

时变路网下带时间窗的易腐食品生产 - 配送问题

吴 瑶, 马祖军

(西南交通大学 经济管理学院 物流与应急管理研究所, 成都 610031)

摘要 为减少易腐食品在生产配送过程中的价值损耗, 通常按订单确定的交货期合理组织生产并立即配送。为此, 针对多品种易腐食品的集成生产 - 配送问题, 考虑实际配送过程中路网交通的时变特性, 建立了以系统总成本最小为目标、带时间窗的易腐食品集成生产 - 配送问题优化模型。根据该模型特点设计了一种混合遗传算法, 并通过数值算例验证了所提模型和算法的有效性。结果表明, 路网的时变特性显著影响客户服务水平, 优化投入配送的车辆数可有效降低系统总成本和易腐食品的价值损耗。

关键词 易腐食品; 生产调度; 车辆路径; 时变; 混合遗传算法

Time-dependent production-delivery problem with time windows for perishable foods

WU Yao, MA Zujun

(Institute for Logistics and Emergency Management, School of Economics and Management, Southwest Jiaotong University, Chengdu 610031, China)

Abstract To reduce the decay of perishable products in the process of production and delivery, it is necessary to appropriately arrange the production and delivery of perishable products subject to their deadlines. Therefore, to minimize the total operational costs of producing and delivering perishable foods, an optimization model was developed for joint production-delivery problem with time-windows, by considering the time-dependent characteristics of road networks in real world. According to the characteristics of the model, a hybrid genetic algorithm was proposed and validated by using a numerical experiment. The results show that the time-dependent characteristics of road networks have a remarkable impact on customer service level. The total operational costs and the loss of product value can be reduced further by optimizing the number of delivery vehicles in use.

Keywords perishable foods; production scheduling; vehicle routing; time-dependent; hybrid genetic algorithm

1 引言

近年来, 随着人们生活水平的提高和生活节奏的加快, 绿色、安全和便捷的食品越来越受到消费者的青睐。消费者在选购易腐食品时, 不仅考虑价格, 而且更关注产品的质量特征(即新鲜程度)。尤其是保质期极短的生鲜食品, 如鲜奶、鲜肉、快餐、烘焙食品(如面包、蛋糕)、鲜切果蔬等, 其新鲜度是影响产品价值的关键。针对这类快速易腐性食品, 实际上随着生产加工的结束, 其新鲜度即开始衰减, 产品价值也不断下降, 因

收稿日期: 2015-06-09

作者简介: 吴瑶(1984-), 男, 汉, 湖北十堰人, 博士研究生, 研究方向: 物流系统优化, E-mail: yaowumail@126.com; 马祖军(1974-), 男, 汉, 浙江开化人, 教授, 博士生导师, 研究方向: 物流与供应链管理、应急管理, E-mail: zjma@swjtu.edu.cn。

基金项目: 国家自然科学基金重大研究计划培育项目(90924012); 国家自然科学基金面上项目(71672154); 教育部人文社会科学研究规划基金项目(16YJA630038)

Foundation item: Major Research Plan of National Natural Science Foundation of China (90924012); National Natural Science Foundation of China (71672154); Humanities and Social Science Foundation of Ministry of Education of China (16YJA630038)

中文引用格式: 吴瑶, 马祖军. 时变路网下带时间窗的易腐食品生产 - 配送问题[J]. 系统工程理论与实践, 2017, 37(1): 172-181.

英文引用格式: Wu Y, Ma Z J. Time-dependent production-delivery problem with time windows for perishable foods[J]. Systems Engineering — Theory & Practice, 2017, 37(1): 172-181.

而通常采取零库存生产方式, 即按订单确定的交货期合理组织生产并立即配送, 以尽可能缩短产品的交付时间。因此, 如何合理地安排易腐食品的生产计划和配送过程, 以尽量保持产品交付时的新鲜度, 直接关系着企业的盈利能力和服务水平。

目前有关易腐产品生产调度和物流配送的研究, 基本上都是针对其中之一进行研究。在易腐产品生产调度研究方面, Arbib 等^[1]假设中间转换阶段不影响产品的相对易腐性和开始 - 完成时间对的可行性, 建立了易腐产品生产调度的三维匹配模型。Sana 等^[2]考虑具有时变需求率的易腐品柔性生产系统, 采用内罚函数法研究了允许缺货情况下的生产批量问题。Lütke Entrup 等^[3]在酸奶生产计划和调度中考虑产品保质期, 基于分批计划 (block planning) 原则建立了三种周生产计划的混合整数规划模型。周泓等^[4]针对易变质产品生产批量计划与作业排序的集成优化问题, 建立了以总成本最小为目标的混合整数规划模型, 并采用协同进化遗传算法进行求解。

在易腐产品配送研究方面, Tarantilis 等^[5]以物流总成本最小为目标, 分别建立了鲜奶和鲜肉配送的多车型车辆路径问题模型, 并设计了门槛接受算法 (threshold-accepting algorithm) 进行求解。Hsu 等^[6]考虑了易腐食品配送过程中的随机性, 以总成本 (包含能源消耗成本和客户时间窗违反成本) 最小为目标, 建立了随机、带时间窗的车辆路径问题 (vehicle routing problem with time-windows, VRPTW) 优化模型, 采用启发式算法求得最优车辆配送路线和调度方案。Osvald 等^[7]针对新鲜蔬菜配送问题建立了时变 VRPTW 模型, 并设计了启发式算法对模型进行求解, 分析了产品价值损耗对配送总成本的影响。Amorim 等^[8]通过建立多目标优化模型, 对不同配送环境、不同时间窗要求以及不同产品变质系数下的产品新鲜度与配送成本之间的关系进行了研究。

上述有关易腐产品生产和配送的研究均未考虑两者的集成调度优化, 但针对时效性较强的易腐食品, 单独考虑产品的生产调度或者物流配送优化显然不足。例如, 为了降低易腐食品交付时的价值损耗, 若仅考虑产品的生产调度优化, 尽量推迟订单的生产开始时间, 则可能使产品的配送方案难以达到最优甚至不可行。相反, 若仅考虑订单的配送优化, 则可能引起过多的提前生产, 导致较大的价值损耗。因此, 针对时效性较强的易腐食品供应链, 有必要统筹优化产品的生产和配送过程, 即研究产品生产 - 配送集成调度问题 (integrated scheduling of production and delivery problem, ISPDP)。

近年来, ISPDP 引起了一些学者的关注, 但主要针对直达运输模式下运输时间确定的一般产品 ISPDP^[9-12]。对于易腐产品, 只有 Naso 等^[13]研究了多工厂、多工地情形下的预拌混凝土 ISPDP, 并设计了遗传算法和构造启发式算法对该问题进行求解。目前研究巡回配送模式下 ISPDP 的文献较少^[14,15], 特别是针对易腐性产品的研究更为鲜见。Farahani 等^[16]采用层次建模 (hierarchical modeling) 方法将快餐的 ISPDP 分解为生产调度和配送这两个子问题进行迭代求解, 其中前者采用基于分批计划的方法建立混合整数规划模型, 可用 CPLEX 软件求解。而后者可看成 VRPTW, 采用大规模邻域搜索算法求解。Chen 等^[17]研究了随机需求情况下具有常数变质率的易腐食品 (如蔬菜) ISPDP, 以供应商的收益最大为目标, 建立了非线性规划模型, 设计了以单纯形算法和插入算法为基础的迭代求解算法, 求得客户订单的最佳生产时间、生产量和配送车辆最优行驶路线。然而, 上述有关 ISPDP 的研究均未考虑物流配送过程中的时变性, 而在实际城市路网中, 由于受到交通管理、交通流量、交通事故、上下班高峰期等因素的影响, 车辆行驶速度总在不停变化^[18-20]。若不考虑这一因素, 可能会导致配送车辆在途等待时间延长而产生过多的价值损耗或者使产品交付延迟, 从而降低企业的盈利能力和服务水平。

为此, 本文在文献 [17] 的基础上, 进一步考虑路网交通状况的时变性, 研究时变路网环境下带时间窗的易腐食品生产 - 配送集成调度问题, 构建以配送车辆调用成本和行驶成本以及产品价值损耗总和最小为目标的优化模型, 以确定最佳的客户订单生产开始时间、配送车辆离开配送中心的时间及其配送路径。

2 问题描述

考虑多客户 (如超市、酒店) 订单情境下易腐食品加工配送中心的生产配送计划。加工配送中心根据自身能力接受客户订单后, 按照各订单的交付时间和需求数量组织生产, 然后及时安排车辆将完工的产品分批配送至客户。在此过程中, 影响订单盈利水平的主要因素包括: 产品生产和配送过程中的价值损耗以及物流

配送成本。因此,为了使总成本达到最低,需协同决策以下问题: 1) 如何安排各订单的生产顺序和开始时间,即订单的生产排序; 2) 各订单的配送批次及同批次订单的配送顺序,即配送车辆路径问题。

在产品配送过程中,考虑到城市路网交通的时变特性,车辆行驶速度会随时间变化,各路段的行驶时间取决于路段长度和车辆出发时间。用分段连续函数表示车辆行驶速度的时变性,一定时段内的行驶速度近似恒定,如图 1 所示。用 $T = \{T_0, T_1, T_2, \dots, T_L\}$ 表示时段集合, $[T_{l-1}, T_l]$ 为第 l 个时段, $v_{ij}(t)$ 表示 t 时刻路段 (i, j) 上的车辆行驶速度。为了使车辆行驶满足 FIFO 准则,可采用式(1)计算车辆在路段 (i, j) 上的行驶时间:

$$T_{ij}(d_i, t) = \begin{cases} t_{re} + T_{ij}(d_i + d_{re}, t_{re} + t), & d_i + d_{re} < d_{ij} \\ (d_{ij} - d_i)/v_{ij}(t), & d_i + d_{re} \geq d_{ij} \end{cases} \quad (1)$$

式中, $T_{ij}(d_i, t)$ 表示 t 时刻在路段 (i, j) 上已行驶了距离 d_i 的车辆继续驶完全程所需的时间; t_{re} 为 t 时刻所在时段的剩余时间,即 $t_{re} = T_l - t$ 且 $t \in [T_{l-1}, T_l]$; d_{ij} 表示路段 (i, j) 的长度; d_{re} 表示以 t 时刻的恒定速度驶完当前时段对应的行驶里程, $d_{re} = t_{re}v_{ij}(t)$ 。

3 模型建立

3.1 模型假设

建模之前做如下假设:

- 1) 易腐食品加工配送中心只有一条生产线,可混流生产客户订单指定的各种产品,但任意时刻只能生产一种产品,不考虑生产转换时间和成本^[17,21],同批次生产的订单按产品的易腐性由低到高进行连续生产;
- 2) 加工配送中心拥有有限数量的同型配送车辆,装载容量(重量或体积)为定值。不考虑配送车辆装载时间,从加工配送中心出发完成巡回配送任务后返回加工配送中心;
- 3) 每个客户的订单数量不大于车辆容量限制,且一次性交付,即不能分割配送。

基于以上假设 1) 和 2),为了使易腐食品的生产配送总成本最小,应尽量减少产品生产完成后等待配送的时间,故需满足如下最优化条件: 同一批次配送的订单应在同一批次生产,且同批次产品连续无空闲生产。

3.2 符号定义

用 $G = (N_0, A)$ 表示由加工配送中心和客户构成的配送网络图,其中 $N_0 = N \cup \{0\}$, 0 表示加工配送中心, $N = \{1, 2, \dots, n\}$ 表示客户点集合; $A = \{(i, j) | i, j \in N_0, i \neq j\}$ 为路段集合。加工配送中心的生产时间窗为 $[e_0, l_0]$; H 为加工配送中心可生产的产品类型集合,各产品的价值损耗系数(即单位产品在单位时间内的价值损耗)为 $\rho_1 \leq \rho_2 \leq \dots \leq \rho_{|H|}$; 单位产品 $h (h \in H)$ 的生产时间为 b_h ; V 表示容量为 Q 的同型配送车辆集合。客户 i 订购产品 h 的数量为 q_{ih} ,客户 i 的订单交付时间窗为 $[e_i, l_i]$ 。此外,定义如下参数和变量:

f 表示配送车辆启用的固定成本;

k 表示订单配送批次的编号,与配送车辆编号一致, $k \in V$;

$t_{k,h}^s$ 表示 k 批次订单中产品 h 的生产开始时刻;

$t_{k,h}^c$ 表示 k 批次订单中产品 h 的生产完工时刻;

t_k^d 表示车辆 k (即 k 批次订单)从加工配送中心出发的时刻;

q_i 表示客户 i 订购的产品数量, $q_i = \sum_{h \in H} q_{ih}$;

a_i 表示配送车辆抵达客户 i 的时刻;

t_i 表示客户 i 订单交付的开始时刻;

s_i 表示客户 i 订单交付的服务时间;

θ_{ih} 表示客户 i 订单交付时,单位 h 产品在生产和配送过程中的价值损耗;

w 表示配送车辆行驶单位里程的成本;

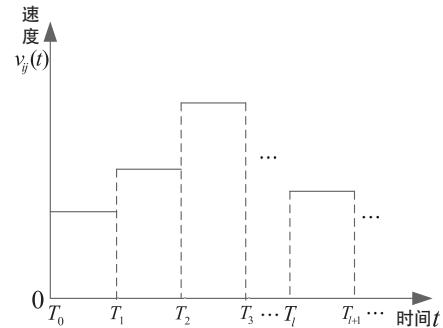


图 1 路段的时变行程速度

M 表示很大的正数;

$$\begin{aligned} x_{ik} &= \begin{cases} 1, \text{客户 } i \text{ 由车辆 } k \text{ 负责配送}, \\ 0, \text{否则,} \end{cases} \quad x_{ijk} = \begin{cases} 1, \text{车辆 } k \text{ 从客户 } i \text{ 到客户 } j, \\ 0, \text{否则,} \end{cases} \\ y_{kk'} &= \begin{cases} 1, \text{批次 } k \text{ 紧前于批次 } k' \text{ 加工,} \\ 0, \text{否则,} \end{cases} \quad y_{k0} = \begin{cases} 1, \text{批次 } k \text{ 为最后加工批次,} \\ 0, \text{否则.} \end{cases} \end{aligned}$$

3.3 数学模型

至此, 可建立时变路网环境下带时间窗的易腐食品生产 - 配送集成调度问题 (ISPDP) 优化模型如下:

$$\min Z = \sum_{i \in N} \sum_{h \in H} \theta_{ih} q_{ih} + w \sum_{i \in N} \sum_{j \in N} \sum_{k \in K} d_{ij} x_{ijk} + f \sum_{j \in N} \sum_{k \in K} x_{0jk} \quad (2)$$

$$\text{s.t.} \quad t_{k,h}^c = t_{k,h}^s + b_h \sum_{i \in N} x_{ik} q_{ih}, \quad \forall k \in V, \forall h \in H \quad (3)$$

$$t_{k,h}^s = t_{k,h-1}^c, \quad h \in H \setminus \{1\}, \quad \forall k \in V \quad (4)$$

$$t_{k,1}^s \geq e_0, \quad \forall k \in V \quad (5)$$

$$t_{k,|H|}^c \leq l_0, \quad \forall k \in V \quad (6)$$

$$t_{k,|H|}^c \leq t_k^d, \quad \forall k \in V \quad (7)$$

$$t_{k,|H|}^c \leq t_{k',1}^s + M(1 - y_{kk'}), \quad \forall k, k' \in V \quad (8)$$

$$\sum_{k' \in V \cup \{0\}} y_{kk'} = 1, \quad \forall k \in V \quad (9)$$

$$\sum_{i \in N_0} \sum_{k \in V} x_{ijk} = 1, \quad \forall j \in N \quad (10)$$

$$\sum_{j \in N_0} \sum_{k \in V} x_{ijk} = 1, \quad \forall i \in N \quad (11)$$

$$\sum_{i \in N_0} x_{ijk} - \sum_{i \in N_0} x_{jik} = 0, \quad \forall j \in N, \forall k \in V \quad (12)$$

$$\sum_{j \in N_0} x_{ijk} \leq x_{ik}, \quad \forall i \in N, \forall k \in V \quad (13)$$

$$\sum_{i \in N} q_i x_{ik} \leq Q, \quad \forall k \in V \quad (14)$$

$$\sum_{j \in N} x_{0jk} \leq |V|, \quad \forall k \in V \quad (15)$$

$$\sum_{k \in V} t_k^d x_{ik} < t_i \leq l_i, \quad \forall i \in N \quad (16)$$

$$t_i = \max\{a_i, e_i\}, \quad \forall i \in N \quad (17)$$

$$\{t_i + s_i + T_{ij}(0, t_i + s_i)\} - a_j \leq M \left(1 - \sum_{k \in V} x_{ijk}\right), \quad \forall i, j \in N_0 \quad (18)$$

$$\theta_{ih} = \rho_h \cdot \left[t_i - \frac{1}{2} \sum_{k \in V} (t_{k,h}^s + t_{k,h}^c) x_{ik} \right], \quad \forall i \in N, \forall h \in H \quad (19)$$

$$x_{ik}, x_{ijk}, y_{kk'}, y_{k0} \in \{0, 1\}, \quad \forall i, j \in N_0, \forall k, k' \in V \quad (20)$$

目标式 (2) 表示系统总成本最小, 包括订单交付时的产品价值损耗、配送车辆启用成本和行驶成本. 约束式 (3)~(9) 为订单生产过程约束, 其中式 (3) 表示批次产品连续生产所需的时间; 式 (4) 表示同批次产品连续无空闲生产; 式 (5) 和 (6) 表示所有批次的生产开始及结束时间不得违背加工配送中心的生产时间窗要求; 式 (7) 表示执行某批次配送任务的车辆只有在该批次所有订单产品生产完工后才能开始配送; 式 (8) 表示连续批次的生产开始时间约束; 式 (9) 表示连续批次的次序关系. 约束式 (10)~(18) 为订单配送过程约束, 其中式 (10) 和 (11) 表示所有客户的需求均得到满足; 式 (12) 表示送达客户的车辆完成订单交付后必须离

开该客户; 式(13)表示车辆仅对分配给它的批次订单客户进行配送; 式(14)表示每辆车配送的产品数量不超过车辆容量; 式(15)表示配送过程中使用的车辆数量不超过车辆总数; 式(16)表示客户的服务开始时刻满足其时间窗要求; 式(17)表示客户订单交付的开始时刻不早于该客户时间窗的开始时刻; 式(18)表示配送车辆抵达客户的时刻表达式, 同时消除了配送路线上可能存在的子回路; 式(19)表示订单交付时单位产品的价值损耗^[17]; 式(20)表示变量的取值范围.

3.4 模型分析

上述时变路网环境下带时间窗的易腐食品 ISPDp 优化模型可划分为三个问题, 即订单的批次分配 (batch allocation problem, BAP)、不同批次间的生产排序 (batch scheduling problem, BSP) 以及同批次订单的配送顺序, 即带时间窗的时变旅行商问题 (time-dependent traveling salesman problem with time-windows, TDT-SPTW). 一旦 BAP 和 BSP 确定后 (即确定了 x_{ik} 和 $ct_{k,|H|}$), 原问题即转化为求解 TDT-SPTW, 如图 2(a) 所示; 而当 BAP 和 TDT-SPTW 确定后, 即给定了带时间窗的时变车辆路径问题 (time-dependent vehicle routing problem with time-windows, TDVRPTW) 解决方案, 原问题即转化为求解 BSP, 如图 2(b) 所示. BSP 需确定各批次的生产顺序和开始时间, 由于决策变量是连续型变量, 不便于表示, 尤其是考虑了订单配送环节路网的时变特性, 使客户订单很难如文献 [16] 那样按照配送时间和客户时间窗进行组批, 另外即使生成了 BSP 调度方案, 也很难满足 TDVRPTW 中的强约束条件. 而 TDVRPTW 为典型的组合优化问题, 可以借鉴现有相关研究成果进行求解, 且生成的订单批次容易满足 BSP 的约束条件. 因此, 按照图 2(b) 所示将 ISPDp 分解为 TDVRPTW 和 BSP 这两个子问题进行迭代求解.

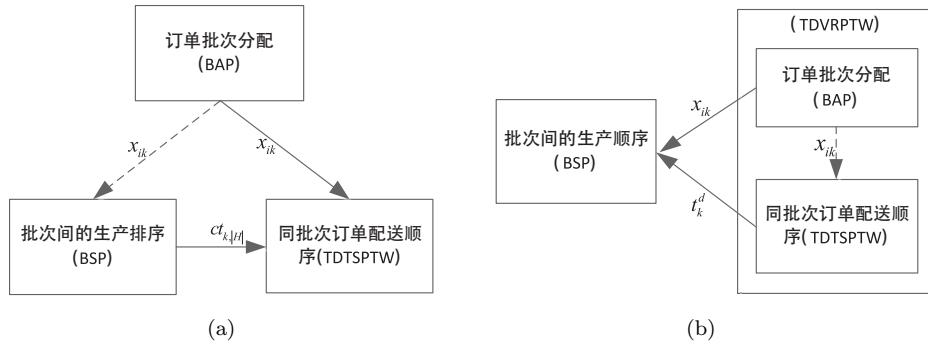


图 2 ISPDp 各子问题间的关系

客户 i 订单交付时单位产品 h 的价值损耗 θ_{ih} 可分解为两部分, 即生产完成后等待配送期间 ($t_{i,h}^1$) 的损失 θ_{ih}^1 和订单配送期间 ($t_{i,h}^2$) 的损失 θ_{ih}^2 , 其中 $\theta_{ih}^1 = \rho_h \sum_{k \in V} [t_k^d - \frac{1}{2}(t_{k,h}^s + t_{k,h}^c)] x_{ik}$, $\theta_{ih}^2 = \rho_h (t_i - \sum_{k \in V} t_k^d \cdot x_{ik})$. 相应地, 所有客户订单交付时总的价值损耗可分解为 $\sum_{i \in N} \sum_{h \in H} \theta_{ih}^1 q_{ih}$ 和 $\sum_{i \in N} \sum_{h \in H} \theta_{ih}^2 q_{ih}$. 因此, 可将 ISPDp 解耦为如下两个子问题:

TDVRPTW:

$$\begin{aligned} \min Z_1 &= \sum_{i \in N} \sum_{h \in H} \theta_{ih}^2 q_{ih} + w \sum_{i \in N} \sum_{j \in N} \sum_{k \in K} d_{ij} x_{ijk} + f \sum_{j \in N} \sum_{k \in K} x_{0jk} \\ \text{s.t.} & (10) \sim (18) \end{aligned} \quad (21)$$

$$\theta_{ih}^2 = \rho_h \left(t_i - \sum_{k \in V} t_k^d \cdot x_{ik} \right) \quad (22)$$

BSP:

$$\begin{aligned} \min Z_2 &= \sum_{i \in N} \sum_{h \in H} \theta_{ih}^1 q_{ih} \\ \text{s.t.} & (3) \sim (9) \end{aligned} \quad (23)$$

$$\theta_{ih}^1 = \rho_h \sum_{k \in V} \left[t_k^d - \frac{1}{2}(t_{k,h}^s + t_{k,h}^c) \right] x_{ik} \quad (24)$$

4 模型求解

原问题 ISPDP 分解后的 BSP 为混合 0-1 规划模型, 在订单批次较少时可用分支定价 (branch and price) 算法求解, 而 TDVRPTW 为 NP-hard 问题。原问题 ISPDP 比 TDVRPTW 更复杂, 也属于 NP-hard 问题, 规模较大时很难在有限的时间里获得精确解, 通常采用启发式算法求得问题的满意解。根据图 2(b) 所示, 可先求出 TDVRPTW 的解, 再求得对应的 BSP 的解, 然后利用目标函数式 (21) 和 (22) 对解的质量进行评价; 通过反馈调整 TDVRPTW 的解, 并反复迭代求解, 以不断提高解的质量。为了提高解空间的搜索效率, 可借助具有全局搜索能力的并行算法来求解 TDVRPTW。而遗传算法是借鉴生物进化规律, 通过对染色体的评价、选择、杂交和变异来逐步提高解的质量的一种并行随机搜索算法, 已广泛应用于组合优化问题, 且取得了良好的效果。因此, 本文设计了一种混合遗传算法 (hybrid genetic algorithm, HGA) 用于求解 ISPDP, 如图 3 所示。

4.1 染色体编码和种群初始化

为了表示总数有限的车辆实际启用数的不确定, 且方便染色体的交叉和变异, 本文采用两个长度均为 $|N|$ 的子串对染色体进行编码。子串 1 为 $1 \sim |N|$ 的不重复自然数, 表示客户的随机序列。子串 2 为路径编号序列 (即订单批次编号), 各基因位的取值为 $1 \sim |V|$ 的任意自然数, 对应表示子串 1 相同基因位的客户所属的路径。子串 1 中同一路径上的客户在序列中出现的次序, 表示该车辆访问客户的先后顺序。例如, 假设加工配送中心有 5 个待服务的客户, 配送车辆数为 2, 若生成染色体为 $(\underbrace{5 - 2 - 3 - 1 - 4}_{\text{客户子串 1}} | \underbrace{1 - 2 - 1 - 1 - 2}_{\text{客户子串 2}})$,

由于子串 2 的 1、3 和 4 位均为 1, 子串 1 对应位置为 5、3 和 1, 表示这些客户均由车辆 1 进行配送, 顺序为 5-3-1; 同理, 客户 2 和 4 均由车辆 2 进行配送, 顺序为 2-4。

设种群规模为 $popsize$, 按照上述方式随机产生 1 条染色体, 检验各子路径是否满足车辆容量约束; 若满足, 则将其加入种群; 否则, 重新生成 1 条染色体。直到生成的可行染色体数达到种群规模为止。

4.2 适应度函数

TDVRPTW 中含有车辆容量约束式 (14) 和客户时间窗约束式 (16), 为了扩大邻域搜索范围, 提高解的质量, 对不可行解进行惩罚, 将其分别转化为惩罚项加入目标函数 Z_1 , 使 Z_1 变为 Z'_1 , 即 $Z'_1 = Z_1 + \alpha_1 \cdot \sum_{k \in V} \max\{\sum_{i \in N} q_i x_{ik} - Q, 0\} + \alpha_2 \cdot \sum_{i \in N} \max\{a_i - l_i, 0\}$, 其中 α_1 和 α_2 为惩罚系数。同理, 在 BSP 中订单批次排序需满足约束式 (5)~(7), 对于任何不可行排序方案, 都可通过保证约束式 (6) 和 (7) 得到满足, 而使约束式 (5) 不能被满足, 因此将约束式 (5) 转化为目标函数 Z_2 的惩罚项, 使 Z_2 变为 Z'_2 , 即 $Z'_2 = Z_2 + \beta \cdot \max\{e_0 - t_{k,1}^s, 0\}$, β 为惩罚系数。

松弛上述约束条件后, 染色体对应 ISPDP 解的目标函数为 $Z' = Z'_1 + Z'_2$ 。对任意染色体 P , 目标函数值 Z' 的计算步骤如下:

Step 1 染色体 P 解码成路径集合 $R = \{r_1, r_2, \dots, r_m\}$, 初始路径标号为 $k = 1$;

Step 2 确定路径 r_1 的车辆最迟离开加工配送中心的时间 t_k^d , 使车辆配送满足客户时间窗且在途等待时间最少。其中 t_k^d 的计算步骤如下 (定义 $T_{ij}^{-1}(d_{ij}, T_{ij})$ 为 $T_{ij}(0, t)$ 的反函数, 则 $t_i^d = T_{ij}^{-1}(d_{ij}, t_j)$ 表示从 i 行驶至 j 的车辆到达 j 的时间为 t_j 时车辆最迟离开点 i 的时间);

Step 2.1 按路径逆序计算车辆尽可能在当前节点 j 的时间窗内进行服务时, 最迟离开上一节点 i 的时间, 即 $t_i^d = T_{ij}^{-1}(d_{ij}, t_j^l)$, t_j^l 为允许最迟到达当前节点的时间, $t_0^l = l_0$. t_i^a 的计算如下:

$$T_{ij}^{-1}(d_j, t) = \begin{cases} t_{pri} + T_{ij}^{-1}(d_j + d_{pri}, t - t_{pri}), & d_j + d_{pri} < d_{ij} \\ (d_{ij} - d_j)/v_{ij}(t), & d_j + d_{pri} \geq d_{ij} \end{cases} \quad (25)$$

式中, d_j 表示 t 时刻车辆距节点 j 的距离; t_{pri} 表示从当前时段开始到 t 时刻经过的时间, 即 $t_{pri} = t - T_{l-1}$

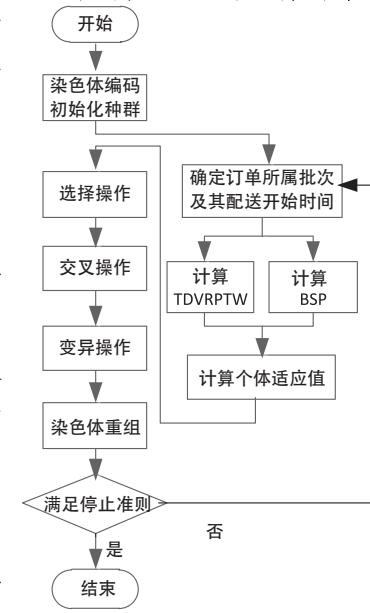


图 3 HGA 流程图

且 $t \in [T_{l-1}, T_l]$; d_{pri} 表示以当前 t 时刻的速度行驶 t_{pri} 时间可行驶的路程, 即 $d_{pri} = t_{pri}v_{ij}(t)$.

Step 2.2 记当前节点为 i , 计算最迟到达该点的时间 $t_i^l = \max\{\min\{l_i, t_i^d - s_i\}, e_i\}$;

Step 2.3 当前节点为路径起始节点 0 时, $t_k^d = t_0^l$, 转 Step 3; 否则, 转 Step 2.1;

Step 3 当 $k < m$ 时, $k = k + 1$, 转 Step 2; 否则, 可获得所有路径车辆最迟离开配送中心的时间 $(t_1^d, t_2^d, \dots, t_m^d)$, 转 Step 4;

Step 4 根据 $(t_1^d, t_2^d, \dots, t_m^d)$, 计算 BSP 的解. 为了提高运算效率, 在此采用如下启发式算法进行求解:

Step 4.1 计算所有订单批次生产过程所需的时间 $PT_k = \sum_{i \in N} \sum_{h \in H} b_h q_{ih} x_{ik}$, 以及生产完成后等待单位时间的价值损耗 $U_k = \sum_{i \in N} \sum_{h \in H} \rho_h q_{ih} x_{ik}$;

Step 4.2 对 t_k^d 值按升序进行排序, 假设排序结果为 $(t_{k_1}^d, t_{k_2}^d, \dots, t_{k_m}^d)$, 记当前批次 $k_i = k_m$, 计算批次 k_i 生产开始时间 $t_{k_i,1}^s = t_{k_i}^d - PT_{k_i}$, 以及生产完工时间 $t_{k_i,|H|}^c = t_{k_m}^d$;

Step 4.3 记 $k_i = k_{i-1}$, 当 $i = 0$ 时, 转 Step 4.4; 否则计算批次 k_i 的生产开始时间为 $t_{k_i,1}^s = \min\{t_{k_i}^d, t_{k_{i+1},1}^s\} - PT_{k_i}$, 以及生产完工时间为 $t_{k_i,|H|}^c = t_{k_i,1}^s + PT_{k_i}$, 循环执行该步骤;

Step 4.4 根据排序 $(t_{k_1}^d, t_{k_2}^d, \dots, t_{k_m}^d)$ 把所有订单生产完成时间分为 m 个区间, 按照 Step 4.1 中的 U_k 值, 把生产完成时间在同一区间内的订单批次按其 U_k 值由小到大的顺序进行生产重排序, 调整批次的生产开始和完成时间 $t_{k,1}^s, t_{k,|H|}^c$, 更新车辆离开加工配送中心时间 $t_k^d = t_{k,|H|}^c$;

Step 5 分别根据 t_k^d 和 $t_{k,1}^s$ 值计算 TDVRPTW 的目标函数值 Z'_1 和 Z'_2 , 则 ISPDp 的目标函数为 $Z' = Z'_1 + Z'_2$.

4.3 遗传算子

采用精英保留策略和锦标赛选择方式选择适应度高的染色体构成下一代种群.

为保持种群的多样性, 并防止产生误码, 对染色体的各子串分别进行交叉和变异操作. 子串 1 随机选择顺序交叉和部分匹配交叉操作, 以及逆序变异; 子串 2 采用均匀交叉操作, 并对 1~2 个基因座上的基因进行随机变异, 取值为已启用的车辆编号.

4.4 终止条件

令算法的最大进化代数为 gen_1 , 在连续 gen_2 代最优值无变化时, 算法终止.

5 算例分析

5.1 算例构造

目前尚无 ISPDp 标准测试算例, 为此本文借鉴 VRP 的 Solomon 标准测试算例中的 R201 算例构造包含 50 个客户点的算例. 设加工配送中心的坐标为 [17.5, 17.5], 工作时间窗为 [7 : 00, 17 : 00], 拥有 10 辆容量为 300 单位的同型车辆, 行驶成本为 1 元/km, 启用一辆车的固定成本为 150 元. 各客户间距在 20km 以内, 客户订单信息如表 1 所示, 客户需求的时间窗在 [8 : 00, 17 : 00] 内取值 (表 1 中时间窗取值以 8 : 00 为零时刻), 时间窗宽度为 0.5~1.0h, 各客户的服务时间为 0.15h. 假设客户订购共有 A、B 和 C 三种易腐食品, 对各产品的需求量服从 [0, 15] 的离散均匀分布, 单位产品的生产时间为 0.005h、0.006h 和 0.007h, 单位产品价值损耗系数分别为 0.5、1.0 和 1.5 元/h.

根据城市路网的早晚高峰时变特征, 将一天的工作时间分为三个时段, 即 [7 : 00, 10 : 00]、[10 : 00, 16 : 00] 和 [16 : 00, 19 : 00], 分别对应早高峰、正常行驶和晚高峰时段. 设车辆的日常平均行驶速度为 27km/h, 车辆在各时段内的行驶速度为 $[27(1 - \lambda), 27(1 + \lambda), 27(1 - \lambda)]$. 令 λ 分别取 [0.2, 0.3, 0.4], 构造出三种时变路段类型, 当 $\text{mod}(i + j, 3) = 0$ 时, 路段 (i, j) 为第 3 种类型, 其它情况按照 $\text{mod}(i + j, 3)$ 的值分配路段类型, 构造时变交通路网.

令 TDVRPTW 和 BSP 这两个子问题中约束转化的惩罚系数分别为 $\alpha_1 = 100$, $\alpha_2 = \beta = 150$. 混合遗传算法参数设置为: 种群规模 $popsize = 100$, 最大进化代数 $gen_1 = 1500$, 最优值不改变的最大允许代数 $gen_2 = 300$, 交叉概率 $p_c = 0.90$, 变异概率 $p_m = 0.08$. 采用 MatlabR2014a 语言编程实现本文算法, 在 Intel Core i7 3.6 GHz CPU、8GB RAM 计算机和 Windows 7 平台上运行.

表1 客户订单信息

客户	坐标 (km)	需求量 (A, B, C)	时间窗 (h)	客户	坐标 (km)	需求量 (A, B, C)	时间窗 (h)
1	(20.5, 24.5)	(14, 15, 13)	[6.4, 7]	26	(8.5, 17)	(6, 8, 12)	[7.1, 7.8]
2	(17.5, 8.5)	(10, 0, 2)	[1.3, 2.3]	27	(6, 12)	(12, 10, 5)	[2.9, 3.4]
3	(27.5, 22.5)	(7, 13, 15)	[4.7, 5.2]	28	(12, 29)	(11, 6, 4)	[2, 2.7]
4	(27.5, 10.0)	(10, 9, 8)	[6.5, 7.2]	29	(28, 19.5)	(10, 2, 8)	[5.7, 6.4]
5	(7.5, 15.0)	(8, 15, 11)	[0.3, 1.1]	30	(18.5, 23.5)	(7, 9, 11)	[2.1, 2.6]
6	(12.5, 15)	(10, 8, 15)	[3.7, 4.4]	31	(28.5, 28)	(13, 9, 11)	[7.1, 7.7]
7	(10, 25)	(8, 7, 4)	[3.2, 3.7]	32	(23.5, 8)	(3, 14, 4)	[0.3, 1.1]
8	(25, 17.5)	(2, 9, 2)	[2.1, 2.7]	33	(22, 8.5)	(13, 11, 10)	[2.8, 3.4]
9	(15, 12.5)	(2, 13, 3)	[6.8, 7.4]	34	(24.5, 5.5)	(8, 1, 6)	[2.1, 2.6]
10	(10, 20)	(11, 1, 3)	[3.4, 4.1]	35	(24.5, 21)	(13, 13, 1)	[2.6, 3.5]
11	(22.5, 10)	(14, 11, 10)	[2, 3]	36	(26.5, 21.5)	(9, 14, 12)	[7.8, 8.6]
12	(22.5, 15)	(1, 14, 11)	[5.3, 6]	37	(28.5, 24)	(3, 13, 9)	[3.5, 4.2]
13	(17.5, 20)	(8, 9, 8)	[1.5, 2.3]	38	(27.5, 27)	(11, 8, 2)	[3.1, 4.1]
14	(20.5, 18.5)	(8, 0, 3)	[1.6, 2.3]	39	(7.5, 23.5)	(12, 3, 2)	[1.5, 2.5]
15	(15.5, 26)	(6, 7, 2)	[2, 2.5]	40	(7, 18.5)	(12, 6, 8)	[1.2, 2.2]
16	(26.5, 26)	(11, 3, 13)	[0.4, 1.2]	41	(14, 9)	(2, 5, 11)	[3.1, 4]
17	(20, 12.5)	(11, 7, 3)	[3.2, 4.1]	42	(13, 26)	(2, 4, 1)	[2.5, 3.4]
18	(12, 6)	(7, 2, 5)	[1, 1.5]	43	(13, 17.5)	(10, 5, 1)	[6.8, 7.5]
19	(6.5, 26)	(11, 4, 1)	[7.4, 8.2]	44	(7.5, 9.5)	(3, 10, 12)	[6.9, 7.8]
20	(23.5, 23.5)	(0, 10, 1)	[5, 5.8]	45	(11, 11)	(7, 0, 14)	[0.2, 1.1]
21	(13.5, 21.5)	(6, 8, 11)	[1.8, 2.7]	46	(9, 12)	(2, 13, 10)	[7.3, 8.1]
22	(18.5, 15.5)	(4, 7, 9)	[3.7, 4.4]	47	(11, 13.5)	(8, 5, 2)	[5.5, 6.2]
23	(28.5, 14.5)	(3, 2, 1)	[5.8, 6.4]	48	(12.5, 10.5)	(14, 7, 2)	[5.5, 6.1]
24	(16, 6)	(13, 13, 14)	[4.1, 4.8]	49	(10, 13)	(9, 3, 5)	[3.3, 3.9]
25	(18, 13)	(6, 4, 8)	[8.1, 8.8]	50	(9, 9)	(12, 10, 10)	[8, 8.7]

为了清晰显示计算结果, 以表1中的前20个客户构成的算例为例, 算法运行5次, 平均运行时间为49.5s, 最优结果的收敛过程如图4所示, 可以看出算法在130代左右已收敛到最优解附近, 后又经过将近100代的搜索跳出局部最优, 表明算法具有良好的收敛性和搜索能力。最终求得的目标函数值为1547.4元, 产品价值损耗值为580.2元, 对应的生产配送方案如图5和表2所示。

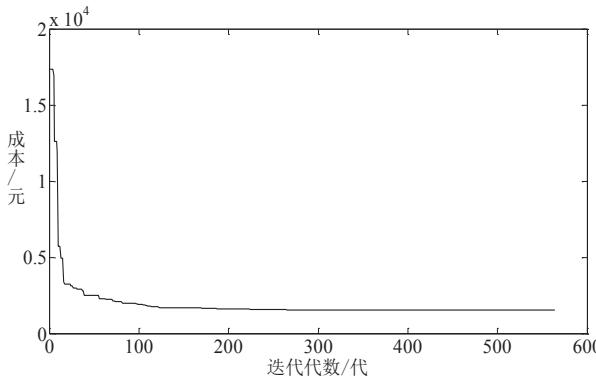


图4 算法收敛过程图

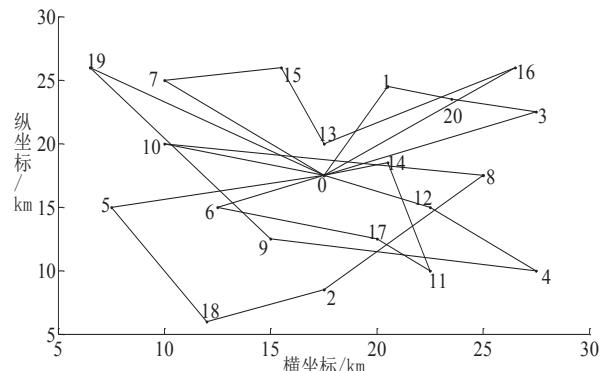


图5 最优生产 - 配送方案

表2 同批次订单生产 - 配送方案

批次	配送路线	生产开始时间 (min)	配送开始时间 (min)
1	0-3-20-1-0	298.4	330.6
2	0-12-4-9-19-0	368.0	399.3
3	0-14-11-17-6-0	152.7	188.0
4	0-5-18-2-8-10-0	47.7	78.5
5	0-16-13-15-7-0	78.5	109.1

表3 不同规模算例求解时间

算例规模	计算时间 (s)	算例规模	求解时间 (s)
10	26.3	60	282.4
20	48.6	70	332.5
30	108.0	80	405.3
40	156.2	90	526.9
50	231.1	100	589.4

5.2 计算结果

按照 5.1 节的算例构造过程把客户数量扩展到 100 个, 在算法进化代数和种群规模不变的情况下, 依次求解不同规模的算例, 计算时间如表 3 所示。可以看出, 随着客户数量的扩大, 算法运行时间并不随客户数量呈指数增长, 表明该算法具有较好的计算性能。

5.3 路网时变特性的影响分析

在此分析不考虑时变时(假设以车辆的日常平均行驶速度行驶)得到的生产 - 配送方案(简称“静态方案”)在时变路网中执行时客户服务时间窗的满足情况。按照 5.1 节所述方式构造三种时变强度依次增强的路网, 即 λ 依次取 $[0.1, 0.2, 0.3]$ 、 $[0.2, 0.3, 0.4]$ 和 $[0.3, 0.4, 0.5]$ 三组数据, 产生的路况对应定义为“路况 1”、“路况 2”、“路况 3”。由表 1 中的客户数据, 构造 10 个不同规模的算例(客户数从 5 开始依次增加 5, 直到 50), 计算各算例的静态方案在三种时变路况下执行时客户时间窗的满足情况, 结果如图 6 所示。

可以看出, 随着服务客户数量的增加, 三种类型路网下违背服务时间窗的客户数量整体均呈上升趋势, 其中出现少量波动是由于伴随客户数量的增加, 投入的配送车辆数也有所增加, 使得部分路径客户服务时间变得宽松。此外, 当服务相同数量的客户时, 路网的时变特性越强, 则违背服务时间窗的客户数量越多, 即违背服务时间窗的客户数量随路网时变强度呈上升趋势。

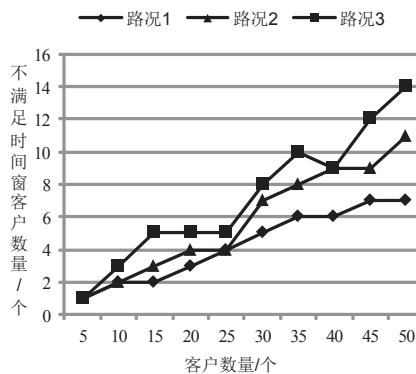


图 6 静态方案在时变路网中执行时客户时间窗不满足的数量与路况的关系

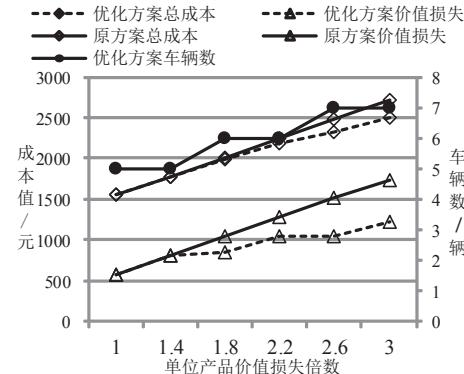


图 7 单位产品价值损耗系数的影响

5.4 产品价值损耗系数的影响分析

为了分析产品价值损耗系数对最优生产 - 配送方案的影响, 令单位产品的价值损耗系数增大到原来的 m 倍, 研究最优方案的成本变化情况。在 λ 取 $[0.2, 0.3, 0.4]$ 构成的时变路网中, 以服务表 1 中前 20 个客户为例, 结果如图 7 所示。可见, 随着单位产品价值损耗系数的增大, 系统总成本和产品价值损耗值均逐渐增大, 但与原方案(即 $m=1$ 时)相比, 成本增加幅度均明显减小, 主要原因在于启动的配送车辆数的增加, 减少了产品在途运输时间, 使产品价值损耗值减小幅度显著大于新投入的车辆启动成本和运输费用的增加。

因此, 对易腐食品生产配送企业而言, 应权衡增加启动的配送车辆数而增加的成本, 优化投入使用的车辆数量。不过, 易腐产品价值损耗不仅致显性成本增加, 而且从长远来看还会影响企业的声誉, 因此企业也要合理评估其权重, 可适当增加投入使用的配送车辆数来降低产品价值损耗, 提高客户服务水平。

6 结论

本文针对带时间窗的易腐食品生产 - 配送集成调度问题, 考虑路网交通的时变特性, 以客户订单交付时的价值损耗以及配送车辆启用成本和运输成本总和最小为目标, 建立了易腐食品生产 - 配送集成调度优化模型, 并根据该模型的特征设计了一种混合遗传算法。通过对不同规模的算例分析, 以及对不考虑时变时得到的“静态”生产 - 配送方案在不同时变强度的路网中进行测试, 结果表明客户时间窗违背数量随着路网时变强度的增强而增大。此外, 通过分析产品价值损耗系数的变化对总成本和产品价值损耗的影响, 结果表明适当增加投入配送的车辆数, 可以有效降低系统总成本和产品价值损耗。

本文考虑了加工配送中心只有一条生产线的情况, 进一步的研究可考虑有多条生产线的情形。此外, 针

对加工配送中心生产运营中可能面临的干扰因素, 如实时到达的订单、机器故障等现象, 以及配送途中可能出现的天气变化、交通管制和交通事故等干扰因素, 需对生产 - 配送方案进行实时调整, 都有待进一步研究。

参考文献

- [1] Arbib C, Pacciarelli D, Smriglio S. A three-dimensional matching model for perishable production scheduling[J]. *Discrete Applied Mathematics*, 1999, 92(1): 1–15.
- [2] Sana S, Chaudhuri K S. On a volume flexible production policy for a deteriorating item with time-dependent demand and shortages[J]. *Advanced Modeling and Optimization*, 2004, 6(1): 57–74.
- [3] Lütke Entrup M, Günther H O, Van Beek P, et al. Mixed-integer linear programming approaches to shelf-life-integrated planning and scheduling in yoghurt production[J]. *International Journal of Production Research*, 2005, 43(23): 5071–5100.
- [4] 周泓, 夏晓雯. 易变质产品的生产计划与作业排序集成优化研究 [J]. *计算机工程与应用*, 2010, 46(18): 192–195.
Zhou H, Xia X W. Study of integrated optimization of lot-sizing and scheduling problem based on perishable products[J]. *Computer Engineering and Applications*, 2010, 46(18): 192–195.
- [5] Tarantilis C D, Kiranoudis C T. A meta-heuristic algorithm for the efficient distribution of perishable foods[J]. *Journal of Food Engineering*, 2001, 50(1): 1–9.
- [6] Hsu C I, Hunga S F, Lia H C. Vehicle routing problem with time-windows for perishable food delivery[J]. *Journal of Food Engineering*, 2007, 80(2): 465–475.
- [7] Osvald A, Stirn L Z. A vehicle routing algorithm for the distribution of fresh vegetables and similar perishable food[J]. *Journal of Food Engineering*, 2008, 85(2): 285–295.
- [8] Amorim P, Almada-Lobo B. The impact of food perishability issues in the vehicle routing problem[J]. *Computers & Industrial Engineering*, 2014, 67(1): 223–233.
- [9] Chen Z L, Pundoor G. Order assignment and scheduling in a supply chain[J]. *Operations Research*, 2006, 54(3): 555–572.
- [10] Geismar H N, Dawande M, Sriskandarajah C. Pool-point distribution of zero-inventory products[J]. *Production and Operations Management*, 2011, 20(5): 737–753.
- [11] 蒋大奎, 李波. 基于混合禁忌搜索算法的供应链排序问题 [J]. *机械工程学报*, 2011, 47(20): 53–59.
Jiang D K, Li B. Supply chain scheduling based on hybrid taboo search algorithm[J]. *Journal of Mechanical Engineering*, 2011, 47(20): 53–59.
- [12] 李政道, 周泓. 基于限时送达服务的订单分派与生产运送调度的集成优化 [J]. *计算机集成制造系统*, 2014, 20(7): 1643–1653.
Li Z D, Zhou H. Integrated optimization of order assignment and joint production-transportation scheduling with commit-to-delivery mode[J]. *Computer Integrated Manufacturing Systems*, 2014, 20(7): 1643–1653.
- [13] Naso D, Surico M, Turchiano B. Scheduling production and distribution of rapidly perishable materials with hybrid gas[J]. *Studies in Computational Intelligence*, 2007, 49: 465–483.
- [14] Chen Z L, Vairaktarakis George L. Integrated scheduling of production and distribution operations[J]. *Management Science*, 2005, 51(4): 614–628.
- [15] Ullrich C A. Integrated machine scheduling and vehicle routing with time windows[J]. *European Journal of Operational Research*, 2013, 227(1): 152–165.
- [16] Farahani P, Grunow M, Günther H O. Integrated production and distribution planning for perishable food products[J]. *Flexible Services and Manufacturing Journal*, 2012, 24(1): 28–51.
- [17] Chen H K, Hsueh C F, Chang M S. Production scheduling and vehicle routing with time window for perishable food products[J]. *Computers and Operations Research*, 2009, 36(7): 2311–2319.
- [18] Haghani A, Jung S. A dynamic vehicle routing problem with time-dependent travel times[J]. *Computers & Operations Research*, 2005, 32(11): 2959–2986.
- [19] 李妍峰, 李军, 高自友. 大规模邻域搜索算法求解时变车辆调度问题 [J]. *管理科学学报*, 2012, 15(1): 22–32.
Li Y F, Li J, Gao Z Y. Very large scale neighborhood search algorithm for solving time dependent vehicle routing problem[J]. *Journal of Management Sciences in China*, 2012, 15(1): 22–32.
- [20] 李妍峰, 李军, 高自友. 动态规划启发式算法求解时变车辆调度问题 [J]. *系统工程理论与实践*, 2012, 32(8): 1712–1718.
Li Y F, Li J, Gao Z Y. Dynamic programming heuristics for solving time dependent vehicle routing problem[J]. *Systems Engineering — Theory & Practice*, 2012, 32(8): 1712–1718.
- [21] Low C, Chang C M, Li R K, et al. Coordination of production scheduling and delivery problems with heterogeneous fleet[J]. *International Journal of Production Economics*, 2014, 153(7): 139–148.