

供应网络中越库转运中心仓门分配问题研究

强 瑞¹, 缪朝炜², 吴为民²

(1. 福州大学管理学院, 福建 福州 350108; 2. 厦门大学管理学院, 福建 厦门 361005)

摘要: 本文探讨一种带有时间窗口的仓门分配问题——车辆在转运中心进行货物装卸作业时如何在其时间窗口限制内有效的分配有限的仓门资源, 以达到最佳运作效率。以往的研究结果表明该问题是强 NP 难题, 因此本文针对该问题的特殊结构, 提出一种新颖的整合了贪婪算法、遗传算法以及禁忌算法思想的混合启发式算法来有效的解决该问题。我们将该混合启发式算法与遗传算法、禁忌算法以及 CPLEX 这三种方式的求解效果进行对比, 其数值实验结果表明混合启发式算法在求解效果上有明显的优势。

关键词: 混合启发式算法; 越库; 转运; 仓门分配

中图分类号: F224.3 **文献标识码:** A **文章编号:** 1004-6062(2011)01-0209-07

0 引言

21 世纪随着企业规模的越来越大, 企业生产经营的产品种类越来越丰富, 转运中心对货物的处理变得愈加复杂与重要。转运中心作为一个物流结点, 担当着衔接、信息以及管理等功能。因此, 不少学者开始加入了对转运问题的研究之中^[1~3]。这其中, 对越库转运中心(crossdock)的研究是一个热点。越库转运中心作为一个快速高效的物流节点, 通常货物在其间的停留不超过 24 小时, 有时甚至短于 1 小时, 这就极大地提高了顾客响应速度。随着全球化进程中货物量的增长, 这些优势使得越库转运成为一种重要的物流策略。许多知名企业采取了越库转运模式以获得企业供应链的竞争优势, 比如沃尔玛^[4], UPS^[5], Toyota^[6]等。

对于越库转运中心的研究, 目前主要集中在多个越库网络优化以及单个越库中心内部运作两大方面。越库网络优化问题是传统的最小费用网络流问题的一个延伸^[7], 主要研究问题集中在越库网络物流配送的优化问题上, 如文献[8]研究了一个基于送货和分拣时间窗限制的越库网络问题, 以最小化运输成本与存储成本为目标。在该问题中, 含有多个越库节点, 考虑了由供应商、客户以及越库组成的一个三级物流配送网络, 并将运输与分拣作业的时间设定在某一范围内, 即给定一个时间窗。为了获得该配送网络的最优的运输配送方案, 文中建立了整数规划模型, 并证明了该问题为强 NP 难问题, 同时给出了求解该问题的贪婪算法和禁忌搜索算法。文献[9]研究了供应商和顾客带有时间窗口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。文献[10]将此类问题转化为一类机器排程问题, 并提出了相应的禁忌算法以及模拟退火等启发式算法来进行求解。

在越库中心内部运作研究方面, 研究的问题主要集中在如何最小化单个越库节点内部的运作成本或操作时间, 主要包括转运操作成本和劳动力成本, 以及转运操作时间等。比如, 文献[11]通过设计三阶段启发式方法以及遗传算法这两种方法来求解转运系统内部的车辆调度问题以及内部货物移动的成本最小化模型。文献[12]建立了车辆仓门分配数学规划模型来刻画越库转运中心内部运输距离最小化问题, 并设计了高效的分支定界算法。文献[2]研究了在越库中心内部运作中如何最小化其劳动力成本, 并考虑到了内部堵塞等因素对成本造成的影响, 但他们没有给出仓门安排以及车辆调度的具体策略。在近期的研究中, 文献[13]研究了一个带有临时仓储空间的越库转运系统, 并为此类车辆-仓门的安排问题建立了混合整数规划模型来解决最小化总体运作时间的问题, 不过现实应用中一些重要的限制条件比如转运中心物理布局, 仓库总体容量, 以及车辆的到达与离开时间都未予考虑。文献[3]针对带有时间窗口的越库中心仓门分配问题进行了研究, 文中考虑了一个越库在仓门数量有限的前提下, 针对到来的每辆已分配了服务时间窗口的卡车进行仓门的有效分配, 使得尽量多的卡车可以完成转运任务, 同时在越库内部的总操作时间最短, 具体考虑最小化由内部转运成本以及由于仓门有限而使得转运无法完成时所引发的惩罚成本构成的总成本, 并通过禁忌算法以及遗传算法进行求解, 但是模型中对特定参数的取值有一定限制, 只适用于部分问题。

本文研究的越库转运中心仓门分配模型是对文献[3]中模型的改进, 使之可以避免原模型对问题参数的特殊设置, 使得模型更加具有普遍性。该问题是一个强 NP 难题, 文中

收稿日期: 2010-04-09 修回日期: 2010-06-23

基金项目: 国家自然科学基金资助项目(70802052, 70901021); 教育部人文社科资助项目(07JC630047); 福建省高校杰出青年科研人才计划资助项目

作者简介: 强瑞(1963—), 男, 汉, 福建福州人, 福州大学管理学院副教授, 研究方向: 供应链管理以及质量控制研究。

我们将设计一种新颖的整合了贪婪算法、遗传算法以及禁忌算法思想的混合启发式算法来有效的解决该问题。我们将将该混合启发式算法与遗传算法、禁忌算法以及 CPLEX 这三种方式的求解效果进行对比, 来分析其在效率方面的优势。

1 越库转运中心仓门分配问题

传统的仓储中心管理主要集中在库存调配与管理问题上, 但是现在大力提倡的无论快速反应策略, 准时制还是敏捷制造, 讲究的都是时间的竞争, 企业与客户十分注重货物或者商品的周转速度, 所以越库转运中心的即时配送或者直接转运功能是一个关键因素。为了可以充分利用转运中心的有限资源, 每辆抵达的车辆将预先分配一个固定的服务时间窗口, 车辆要求在时间窗口的起始点到达, 结束点离开, 在时间窗口内进行装卸作业。由于转运中心的仓门数量有限, 仓门间的距离、每台车辆所转运的货物数量以及所分配的时间窗口都不尽相同, 因此如何将仓门合理的分配给这些车辆, 使得尽可能多的车辆可以进行转运作业并且货物在转运中心内部的运输路线尽可能短, 这都会影响到越库转运的效率以及成本。本文探讨的车辆与仓门分配问题有以下几点合理假设: 其一, 我们没有考虑车辆的装卸货时间以及其他缓冲时间, 因为我们对车辆设置了在转运中心的服务时间窗口, 这些时间很容易在时间窗口中进行再分配; 其二, 转运中心有容量限制, 也就是说任何时刻在转运仓库内的货物量不能超过最大容量; 其三, 对于时间窗口有重叠的任意两辆车不可占用同一个仓门; 其四, 任意两个仓门之间的转运时间与它们之间的距离成正比; 其五, 车辆的数量大大超过仓门的数量; 其六, 当由于条件的限制, 车辆间货物的转运无法实现时, 将产生处罚成本, 因为这些车辆需要重新排定以便完成它们的任务, 在这过程中, 将会耽搁时间, 使得货物无法按时送达指定地点, 因此设置一个处罚成本是合理的。

下面介绍一下模型中用到的参数:

- n : 车辆总数;
- m : 转运中心仓门的数量;
- a_i : 车辆 i 的到达时间;
- d_i : 车辆 i 的离开时间;
- t_{kl} : 货物从仓门 k 到仓门 l 的运行时间;
- f_{ij} : 车辆 i 转运给车辆 j 的货物量;
- c_{kl} : 货物从仓门 k 到仓门 l 的单位时间运行成本;
- p_{ij} : 车辆 i 无法将货物转运给车辆 j 时单位货物的处罚成本;
- C : 转运中心的容量, 也就是转运中心在任一时间点的货物最大容量限制;

x_{ij} : 如果车辆 i 离开时间不晚于车辆 j 的到达时间则取值为 1 (此时车辆 i 与车辆 j 时间窗口一定不重叠); 否则取值为 0。

决策变量定义如下:

$z_{ijkl} \in \{0, 1\}$, 取值为 1 则车辆 i 分配到仓门 k 并且车辆 j

分配到仓门 l ; 否则为 0。

$w_{ijkl} \in \{0, 1\}$, 取值为 1 则在仓门 k 的车辆 i 可以转运货物给在仓门 l 的车辆 j ; 否则为 0。

$y_{ik} \in \{0, 1\}$, 取值为 1 表示车辆 i 被安排在仓门 k , 否则为 0。

此外, 为了建模方便, 我们对 n 辆车的到达与离开的时间点进行排序, 让 $t_r (r=1, 2, \dots, 2n)$ 对应于这 $2n$ 个时间点, 使得 $t_1 \leq t_2 \leq \dots \leq t_{2n}$ 。

我们的目标值是要最小化总成本: 包括货物在转运中心内部的运行成本和未被转运的货物的处罚成本。

$$\text{Min} \sum_{k=1}^m \sum_{l=1}^m \sum_{i=1}^n \sum_{j=1}^n c_{kl} t_{kl} w_{ijkl} + \sum_{i=1}^n \left(\sum_{j=1}^n p_{ij} f_{ij} \left(1 - \sum_{k=1}^m \sum_{l=1}^m w_{ijkl} \right) \right)$$

$$\text{s. t.} \quad \sum_{k=1}^m y_{ik} \leq 1 \quad (1 \leq i \leq n) \quad (1)$$

$$z_{ijkl} \leq y_{ik} \quad (1 \leq i, j \leq n, 1 \leq k, l \leq m) \quad (2)$$

$$z_{ijkl} \leq y_{jl} \quad (1 \leq i, j \leq n, 1 \leq k, l \leq m) \quad (3)$$

$$y_{ik} + y_{jl} - 1 \leq z_{ijkl} \quad (1 \leq i, j \leq n, 1 \leq k, l \leq m) \quad (4)$$

$$w_{ijkl} \leq z_{ijkl} \quad (1 \leq i, j \leq n, 1 \leq k, l \leq m) \quad (5)$$

$$x_{ij} + x_{ji} \geq z_{ijik} \quad (1 \leq i, j \leq n, i \neq j, 1 \leq k, l \leq m) \quad (6)$$

$$\sum_{k=1}^m \sum_{l=1}^m \sum_{i \in (i: a_i < = t_r)} \sum_{j \in (j: d_j < = t_r)} f_{ij} w_{ijkl} - \sum_{k=1}^m \sum_{l=1}^m \sum_{i=1}^n \sum_{j=1}^n f_{ij} w_{ijkl} \leq C \quad (1 \leq r \leq 2n) \quad (7)$$

$$f_{ij} w_{ijkl} (d_j - a_i - t_{kl}) \geq 0 \quad (1 \leq i, j \leq n, 1 \leq k, l \leq m) \quad (8)$$

$$y_{ik}, z_{ijkl}, w_{ijkl} \in \{0, 1\} \quad (1 \leq i, j \leq n, 1 \leq k, l \leq m) \quad (9)$$

约束条件(1)保证每辆车最多只能分配到一个仓门。约束条件(2)~(5)联合定义了变量 y_{ik} , z_{ijkl} , w_{ijkl} 之间的逻辑关系, 其中(2)和(3)表示只有当车辆 i 分配给仓门 k 并且车辆 j 分配给仓门 l 的时候, 变量 z_{ijkl} 才能取值 1, 同时也限制了如果 z_{ijkl} 的取值为 1 时, y_{ik} 以及 y_{jl} 必须取值 1; 约束条件(4)表示当 z_{ijkl} 取值为 0 时, y_{ik} 与 y_{jl} 至少有一个取值为 0; 约束条件(5)表示只有当车辆 i 分配给仓门 k 并且车辆 j 分配给仓门 l 的时候, 这两辆车之间的转运才有可能; 约束条件(6)限定一个仓门不能同时被两辆车同时占用; 约束条件(7)是转运中心容量限制, 意味着在任何时间点 t_r , 在转运中心内的总货量不能超过库存最大容量限制 C ; 约束条件(8)表示如果在仓门 k 的车辆 i 要转运货物给在仓门位 l 的车辆 j 时, 车辆 i 的到达时间加上两仓门之间的转运时间要小于于车辆 j 的离开时刻点, 即车辆 j 尚未离开时转运才能发生。

在文献[3]建立的模型中, 参数 f_{ij} 与 z_{ijkl} 之间存在的逻辑关系, 使得 f_{ij} 的取值要满足特殊条件模型才成立。本文中此改进的模型最大的特点就是由于引入了决策变量 w_{ijkl} , 使得变量 z_{ijkl} 与 f_{ij} 之间无逻辑关系, 避免了原模型对参数 f_{ij} 设置上的特殊要求, 对各种取值的 f_{ij} , 该模型都适用, 使模型更加具有一般性。

由于该问题已经在文献[3]中证明是一个强 NP 难题，此外我们发现随着问题规模的增加，该整数规模模型中决策变量以及约束条件的数量呈几何规模递增。因此，要有效率的处理实际应用中的大规模问题，我们有必要借助与启发式算法。

2 混合启发式算法设计

在我们的问题中，由于车辆时间窗口以及转运中心容量等限制会使得某辆车的仓门安排会影响到其他车辆的安排以及相互之间的货物转运时间。首先，由于处罚成本相对内部转运成本要高，因此我们要尽可能将车辆安排到有限的仓门上，而分配到仓门的车辆数量与其时间窗口密切相关。为此，我们针对时间窗口采取贪婪算法对车辆依次进行仓门安排来生成一组初始解。其次，遗传算法^[14]与禁忌算法^[15]常用来解决复杂的组合优化问题。为了克服遗传算法的局部搜索缺陷以及禁忌算法的全局搜索能力问题，我们综合了这两种算法的优势，采取在遗传算法最优解基础上进行禁忌邻域搜索策略，因此我们将该算法称为混合启发式算法。图1给出了混合启发式算法框架。

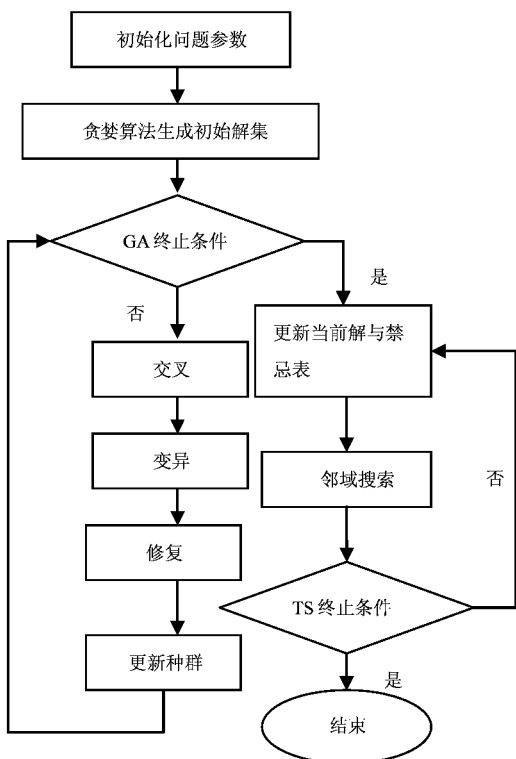


图 1 混合启发式算法框架

2.1 编码方案

在该问题中，我们采用自然数编码，我们设计一个位串长度为 n 的染色体序列 (s_1, s_2, \dots, s_n) ，每个基因位代表着每辆车分配到的仓门的情况。基因位的取值范围为 $\{0, 1, 2, \dots, m\}$ ，其中若基因位 j $(1 \leq j \leq n)$ 取值为 k $(1 \leq k \leq m)$ 则表示车辆 j 分配到了仓门 k ；若取值为 0 表示没有分配到仓门。

2.2 初始种群

我们采用贪婪算法生成初始群体。按照车辆的随机顺

序依次安排给仓门，在满足同一仓门内的车辆时间窗口不发生冲突的前提下尽可能多的安排车辆，对无法安排的车辆对应的基因位赋值 0。贪婪算法生成初始群体的具体步骤如下：

- (1) 随机选取车辆 i ，为其安排仓门 k ；
- (2) 再从未安排的车辆中随机选取车辆 j ，先寻找在该车时间窗口内空闲但之前已被分配的仓门，若存在此类仓门，则将该仓门分配给该车，否则寻找未被分配过的仓门并将其分配给该车；如果由于时间冲突或容量限制，以上两类仓门都未找到，则不予分配，赋值为 0；

(3) 按照步骤 (2) 依次为剩下的车辆分配仓门，直到所有车辆均安排完毕。

2.3 遗传算法

2.3.1 交叉算子

有效交叉算子不仅取决于问题的结构，种群大小等因素，还取决于当前种群的状况。因此，我们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，以一定概率进行交叉运算。由于单一交叉算子具有其局限性，因此我们延用了 [3] 中的单点交叉以及双点交叉算子。

在单点交叉中，首先随机选择一个交叉点。第一个子个体由父个体的第一部分与母个体的第二部分组成。同理，第二个子个体由父个体的第二部分与母个体的第一部分组成。如图 2 所示。

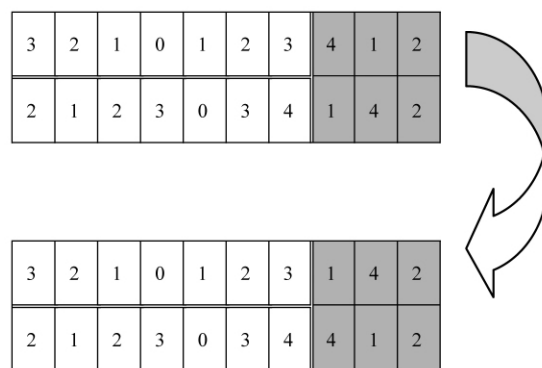


图 2 单点交叉算子

在双点交叉中，随机生成两个交叉点。双亲染色体中两个交叉点之间的部分进行交换生成两个子个体。交叉过程如图 3 所示。

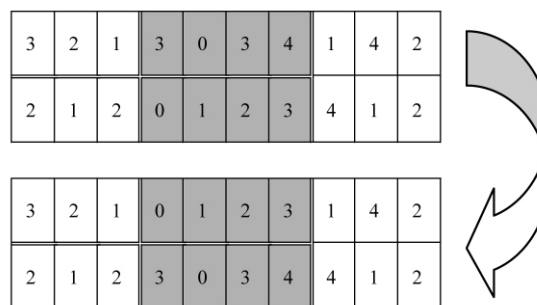


图 3 双点交叉算子

2.3.2 变异算子

我们沿用 [3] 中的“互换变异”作为我们的变异策略。即以一定概率随机选择两个变异点,然后把这两个变异点上的值进行互换。如图 4 所示。

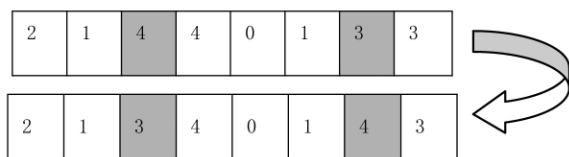


图 4 变异算子

2.3.3 修复策略

在交叉以及变异过程中,均会产生不可行解,因此为了保证子代的可行性,我们需要采用修复策略。事实上,由于父代都是可行解,因此双亲个体产生的子代的不可行区域并非全部。对交叉算子而言,不可行的原因是子代继承的两个父代基因片段之间产生冲突(比如时间窗口的冲突或者不满足容量限制条件)。而对变异算子而言,不可行的原因是子代的变异点对应的新的仓门分配方案同样产生了时间窗口的冲突或者违背了容量限制的条件。基于以上原因,我们对子代进行修复时,针对发生不可行问题的基因,采取替换策略,即类似于初始种群生成步骤(2)的方法,先寻找可分配的仓门,如果未找到,则赋值 0。通过该修复策略,可以保证所有生成的子代的可行性。

2.3.4 遗传算法步骤

遗传算法具体步骤如下,具体参数设置见本文数值实验部分:

- (1) 设定遗传算法的参数,并使用贪婪算法生成初始种群;
- (2) 对种群进行交叉运算,反复此过程,直到生成要求数量的子代;
- (3) 对子代进行变异运算;
- (4) 对子代中不可行的解进行修复处理;
- (5) 在父代与子代的合集中选择适应值(即对应的目标函数)最佳的初始种群数量的个体作为新的父代,并更新当前最优解;
- (6) 如果迭代次数达到规定世代数,则转到下一步,否则迭代数加 1,并转到(2);
- (7) 将当前最优解作为禁忌算法的初始解,终止遗传算法。

2.4 禁忌算法

在通过遗传算法对初始群体的全局寻优后,我们将在其基础上通过禁忌算法进行进一步的局部寻优。通过大量数值模拟实验,我们发现将禁忌表长度设置为 $n+30$,算法的总体效率较高,虽然耗时比通常采用的表长 \sqrt{n} 相对较多,但是最终解的质量有了很大提高。此外,在禁忌规则中我们采用了目标值原则,即对目标值最好的解进行禁忌操作。下面我们重点介绍领域搜索以及禁忌算法步骤

2.4.1 邻域搜索

下面分别介绍两种方式来构建领域。

(1) 优先安排方式:将某辆未安排仓门的车辆以及其余任一车辆的仓门进行重新安排。该方式有助于生成的解减少处罚成本,但是新生成的解有可能不可行,因此要采用上面所提到的修复策略来获得可行性。图 5 给出了一个例子,对车辆 1 和车辆 8 重新安排仓门,导致原先在仓门 3 的车辆 4 无法安排,经过搜寻无法为其重新安排后,将其赋值为 0。

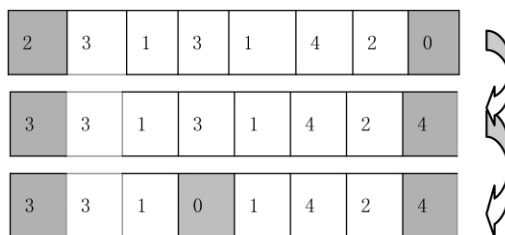


图 5 优先安排方式

(2) 区间交换方式:将任意两个仓门所分配的所有车辆进行互换。即将安排给仓门 k 的所有车辆都重新安排给仓门 l,而将安排给仓门 l 的所有车辆都安排给仓门 k。该方式在 [3] 中使用过,由于其不影响解的可行性的特性,我们也采用这一方式。如图 6 所示。

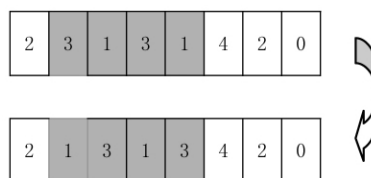


图 6 区间交换方式

2.4.2 禁忌算法步骤

禁忌算法具体步骤如下:

- (1) 设定禁忌算法的参数,将遗传算法的最优解作为禁忌算法的当前解以及当前最优解;
- (2) 通过以上领域搜索的两种方式生成当前解的领域;
- (3) 在领域中,寻找目标值最小的那个解,如果该解好于当前最优解,则加入禁忌表中,并将其作为新的当前最优解以及当前解。否则,将该解作为禁忌算法的当前解,更新禁忌表;
- (4) 如果满足以下终止条件之一:第一,当前最优解在一定迭代次数内未有改善;第二,达到最大迭代次数。则终止禁忌算法,输出当前最优解,否则转到(2)。

3 数值实验

由于针对该问题,国际上尚无标准测试数据,因此我们根据实际情况随机产生模型的参数,通过大量的数值实验,将混合启发式算法的实验结果与 ILOG CPLEX,以及单独使用遗传算法以及禁忌算法的求解结果进行对比分析,来检测该混合启发式算法的有效性。实验在 Intel(R) Pentium(R) Dual CPU T2410 2.00GHz 和 2GB 内存的计算机上进行。

3.1 数据生成与参数设定

我们研究的越库转运中心是一个矩形结构,仓门平均分布在两条长边上,这样的设计比较常见。此外,我们假设同

一条边上的两个相邻仓门距离为 1 个单位,而分属于两条边上的相对的两个仓门的距离为 3。不失一般性,我们假设转运中心内的叉车只能在平面上做垂直或者水平移动。这样我们可以很容易计算出其它各个仓门之间的距离。表 1 是其模型参数的随机产生范围。表 2 为混合算法的参数设定。

3.2 实验结果与分析

我们根据车辆与仓门的数量将数值实验的算例分为大、中、小规模三大类,每一类我们又按照车辆与仓门的数量分成 6 组,每组各做 4 次随机实验。表 3 ~ 表 5 给出了每个小组 4 次随机实验分别采用混合启发式算法、CPLEX、遗传算法以及禁忌算法来求解得到的平均实验结果,包括平均目标值以及平均求解时间(以秒为单位)。其中 $n \times m$ 表示该问题规模为 n 车辆 m 个仓门,这 18 组实验规模从 10×4 到 40×8 。从表 3 可以看出,对于小规模算例,混合启发式算法与其他三种方法在解的质量上无明显差异,在求解时间方面与遗传算法相当,稍逊于禁忌算法,但是比 CPLEX 有明显优势。从表 4 上看,对于中等规模算例,混合启发式算法在解的质量上表现最为突出,耗时方面的对比结果与小规模实验

的对比结果一致。从表 5 上看,对于大规模算例,虽然混合启发式算法在求解时间上逊色与禁忌算法与遗传算法,但是仍然明显好于 CPLEX,此外混合启发式算法的解的质量与禁忌算法同样出色。总而言之,混合启发式算法在求解质量上的表现总体最佳,在耗时上与遗传算法相当,逊色与禁忌算法,明显优于 CPLEX。

4 结论

本文研究了带有时间窗口以及仓库容量限制的仓门分配问题,在文献[3]的基础上对模型进行了改进,使得模型更具有普遍性,并设计了一种新颖的混合启发式算法来解决该问题,比原来文献中的方法更加有效。数值实验结果表明该混合启发式算法在解的质量方面表现优异。该算法可以有效地解决此类问题,特别是车辆与仓门数量较多的情况下,优化求解的优势比较明显。今后的研究将致力与此类问题模型的优化,建立更加简洁的数学模型,使得算法的性能可以进一步提高。

表 1 模型数据范围

模型参数	a_i	d_i	f_{ij}	c_{kj}	p_{ij}	b	C
范围	[1, 70n/m]	$a_i + [45, 74]$	[6, 60]	[1, 4]	[8, 12]	[0.6, 0.9]	$b \sum_{1 \leq i, j \leq n} f_{ij}$

表 2 混合启发式算法的参数设定

参数	迭代次数	最优解无法改善的迭代次数	交叉概率	变异概率	禁忌表长	种群数量
数值	1000	60	0.8	0.01	n + 30	300

表 3 小规模问题数值实验结果对比

规模 $n \times m$	10×4	12×4	14×4	16×4	18×4	12×6
禁忌算法	11111	17419	14962.5	21179	29301.5	5785.4
禁忌算法耗时	1.14	2.3	4.516	7.8	13	5.6
遗传算法	11110.9	17419	14941.3	21208	29334	5783.8
遗传算法耗时	1.141	20.78	32.5	45.6	60.4	43.3
混合启发式算法	11110.9	17419	14941.2	21179	29301.5	5774.8
混合启发式算法耗时	11.9	23.1	37	53.3	73.5	48.9
CPLEX	11111	17419	14941	21179.42	29306.6	5820.4
CPLEX 耗时	401.34	986.45	7028.2	> 8000	> 8000	> 8000

表 4 中等规模问题数值实验结果对比

规模 $n \times m$	14×6	16×6	18×6	20×6	24×6	25×6
禁忌算法	8338.7	23957	23238.8	37003	26655	40230
禁忌算法耗时	11.5	22.8	37	62.7	138.4	178.1
遗传算法	8335.9	23956	22880	37082	27384	40473
遗传算法耗时	67.5	98.2	139	190.1	325	366.9
混合启发式算法	8332	23891	22860	37035	26648	40127
混合启发式算法耗时	79	120.9	176.1	252.5	461.4	541.9
CPLEX	8366.7	24607.6	23563.9	37232.7	36951.8	41378.4
CPLEX 耗时	> 8000	> 8000	> 8000	> 8000	> 8000	> 8000

表5 大规模问题数值实验结果对比

规模 n × m	30 × 6	20 × 8	25 × 8	30 × 8	35 × 8	40 × 8
禁忌算法	50598.4	45256	50107.5	48355	107348	53409
禁忌算法耗时	435.8	126.1	307.9	610.9	1270.1	2697
遗传算法	51840	46504	52915.2	50227	108691	56382
遗传算法耗时	611	315.7	607.6	1046	1270.1	2507.4
混合启发式算法	50555	45062	51040.4	48327	107519	54936
混合启发式算法耗时	1047.2	442.4	915.5	1657	1628.9	5233.6
CPLEX	53746.3	46008.3	54439.7	51500.3	278884	339897
CPLEX 耗时	>8000	>8000	>8000	>8000	>8000	>8000

参 考 文 献

- [1] Napolitano M. Making the Move to Cross Docking [M]. Warehousing Education and Research Council , 2000.
- [2] Bartholdi J , Gue K. Reducing labor costs in an LTL cross-docking terminal [J]. Operations Research , 2000 , 48 , 823 ~ 832.
- [3] Miao ZA. Lim H. Ma. , Truck dock assignment problem with operational time constraint within transshipments [J]. European Journal of Operational Research , 2009 , 192 , 105 ~ 115.
- [4] Stalk GP. Evans LE. Shulman. Competing on capabilities: the new role of corporate strategy [J], Harvard Business Review , 1992 , 70 (2) , 57 ~ 69.
- [5] Forger G. UPS starts world's premiere cross-docking operation [J]. Modern Material Handling ; 1995 , 36 ~ 38.
- [6] Witt CE. Transshipping: Concepts demand choice [J], Material Handling Engineering , 1998 , 53 (7) , 44 ~ 49.
- [7] Aronson J. A survey on dynamic network flows [J]. Annals of Operations Research , 1989 , 20 , 1 ~ 66.
- [8] Chen P , Guo YS. , Lim , A. , et al. Multiple crossdocks with inventory and time windows [J]. Computers & Operations Research , 2006 , 33 (1) 43 ~ 63.
- [9] Lim A , Miao Z , Rodrigues B , et al. Transshipment through transshipments with inventory and time windows [J]. Naval Research Logistics , 2005 , 52 (8) . 724 ~ 733.
- [10] Li Y , Lim A , Rodrigues B. Crossdocking: JIT scheduling with time windows [J], Journal of the Operational Research Society , 2004 , 55 (12) , 1342 ~ 135.
- [11] Oh Y , H. Hwang , C. Cha , et al. A dock-door assignment problem for the Korean mail distribution center [J]. Computers & Industrial Engineering , 2006 , 288 ~ 296.
- [12] Tsui L , Chang C. An optimal solution to a dock door assignment problem [J]. Computers & Industrial Engineering , 1992 , 23 (1-4) , 283 ~ 286.
- [13] Yu W , Pj Egbelu. Scheduling of inbound and outbound trucks in cross docking systems with temporary storage [J], European Journal of Operational Research , 2008 , 184 (1) , 377 ~ 396.
- [14] Davis L. Handbook of Genetic Algorithms [M]. Van Nostrand Reinhold , New York , 1991.
- [15] Glover F , Laguna M. Tabu Search [M]. Kluwer Academic Publishers , Dordrecht , The Netherlands , 1998.

Truck Dock Assignment Problem for Crossdocks in Supply Networks

QIANG Rui¹ , MIAO Zhao-wei² , WU Wei-min²

- (1. School of Management , Fuzhou University , Fuzhou 350108 , China;
2. School of Management , Xiamen University , Xiamen 361005 , China)

Abstract: An effective supply chain can help companies achieve international competitiveness. Many companies have tried to optimize their distribution activities in their supply chains by adopting the cross-docking logistics strategy. This strategy can effectively integrate inventory management and distribution activities. Cross-docking is a logistics practice of unloading materials from incoming transportation vehicles (e. g. truck , trailer or rail car) and loading these materials directly into outgoing transportation vehicles without holding storage in between. Cross-docking can significantly reduce inventory levels , inventory costs and cargo loss rates , speed up cash flows , and increase response time to market demand. Consequently , an effective cross-docking strategy can improve customer satisfaction and have significant , lasting impacts on the operational efficiency of supply chains.

This study investigates cross-docking problems limited with a set of constraints. An Integer Programming Model is setup to analyze cross-docking problems. A Hybrid Meta-heuristic Algorithm (HMA) is designed to help solve these problems. A few constraints are

constructed in order to effectively use these models to solve cross-docking problems. Each truck has a service time window, only within which the truck can occupy a dock. The number of docks is limited. The distance between docks varies. The number of cargos and the time window for each truck are different. The shipping distance between docks needs to be kept at the minimum level. The goals of these models are to assign docks to trucks so that cargos can be processed as efficiently as possible. Constraints, such as capacity and time window, will cause some cargos not to be shipped out. Resultant delayed deliveries can increase total logistics costs. An effective assignment of trucks can have impact on cross-docking operational efficiency and cost. The objective of this study is to design a cross-docking assignment strategy so that the optimal efficiency of cross-docking operations can be achieved while meeting operation constraints.

We first describe cross-docking assignment problems for trucks, constraints and assumptions. We then introduce an Integer Programming Model, its notations and decision variables. Second, we introduce the idea of designing a HMA and apply a Genetic Algorithm (GA) to locate a near optimal solution. Neighborhood Search (NS) and Tabu Search (TS) are applied in order to solve problems and improve the quality of solutions to these problems. After doing so, we explain how HMA works, including single-point crossover operator, two-point crossover operator, exchange mutation operator, repair strategy, neighborhood search strategy, etc.

Numerical examples are used to compare the performance of HMA, CPLEX, GA and TS with respect to the efficiency of HMA. We not only show the setting of all parameters, but also develop three categories of experiments including large, medium and small scale instances. The results show that all of these four methods have similar performance for small scale problems in conditions that the time consumed of HMA is as much as GA, a little bit longer than TS, but shorter than CPLEX. For medium scale problems, the solution quality of HMA outperforms the others, and the ranking of time consumed is as the same as that of small scale problems. For large scale problems, HMA is slower than TS and GA, but still much faster than CPLEX. Moreover, both HMA and TS have the highest solution quality. In general, HMA has the best performance in the solution quality, and the time consumed by it is as much as GA, longer than TS, but much shorter than CPLEX.

In summary, this paper investigates cross-docking assignment problems with the constraints of limited time window and cross-docking capacity constraints. Our proposed model develops an efficient HMA to overcome the weakness of previous cross-docking models. The numerical experiments show that HMA has the best performance in the solution quality. The finding indicates that this HMA is an efficient way to solve cross-docking problems, especially for a large numbers of trucks and docks.

Key words: hybrid meta-heuristic; crossdock; transshipment; dock assignment

中文编辑：杜 健；英文编辑：Charlie C. Chen