

2017年度重庆市出版专项基金资助栏目

运筹学与控制论

DOI:10.11721/cqnuj20180220

带相容性约束的车辆路径问题及其混合蚁群算法*

余良¹, 秦虎²

(1. 武汉理工大学 经济学院, 武汉 430070; 2. 华中科技大学 管理学院, 武汉 430074)

摘要:【目的】着力设计带相容性约束的车辆路径问题的高效启发式算法。【方法】针对带相容性约束的车辆路径问题的特点,提出了一种混合蚁群算法。该算法的核心由蚁群搜索和禁忌搜索组成,对蚁群搜索的状态转移公式和信息素更新规则进行了改进,并在蚁群搜索过程中加入了一个扰动机制,同时在禁忌搜索部分采用了新的邻域结构和禁忌规则。【结果】得到了关于带相容性约束的车辆路径问题的混合蚁群算法。【结论】通过多个算例对算法进行了测试,计算结果表明该算法具有很高的求解效率。

关键词:车辆调度问题;冷链物流;蚁群算法;禁忌搜索;相容性约束

中图分类号: TP301

文献标志码: A

文章编号: 1672-6693(2018)02-0001-09

车辆路径问题(Vehicle routing problem, VRP)是物流领域中一个重要的组合优化问题,与企业的实际操作联系紧密,已有数十年的研究历史^[1]。近些年来,随着冷链物流的日益兴起,带相容性约束的车辆路径问题(Vehicle routing problem with compatibility constraints, VRPCC)日渐得到业内重视。在冷链物流的过程中,由于冷链产品的特殊性,不同种类的冷链产品可能不允许同时装载进一辆冷藏车进行配送,然而客户却可能同时对多种产品存在需求,这样就使得冷链物流的车辆路径设计变得更为复杂。例如,某个冷链物流企业的客户有水果经销商和冰激凌经销商,该企业需要同时为这两家客户运送各自需要的货物。但是,由于水果只能在4℃以上的环境下运输,而冰激凌的运输温度则要在-15℃以下,两种货物适温带不存在交集,因而不能装入同一辆冷藏车配送。如果某一客户(例如超市门店)同时需要水果和冰淇淋,那么只能由一辆冷藏车进行配送。除了温带以外,包装、化学性质等各类因素也都可能成为产品之间的一些相容性约束,这就要求企业在进行车辆路径设计求费用最小化时,还要考虑到货物能否同时运输的问题,增加了车辆路径设计的复杂度。在传统车辆路径问题的基础上引入相容性约束,便衍生出了带相容性约束的车辆路径问题。

VRP是著名的NP-难问题,因此相应的VRPCC也是NP-难问题。文献中研究VRPCC的文章非常少,并且没有算法可以用来求解VRPCC,因此本文着力于设计高效的启发式算法求解VRPCC,为冷链物流企业提供一个切实可行的解决方案。

蚁群算法^[2]是一种仿生类随机搜索算法,该算法凭借特有的搜索机制,在多类组合优化问题中表现出了在求解速度和求解稳定性等多方面的优势。刘志硕等人应用蚁群算法求解路径优化问题^[3],并取得了一定的研究成果。禁忌搜索算法则是一种亚启发式随机搜索算法,该算法具有很强的局部搜索能力,Gendreau、Jiefeng、Barbarosoglu、郎茂祥等人也都曾利用禁忌搜索算法求解路径优化问题^[3-8],并取得了丰富的研究成果。本文在充分考虑蚁群算法和禁忌搜索算法的特点以及VRPCC结构特征的基础上,针对大规模VRPCC算例设计出了一种混合蚁群算法。

1 蚁群算法基本原理与VRPCC问题描述

1.1 VRPCC问题

VRPCC指的是在一个物流系统中,有一个配送中心和区域内其服务的多个客户,各个客户之间以及各个客

* 收稿日期:2017-12-05 修回日期:2018-03-21 网络出版时间:2018-03-27 16:45

资助项目:国家自然科学基金面上项目(No. 71571077)

第一作者简介:余良,男,高级工程师,博士研究生,研究方向为运筹优化, E-mail: yuliang@szum.gov.cn; 通信作者:秦虎,教授, E-mail: tiger-qin1980@gmail.com

网络出版地址: <http://kns.cnki.net/kcms/detail/50.1165.N.20186327.1645.002.html>

户与配送中心之间均存在一条边,该边的长度满足三角不等式关系。配送中心有多类待运输的货物和足够数辆同型号的运输车,每个客户可能对若干或全部类别的货物存在需求,某些种类的货物不允许同时运输(即相容性约束),一辆运输车每次必须至少为一个客户运输一类货物并保证完全满足客户对这类货物的需求(即某一客户的同类货物不可拆分运输),配送中心可派多辆车为每个客户运输货物,最终满足所有客户的需求。VRPCC 的目标是找到一套运输方案,在满足各类约束条件的前提下,首先最小化所使用的配送车辆的数目,然后最小化所有车辆总的行驶距离。

为了建立问题的数学模型,将每个客户抽象成多个虚拟客户,例如某个客户需求 t 种货物,那么就将该客户抽象成 t 个虚拟客户,这些虚拟客户的物理位置相同,但是每个客户只需求某一种货物。在给出问题的模型之前,首先定义参数和决策变量。

参数: V 指的是配送中心及顾客所在的节点的集合,即 $V = \{0, 1, 2, \dots, N\}$; 配送中心的编号为 0; 服务车辆的编号为 $1, 2, \dots, K$; 客户的编号为 $1, 2, \dots, N$, 所有客户用集合 V_c 表示; 货物种类的编号为 $1, 2, \dots, M$; A_{im} 表示如果客户 i 需求货物 m , 那么 $A_{im} = 1$, 否则 $A_{im} = 0$; 集合 $P = \{P_1, P_2, \dots, P_t\}$, 其中每个元素 $P_u = \{a, b\}$ ($1 \leq u \leq t$) 表示货物 a 不能与货物 b 一起运输的货物的集合; D_i 表示客户 i 货物的需求量; C 表示车的容量; d_{ij} 表示客户 i 和客户 j 之间的距离; L 为一个足够大的正数。

决策变量: x_{ijk} 表示如果车 k 从客户 i 到客户 j , 那么 $x_{ij} = 1$, 否则 $x_{ij} = 0$; y_{ik} 表示如果车辆 k 服务顾客 i , 那么 $y_{ik} = 1$, 否则 $y_{ik} = 0$; z_{mk} 表示如果车辆 k 装载货物 m , 那么 $z_{mk} = 1$, 否则 $z_{mk} = 0$ 。

当车辆的数目给定时,可以用下面的模型来求得最小的行驶总距离。

$$\min \sum_{i \in V} \sum_{j \in V} d_{ij} \sum_{k=1}^K x_{ijk}, \quad (1)$$

$$\text{s.t.} \quad \sum_{k=1}^K y_{ik} = 1, \forall i \in V \setminus \{0\}, \quad (2)$$

$$\sum_{k=1}^K y_{0k} = K, \quad (3)$$

$$\sum_{j \in V} x_{ijk} = \sum_{j \in V} x_{jik} = y_{ik}, \forall i \in V, k = 1, \dots, K, \quad (4)$$

$$\sum_{i \in V} D_i y_{ik} \leq C, \forall k = 1, \dots, K, \quad (5)$$

$$\sum_{i \in S} \sum_{j \notin S} x_{ijk} \geq y_{hk}, \forall S \subseteq V \setminus \{0\}, h \in S, k = 1, \dots, K, \quad (6)$$

$$A_{im} y_{ik} \leq L z_{mk}, \forall m = 1, \dots, M, k = 1, \dots, K, i \in V \setminus \{0\}, \quad (7)$$

$$\sum_{m \in P_u} z_{mk} \leq 1, \forall P_u \in P, k = 1, \dots, K, \quad (8)$$

$$y_{ik} \in \{0, 1\}, \forall i \in V, k = 1, \dots, K, \quad (9)$$

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

$$z_{mk} \in \{0, 1\}, \forall m = 1, \dots, M, k = 1, \dots, K, \quad (11)$$

(1)式的目标就是最小化所有车辆总的行驶路径;约束(2)保证每个客户都会被车辆服务;约束(3)规定有 K 辆车从配送中心派遣出去服务顾客;约束(4)保证当边 (i, j) 被车 k 访问时,车辆 k 必定服务顾客 i ;约束(5)保证每辆车的容量限制必须遵守;约束(6)是肖子环约束;约束(7)规定如果顾客 i 被车辆 k 服务,并且顾客 i 需求的货物的种类为 m ,那么车辆 k 必须装载有种类为 m 的货物;约束(8)为不相容性约束。

为更好适应混合蚁群算法的求解,对 VRPCC 应用模型做如下定义和优化。

定义 1 将整个物流网络抽象为一个无向赋权完全图 $G_1 = \langle V_1, E_1 \rangle$ 。节点集合 $V_1 = \{v_0, \{v_{1k} \mid D_{1k} > 0, k = 1, 2, \dots, m\}, \dots, \{v_{nk} \mid D_{nk} > 0, k = 1, 2, \dots, m\}\}$, 其中 v_0 代表配送中心,客户 i 对 j 类货物的需求抽象为节点 v_{ij} ($i = 1, 2, \dots, n; j = 1, 2, \dots, m$), D_{ij} 代表客户 i 对 j 类货物的需求量, D_{ij} 可被称为 v_{ij} ($i = 1, 2, \dots, n; j = 1, 2, \dots, m$) 的点权。各节点之间边的边权等于各节点对应的客户或配送中心之间通路的长度,若两个节点对应同样一个客户,则这两个节点的边权为 0。

定义 2 将各类货物之间的相容性约束抽象为一个无向图 $G_2 = \langle V_2, E_2 \rangle$ 。节点集合为 $V_2 = \{v_1, v_2, \dots, v_m\}$, 每类货物 i 对应一个节点 v_i ($i = 1, 2, \dots, m$), 当货物 i 与货物 j 可以同时运输时, v_i 和 v_j 之间存在一条边。

对应于蚁群算法中的概念,图 G_1 中的 v_0 和其他各节点分别代表“蚂蚁巢穴”和不同的“食物源”,在本问题中,蚁群需要从“蚂蚁巢穴”出发找到所有的“食物源”并带回各个“食物源”中的食物,最终尽量使整体成本最小。

此外,蚁群在寻找“食物源”的过程中还需要满足 VRPCC 的各项约束,其中由相容性约束可以得出如下推论。

推论 设全体货物集合为 M ,若 M 存在 k 个子集 M_1, M_2, \dots, M_k (以下简称“子集”),对于其中任意两个不同子集 M_i 和 M_j ,均有 $\forall x \in M_i, \forall y \in M_j, x$ 与 y 不能同时运输;同时对于其中任意一个子集 M_i ,均有对于 $\forall x_1 \in M_i$,一定存在至少一个 $x_2 \in M_i$ 可以与 x_1 同时运输。那么,原针对 m 类货物的 VRPCC 一定可以分解为 k 个小规模的 VRPCC,其中每一个小规模 VRPCC 对应 M 的一个子集。

由以上推论,可以进一步对原 VRPCC 进行如下优化。

优化 G_2 中的每一个连通部分视为一个子集,根据 G_2 中所有的 k 个连通部分,可将原物流网络图 G_1 重新分解为 k 个子图 $G_{11}, G_{12}, \dots, G_{1k}$,对于任意子图 $G_{1i} = \langle V_{1i}, E_{1i} \rangle$,均有 $V_{1i} = \{V_0, \{V_{1j} | j \in M_i, D_{1j} > 0\}, \dots, \{V_{nj} | j \in M_i, D_{nj} > 0\}\}$ 。

1.2 蚁群算法基本原理

Dorigo 等于 1991 年首先提出了蚁群算法^[8],并随后将该算法应用到求解旅行商问题、分配问题、Job-Shop 调度问题等组合优化问题上,取得了较好的效果。由此,诸多学者开始关注蚁群算法,并将蚁群算法应用到了其它各个研究领域中,取得了大量的研究成果。

蚁群算法的原理基于蚂蚁觅食过程中的正反馈效用:假设从蚂蚁巢穴到食物源共有两条路径 A, B,不妨假设路径 A 的长度大于路径 B,如图 1 所示。现有两只蚂蚁 X, Y 同时从巢穴出发,分别沿 A 路径和 B 路径寻找食物。由于 B 路径较 A 路径短,因而在蚂蚁行进速率一致的前提下,蚂蚁 Y 将更早的返回巢穴。随后从巢穴出发寻找食物的蚂蚁,会首先发现由蚂蚁 Y 在 B 路径上留下的信息素,并据此指引经由 B 路径出发进行觅食,并再次在 B 路径上留下信息素,这样就形成了一种正反馈的机制,最终使得蚁群整体会选择相对较优的路径。

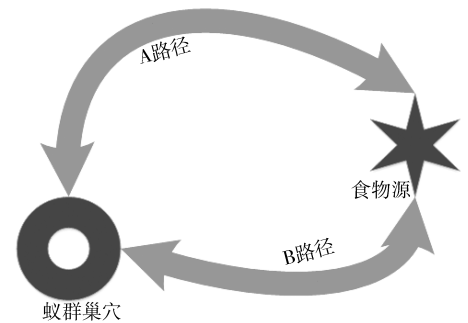


图 1 蚁群寻找巢穴和食物之间最优路径

Fig. 1 An ant colony searching an optimal path between the food and the nest

2 混合蚁群算法设计

2.1 混合蚁群算法的设计思路

蚁群算法具有很好的全局搜索能力,可以快速地搜索到某个较优解,在对解的质量要求不高时有很好的求解效率。但是传统蚁群算法的局部搜索能力较弱,搜索过程易于出现停滞或快速收敛至某个局部最优解,因此很难为大规模路径优化问题找到高质量的解。禁忌搜索算法则有着很强的局部搜索能力,善于挖掘邻域内的最优解,但在求解大规模路径优化问题时往往因为邻域规模过大而导致求解速度缓慢。

本文提出的混合算法根据两类搜索算法的特性,取长补短,将两类算法分别在求解速度和求解精度方面的优势相结合,先采用蚁群算法快速搜索到某个局部较优解,再采用禁忌搜索算法对解进行进一步地优化,从而获得更高质量的解。同时,为了更好地适应 VRPCC 的特性,本文也对蚁群算法和禁忌搜索算法进行了多方面的改进和优化。

2.2 信息素更新及蚁群状态转移策略的改进

传统蚁群算法的信息素更新及蚁群状态转移策略在求解速度、局部跳出能力等方面具有一定的缺陷,针对这些问题,很多学者提出了若干改进方案。Gambardella 等人^[9]于 1996 年第一次提出了蚁群系统(Ant colony system, ACS),随后诸多学者又在此基础上发展出了精英蚁群系统(Elitist ant system, EAS),有效帮助解决了基本蚁群算法求解大规模问题时收敛速度慢、不容易产生优化解的问题;德国学者 Thomas Stutzle^[10]提出了最大最小蚁群系统(Max-Min ant system, MMAS),该策略有效地防止了蚁群算法由于信息素浓度分布差异过大而导致算法停滞的问题。

本文综合考虑了各种策略的优缺点,提出了一种改进的混合策略,以满足快速搜索局部较优解的目标。具体对策略的改进包括:

1) 根据当前解相对于算法执行以来最优解的质量,选择不同权值的信息素增量对当前解对应路径上的信息素浓度进行更新。改进后的信息素更新规则为 $\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \lambda(t) \cdot \Delta\tau$ 。其中 $\Delta\tau$ 为常数, $\lambda(t) =$

$$\begin{cases} \theta_1, W(t) < W_{\text{best}}, K(t) < K_{\text{best}} \\ \theta_2, W(t) < W_{\text{best}}, K(t) = K_{\text{best}} \\ \theta_3, W(t) > W_{\text{best}}, K(t) = K_{\text{best}} \\ \theta_4, W(t) > W_{\text{best}}, K(t) > K_{\text{best}} \end{cases}, \theta_1 > \theta_2 > \theta_3 > \theta_4, W(t) \text{ 和 } K(t) \text{ 分别为第 } t \text{ 次迭代中最优解的总距离和总车次}$$

次, W_{best} 和 K_{best} 分别为当前最优解的总距离和总车次。

2) 以概率选择的方式决定蚂蚁转移到哪个目标点。具体方式是在蚂蚁每一次转移时,随机抽取一个可行的目标点(即满足相容性约束和容量约束的目标点),根据改进的概率转移公式按概率判定蚂蚁是否转移到该点,如此循环直至不存在任意一个可行的目标点或者全部可行的目标点均被抽取过。

改进后的蚁群状态转移公式为 $P_{ij} = \min((\alpha\eta_{ij}^1)^{\beta\eta_{ij}^2} \times \log_{\gamma}(\tau_{ij} + 1), 1)$, 其中 $\eta_{ij}^1 = \frac{D_j}{\max\{D_k | k \in A_{\text{allowed}}\}}$, $\eta_{ij}^2 = \frac{L_{ij}}{\max\{L_{ik} | k \in A_{\text{allowed}}\}}$; α 为 η_{ij}^1 的调节因子,反映需求启发信息对于蚁群路径选择的影响; β 为 η_{ij}^2 的调节因子,反映距离启发信息对于蚁群路径选择的影响; γ 为 \log 函数的调节因子,反映信息素浓度对于蚁群路径选择的影响; A_{allowed} 表示所有满足要求的集合。

3) 对各路径上的信息素浓度设置最大最小限制。改进后的状态转移策略使算法具备了更快的收敛速度,而改进的信息素更新规则和最大最小限制则可以有效防止算法的求解陷入停滞,配合适当的调节因子,改进后的蚁群算法整体具有快速求解能力。

2.3 扰动机制

在保证求解速度的基础上,为了保证改进的蚁群算法在有限迭代次数内具有更强的全局搜索能力,同时引入了一个扰动机制。具体规则是当蚂蚁在前一次转移过程中选择了信息素浓度很低的路径时,随后一次转移采用等概率随机的方法选择蚁群转移的目标节点。通过这样的方式可以有效提高算法的全局搜索能力,防止算法过于快速地收敛于局部最优解。

2.4 禁忌搜索规则

1) 初始解。以上文所述的改进蚁群算法为禁忌搜索算法提供初始解。

2) 邻域结构。采用以下 3 种操作确定邻域。

① 插入操作:在保证新生成解可行(即不违反相容性约束和容量约束)的前提下,将某个点从原路径中移出并插入到另外一条路径上,如图 2 所示。

② 删除操作:对于除了只包含一个客户的路径外,将位于某条路径上的某个点删除,重新建立一条只包含该点的新路径,如图 3 所示。

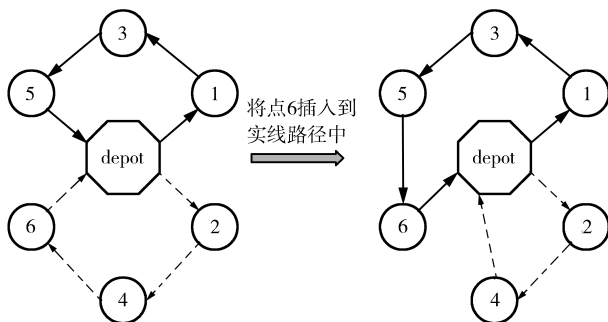


图 2 插入操作

Fig. 2 Relocate operator

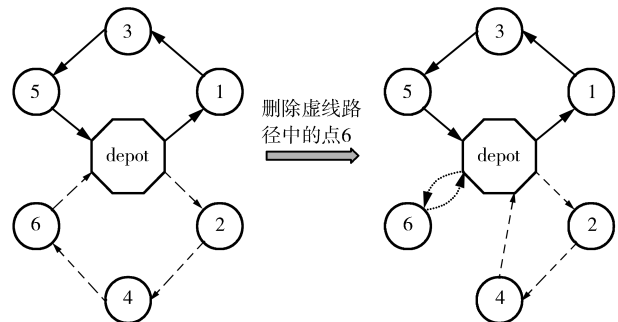


图 3 删除操作

Fig. 3 Delete operator

③ 2-opt 操作:选择位于同一条路径上的两个点,调转这两点之间路径上点的顺序,如图 4 所示。

3) 禁忌表。设置禁忌表的目的是为了防止“震荡”现象的发生,即防止算法重复搜索到相同解。但由于 VR-PCC 解的结构较为复杂,记录全部解的变化过程并不现实,所以本文采用了一种简单的禁忌规则,目的是以最小的

成本最大概率避免搜索到相同解,具体规则为:当某个点被从当前所在路径移除(具体方式包括删除操作以及插入操作中的移出)后,对该路径设置如下禁忌:在此后若干步迭代内,不允许被移除的点再次被插入到当前路径中。

这样的规则可以使禁忌表具有较低的空间复杂度,且将每次对于禁忌表进行查询的时间复杂度降至 $O(1)$ 级别。考虑到插入操作相较于删除操作选择到同一点的概率要低很多,进而一条路径被执行若干次删除及插入操作后在另外的位置生成完全相同的路径的概率是非常低的,于是可以说明这样的禁忌规则是有效的。

4) 解的接收准则。若新生成的解 S^* 优于当前解 S , 则用 S^* 替代当前解进入下次迭代, 否则继续搜索 S 的邻域。若已搜索到的解的数量达到邻域规模上限, 但仍未找到优于 S 的 S^* , 则在所有已找出的 S^* 中选择最好的一个替代当前解进入下次迭代。

5) 停止准则。当禁忌搜索迭代次数达到上限时, 算法停止。

2.5 算法步骤

混合蚁群算法求解 VRPCC 的流程如图 5 所示。

3 算例测试及结果分析

为验证混合蚁群算法的有效性, 采用标准算例测试库 Augerat et al.。Set A 中部分算例的位置信息, 结合随机生成的需求信息和相容性约束信息构造的若干个新算例, 对算法在求解质量、求解速度和求解稳定性方面的表现进行了全面地测试。

本研究生成了 4 组 12 个 VRPCC 算例对算法进行了测试, 各组测试算例的客户及配送中心位置信息分别取自 VRP 标准算例测试库 Augerat et al. Set A (下载网址为 [http://neo.lcc.uma.es/vrp/vrp-instances/](http://neo.lcc.uma.es/vrp/vrp-instances/capacitated-vrp-instances/)) 中的 A-n32-k5, A-n45-k7, A-n53-k7 和 A-n80-k10, 客户需求和相容性约束信息随机生成。测试算例的命名规则为:“VRP-客户及配送中心数量和-货物种类-车辆容量-相容性约束关系图 G_2 的疏密程度”, 其中相容性约束关系图 G_2 的疏密程度为 dense 和 sparse 分别表示 G_2 的边的数量为与之对应的完全图边数的 60% 和 20%。本文使用的算例可以从网页 <http://www.msor-world.com/> 获取。

算法的参数设置如下: 蚁群搜索最大迭代次数为 300, 蚂蚁数量 $M=10$, 信息素挥发速度 $\rho=0.01$, 信息素增量权重 $\theta_1=2, \theta_2=1.2, \theta_3=0.8, \theta_4=0.4$, 信息素增量 $\Delta\tau=1$, 最大信息素浓度 $\tau_{max}=15$, 最小信息素浓度 $\tau_{min}=0.01$, 需求影响调节因子 $\alpha=0.98$, 距离影响调节因子 $\beta=1$, 信息素浓度影响调节因子 $\gamma=e$, 禁忌搜索最大迭代次数为 100, 邻域规模 $N=30$, 禁忌长度为 10。计算实验

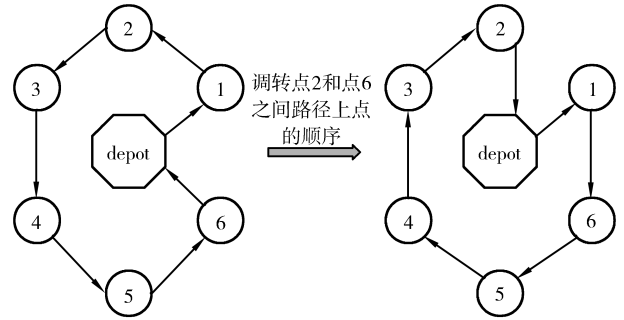


图 4 2-opt 操作

Fig. 4 2-opt operator

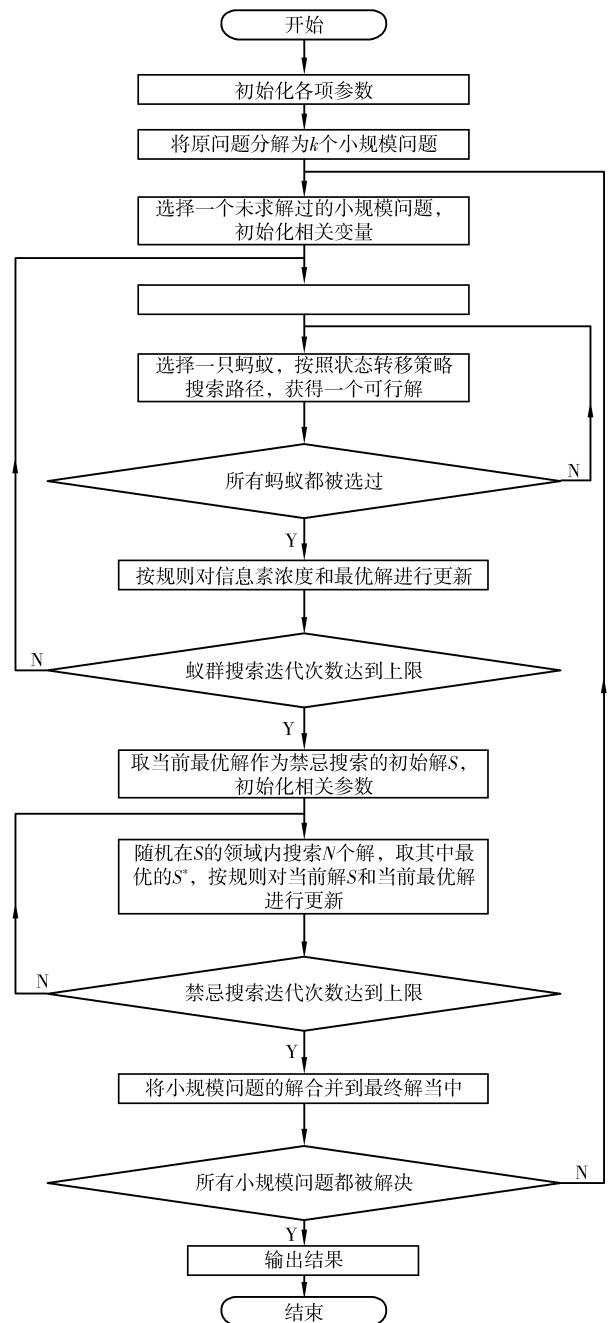


图 5 混合蚁群算法求解 VRPCC 流程

Fig.5 Process of solving VRPCC by the hybrid ant colony algorithm

环境为:DEV C++,CPU 为 Intel(R) Core(TM) i3 M 370,内存为 4G。

运行混合蚁群算法求解每个算例 5 次,计算结果如表 1、图 6-7 所示。在表 1 中,算例规模指将算例对应的问题抽象为无向赋权图 G_1 后图中节点的数量,变异系数的值由标准差除以均值获得。使用车辆数目小的解始终优于车辆数目多的解,如果车辆数目一样,则总行驶距离小的解较优。

表 1 混合蚁群算法对 VRPCC 的求解表现

Tab. 1 Results of VRPCC instances produced by the hybrid ant colony algorithm

组别	算例	算例规模	算法求解结果							
			平均解		最好解		平均用时/ms	标准差		总距离变异系数
			车辆数	总距离/km	车辆数	总距离/km		车辆数	总距离/km	
1	VRP-32-10-100-dense	311	62	16 727	62	16 429	87 152	0	365	2.18%
	VRP-32-10-100-sparse	37	69	17 083	69	16 543	63 629	0	524	3.07%
	VRP-45-10-100-dense	441	91	23 311	91	22 936	218 046	0	400	1.72%
2	VRP-45-10-100-sparse	44	89.6	21 856	88	219 86	125 311	0.89	162	0.74%
	VRP-45-20-100-dense	881	174.8	48 060	174	49 299	765 714	0.45	905	1.88%
	VRP-45-20-100-sparse	64	184.4	48 328	184	47 305	498 009	0.55	757	1.57%
3	VRP-53-10-100-dense	521	109	26 857	109	26 527	272 469	0	398	1.48%
	VRP-53-10-100-sparse		111.8	25 792	111	25 752	161 389	0.45	295	1.14%
	VRP-53-20-100-dense	1 041	215.2	56 446	215	55 951	1 131 513	0.45	843	1.49%
	VRP-53-20-100-sparse		215.4	56 081	215	55 583	738 425	0.55	741	1.32%
4	VRP-80-10-100-dense	791	156	48 112	156	47 226	573 264	0	630	1.31%
	VRP-80-10-100-sparse		162.4	45 515	162	44 511	315 876	0.55	905	1.99%

表 1 和图 7 的反映出本算法具有良好的稳定性,同时图 6 又显示本算法对不同规模算例的求解时间呈近似线性的变化趋势。结合本算法的终止机制考虑,可以看出本算法对于解的优化程度同算法的迭代次数具有一定的相关性。

为找到更佳参数配置、更好地分析本算法的求解能力,本文选取 VRP-45-20-100-sparse 算例,采用多组不同的参数组合进行了计算,结果如表 2、图 8-10 所示。计算结果显示,迭代次数以及邻域规模大小对于混合蚁群算法的求解质量和求解时间具有显著的影响。

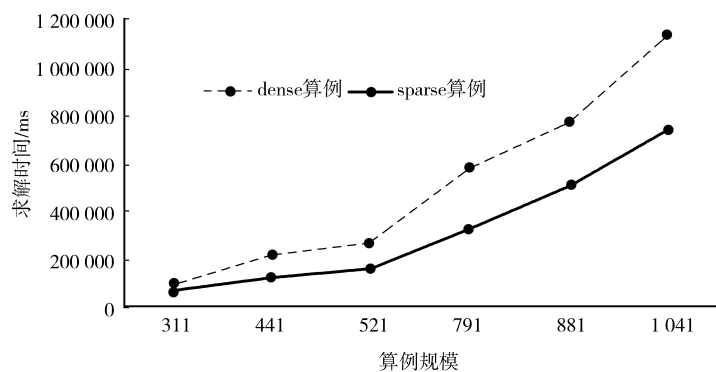


图 6 混合蚁群算法对 VRPCC 算例的计算时间
Fig. 6 Computation times for solving VRPCC instances by the hybrid ant colony algorithm

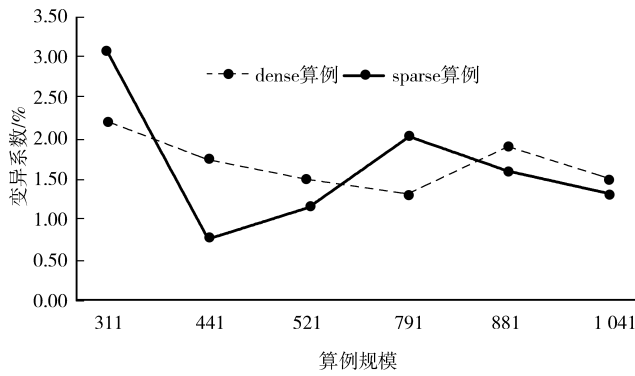


图 7 混合蚁群算法对 VRPCC 算例求解结果中总距离的变异系数

Fig. 7 Coefficient of variation of the total distances calculated by the hybrid ant colony algorithm for VRPCC instances

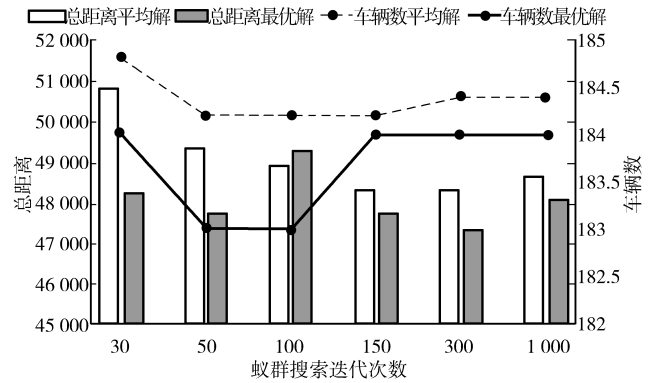


图 8 蚁群搜索迭代次数对于算法求解质量的影响

Fig. 8 The influence of the number of ant colony search iterations on the final results

图 8 反映出在其他条件相同的情况下,增加蚁群搜索迭代次数对于求解质量的影响是先利后弊,在迭代次数较小时,迭代次数的增加将会改善解的质量,而当迭代次数较大时继续增加迭代次数将会导致结果恶化。经过分析,出现这一结果的原因是蚁群搜索迭代次数会对算法的收敛速度产生影响,而算法收敛速度过快或过慢都不利于改善最终的求解质量。具体在蚁群搜索迭代次数适当的情况下,蚁群搜索可以为禁忌搜索提供适当的初始解,优化禁忌搜索的候选解质量,提高算法的整体求解效率。而当蚁群搜索迭代次数过多时,蚁群搜索将会收敛向某个局部,这反而增加了算法禁忌搜索部分跳出局部、进一步对解进行优化的难度,最终使得算法的求解质量下降。

表 2 不同迭代次数组合下混合蚁群算法对算例 VRP-45-20-100-sparse 的求解表现

Tab.2 Results of instance VRP-45-20-100-sparse produced by the hybrid ant colony algorithm with different iterations

算法参数			求解结果						
蚁群迭代次数	禁忌搜索		平均解		最好解		平均用时 /ms	标准差	
	迭代次数	邻域规模	车辆数	总距离/km	车辆数	总距离/km		车辆数	总距离/km
1 000	100	30	184.4	48 644	184	48 080	969 770	0.55	483
300	100	30	184.4	48 328	184	47 305	498 009	0.55	757
150	100	30	184.2	48 300	184	47 735	300 743	0.45	1 174
100	100	30	184.2	48 909	183	49 292	185 370	0.84	1 033
50	100	30	184.2	49 312	183	47 727	81 522	0.84	1 039
30	100	30	184.8	50 837	184	48 209	48 274	0.45	2 556
50	100	60	185	48 072	185	47 015	107 896	0	1 078
50	100	15	184	49 509	183	48 831	76 670	0.71	534
50	500	30	184	47 421	183	46 883	142 414	0.71	623
50	500	15	184	48 101	183	48 163	101 226	0.71	338
50	1 000	15	184.4	46 835	184	46 043	126 087	0.55	780
50	1 000	30	184	46 130	183*	46 126*	187 792	1	396
50	2 000	15	184.25	45 623	184	45 289	181 500	0.5	361

注: * 表示该算例的最好解。

图 9 中的数据同样表明,单独扩大影响算法局部搜索能力的邻域规模也可能会对算法的求解质量产生不利的影 响,但结合表 2 中的其他数据,发现禁忌搜索的邻域规模和迭代次数对于最终解质量的影响是交互的,在迭代次数较多的情况下邻域规模越大越好,而在迭代次数较少的情况下则恰恰相反。本研究认为,这是因为当迭代次数较少时,受益于小邻域规模禁忌搜索对于劣解更好的包容性,算法更容易在更大的范围内找到优于某个小局部的较优解,而当迭代次数较多时,较大邻域规模的禁忌搜索则可以覆盖到很大的局部,选择到更优解的概率则要更大。

图 10 中的数据表明,增加禁忌搜索迭代次数可以持续改善最终解的质量,而从表 2 的数据中又可以看出,同比例提高禁忌搜索迭代次数对于求解时间的影响要远小于另外两者。进而可以判断,增加禁忌搜索迭代次数可以最有效的提高算法的求解质量。

通过以上分析可以看出,VRPCC 对于算法的全局搜索能力和局部搜索能力均有着较高的要求。混合蚁群算法结合了蚁群算法和禁忌搜索算法各自的优势,具有灵活调整算法局部搜索和全局搜索深度的能力,很好的迎合了 VRPCC 的特点,可以有效地用于解决 VRPCC。

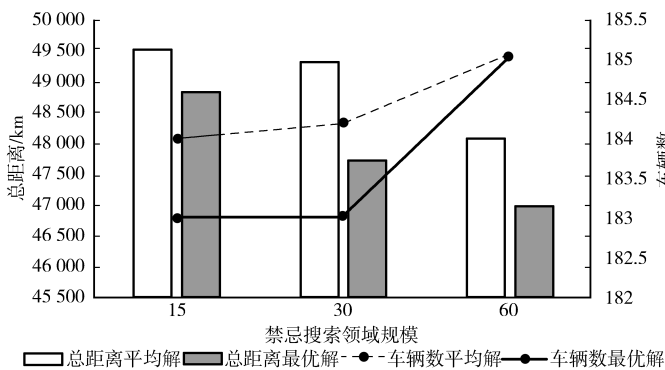


图 9 禁忌搜索邻域规模对于算法求解质量的影响

Fig. 9 The influence of the scale of tabu search neighborhood on the final results

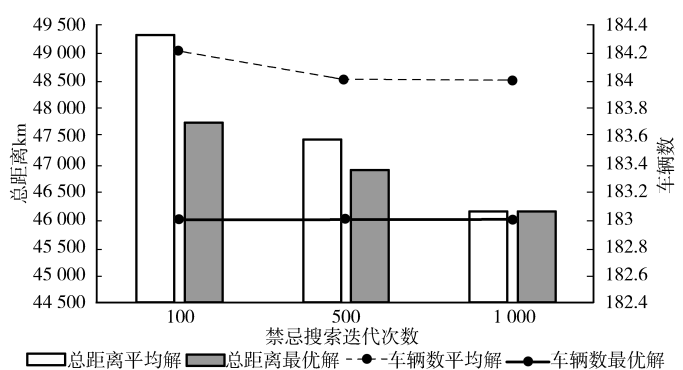


图 10 禁忌搜索迭代次数对于算法求解质量的影响

Fig. 10 The influence of the number of tabu search iterations on the final results

4 结束语

本文针对 VRPCC,结合蚁群算法和禁忌搜索算法两者的优势,提出了一种混合蚁群算法。该算法在蚁群搜索部分采用了一种改进的状态转移公式,使蚁群可以综合考虑距离、需求、载重和信息素浓度因素,更有效地决定转移目标;结合精英蚁群系统和最大最小蚁群系统,改进了一种信息素集权更新策略,可以有效防止算法过快收敛;加入了扰动机制,增强了算法的全局搜索能力。该算法同时在禁忌搜索部分对邻域结构进行了改进,通过插入、删除和 2-opt 这 3 种操作,配合随机选择的方式确定禁忌搜索的候选解,有效的提高了禁忌搜索算法的效率;改进了禁忌规则,仅对重复执行的插入操作进行限制,减少了实行禁忌的空间和时间复杂度,同时仍可以有效防止“震荡”的发生。通过测试证明,该算法具有良好的求解能力,在求解速度、质量和稳定性等方面均有着较好的表现,可以很好地适用于 VRPCC 的求解。

参考文献:

- [1] LAPORTE G. The vehicle routing problem: An overview of exact and approximate algorithms[J]. European Journal of Operational Research, 1992, 59(3): 345-358.
- [2] DORIGO M, Maniezzo V, Colorni A. Ant system: optimization by a colony of cooperating agents[J]. IEEE Transactions on Systems, Man, and Cybernetics, Part B(Cybernetics), 1996, 26(1): 29-41.
- [3] 刘志硕,申金升,柴跃廷.基于自适应蚁群算法的车辆路径问题研究[J].控制与决策,2005,20(5):562-566.
- [4] LIU Z S, SHEN J S, CHAI Y T. Vehicle routing problem based on an adaptive ant colony algorithm[J]. Control and Decision, 2005, 20(5): 562-566.
- [5] GENDREAU M, LAPORTE G. A tabu search heuristic for the vehicle routing problem[J]. Management Science, 1991, 40(10): 1276-1290.
- [6] GENDREAU M, LAPORTE G, SÉGUIN R. A Tabu search heuristic for the vehicle routing problem with stochastic demands and customers[J]. Operations Research, 1996, 44

- (3):469-477.
- [6] XU J F, KELLY J P. A network flow-based tabu search heuristic for the vehicle routing problem[J]. *Transportation Science*, 1998, 30(4):379-393.
- [7] BARBAROSOGLU G, OZGUR D. A tabu search algorithm for the vehicle routing problem[J]. *Computers & Operations Research*, 1999, 26(3):255-270.
- [8] 郎茂祥, 胡思继. 车辆路径问题的禁忌搜索算法研究[J]. *管理工程学报*, 2004, 18(1):81-84.
LANG M X, HU S J. Study on the Tabu search algorithm for vehicle routing problem[J]. *Journal of Industrial Engineering/Engineering Management*, 2004, 18(1):81-84.
- [9] COLORNI A, DORIGO M, MANIEZZO V. Distributed optimization by ant colonies[C]//*European Conference on Artificial Life*. Paris: Elsevier Publishing, 1991:134-142.
- [10] DORIGO M, GAMBARDELLA L M. Ant colony system: a cooperative learning approach to the traveling salesman problem[J]. *IEEE Transactions on Evolutionary Computation*, 1997, 1(1):53-66.
- [11] STÜTZLE T, HOOS H. MAX-MIN ant system and local search for the traveling salesman problem[C]//*IEEE International Conference on Evolutionary Computation*, 13-16 April, 1997, Indianapolis:309-314.

Operations Research and Cybernetics

A Hybrid Ant Colony Algorithm for the Vehicle Routing Problem with Compatibility Constraints

YU Liang¹, QIN Hu²

(1. School of Economics, Wuhan University of Technology, Wuhan 430070;

2. School of Management, Huazhong University of Science and Technology, Wuhan 430074, China)

Abstract: [Purposes]The vehicle routing problem with compatibility constraints (VRPCC) that appears in the cold-chain logistics is a type of complicated vehicle routing problem. So far, research on this problem is scarce. [Methods]In this study, the VRPCC is described in detail and then a hybrid ant colony algorithm is proposed for it according to its characteristics. The main components of our algorithm are ant colony search and tabu search. In the ant colony search, new state transition probability formula and pheromone updating rule are applied and a perturbation phase is added. [Findings]Moreover, a new neighborhood structure and tabu rules are designed for the tabu search. [Conclusions]The algorithm is evaluated based on several test instances, and the results show that our hybrid ant colony algorithm is effective and efficient.

Keywords: vehicle routing problem; cold-chain logistics; ant colony algorithm; tabu search; compatibility constrain

(责任编辑 黄颖)