

# 越库转运问题的自适应遗传算法研究<sup>①</sup>

缪朝炜<sup>1</sup>, 杨 凤<sup>1</sup>, 徐东升<sup>2</sup>, 石 宁<sup>2</sup>

(1. 厦门大学管理学院, 厦门 361005; 2. 中山大学管理学院, 广州 510275)

**摘要:** 探讨一种固定运输模式下的越库转运问题——采用运输量不可拆分的单次运送方式以最小费用通过选择固定的运输路径将货物经过越库转运到目的地, 其货物将可能在越库中停留甚至无法运到目的地, 这将会导致库存成本和惩罚成本. 文中证明了此类越库转运问题是强 NP 难题, 因此本文针对该问题的特殊结构, 提出一种采用了邻域搜索技术的自适应遗传算法 (AGA with NS) 来有效的解决该类问题, 数值试验结果表明该算法比 CPLEX 求解更加高效. 此外文中还分别比较了在不采用邻域搜索或者自适应策略的情况下的三种遗传算法, 其数值实验结果表明邻域搜索策略以及自适应策略对提高算法的效率有显著的影响.

**关键词:** 遗传算法; 转运; 越库; 供应链

**中图分类号:** F224.3; F253.4 **文献标识码:** A **文章编号:** 1007-9807(2011)06-0024-11

## 0 引言

随着经济全球化不断加深, 越来越多的公司将时间视为影响企业供应链管理的关键要素, 时间不仅影响企业的资金周转速度, 还影响企业的顾客服务水平, 因此为了降低安全库存、提高库存周转率, 各企业都在寻求适合自身发展的供应链管理策略, 如: 准时制策略、库存采购管理策略以及物流配送网络优化策略等. 越库 (crossdocking) 策略集库存管理策略与配送策略为一体, 越来越被各类企业用于优化配送网络的准时制物流策略.

越库至今还没有一个完整的定义, 从国内外文献<sup>[1-3]</sup>可以看出在越库物流系统中, 配送中心充当库存的协调点而不是存储点. 越库作业的特点在于消除仓库中的存储与订单拣选功能, 只实现其进货及出货功能, 货物直接从入货去转移到出货车上, 货物在越库中心的停留时间不超过 24

小时, 有时甚至不到 1 小时. 越库技术被广泛地应用在易变质或时间敏感性货物行业、物流配送行业, 如沃尔玛、Home Depot, Costco, Canadian Tire 以及 FedEx 等企业. Wal-Mart<sup>[4]</sup> 成功地完成由大量仓库、越库以及零售终端组成的越库运输网络上的 2 000 专车调度.

针对越库技术的研究主要集中在越库配送中心的布局及货物整合方面: Axsäter<sup>[5-6]</sup> 集中研究设施选址、越库策略, 即如何通过处理货物达到快速转运的目标. Bartholdi 和 Gue<sup>[7]</sup> 根据运用随机过程的方法来分析越库操作, 以达到减少人工成本的目标. Miao 等<sup>[8]</sup> 研究了越库内车辆调度问题, 在证明该类问题为强 NP 难题的基础上设计出高效的解决该类问题的禁忌算法和遗传算法. 俞亮等<sup>[9]</sup> 研究了单个越库内的两阶段越库操作问题, 并提出了基于最早交货期 (EDD) 的启发式算法. 衣方磊等<sup>[10]</sup> 提出了动态配送车辆调度优化问题, 利用局内问题及竞争分析理论对不同的管

① 收稿日期: 2010-10-07; 修订日期: 2011-04-21.

基金项目: 国家自然科学基金资助项目 (70802052; 71001109); 福建省高校杰出青年科研人才计划资助项目 (JA10001S); 中央高校基本科研业务费资金资助项目 (2010221025); 教育部“新世纪优秀人才支持计划”资助项目 (NECT).

作者简介: 缪朝炜 (1980—), 男, 福建人, 博士, 教授; 通讯作者: 石宁, Email: shiningchina08@gmail.com.

理策略进行对比来得到最佳策略.姚建明等<sup>[11]</sup>针对物流供应链调度的基本运作特征,设计出有效的蚁群算法来进行寻优,并通过实验结果验证了算法的可行性.

在越库调度模型研究方面,Chen和Lee<sup>[12]</sup>研究了一类基于排程问题的越库调度模型,证明该问题为强NP-难题特征的基础上给出了求解该类问题的近似算法和分支定界算法.马东彦<sup>[13]</sup>则在Chen和Lee<sup>[12]</sup>的模型基础上提出求解该类问题的动态规划算法用于解决小规模问题.陈峰等人<sup>[14]</sup>针对这类问题提出解决大规模问题的近似启发式算法的精确度分析和小规模分支定界算法.王长军等<sup>[15]</sup>总结了此类调度问题所具有的各任务的自利性产生的无序竞争导致系统全局目标恶化的无秩序特征,分析了不同情况下Nash均衡调度的无秩序代价.宋莉波等<sup>[16]</sup>提出了一种基于混合遗传算法的求解方案来解决柔性车间调度问题,其算法设计思想与本文有相似点.Lim等<sup>[17]</sup>研究了制造商和顾客带有时间约束的越库运输网络,并证明不同运输模式下该问题的算法复杂度,但是没有给出问题的具体数学模型以及求解方法.此外,Chen等<sup>[18]</sup>研究了带有时间窗口和容量限制的越库网络运输问题,由于无法确定具体的运输时间点,该问题采用了离散化时间窗口的方法,将连续性的问题简化为离散问题来解决,使得新问题的解空间只是原问题解空间的一个子集,因此新模型是原模型的一种近似形式,同时采用禁忌算法和模拟退火算法求解该近似问题的近似最优解,会影响到近似最优解的质量.Ma等<sup>[19]</sup>在Chen等<sup>[18]</sup>的研究基础上,对模型进行了扩展,考虑了运输车辆的数量以及运输的固定成本,并采用两阶段启发式算法来进行求解,其中迭代寻优过程采用了遗传算法.遗传算法是基于自然选择的思想,通过不断地更新换代,来逐步实现解的质量的提高.该方法目前被广泛应用在求解优化的问题中,尤其对于结构复杂的组合优化问题有很好的效果,因此本文采用遗传算法来进行优化求解.本文研究的问题考虑了实际运作中常见的情况作为建模依据,建立更加切合实际的固定运输模式下的越库转运模型,并且设计的采用邻域

搜索策略的自适应遗传算法可在解空间中搜寻高质量的近似最优解,因此避免了采用近似模型而使得求解质量下降的问题.

## 1 固定运输模式的越库转运问题

### 1.1 问题描述

以往的研究主要集中在自由运输方式的越库运输问题上,即运输时刻是自行决定的.这在现实中比较符合卡车运输的模式,但是在现在大力提倡的多式联运运输网络中,除了卡车运输外,还有诸如火车运输和航空运输之类的运输方式,后者有一个很大的特点就是运输班次是固定的,即发车或起飞时间以及到达时刻是固定的,本文中称之为固定运输模式.在这样的固定运输模式下,该如何求得此类越库转运问题的最优解就具有很高的实际应用价值,因此该问题的研究具有重要意义.

本文探讨了一种固定运输模式下的越库转运优化问题,该越库运输网络由大量供应商、越库和顾客组成的一个三层级网络系统,供应商与越库以及越库与顾客之间采用固定运输模式,当运输调度不能满足顾客需求时,货物需要通过较高运输成本的外部渠道运输,在模型中把这种较高外部渠道运输成本定义为库存惩罚成本.此外,该运输模型采用了运输量不可拆分的单次运送方式目的是为了减少多次少量运输带来的高固定准备成本.模型的目标是优化三级节点之间的运输调度,使得总成本最低,包括运输成本、库存成本以及惩罚成本.

### 1.2 数学模型

本节中,将把上述的越库转运问题表述为混合整数线性规划模型.首先将引进模型的参数符号说明:

$(n, m, l)$ : 分别表示供应商,顾客以及越库数量;

$s_i$ : 表示供应商  $i$  可提供的货物量;

$d_k$ : 表示顾客  $k$  的需求量;

$S_{ij}$ : 供应商  $i$  与越库  $j$  之间的固定运输班次集合,其中  $|S_{ij}| = \gamma_{ij}$ ;

$S'_{jk}$ : 越库  $j$  与顾客  $k$  之间固定运输班次集合,

其中  $|S'_{jk}| = \gamma'_{jk}$ ;

$(b'_{ijq}, e'_{ijq})$ : 供应商  $i$  与越库  $j$  第  $q$  条固定运输班次, 其中  $b'_{ijq}$  表示始发时间  $e'_{ijq}$  表示到达时间;

$(b^a_{jkq}, e^a_{jkq})$ : 越库  $j$  与顾客  $k$  第  $q$  条固定运输班次, 其中  $b^a_{jkq}$  表示始发时间  $e^a_{jkq}$  表示到达时间;

$c_{ijq}$ : 供应商  $i$  与越库  $j$  第  $q$  条固定运输班次的单位运输成本;

$c'_{jkq}$ : 越库  $j$  与顾客  $k$  第  $q$  条固定运输班次的单位运输成本;

$P_j$ : 越库  $j$  在最后时刻单位库存惩罚成本;

$h_j$ : 越库  $j$  的单位时间单位库存成本;

$T_j$ : 供应商到越库  $j$  所有固定运输班次的到达时刻集合, 其中  $T_j = \{e'_{ijq} : 1 \leq i \leq n, 1 \leq q \leq \gamma_{ij}\}$ ;

$T'_j$ : 越库  $j$  到顾客的所有固定运输班次的始发时刻集合,  $T'_j = \{b^a_{jkq} : 1 \leq k \leq m, 1 \leq q \leq \gamma'_{jk}\}$ ;

$\hat{T}_j$ : 越库  $j$  内所有库存可能发生变化的时刻点集合  $\hat{T}_j = T_j \cup T'_j$ , 令  $|\hat{T}_j| = \tau_j$ ,  $\hat{T}_j$  内所有的元素  $\tau_j$  消除重复时刻点并按升序排列后表示为,  $t_{jg} (g = 1, 2, \dots, \tau_j), t_{j1} \leq t_{j2} \leq \dots \leq t_{j\tau_j}$ ;

决策变量如下:

$$x_{ijq} = \begin{cases} 1, & \text{供应商 } i \text{ 通过第 } q \text{ 条班次将货运到越库 } j \\ 0, & \text{否则} \end{cases}$$

$$x'_{jkq} = \begin{cases} 1, & \text{顾客 } k \text{ 通过第 } q \text{ 条班次从越库 } j \text{ 收到货} \\ 0, & \text{否则} \end{cases}$$

$y_{jt_{jg}}$ : 整数变量, 表示越库  $j$  在时刻点  $t_{jg}$  的库存量, 其中  $t_{jg} \in \hat{T}_j$ .

目标函数:

$$\text{Min } COST_{\text{transportation}} + COST_{\text{penalty}} + COST_{\text{inventory}} \quad (1)$$

其中

$$COST_{\text{transportation}} = \sum_{i=1}^n \sum_{j=1}^l \sum_{q=1}^{\gamma_{ij}} c_{ijq} s_i x_{ijq} + \sum_{k=1}^m \sum_{j=1}^l \sum_{q=1}^{\gamma'_{jk}} c'_{jkq} d_k x'_{jkq} \quad (2)$$

$$COST_{\text{penalty}} = \sum_{j=1}^l P_j y_{jt_{j\tau_j}} \quad (3)$$

$$COST_{\text{inventory}} = \sum_{j=1}^l \sum_{g=1}^{\tau_j} h_j (t_{jg} - t_{jg-1}) y_{jt_{jg-1}} \quad (4)$$

s. t.

$$\sum_{j=1}^l \sum_{q=1}^{\gamma_{ij}} x_{ijq} = 1 \quad (1 \leq i \leq n) \quad (5)$$

$$\sum_{j=1}^l \sum_{q=1}^{\gamma'_{jk}} x'_{jkq} \leq 1 \quad (1 \leq k \leq m) \quad (6)$$

$$y_{jt_{j0}} = 0 \quad (1 \leq j \leq l, t_{j0} = 0) \quad (7)$$

$$y_{jt_{jg}} = y_{jt_{jg-1}} + \sum_{i=1}^n \sum_{\{q: e'_{ijq}=t_{jg}\}} s_i x_{ijq} - \sum_{k=1}^m \sum_{\{q: b^a_{jkq}=t_{jg}\}} d_k x'_{jkq} \quad (1 \leq j \leq l, 1 \leq g \leq \tau_j) \quad (8)$$

$$x_{ijq} \in \{0, 1\} \quad (1 \leq i \leq n, 1 \leq j \leq l, 1 \leq q \leq \gamma_{ij}) \quad (9)$$

$$x'_{jkq} \in \{0, 1\} \quad (1 \leq k \leq m, 1 \leq j \leq l, 1 \leq q \leq \gamma'_{jk}) \quad (10)$$

$$y_{jt_{jg}} \in I^+ \cup 0 \quad (1 \leq j \leq l, 1 \leq g \leq \tau_j) \quad (11)$$

目标函数是最小化总成本, 包括运输成本、库存持有成本以及惩罚成本. 约束条件(5) 和(6) 表示该问题采用了运输量不可拆分的单次运输方式, 即约束条件(5) 要求供应商必须通过一次运输将货物全部运出, 而约束条件(6) 要求顾客至多能通过一次运输接受货物; 约束条件(7) 表明越库初始库存为零, 越库内每个时刻库存变化由约束(8) 给出, 约束(9) 到(11) 是限制了决策变量的取值区域. 对于该问题有以下命题.

命题 1 此类越库转运问题是强 NP 难题.

证明 见附录.

由命题 1 可知此类问题不存在多项式算法, 下面将详细介绍求解该问题的采用邻域搜索策略的自适应遗传算法的设计.

## 2 算法设计

在上述问题中, 供应商的运输策略起重要的

作用, 供应商的越库以及路径选择不仅影响了解的可行性, 而且影响顾客的运输路径的选择, 因此, 在该遗传算法的初始种群生成部分, 将会分成三个阶段并在每个阶段采用不同的启发式策略来产生初始种群, 具体策略将在 2.1.2 中介绍。随着问题的规模增大, 单一的交叉算子与变异算子对遗传算法的寻解速度影响较大, 基于这点, 文中将设计两种交叉算子, 并在整个迭代过程中采用自适应策略选择最佳的交叉算子, 变异算子则采用基因变异概率和基因位变异概率组合对子代进行变异运算; 遗传算法是一种全局优化算法, 缺点在于局部搜索能力和迭代效率较低, 针对这一缺陷, 本文采用了邻域搜索策略来改善解的质量。下面将具体介绍该遗传算法。

### 2.1 自适应遗传算法的基本要素设计

#### 2.1.1 基因表达

在该问题中, 基因由两部分组成, 第一部分是越库分配策略, 越库分配策略表示为  $\chi_1, \dots, \chi_n, \chi_{n+1}, \dots, \chi_{n+m}$ , 其中  $n$  和  $m$  分别表示供应商和顾客的数量,  $\chi_i \in \{1, 2, \dots, l\}$  ( $l$  为越库的标号) 表示供应商  $i$  ( $i = 1, 2, \dots, n$ ) 或者顾客  $j$  ( $i = n+1, n+2, \dots, n+m; j = i-n$ ) 分配到越库  $\chi_i$ ; 第 2 部分为路径分配策略, 表示为  $\varphi_1, \dots, \varphi_n, \varphi_{n+1}, \dots, \varphi_{n+m}$ , 其中对于供应商  $i$  ( $i = 1, 2, \dots, n$ ) 而言,  $\varphi_i \in \{1, \dots, \gamma_{i, \chi_i}\}$ , 表示供应商  $i$  选择运输班次  $\varphi_i$  运输, 顾客部分的表达意思与供应商相似。这个染色体可以表示为

$$\text{Chromosome} = \underbrace{\{\chi_1, \chi_2, \dots, \chi_{n+m}\}}_{\text{crossdocks}} \underbrace{\{\varphi_1, \varphi_2, \dots, \varphi_{n+m}\}}_{\text{routes}}$$

#### 2.1.2 初始种群

在目标函数中, 总成本包括由供应商以及顾客的运输成本, 越库内货物流动快慢造成的库存成本, 以及因转运中不匹配导致的惩罚成本。根据成本构成特点, 设计了阶段分配启发式策略和调整策略来产生初始种群。在阶段分配启发式策略中, 首先采用基于运输成本的概率启发策略产生供应商的运输策略, 再根据供应商的运输策略, 采用基于运输成本的概率启发策略产生顾客的运输策略; 在调整策略中, 采用贪婪思想重新匹配每个越库内运输分配。

1) 基于供应商运输成本的概率启发策略。该

策略是一种基于概率的贪婪策略, 用于产生供应商的运输策略。该策略的基本思想是每个供应商依照一定的概率选择运输班次和越库, 对于  $1 \leq i \leq n$ , 选择概率基于运输成本的大小可以由式 (12) 得出, 从该式可以看到, 运输成本高的运输班次被选择的概率小, 相反, 运输成本低的运输班次被选择的概率大。

$$pr_i^v = \text{coe} \times \frac{\max_{j \neq i} c_{i j q} - c_{i p q}}{\max_{j \neq i} c_{i j q} - \min_{j \neq i} c_{i j q}} \quad (12)$$

2) 基于顾客运输成本的概率启发策略。与上面的策略类似, 对于  $1 \leq k \leq m$ , 顾客选择运输班次的概率由式 (13) 得出。

$$pr_k^u = \text{coe} \times \frac{\max_{j \neq k} c'_{j k q} - c'_{u k q}}{\max_{j \neq k} c'_{j k q} - \min_{j \neq k} c'_{j k q}} \quad (13)$$

在式 (12) 和 (13) 中,  $\text{coe}$  ( $0 < \text{coe} < 1$ ) 为常量, 用于控制不同运输班次的选择概率。

3) 基于运输计划的确定性启发策略。该策略基于贪婪思想, 在完成供应商和顾客的运输分配后, 该策略用于调整每个越库的运输分配计划, 让进出越库的货物达到匹配。

#### 2.1.3 自适应交叉运算

有效交叉算子不仅取决于问题的结构, 种群大小和遗传算法的其他参数, 还取决于当前种群的状况。因此, 采用精英选择策略选择种群中一定数量的最好个体作为交叉运算的父代, 随着问题规模的增大, 单一交叉算子有一定的局限性, 因此特别设计两种交叉算子, 并在进化迭代过程中根据交叉算子的优越性自适应的选择交叉算子。第一种交叉算子称之为 PUCO (probabilistic-uniform crossover operator), 第二个交叉算子称为 PACO (preserve-assignment crossover operator)。由于该问题可行解的结构复杂, 供应商的运输路径选择影响顾客的运输路径选择, 因此设计的交叉运算和变异运算只作用于基因段内表示供应商越库分配以及运输路径选择的基因位, 一旦供应商越库分配和路径选择发生变化, 将按照 2.1.2 中的策略重新调整顾客的越库分配以及运输路径, 以此生成新的可行解。下面将具体介绍这两种交叉算子和自适应交叉运算。

1) PUCO

该算子与一般的一致交叉算子不同,交叉的概率是由父代的目标值决定的,目的是让子代得到更多适应度好的父代的优良基因位. 交叉概率如下.

$$p_i = 1 - \frac{OBJ_{parent_i}}{OBJ_{parent_1} + OBJ_{parent_2}} \quad (i = 1, 2) \quad (14)$$

首先从种群池中选择两个父代,计算出基因位交叉概率,以该交叉概率交叉父代的基因位产生子代.

### 2) PACO

该交叉算子是两点交叉算子,该算子能尽可能地使得子代从父代中保存相同的基因位. 首先,从种群池中随机选择两个个体作为父代,其次随机选择两个点用于父代基因交叉. 第一个子代直接遗传第一个父代的两个点之间的基因位,其它基因位从第二个父代遗传得到,同理,改变两个父代的交叉角色得到第二个子代个体. 图1为8个表示越库的基因位的 PACO 算子的交叉运算,表示运输班次的基因位交叉运算相同.

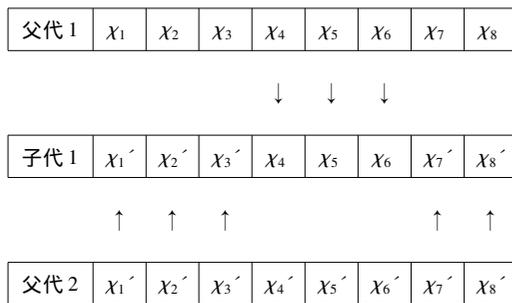


图1 PACO 算子  
Fig. 1 PACO operator

### 3) 自适应交叉策略

PACO 算子能尽可能多地保存父代的基因位,其缺点是子代与父代的优劣性关系不大;相反,PUCO 算子能保证子代尽可能从优良的父代中继承基因位,但是不能较完整地保存父代的基因结构,因此采用自适应交叉策略能够克服这两个算子在进化过程的缺陷,使得进化过程具有较强的鲁棒性.

自适应交叉策略的基本思想是根据交叉算子在进化过程的优劣性不断更新交叉算子在下次交叉过程被选择的概率. 首先,在第一代交叉过程

中,给两个交叉算子设置相同的概率分布,使得每个算子具有相同的参与交叉运算的可能性,随着进化过程的进行,不断地更新交叉算子的概率分布,并根据概率分布在下次交叉过程中奖励或者惩罚交叉算子. 自适应交叉策略使得算法偏向具有较强进化能力和鲁棒能力的交叉算子.

本文中,在自适应交叉运算中采用 Kaelo 和 Ali<sup>[20]</sup> 提出的一种非线性反馈学习策略,数值实验结果表明该策略对本文研究的越库转运优化问题有良好的效果. 下面对该策略作简要介绍. 首先为两个交叉算子设置相同的概率分布  $(p_1^{cr1}, p_2^{cr2}) = (0.5, 0.5)$  这里  $cr1$  和  $cr2$  分别代表 PACO 和 PUCO 这两种交叉算子. 在随后根据两个算子在迭代过程的相对表现更新下一次迭代的概率分布,其中  $p_k^{cr1} + p_k^{cr2} = 1$ . 如果交叉算子在当前的交叉运算过程中比另外一个算子表现要好,则该算子的概率采用式(15)进行更新,其中  $\beta_0 \in (0, 1)$  是奖励因子

$$p_k = p_{k-1} + \beta_0 p_{k-1} (1 - p_{k-1}) \quad (15)$$

相反,则采用式(16)更新概率,其中  $\beta_1 \in (0, 1)$  为惩罚因子

$$p_k = p_{k-1} - \beta_1 p_{k-1} (1 - p_{k-1}) \quad (16)$$

由于  $p_k^{cr1} + p_k^{cr2} = 1$ ,每次只要更新  $p_k^{cr1}$ .

自适应交叉策略的另一个关键要素是设计奖惩机制,根据 Kaelo 和 Ali<sup>[20]</sup> 的研究结果,给交叉算子产生的每个后代  $i$  分配权重  $p_i, i = 1, 2, \dots, m$ ,其中  $m$  等于交叉运算次数的两倍. 首先,先将产生的子代按照适应度进行排序,并将  $p_1 = \frac{2-c}{m} (0 < c < 1)$  分给适应度最好的子代,将

$p_m = \frac{c}{m}$  分给适应度最差的子代,其他子代的权重

$p_i$  由式(17)得出

$$p_i = p_m + \frac{m-i}{m-1} (p_1 - p_m) \quad (i = 2, 3, \dots, m-1) \quad (17)$$

其次,计算该交叉算子的产生子代的平均适应度,并将适应度比平均适应度好的子代权重相加作为该算子的和权重,若该算子的和权重大于 0.75,则用式(15)奖励该算子;若和权重小于 0.35,则用式(16)惩罚该算子,若和权重属于

[0.35 0.75] 则概率分布保持不变. 算法 1 描述了基于反馈学习机制的自适应交叉运算.

算法 1: 自适应交叉运算{

#max iter // 最大迭代次数

初始化( $p_1^{cr1}$   $p_2^{cr2}$ ) = (0.5 0.5)

For ( iter = 1 to #max iter) do{

If ( iter = 1) then{

两个交叉算子各自产生  $\frac{m}{2}$  个子代;

用奖惩机制计算个体的权重, 并计算个体的平均适应度以及该交叉算子的和权重, 根据和权重更新概率分布( $p_{iter+1}^{cr1}$   $p_{iter+1}^{cr2}$ );

}

Else if( 以概率  $p_{iter}^{cr1}$  来随机选择算子 cr1, 以概率  $1 - p_{iter}^{cr1}$  来随机选择算子 cr2) then{

用所选择的交叉算子产生  $m$  个子代;

用式(17) 计算个体的权重;

计算个体平均适应度与该算子的和权重;

根据和权重用(15) 或者(16) 更新概率

( $p_{iter+1}^{cr1}$   $p_{iter+1}^{cr2}$ );

} end if

} end if

} end for

}

#### 2.1.4 自适应变异运算

考虑到保存较为优良的基因位, 设计一种自适应变异算子在多样化种群的同时能较多地保存优良的基因. 自适应表现在不仅每个个体以一定的变异概率参与变异, 称该变异概率为体变异概率, 同时每个基因位以一定的变异概率进行变异, 称该变异概率为位变异概率. 体变异概率是常用的给定的变异概率( $p_0$ ). 与交叉运算相同, 位变异运算也只用于表示供应商运输策略的基因位, 因此位变异概率采用式(12) 定义的概率来计算.

#### 2.1.5 邻域搜索策略

邻域搜索(NS) 是一种局部搜索策略, 被广泛地应用于解决各种组合优化问题. 在本文中, 邻域搜索用来改进遗传算法的局部搜索能力, 其基本思想是搜索解的邻域以期获得更好的解.

邻域结构是邻域搜索的关键, 有效的邻域结

构取决于问题的结构. 在该问题中, 总成本包括三大部分, 每个部分的关键影响要素不同. 运输成本通过改变运输网络中的运输策略得以改变; 库存成本可以推迟供应商的运输到达时间以及提前顾客的运输始发时间加以改善; 而惩罚成本则可以通过改变原先顾客的货物分配方案来改变. 基于该问题的成本结构, 设计了以下两种邻域搜索策略.

##### 1) 调整越库的运输计划

该搜索是一种可变邻域分解搜索. 固定属性为越库, 可变属性为供应商与越库以及越库与顾客之间的运输班次. 该搜索技术先固定所选的越库, 然后搜索该越库的所有供应商和顾客的运输班次, 以达到同一越库内供应商和顾客之间运输时间的最佳吻合从而减少库存持有成本. 该搜索技术搜索速度取决于越库的数量, 因此应用于所有的子代和当前最好解.

##### 2) 插入或交换需求未满足的顾客的运输计划

该搜索是一种梯度邻域搜索, 目的在于将可能降低惩罚成本. 相对于运输成本而言, 惩罚成本往往较高, 因此成功地插入一个未满足需求的顾客或者调整需求未满足的顾客与其他需求得到满足的顾客之间运输计划可能会极大地降低惩罚成本. 基于这点, 该搜索技术以未满足需求的顾客的邻域为搜索空间进行搜索. 首先, 随机地选择一个需求未被满足的顾客, 再试探地为该顾客安排运输计划或者将该顾客与其他已安排运输计划的顾客进行交换, 直到邻域内不能得到更好的改善解.

#### 2.2 自适应性遗传算法框架

在下面的遗传算法框架中, #pop 表示种群数, #crossover 表示交叉次数, Offsprings 表示子代, #max iter 表示最大迭代次数, #term iter 表示当前最优解允许的最大未被改善迭代次数.

遗传算法框架: {

参数初始化;

产生初始种群;

for { iter = 1 to #max iter } do{

for ( off = 1 to #crossover) do{

```

调用自适应交叉运算;
调用自适应变异运算;
对 Offsprings 采用邻域搜索;
} end for

```

从所有个体中选择最好  $\#pop$  个体作为下次迭代的种群;

对当前最好解和选择的部分个体采用邻域搜索;

```
更新当前最优解;
```

```
If(当前最好解在 #term iter 代内未改善) then{
```

```
  将当前最好解作为全局最优解;
```

```
  输出结果和迭代时间并终止程序;
```

```
  } end if
```

```
} end for
```

```
输出结果和迭代时间;
```

```
}
```

### 3 数值实验与分析

在数值实验中,通过产生大量的数据来检测遗传算法的有效性.由于文中提出的运输模型是一种较为新型的越库转运模型,国际上无标准测试数据,因此采用学术界常用的方法<sup>[8,18-19]</sup>,将得到的实验结果与 ILOG CPLEX solver 求解结果进行对比分析,来检测该遗传算法的有效性.实验在 MS-Windows XP, Pentium 4 2.93GHz, 1G 内存的环境下测试.

#### 3.1 数据生成与参数设定

由于没有标准测试数据,需要根据实际情况随机产生模型的参数,并给出遗传算法的参数设定.由于越库运输的特殊性,运输网络的时间范围设定在两天 48 小时之内,表 1 是模型参数的随机产生范围,表 2 为遗传算法的参数设定.

#### 3.2 实验结果与分析

在参数设置范围内针对 8 组不同规模问题,每组规模各做 4 次随机实验,规模的大小由供应商、越库以及顾客的数量决定.

表 3 给出了平均实验结果(每个数值都表示该组规模的 4 次随机实验的平均值), $(n \times l \times$

$m)$  表示  $n$  个供应商、 $l$  个越库和  $m$  个顾客.这 8 组 32 次实验规模范围 $(n \times l \times m)$  从 $(30 \times 10 \times 30)$  到 $(40 \times 15 \times 40)$ .表中 LBs 是 CPLEX 软件终止时得到的平均目标值下界,Objective 是平均目标值,Times 表示平均运行时间(以  $s$  为单位),Gap 表示目标值与下界的平均差距百分比.由表 3 实验数据可以看出,CPLEX 不论在运算时间方面还是在解的质量方面,都明显逊于采用邻域搜索的自适应遗传算法的表现,在运行时间方面 CPLEX 都需要超过 3 600s 去求得一个近似最优解,而采用邻域搜索的自适应遗传算法平均在 300s 到 900s 之间就可得到一个更优的近似最优解.总的来说,虽然采用邻域搜索的自适应遗传算法的解仍然与 CPLEX 给出的下界存在一定的差距,不过目前就该问题的复杂程度而言,CPLEX 也很难有效的求解,采用邻域搜索的自适应遗传算法的优势是明显的,该遗传算法在运行时间和解的质量方面都要优于 CPLEX,尤其在求解速度方面有很大的竞争力.

为了验证自适应和邻域搜索策略在遗传算法中所起的作用,本文比较了分别在不采用自适应策略和邻域搜索策略的情况下的三类遗传算法:

- 1) 不采用邻域搜索的自适应遗传算法(AGA without NS): 在该算法中,不采用邻域搜索技术;
- 2) 采用邻域搜索的遗传算法(采用 PACO 算子,GA with NS and PACO): 该算法采用 PACO 交叉算子,并在迭代过程中采用邻域搜索技术;
- 3) 采用邻域搜索的遗传算法(采用 PUCO 算子,GA with NS and PUCO): 该算法采用 PUCO 交叉算子,并在迭代过程中采用邻域搜索技术.

表 3 还给出以上三类遗传算法的比较结果,从给出的不同规模的数值实验结果可以看出,采用邻域搜索的自适应遗传算法性能最佳,此外两种交叉算子中, PUCO 的结果明显优于 PACO,这与算子的设计方法有关,因为 PUCO 算子与父代的成本有关,对子代的性能有较大的影响.

为了更加直观的比较带有邻域搜索的自适应遗传算法比其他算法在求解质量方面有着更显著的优越性,定义了其他算法(Gap<sub>2</sub>)与带有邻域搜索的自适应遗传算法(Gap<sub>1</sub>)之间的相对 Gap

如下:

$$(\text{Gap}_2 - \text{Gap}_1) / \text{Gap}_1 \times 100\%;$$

则根据实验结果可以得到表4.从表4中可以看出清楚的看出带有邻域搜索的自适应遗传算法有非常显著的优势,CPLEX的表现仅次于前者,此外采用PUCO的遗传算法性能明显优于采用PACO的遗传算法,所以邻域搜索策略与自适应策略在提高算法的性能方面发挥了显著作用.

## 4 结束语

本文探索了一种固定运输模式下的越库转运问题,其目标是通过该越库转运网络以最小总成本将商品从供应商经过越库转运给顾客,其中总成本包括运输成本、库存成本以及惩罚成本.文中证明了该问题是强NP难问题,因此不存在多项式算法.通过设计一种采用了邻域搜索策略的自适应遗传算法来解决该问题.从问题结构以及限制条件上看,此类越库转运问题相对传统的转运问题有更高的复杂性.首先,该越库转运网络拥有供应商、越库转运中心以及顾客这样的三层级体系,网络结构相对复杂;其次,越库的功能不仅是转运,而且可以短暂存储货物,将会影响到转运成本,这与一般的网络流问题有很大差异;第三,运输模式方面采取了固定运输模式下的单次运输方式,这个限制条件大大提高了问题的复杂度,在选择可行路径时不仅要考虑固定路径的起始与到达时间,而且要考虑单次运输约束;第四,由于有固定运输模式以及单次运输方式的限制,使得部分客户的需求不能通过该运输网络满足,将导致额外的费用,即惩罚成本.由于该问题具有以上特征,因此文中算法设计需要有很强的针对性.首先,该问题的解不仅要包含越库的分配,即每个供应商该运货给哪个越库,以及每个顾客该由哪个越库来满足需求,此外还要包含路径的分配,即供应商与越库之间以及越库与顾客之间的运输路径的选择.基于解的这些特征,设计的遗传算法的染色体由前后两个部分组成,前部表示供应商与顾客的越库分配,后部表示他们与越库之间的路径分配;其次,总成本包括供应商以及顾客的运输成

本,越库内货物停留造成的库存成本,以及因转运不匹配导致的惩罚成本,由于成本构成与解的结构都相对复杂,因此文中有针对性的设计了一种三阶段初始种群生成方法.在第一阶段,首先采用基于运输成本的概率启发策略产生供应商的运输策略;第二阶段,根据供应商的运输策略,采用基于运输成本的概率启发策略产生顾客的运输策略;第三阶段,采用贪婪思想重新调整每个越库内商品的运输调度,目的在于减少库存以及惩罚成本.通过这三阶段方法,尽可能选择总成本较低的解作为初始解,为提高遗传算法的效率提供支持;第三,设计了PACO和PUCO两种交叉算子,由于该问题可行解的结构复杂,供应商的越库分配以及运输路径选择直接影响顾客的配送方案,因此设计的交叉运算和变异运算只作用于基因段内表示供应商越库分配以及运输路径的基因位,一旦供应商的越库分配以及路径选择发生变化,将按照生成初始解时候采用的策略重新调整顾客配送计划,以此生成新的可行解.这两种交叉算子由于各有优缺点,所以文中采用了一种非线性反馈学习策略作为自适应策略,数值实验结果表明该策略对该问题有良好的效果;第四,设计了自适应变异算子,包含了体变异与位变异,在多样化种群的同时能较多地保存优良的基因;最后,在遗传算法中,采用了两种邻域搜索策略.第一种策略是“调整越库的运输计划”,目的是让同一越库内供应商与客户之间的货物运输时间更加匹配,减少中间的停留时间,从而达到减少库存成本的目的;第二种策略是“插入或交换需求未满足的顾客的运输计划”,目的是通过成功插入一个尚未分配的顾客或者交换一个未分配的和另一个已分配的顾客之间的运输计划,来达到降低惩罚成本的目的.

数值实验结果可以看出,该算法在运行时间和解的质量方面有明显的优势,可以有效的解决此类问题,同时通过与分别不采用自适应策略和邻域搜索策略的遗传算法的实验结果作对比,可以看出采用邻域搜索技术的自适应性遗传算法性能最佳,表明自适应策略与邻域搜索策略对算法性能的提高有显著的作用.

表 1 模型数据范围

Table 1 Parameter range of the model

模型参数	$\gamma_{ij}(\gamma'_{jk})$	$(b'_{ijq}, e'_{ijq})$	$(b''_{jkq}, e''_{jkq})$	$c_{ijq}(c'_{jkq})$	$h_j$	$P_j$	$s_i(d_k)$
随机范围	[4 8]	( [0 ,12] , [13 ,24] )	( [12 ,35] , [36 ,48] )	[10 ,30]	[1 ,3]	[30 ,90]	[100 ,500]

表 2 采用邻域搜索的自适应遗传算法的参数设定

Table 2 Parameter setting of AGA with NS

参数	#maxiter	#termiter	#pop	#crossover	$p_0$	$\beta_0$	$\beta_1$	$c$	coe
数值	100 000	300	300	100	0.02	0.65	0.85	0.02	0.95

表 3 不同算法的随机实验运行结果

Table 3 Results of different algorithms

	$n \times l \times m$	$30 \times 10 \times 30$	$34 \times 10 \times 34$	$36 \times 10 \times 36$	$40 \times 10 \times 40$	$36 \times 15 \times 36$	$40 \times 15 \times 40$
	LB	210 545	236 773	245 666	265 084	245 715	262 005
AGA with NS	Objective	255 505	300 539	302 850	339 698	306 957	337 243
	Time( s)	396.02	696.12	771.25	679.59	860.01	824.82
	Gap	17.60%	21.22%	18.88%	21.96%	19.95%	22.31%
CPLEX	Objective	256 282	314 457	309 248	347 144	313 465	345 372
	Time( s)	> 3 600	> 3 600	> 3 600	> 3 600	> 3 600	> 3 600
	Gap	17.85%	24.70%	20.56%	23.64%	21.61%	24.14%
AGA without NS	Objective	276 925	311 412	318 371	346 268	335 156	358 106
	Time( s)	337.02	337.39	365.96	608.48	342.13	373.53
	Gap	23.94%	23.96%	22.84%	23.45%	26.63%	26.75%
GA with NS And PACO	Objective	284 002	321 859	328 634	356 459	340 545	368 038
	Time( s)	157.14	163.91	268.88	261.71	190.61	242.21
	Gap	25.84%	26.41%	25.21%	25.57%	27.76%	28.73%
GA with NS And PUCO	Objective	277 855	306 525	318 366	346 684	336 237	356 239
	Time( s)	164.18	297.69	282.71	309.69	265.50	374.80
	Gap	24.20%	22.78%	22.81%	23.53%	26.78%	26.24%

表 4 采用邻域搜索策略的自适应遗传算法与其余算法的相对 Gap

Table 4 Percentage gap between AGA with NS and other algorithms

$n \times l \times m$	$30 \times 10 \times 30$	$34 \times 10 \times 34$	$36 \times 10 \times 36$	$40 \times 10 \times 40$	$36 \times 15 \times 36$	$40 \times 15 \times 40$
CPLEX	1.42%	16.40%	8.90%	7.65%	8.32%	8.20%
AGA without NS	36.02%	12.91%	20.97%	6.79%	33.48%	19.90%
GA with NS And PACO	48.82%	24.46%	33.53%	16.44%	39.15%	28.78%
GA with NS And PUCO	37.50%	7.35%	20.82%	7.15%	34.24%	17.62%

参 考 文 献:

[1] Napolitano M. Making the Move to Corss Docking: A Practical Guide to Planning , Designing , and Implementing A

- Cross Dock Operation[M]. Oak Brook, Illinois: Ware Housing Education and Research Council, 2000.
- [2] Apte D M, Viswanathan S. Effective cross docking for improving distribution efficiencies[J]. *International Journal of Logistics: Research and Applications*, 2000, 3(3): 291 - 302.
- [3] 刘志学. 现代物流手册[M]. 北京: 中国物资出版社, 2001.
- Liu Zhixue. *Modern Logistics Handbook*[M]. Beijing: China Logistics Publishing House, 2001. (in Chinese)
- [4] Simchi-Levi D, Kaminsky P, Simchi-Levi E. *Designing and Managing the Supply Chain*[M]. 2<sup>nd</sup> ed., New York: McGraw-Hill, 2003.
- [5] Axsäter S. New decision rule for lateral transshipments in inventory systems[J]. *Management Science*, 2003, (49): 1168 - 1179.
- [6] Axsäter S. Evaluation of unidirectional lateral transshipments and substitutions in inventory systems[J]. *European Journal of Operational Research*, 2003, (149): 438 - 447.
- [7] Bartholdi III J J, Gue K. R. Reducing labor costs in an LTL crossdocking terminal[J]. *Operations Research*, 2000, 48(6): 823 - 832.
- [8] Miao Z, Lim A, Ma H. Truck dock assignment problem with operational time constraint within crossdocks[J]. *European Journal of Operational Research*, 2009, (192): 105 - 115.
- [9] 俞亮, 陈峰. 最小化误工个数的越库调度模型与启发式算法[J]. *上海交通大学学报*, 2009, (12): 1984 - 1988.
- Yu Liang, Chen Feng. Cross docking scheduling model and heuristics to minimize the number of tardy jobs[J]. *Journal of Shanghai Jiaotong University*, 2009, (12): 1984 - 1988. (in Chinese)
- [10] 衣方磊, 徐寅峰, 辛春林. 局内动态配送车调度管理及其竞争策略[J]. *管理科学学报*, 2007, (4): 1 - 8. (in Chinese)
- Yi Fanglei, Xu Yinfeng, Xin Chunlin. On-line management in dynamic distribution-truck scheduling problem and its strategies[J]. *Journal of Management Sciences in China*, 2007, (4): 1 - 8.
- [11] 姚建明, 刘丽文, 蒲云, 等. MC 模式下供应链动态调度的蚁群寻优分析[J]. *管理科学学报*, 2007, (3): 7 - 14.
- Yao Jianming, Liu Liwen, Pu Yun, et al. Analysis on ants optimization algorithm for supply chain dynamic scheduling in mass customization[J]. *Journal of Management Sciences in China*, 2007, (3): 7 - 14. (in Chinese)
- [12] Chen F, Lee C Y. Minimizing the make span in two-machine crossdocking scheduling[J]. *European Journal of Operations Research*, 2005, (5): 102 - 131.
- [13] 马东彦. 越库作业调度问题及其启发式算法研究[J]. *物流技术*, 2007, (6): 57 - 59.
- Ma Dongyan. Heuristic algorithm solving cross docking distribution problem[J]. *Logistics Technology*, 2007, (6): 57 - 59. (in Chinese)
- [14] 陈峰, 宋凯雷. 越库物流调度问题及其近似与精确算法[J]. *工业工程与管理*, 2006, (6): 53 - 58.
- Chen Feng, Song Kailei. Cross docking logistics scheduling problem and its approximative and precise algorithms[J]. *Industrial Engineering and Management*, 2006, (6): 53 - 58. (in Chinese)
- [15] 王长军, 贾永基, 徐琪, 等. 并行机下独立任务调度的无秩序代价分析[J]. *管理科学学报*, 2010, (5): 44 - 50.
- Wang Changjun, Jia Yongji, Xu Qi, et al. Price of anarchy analysis for scheduling selfish tasks on parallel machines[J]. *Journal of Management Sciences in China*, 2010, (5): 44 - 50. (in Chinese)
- [16] 宋莉波, 徐学军, 孙延明, 等. 一种求解柔性工作车间调度问题的混合遗传算法[J]. *管理科学学报*, 2010, (11): 49 - 54.
- Song Libo, Xu Xuejun, Sun Yanming, et al. A hybrid genetic algorithm for flexible job shop scheduling problem[J]. *Journal of Management Sciences in China*, 2010, (11): 49 - 54. (in Chinese)
- [17] Lim A, Miao Z, Rodrigues B, et al. Transshipment through crossdocks with inventory and time windows[J]. *Naval*

Research Logistics ,2005 ,52 ( 8) : 724 - 733.

[18]Chen P , Yun G , Lim A et al. Multiple crossdocks with inventory and time windows[J]. Computers & Operations Research ,2006 ,( 33) : 43 - 63.

[19]Ma H , Miao Z , Lim A. Crossdocking distribution networks with setup cost and time window constraint[J]. Omega-International Journal of Management Science ,2011 ,( 39) : 64 - 72.

[20]Kaelo P ,Ali M M. Integrated crossover rules in real coded genetic algorithms[J]. European Journal of Operational Research ,2007 ,176( 1) : 60 - 76.

## A self-adaptive genetic algorithm for the transshipment problem through crossdocks

MIAO Zhao-wei<sup>1</sup> , YANG Feng<sup>1</sup> , XU Dong-sheng<sup>2</sup> , SHI Ning<sup>2</sup>

1. School of Management , Xiamen University , Xiamen 361005 , China;

2. School of Management , Sun Yat-sen University , Guangzhou 510275 , China

**Abstract:** In this paper we study a kind of transshipment problem , in which the flows through the crossdock are constrained by fixed transportation schedules with single release and single delivery , cargos can be delayed in crossdocks but any delay at the last time point of time horizon will incur inventory penalty cost , and the objective is to find a transshipment scheme with minimum cost. The problem is proved to be NP-hard in the strong sense in this paper. We therefore focus on developing efficient heuristics. Based on the problem structure , we propose a self-adaptive genetic algorithm with neighborhood search ( AGA with NS) to solve the problem efficiently. Computational experiments under different scenarios show that AGA with NS outperforms CPLEX solver , meanwhile , in order to further test the effectiveness of the adaptive scheme and neighborhood search , we also conduct computational experiments by different algorithms such as AGA without NS , GA with NS and PACO , and GA with NS and PUCO. Finally the results show that AGA with NS is the best one among these algorithms for this problem.

**Key words:** genetic algorithm; transshipment; crossdock; supply chain

附录:

命题 1 证明 需要通过已知的强 NP 完全问题 3-Partition 问题来证明文中研究的越库转运问题是强 NP 难的. 3-Partition 问题的描述如下: 给定正整数  $w, D$ , 集合  $T = \{1, 2, \dots, 3w\}$ , 以及正整数  $R_i$ , 其中  $R_i$  满足  $\sum_{i \in T} R_i = wD$  并且  $D/4 < R_i < D/2 (i \in T)$ . 那么集合 T 能否可以划分为  $w$  个不相交的集合  $T_1, T_2, \dots, T_w$  使得  $|T_j| = 3$  并且  $\sum_{i \in T_j} R_i = D (j = 1, 2, \dots, w)$ ? 只要能够通过多项式时间的变换可以将任意一个 3-Partition 问题转换成该越库转运问题的一个特例, 该特例要解决是否存在一个可行解, 使得其目标函数值不大于  $2wD$ ?

首先做如下变换, 在一特定的越库转运问题中, 有  $w$  个供应商, 1 个越库以及  $3w$  个顾客, 令每个供应商提供的货量  $s_j = D (j = 1, 2, \dots, w)$ , 每个顾客的需求量  $d_i = R_i (i = 1, 2, \dots, 3w)$ . 此外, 对于每个供应商  $j (j = 1, 2, \dots, w)$ , 到越库只有一条固定运输路径, 起始时间点是  $j$ , 到达时间点是  $j + 1$  (即  $(j, j + 1)$ ); 对于每个顾客  $i (i = 1, 2, \dots, 3w)$ , 有  $w$  条从越库到他的固定运输路径  $\{(k + 1, k + 2) | k = 1, 2, \dots, w\}$ ; 令每条路径上的单位运输成本都为 1, 越库内单位时间单位库存成本以及单位惩罚成本都为 3.

(下转第 95 页)

## A game model for green supply chain management based on government subsidies

ZHU Qing-hua , DOU Yi-jie

School of Business Management , Dalian University of Technology , Dalian 116024 , China

**Abstract:** This paper establishes a three-stage game model by considering products' green degree and government subsidies. The first stage is that the government determines the subsidies coefficient; the second stage is that manufacturers with various green strategies in supply chains determine their own products' green degree; The third stage is that manufacturers in supply chains determine their own products' prices. Further , a numerical case is presented to test the effects of the variation of different factors. The results provide insights into the decision-making of governments and companies.

**Key words:** green supply chain management; government subsidies; green degree; game model

(上接第 34 页)

下面要证明当且仅当该特定的越库转运问题存在着一个目标函数值不大于  $2wD$  的可行解的时候,对应的 3-Partition 问题存在可行解.

首先,假设该特定的越库转运问题存在着一个目标函数值不大于  $2wD$  的可行解,那么可以证明这个可行解是最优解.因为只有当所有顾客的需求都得到满足,并且无任何货物在转运过程中在越库中停留的条件下,目标函数值才等于  $2wD$ . 否则,由于每条路径上的单位运输成本都为 1,而越库内单位时间单位库存成本以及单位惩罚成本都为 3,因此只要有货物无法运达顾客手中或者停留在库存中,目标函数值都将大于  $2wD$ . 也就是说  $2wD$  是该特定问题的目标值下界,因此这个可行解就是最优解.经过以上分析,可知最优解满足以下两个条件:(1) 无惩罚费用,顾客需求全部被满足;(2) 无任何库存费用,即所有的货物

都是通过越库直接转运给顾客,没有停留.此外,由于该越库问题有单次运输约束,因此可以得到以下结论,对于供应商  $j$  ( $j = 1, 2, \dots, \mu$ ) 运到越库的货物量  $D$ ,都是直接转运给了顾客子集  $T_j$ ,即  $\sum_{i \in T_j} d_i = D$ . 因为  $D/4 < d_i < D/2$  ( $i \in T$ ), 可得  $|T_j| = 3$ . 因此  $T_1, T_2, \dots, T_w$  就是 3-Partition 问题的一个可行解.

其次,假设  $T_1, T_2, \dots, T_w$  是 3-Partition 问题的一个可行解,那么将供应商  $j$  的总货量  $D$  来满足顾客  $i$  ( $i \in T_j$ ) 的需求 ( $j = 1, 2, \dots, \mu$ ), 供应商通过路径  $(j, j+1)$  将货物运往越库,而越库通过路径  $(j+1, j+2)$  及时将货物运往顾客  $i$  ( $i \in T_j$ ). 很容易证明该方案是该越库转运问题的一个可行解,并且目标函数值等于  $2wD$ . 证毕.