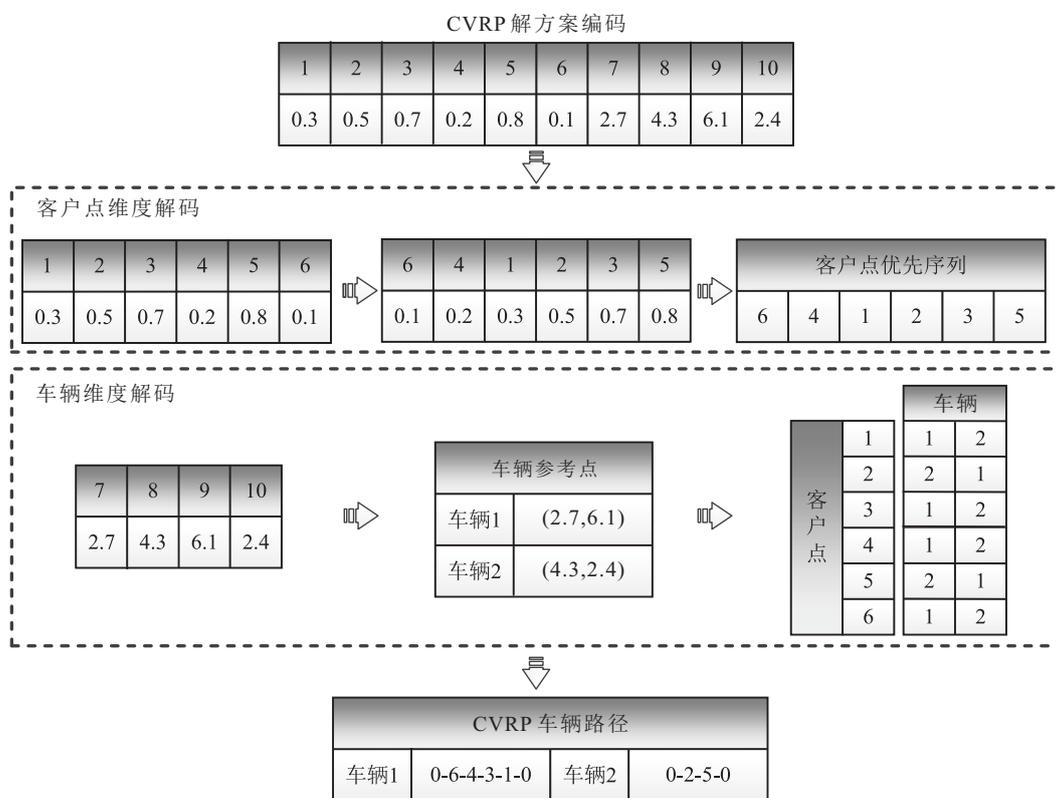


图 2 CVRP解方案编码、解码流程

2.2.2 SR₁解方案解码

编码后的SR₁方案还不能参与HVNSOS算法迭代,必须对客户点及车辆编码信息进行解码,还原其车辆路径方案,解码步骤如下:

1) 生成客户点优先序列. 将客户点维度数据由小到大排列,排序后的数据位即代表相应的客户点顺序,车辆将按照此客户定优先序列对其进行服务,具体可见图2中客户点维度解码环节.

2) 计算客户点车辆优先序列矩阵. 车辆维度数据为相应车辆地理位置参考点x、y坐标数据,客户

点实际坐标数据已知,依次计算客户点到所有车辆参考点的距离,由小到大排列可得客户点车辆优先序列矩阵,详见图2中车辆维度解码环节.

3) SR₁解方案生成. 依据客户点优先序列和客户点车辆优先序列矩阵,按照贪婪准则构造最优车辆路径. 由图2可知,应先对客户点6进行解码,6点车辆优先序列第一位为1车,应首先检验1车对6点进行服务时是否能够满足上文模型中的约束限制,若满足约束则将6点插入其当前路径中,插入位置应满足贪婪准则(选择目标函数最优位置插入),反之按照6点车

辆优先序列递补车辆2进行操作,步骤同上.当所有客户点均指派给配送车辆后,所有车辆路径构成了一个解方案,SR₁方案解码完成.值得注意的是,由于客户点按照贪婪准则安排服务路径,这会导致车辆路径中客户点顺序与客户点优先序列不同.

2.2.3 SR₂解方案生成

SR₂解方案使HVNSOS生态系统中种群更加丰富,SR₂解方案依据改进节约算法原理选择插入法生成,客户点 l 在 i, j 点之间的插入费用 $c_{ijl} = c_{il} + c_{lj} - gc_{ij} + f|c_{il} - c_{lj}|$, g 和 f 是随机数, $g \in [0, 3]$, $f \in [0, 1]$ ^[13-14].随机数 g 和 f 使SR₂解方案生成过程中插入节点之间(客户点之间、客户点与配送中心)费用不再固定,保证了种群多样性优化要求.包含SR₁和SR₂两类解方案的初始生态系统具有良好的种群分散性,个体多样性丰富,能够大范围搜索最优方案.

2.3 共栖搜索算子

算法在生成初始生态系统后对SR₁解方案进行共栖优化,HVNSOS设计了3种不同特点的共栖算子.值得注意的是,共栖算子是对解方案编码数据进行操作产生新的编码,进而解码获得新个体方案;互利共生算子中两个个体互利共赢,共同进化;偏利共生算子仅对单个个体有益,另一个个体不受影响;寄生算子中一方受损,一方获益,且受损个体退出共栖迭代.生态系统中所有个体依次按照互利共生、偏利共生、寄生顺序进行操作,根据CVRP具体问题规模共栖算子会循环运行多次,迭代至最大循环次数时终止.

2.3.1 互利共生算子

互利共生算子模拟了自然界中类似蜜蜂和花朵的共栖关系,两者均从中受益.以 i 个体为例,在执行Mutualism算子计算时,首先随机选择生态系统中另一个体 j , X_i 和 X_j 代表两个体的编码数据向量, $X_{i\text{new}}$ 和 $X_{j\text{new}}$ 则代表通过算子计算所得新个体的编码数据,具体计算过程如下:

$$X_{i\text{new}} = X_i + \text{rand}(0, 1) \cdot (X_{\text{best}} - \text{MV} \cdot \text{BF}_1), \quad (12)$$

$$X_{j\text{new}} = X_j + \text{rand}(0, 1) \cdot (X_{\text{best}} - \text{MV} \cdot \text{BF}_2), \quad (13)$$

$$\text{MV} = (X_i + X_j)/2. \quad (14)$$

其中: X_{best} 是指生态系统SR₁解方案中目标函数最小个体的编码数据; BF_1 和 BF_2 是互利因子,均随机取值1或2;式(14)是 X_i 和 X_j 个体的互利向量数据,与 i, j 编码直接相关.经过互利共生算子所得新个体在经过2.2.2节中介绍的方法对 $X_{i\text{new}}$ 和 $X_{j\text{new}}$ 解

码后,计算其目标函数值,当个体目标函数值得到改进时,可替代原有个体继续参与算法运算,反之则不作改变继续进行迭代,生态系统中所有个体均执行Mutualism算子计算,且满足子程序停止准则后进入下一步共栖算子计算.

2.3.2 偏利共生算子

鲈鱼与鲨鱼是海洋生态系统中典型的偏利共生关系,鲈鱼吸附在鲨鱼身上,依靠鲨鱼的食物残渣生存,对鲨鱼不产生利害影响,偏利共生算子的计算策略与此类似.与Mutualism算子对两个个体进行操作不同,Commensalism算子运行目标是优化一个个体,仍以 i 个体为例,执行偏利共生算子时先在生态系统中随机选中 j 个体, X_j 仅参与优化,不产生新个体,计算公式如下:

$$X_{i\text{new}} = X_i + \text{rand}(-1, 1) \cdot (X_{\text{best}} - X_j). \quad (15)$$

$X_{i\text{new}}$ 解码后计算该解方案目标函数值,目标函数改进的个体取代原个体进入生态系统,反之 X_i 不发生改变, X_j 在参与 $X_{i\text{new}}$ 优化后不作任何改变,可继续参与其他个体的偏利共生优化.

2.3.3 寄生算子

寄生现象在自然界中广泛存在,最严重时寄生源可能会导致宿主死亡,HVNSOS中寄生算子涉及 X_i 、 X_{parasite} 和 X_j , X_{parasite} 在Parasitism算子中的作用与寄生源相同, X_{parasite} 源于 X_i ,个体 j 为宿主.算子运行时,先复制编码数据 X_i ,从中随机任选一维度数据后赋予其新随机值既生成寄生个体(编码数据 X_{parasite}),随后在生态系统中选中任一个体 j ,比较 X_{parasite} 和 X_j 解码后方案的目标函数值,当 X_{parasite} 目标函数值小于 X_j 时,宿主个体 j 被寄生个体取代,反之寄生失败,保留 j 个体,消除寄生个体,寄生算子继续执行迭代.

2.4 变邻域搜索策略

生态系统中个体在执行共栖搜索算子后继续进行局部搜索改进,算法设计了客户点重置(Relocate)、客户点交换(Exchange)和边交换(2-OPT)3种VNS搜索策略,依次对其中所有解方案进行迭代优化.为了减少无效的邻域变化,算法引入了邻域半径减少策略对客户点进行距离检验,与待操作点距离较近的候选点(以距离平均值为标准)才能纳入邻域结构置换环节^[15],当解方案目标函数值累计未改进次数超出预设次数时,当前变邻域策略终止,继续执行后续VNS策略.Relocate和Exchange都是对线路中客户点进行针对性的变邻域搜索,2-OPT则针对方案子线路中部分弧作邻域变换,3种策略均能实现解方案子线路内

和子线路间操作, 邻域搜索范围广. 3种变邻域搜索策略的结合能够深入挖掘解方案邻域内局部最优解, 具体如下:

1) Relocate. 在当前解方案中随机选定某客户点, 将其重新插入在满足邻域半径减少策略检验的其余任一客户点左侧或右侧完成重置, 此外该待操作点还可生成仅有一个客户点的子路径.

2) Exchange. 随机选中当前方案中的某客户点为待操作点, 与该方案子线路中另一客户点直接交换位置即构成新方案.

3) OPT. 与 Relocate、Exchange 策略类似, 2-OPT 首先选中两客户点, 当两客户点在同一子线路内且不相邻时, 逆序两点间路径使其相连组成新线路, 当客户点相邻时不作操作重新选择客户点; 当两客户点分属不同子线路时, 将两点所在子线路从方案中分离, 通过对客户点所在子线路作顺序交叉和逆序交叉的 2-OPT 操作来生成新方案^[16].

2.5 算法计算复杂度分析

对于 HVNSOS, 当算例客户点规模为 n , 生态系统种群规模为 p_{size} , SOS 中共栖算子进化次数为 N , VNS 搜索次数为 N_1 , 通过邻域半径减少策略检验的客户点规模为 m 时, 依据图 1 的算法流程, 对各环节 i 计算复杂度 $O(T_i)$, 分析如下:

- 1) 初始生态系统 SR_1 、 SR_2 种群生成及其多样性优化 $O(T_1) = O(n \cdot p_{size})$;
- 2) 共栖搜索互利共生算子 $O(T_2) = O(N \cdot p_{size})$;
- 3) 偏利共生算子 $O(T_3) = O(N \cdot p_{size})$;
- 4) 寄生算子 $O(T_4) = O(N \cdot p_{size})$;
- 5) 变邻域 Relocate 策略 $O(T_5) = O(2m \cdot N_1 \cdot p_{size})$;
- 6) Exchange 策略 $O(T_6) = O(m \cdot N_1 \cdot p_{size})$;
- 7) 当两客户点分属不同子线路时, 最复杂情况下 2-OPT 策略 $O(T_7) = O(8m \cdot N_1 \cdot p_{size})$.

综上, HVNSOS 算法单次迭代的总计算复杂度为

$$O(T_8) = O((n + 3N + 11m \cdot N_1) \cdot p_{size}) = O(n \cdot p_{size}).$$

3 算例验证及结果分析

本文选取文献[1]中简单算例、CVRP 算例集中 Set A、Set P 以及 Set CMT 部分算例对 HVNSOS 进行测试, 编程采用 Matlab r2010b, 操作系统为 Window7, 电脑内存大小为 4.00 GB, CPU 为双核酷睿 i5-4460, 主频为 3.2 GHz. 经过反复测试, 设置算法参数如下:

生态系统种群规模 $p_{size} = 40 \sim 100$, 最大进化次数为 $10 \sim 50$, 惩罚成本 $p_c = 100 \sim 800$, 算法参数与所选用算例的客户点规模相关, 惩罚费用大小与路径总成本相关(多设为算例最优解路径成本值的 40% 左右).

表 1 文献[8]算例结果比较

算法	最优值	最差值	平均值	求得最优解比例/%
QEA ^[1]	67.5*	71	68.358	54
HGA ^[2]	67.5*	69	67.875	75
IPSO ^[3]	67.5*	70.5	68.425	50
HSSA ^[8]	67.5*	67.5*	67.5*	100
HVNSOS	67.5*	67.5*	67.5*	100

实验 1 文献[1]简单算例中有客户点 8 个, 配送车辆 2 台, 表 1 给出了文献[1-3, 8]中算法及 HVNSOS 的计算结果, 其中 * 表示已知最优解, 算例计算 20 次. 由于该算例规模相对较小, 设置算法种群规模为 20, 进化次数为 10, 惩罚成本为 20. 文献中算法均能求得算例最优解, HSSA 与本文算法性能最优, 求得最优解成功率为 100%.

实验 2 选取 CVRP 算例集 Set A 的 27 个算例, 表 2 给出包括文献[5, 17-18]和本文算法的计算结果, 算例计算 20 次, 其中: BK 表示算例已知最优解, Best、Worst、Average 表示算法最优值、最差值和平均值, % Dev 表示与已知最优解的偏差 ($\% Dev = (Best - BK)/BK$), * 表示已知最优解. 当算例中客户点规模在 $30 \sim 40$ 之间时, 算法种群规模多设置为 50 左右 ($40 \sim 60$), 进化次数为 $10 \sim 20$; 当客户点规模在 $40 \sim 50$ 之间时, 种群规模增加至 60 左右 ($50 \sim 70$), 进化次数相应增加至 20 左右, 不超过 30; 当客户点规模在 $50 \sim 70$ 之间时, 种群规模增加至 70 左右 ($60 \sim 80$), 进化次数为 $20 \sim 40$; 当客户点规模超过 70 时, 算法种群规模在 $80 \sim 100$ 之间设定, HVNSOS 进化次数取 $30 \sim 50$. 客户点规模越大, 算法参数设置相应越大.

由表 2 的结果可知: 对比文献中的算法, LNS-ACO、EBMC-ABC 和 SC-ESA 均能够求得 Set A 的 27 个算例中的 17 个最优解; 相对求解精度而言, SC-ESA 的求解结果与所有算例最优解的平均偏差为 0.16%, EBMC-ABC 的求解结果平均偏差为 0.19%, LNS-ACO 平均偏差为 0.6%, 3 种算法求解结果与最优解平均偏差均小于 1%; 本文设计的 HVNSOS 算法能够求得 Set A 的 27 个算例中的 20 个, 与算例最优解的平均偏差为 0.13%, 能够比所对比的算法求得更多的最优解, 且总体求解精度更高.

表2 CVRP Set A算例集结果比较

算例	BK	LNS-ACO(2016) ^[5]		EBMC-ABC(2017) ^[17]		SC-ESA(2013) ^[18]		HVNSOS			
		Best	% Dev	Best	% Dev	Best	% Dev	Best	Worst	Average	% Dev
A-n32-k5	784*	784*	0	784*	0	784*	0	784*	784*	784*	0
A-n33-k5	661*	661*	0	661*	0	661*	0	661*	661*	661*	0
A-n33-k6	742*	742*	0	742*	0	742*	0	742*	742*	742*	0
A-n34-k5	778*	778*	0	778*	0	778*	0	778*	778*	778*	0
A-n36-k5	799*	799*	0	799*	0	799*	0	799*	799*	799*	0
A-n37-k5	669*	669*	0	669*	0	669*	0	669*	669*	669*	0
A-n37-k6	949*	949*	0	949*	0	949*	0	949*	949*	949*	0
A-n38-k5	730*	730*	0	730*	0	730*	0	730*	730*	730*	0
A-n39-k5	822*	822*	0	822*	0	822*	0	822*	822*	822*	0
A-n39-k6	831*	831*	0	831*	0	831*	0	831*	831*	831*	0
A-n44-k6	937*	937*	0	937*	0	937*	0	937*	937*	937*	0
A-n45-k6	944*	958	1.48	949	0.53	944*	0	944*	953	952.1	0
A-n45-k7	1 146*	1 146*	0	1 146*	0	1 146*	0	1 146*	1 146*	1 146*	0
A-n46-k7	914*	914*	0	914*	0	914*	0	914*	914*	914*	0
A-n48-k7	1 073*	1 084.1	1.03	1 073*	0	1 084	1.03	1 073*	1 073*	1 073*	0
A-n53-k7	1 010*	1 010*	0	1 010*	0	1 011	0.1	1 010*	1 017	1 016.7	0
A-n54-k7	1 167*	1 167*	0	1 167*	0	1 168	0.09	1 167*	1 168	1 167.3	0
A-n55-k9	1 073*	1 073*	0	1 073*	0	1 073*	0	1 073*	1 073*	1 073*	0
A-n60-k9	1 354*	1 354*	0	1 355	0.07	1 355	0.07	1 354*	1 360	1 356.7	0
A-n61-k9	1 034*	1 067	3.19	1 035	0.09	1 034*	0	1 035	1 039	1 038.2	0.09
A-n62-k8	1 288*	1 308	1.55	1 300	0.93	1 298	0.78	1 291	1 308	1 301.4	0.23
A-n63-k10	1 314*	1 329	1.14	1 319	0.38	1 315	0.08	1 319	1 323	1 321	0.38
A-n63-k9	1 616*	1 649	2.04	1 627	0.68	1 624	0.50	1 628	1 639	1 631.3	0.74
A-n64-k9	1 401*	1 415	1	1 412	0.79	1 409	0.57	1 414	1 422	1 419.6	0.93
A-n65-k9	1 174*	1 185	0.94	1 178	0.34	1 178	0.34	1 177	1 180	1 179.1	0.26
A-n69-k9	1 159*	1 170	0.95	1 166	0.60	1 159*	0	1 159*	1 169	1 163.4	0
A-n80-k10	1 763*	1 815	2.95	1 774	0.62	1 776	0.74	1 779	1 789	1 786	0.91
Average			0.6		0.19		0.16				0.13

表3 CVRP Set P算例集结果比较

算例	BK	LNS-ACO(2016) ^[5]		SC-ESA(2013) ^[18]		HVNSOS			
		Best	% Dev	Best	% Dev	Best	Worst	Average	% Dev
P-n16-k8	450*	450*	0	450*	0	450*	450*	450*	0
P-n19-k2	212*	212*	0	219	3.30	212*	212*	212*	0
P-n20-k2	216*	216*	0	218	0.93	216*	216*	216*	0
P-n21-k2	211*	211*	0	212	0.47	211*	211*	211*	0
P-n22-k2	216*	216*	0	216*	0	216*	216*	216*	0
P-n22-k8	603*	—	—	—	—	603*	603*	603*	0
P-n23-k8	529*	529*	0	529*	0	529*	529*	529*	0
P-n40-k5	458*	458*	0	459	0.22	458*	458*	458*	0
P-n45-k5	510*	510*	0	511	0.20	510*	510*	510*	0
P-n50-k10	696*	696*	0	697	0.14	696*	697	696.85	0
P-n50-k7	554*	554*	0	554*	0	554*	554*	554*	0
P-n50-k8	631*	643	1.9	637	0.95	632	632	632	0.16
P-n51-k10	741*	747	0.81	741*	0	744	747	745.1	0.40
P-n55-k10	694*	694*	0	695	0.14	698	699	698.6	0.58
P-n55-k7	568*	568*	0	574	1.06	568*	573	570.9	0
P-n60-k10	744*	755	1.48	745	0.13	748	752	749.9	0.54
P-n60-k15	968*	977	0.93	968*	0	968*	977	973.3	0
Average			0.53		0.67				0.10

实验3 选取CVRP Set P算例集中的17个算例,表3给出了对比文献[5,18]和本文算法的计算结果.算例计算20次,表中符号含义同表2,“-”表示相应文献未求解该算例.当客户点规模在10~30之间时,算法种群规模多设置为40,进化次数为10,其余规模算例算法参数设置与实验2相似.

由表3的对比结果可知:LNS-ACO能够求得Set P的算例中的12个最优解,与最优解平均偏差为

0.53%;SC-ESA能够求得16个算例中的6个最优解,与最优解平均偏差为0.67%;本文HVNSOS能够求得17个算例中的13个最优解,与最优解平均偏差仅为0.10%.在求解精度方面,3种算法与最优解的平均偏差均小于1%,寻优质量高,但本文HVNSOS能够求得算例最优解的比例最高,求解稳定,优于所对比算法.

实验4 选取CVRP Set CMT算例集中的7个算例(即C₁~C₅、C₁₁和C₁₂,算例中无最大路径长度

及服务时间约束),表4给出了对比文献[19-21]和本文算法的计算结果,算例计算20次,%Dev_{ave}表示算法平均值与已知最优解偏差(%Dev_{ave} = (Average - BK)/BK),其他符号含义同表2. 当客户点规模大

于100(101/121/151/200)时,算法种群仍设置为90 ~ 100,进化次数为30 ~ 50,其规模不再增加,其余规模算例算法参数设置与实验2相似.

表4 CVRP Set A算例集结果比较

算例	BK	MPNS-GRASP ^[19]		VNSA ^[20]		ISOS _{SR2} ^[21]			HVNSOS					
		Best	%Dev	Best	%Dev	Best	Average	%Dev _{ave}	%Dev	Best	Worst	Average	%Dev _{ave}	%Dev
CMT-n51-k5	524.61*	524.61*	0	524.61*	0	524.61*	530.93	1.20	0	524.61*	524.61*	524.61*	0	0
CMT-n76-k10	835.26*	836.39	0.13	835.26*	0	835.74	857.68	2.68	0.06	835.26*	854.65	848.99	1.64	0
CMT-n101-k8	826.14*	826.14*	0	826.14*	0	826.14*	845.90	2.39	0	830.42	835.67	833.12	0.84	0.52
CMT-n101-k10	819.56*	821.12	0.19	819.56*	0	819.56*	834.50	1.82	0	819.56*	819.56*	819.56*	0	0
CMT-n121-k7	1 042.11*	1 042.11*	0	1 042.12	0.00	1 042.11*	1 049.95	0.75	0	1 042.11*	1 049.17	1 046.86	0.46	0
CMT-n151-k12	1 028.42*	1 032.24	0.37	1 031.44	0.29	1 028.42*	1 090.86	6.07	0	1 031.44	1 057.41	1 049.04	2.01	0.29
CMT-n200-k17	1 291.29*	1 314.25	1.78	1 305.08	1.07	1 305.49	1 370.64	6.15	1.10	1 294.06	1 363.37	1 335.31	3.41	0.21
Average			0.35		0.19			3.01	0.17				1.19	0.15

由表4的对比结果可知:各算法求解性能均十分优秀,MPNS-GRASP能够求得Set CMT中7个算例的3个最优解,VNSA和本文HVNSOS能够求得其中4个最优解,ISOS_{SR2}能够求得5个最优解;在求解精度方面,4种算法与最优解的平均偏差均小于0.5%,分别为0.35%(MPNS-GRASP)、0.19%(VNSA)、0.17%

(ISOS_{SR2})和0.15%(HVNSOS);此外,与ISOS_{SR2}相比,HVNSOS在算法多次求解的平均值方面更有优势,求解平均值与已知最优解的平均偏差为1.19%.

图3给出了算法SOS、VNS、HVNSOS计算A-n39-k5、P-n40-k5以及CMT-n51-k5算例的收敛曲线对比.由图3可知:当SOS单独计算时,容易在求得算例次优解后陷入局部最优,收敛效果较差;VNS较SOS收敛速度快,经过大量迭代后能够逐步提高解质量,最终求得算例最优解;相对SOS和VNS单独求解而言,HVNSOS能够在迭代初期快速逼近算例已知全局最优解,随后在算法10次迭代内快速收敛.可见,虽然变邻域搜索策略的引入增加了生物共栖搜索算法的计算复杂度,但其在求解精度和算法收敛性等方面的性能均得到了提升,寻优质量较高.

综合上述实验可知,本文HVNSOS在算法精度、求解稳定性等方面整体优于所对比的10种算法,是求解CVRP的有效方法.

4 结 论

针对CVRP的求解,本文提出了一种混合变邻域生物共栖搜索算法.算法采用有序编码方式生成生态系统,并根据改进节约算法生成部分种群,其解方案搜索空间大,多样性高;在共栖搜索算子解方案寻优基础上引入客户点重置、交换及2-OPT策略进行邻域变换,增强其局部搜索能力.算例验证及分析表明,HVNSOS能够有效求解CVRP,寻优质量优于所对比算法,可跳出局部最优.今后将进一步改进该算法,以加强其大规模客户算例求解性能.

参考文献(References)

[1] 赵燕伟,彭典军,张景玲.有能力约束车辆路径问题的量子进化算法[J].系统工程理论与实践,2009,29(2):159-166.

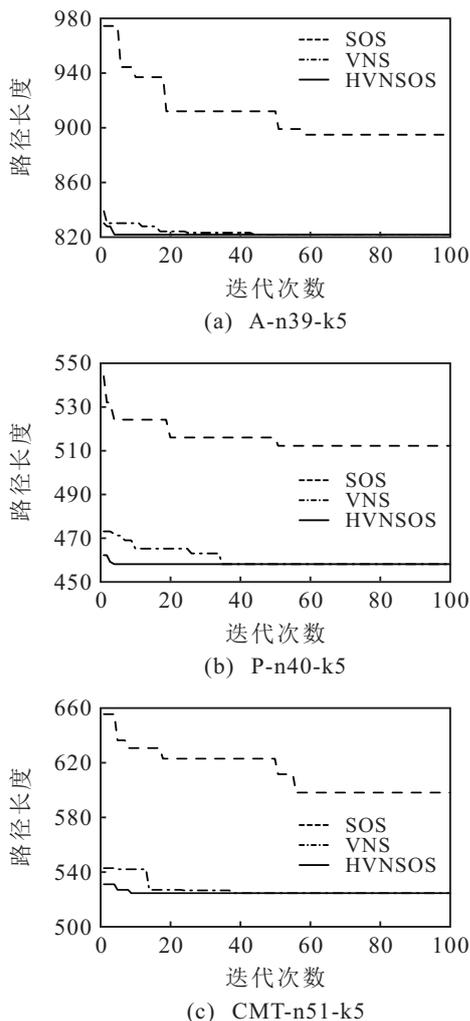


图3 SOS、VNS、HVNSOS算法收敛曲线对比

- (Zhao Y W, Peng D J, Zhang J L. Quantum evolutionary algorithm for capacitated vehicle routing problem[J]. *Systems Engineering — Theory & Practice*, 2009, 29(2): 159-166.)
- [2] 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 C H, Dai S G, Hu Y H. Hybrid genetic algorithm for capacitated vehicle routing problem[J]. *Computer Integrated Manufacturing Systems*, 2007, 13(10): 2047-2052.)
- [3] 罗先国, 侍洪波. 非满载车辆路径问题的改进粒子群优化算法[J]. *华东理工大学学报: 自然科学版*, 2006, 32(7): 767-771.
(Luo X G, Shi H B. Improved particle swarm optimization for vehicle routing problem with non-full load[J]. *J of East China University of Science and Technology: Natural Science Edition*, 2006, 32(7): 767-771.)
- [4] Jeeu F S, Said S, Niaz W. The cumulative capacitated vehicle routing problem with min-sum and min-max objectives: An effective hybridisation of adaptive variable neighbourhood search and large neighbourhood search[J]. *Transportation Research Part B: Methodological*, 2017, 101: 162-184.
- [5] Akpinar S. Hybrid large neighbourhood search algorithm for capacitated vehicle routing problem[J]. *Expert Systems with Applications*, 2016, 61: 28-38.
- [6] Teymourian E, Kayvanfar V, Komaki G M, et al. Enhanced intelligent water drops and cuckoo search algorithms for solving the capacitated vehicle routing problem[J]. *Information Sciences*, 2015, 334(1): 354-378.
- [7] Sze J F, Salhi S, Wassan N. A hybridisation of adaptive variable neighbourhood search and large neighbourhood search: Application to the vehicle routing problem[J]. *Expert Systems with Applications*, 2016, 65: 383-397.
- [8] 张晓楠, 范厚明. 混合分散搜索算法求解带容量约束车辆路径问题[J]. *控制与决策*, 2015, 30(11): 1937-1944.
(Zhang X N, Fang H M. Hybrid scatter search algorithm for capacitated vehicle routing problem[J]. *Control and Decision*, 2015, 30(11): 1937-1944.)
- [9] Cheng M Y, Prayogo D. Symbiotic organisms search: A new metaheuristic optimization algorithm[J]. *Computers & Structures*, 2014, 139: 98-112.
- [10] Boussaid I, Lepagnot J, Siarry P. A survey on optimization metaheuristics[J]. *Information Sciences*, 2013, 237(237): 82-117.
- [11] Ai T J, Kachitvichyanukul V. Particle swarm optimization and two solution representations for solving the capacitated vehicle routing problem[J]. *Computers & Industrial Engineering*, 2009, 56(1): 380-387.
- [12] 陈志刚, 梁涤青, 邓小鸿. Logistic混沌映射性能分析与改进[J]. *电子与信息学报*, 2016, 38(6): 1547-1551.
(Chen Z G, Liang D Q, Deng X H. Analysis and improvement of Logistic chaotic mapping[J]. *J of Electronics and Information Technology*, 2016, 38(6): 1547-1551.)
- [13] Zachariadis E, Tarantilis C, Kiranoudis C. A hybrid metaheuristic algorithm for the vehicle routing problem with simultaneous delivery and pick-up service[J]. *Expert Systems with Applications*, 2009, 36(2): 1070-1081.
- [14] Altnel I K, Oncan T. A new enhancement of the Clarke and Wright savings heuristic for the capacitated vehicle routing problem[J]. *J of the Operational Research Society*, 2005, 56(8): 954-961.
- [15] Hansen P, Mladenovi N. Variable neighborhood search: Principles and applications[J]. *European J of Operational Research*, 2001, 130(3): 449-467.
- [16] Chen P, Huang H K, Dong X Y. Iterated variable neighborhood descent algorithm for the capacitated vehicle routing problem[J]. *Expert Systems with Applications*, 2010, 37(2): 1620-1627.
- [17] Ng K K H, Lee C K M, Zhang S Z, et al. A multiple colonies artificial bee colony algorithm for a capacitated vehicle routing problem and re-routing strategies under time-dependent traffic congestion[J]. *Computers & Industrial Engineering*, 2017, 109: 151-168.
- [18] Stanojevic M, Stanojevic B, Vujosevic M. Enhanced savings calculation and its applications for solving capacitated vehicle routing problem[J]. *Applied Mathematics & Computation*, 2013, 219(20): 10302-10312.
- [19] Marinakis Y. Multiple phase neighborhood search-GRASP for the capacitated vehicle routing problem[J]. *Expert Systems with Applications*, 2012, 39: 6807-6815.
- [20] Xiao Y, Zhao Q, Kaku I, et al. Variable neighborhood simulated annealing algorithm for capacitated vehicle routing problems[J]. *Engineering Optimization*, 2014, 46(4): 562-579.
- [21] Yu V F, Redi A A N P, Yang C L, et al. Symbiotic organisms search and two solution representations for solving the capacitated vehicle routing problem[J]. *Applied Soft Computing*, 2017, 52: 657-672.

(责任编辑: 齐 霖)