

一种新的集卡动态调度模型及算法

李尤丰¹, 李勤丰^{2,3}, 刘玉霞⁴, 周晓峰³

- (1. 金陵科技学院信息技术学院, 江苏 南京 211169)
- (2. 金陵科技学院公共基础课部, 江苏 南京 211169)
- (3. 河海大学计算机与信息学院, 江苏 南京 211100)
- (4. 解放军国际关系学院基础部, 江苏 南京 210039)

[摘要] 港口物流环节中,集卡的调度直接影响着港口的工作效率.目前,对集卡装和卸同时进行的调度模型和算法鲜有研究.本文以一艘船待装集装箱、一艘船待卸集装箱、装卸同时进行为例,提出一种动态优化模型,并用遗传算法求解.但其不同于一般的遗传算法,这里创造性地引入 N_6 邻域方法来减少解空间搜索范围,提高遗传算法效率;设计了多种交叉操作和变异操作,来增加解的多样性,避免算法陷入早熟;设计了变异概率 P ,保证了算法的收敛性.实验结果验证了该模型和算法的有效性,从而有效提高集卡运输效率.

[关键词] 改进混合遗传算法,集卡调度,动态优化, N_6 邻域

[中图分类号] TP301.6; O29 [文献标志码] A [文章编号] 1001-4616(2014)01-0104-08

Dynamic Dispatch Model of Container Trucks Based on Hybrid Genetic Algorithm

Li Youfeng¹, Li Qinfeng^{2,3}, Liu Yuxia⁴, Zhou Xiaofeng³

- (1. College of Information Technology, Jinling Institute of Technology, Nanjing 211169, China)
- (2. Department of Basic Course, Jinling Institute of Technology, Nanjing 211169, China)
- (3. College of Information and Computer Engineering, Hohai University, Nanjing 211100, China)
- (4. Department of Basic Course, The University of International Relations of Chinese PLA, Nanjing 210039, China)

Abstract: The truck dispatching is an important way to improve the efficiency of port logistics. In order to meet the requirements of rapid loading or unloading containers, loading and unloading of container truck scheduling simultaneously emerged. However, the problem is seldom considered. Here, in case two boats, an improved model about loading and unloading of trucks at the same time was given. And based on it, a new GA was presented. In order to reduce the time during seeking the best solution, the method of N_6 neighborhood was proposed to improve the efficiency of GA. To increase the diversity and avoid the premature convergence, a combination of several crossover and mutation operations was introduced. The convergence was assured by the mutation probability P . The experimental results show the correctness and effectiveness of the hybrid genetic algorithm.

Key words: hybrid genetic algorithm, truck scheduling, dynamic optimization, N_6 neighborhood

龙门吊和集卡是港口物流的重要设备,对集卡的调度直接影响码头的整体作业效率.在实际操作中,由于各种原因,集卡作业不均衡,经常会出现龙门吊待机或集卡排队等现象.如何在不增加基础设施投入的前提下,提高集卡调度效率成为提高港口竞争力的关键因素.

目前,集卡调度方法分为两种:静态调度法和动态调度法.静态调度法是基于“作业线”的调度法,即每辆集卡只为某个固定的装卸桥服务.当集卡将所卸集装箱从岸边运到堆场后,空驶回先前的装卸桥,继续下一个卸船集装箱的运送.实际中,由于这种调度方法易于操作和实现,多数港口、码头采用静态调度法.多数研究也是针对集卡静态调度基础上的装船、卸船研究^[1-4].动态调度法是基于“作业面”的调度法,

收稿日期:2013-10-20.

基金项目:江苏省现代教育技术研究课题(2011-R-19470).

通讯联系人:李尤丰,讲师,硕士,研究方向:图像分析和大型软件系统架构、管理. E-mail: fllyf@126.com

即集卡不一定只为某个装卸桥服务,而是可以根据现场情况在不同作业线上动态分配.静态调度法中,集卡空驶严重,使集卡运输效率降低,从而使整个码头所需的集卡数量增加,进而影响码头内道路交通.因此,动态调度集卡成为提高码头效率的重要途径.基于作业面的装卸同时进行的集卡调度情况非常复杂,目前对装卸同时进行的集卡动态调度研究^[5]较少.本文的目标就是建立一个装卸同时进行的动态集卡调度模型,使集卡运输效率提高,从而实现快速装卸,来减少作业面上的作业总时间.

文献[5]也给出了装卸同时进行的集卡动态调度模型,但它以码头船只和堆场外围为集卡行走路线.本文在此基础上作了改进,充分利用堆场之间的道路,减少集卡空驶距离.遗传算法是可用于复杂系统优化计算的启发式算法,鲁棒性好,近年来被广泛地应用于 NP 问题^[6-8].本文就是在标准遗传算法的基础上,结合 N6 邻域法(见 2.1 节),来降低解空间的搜索范围,提高算法效率.同时构造变异概率 P ,结合几种不同的变异操作和交叉操作来保证种群多样性,避免算法收敛于局部解,变异概率 P 的取法保证了算法收敛性.最后通过算例对标准遗传算法和结合了 N6 邻域的改进遗传算法的求解性能进行了比较,从而验证本文提出之算法有效性,验证该算法可以提高集卡运输效率.

1 基于空驶最小的集卡动态调度问题描述和建模

1.1 问题描述

动态调度集卡可分为两种:一种是集卡为同一船舶的装和卸服务,另一种是集卡为不同船舶的装和卸服务,即装箱船舶和卸箱船舶不是同一船舶.这里研究第二种情形,以一艘船待装集装箱、一艘船待卸集装箱、装卸同时进行为例.

由于船舶到港时间具有随机性,运输过程中,道路的承载能力限制、集卡司机的自主性、物流系统安排的局限性等使得集卡的运输情况具有不确定性,集卡的分配也就具有不确定性.

1.2 模型假设

为建立合理的动态集卡优化调度模型,给出几个假设:

假设 1:不存在错装、错卸和错发等情况.

假设 2:不存在车祸等交通情况,允许集卡排队.

假设 3:集卡每次装载一个集装箱匀速行驶,但重卡和空卡时速度分别为 v_1, v_2 ,显然 $v_1 < v_2$.

假设 4:堆存位置已知.

假设 5:装车和卸车时间相同,均为 t .

假设 6:两艘船同时靠岸,并记靠岸时间 $t_0 = 0$.

1.3 模型建立

设 S_1, S_2 分别表示卸船和装船船舶, N_1, N_2 分别表示进口和出口的集装箱集合, n_1, n_2 分别表示进口和出口集装箱的数量. L_1, L_2 分别表示进口和出口集装箱的堆存位置, $t_{S_1, L_1}, t_{L_1, L_2}, t_{L_2, S_2}, t_{S_2, S_1}$ 分别表示集卡从卸船到堆场 L_1 、从堆场 L_1 到堆场 L_2 、从堆场 L_2 到装船 S_2 、从装船 S_2 到卸船 S_1 的时间, $t_{L_1, S_1}, t_{S_2, L_2}$ 分别表示集卡从堆场 L_1 到卸船、从装船 S_2 到堆场 L_2 的时间,则整个过程中有 6 条路径. 设 $F_{ij} = 1$ 表示路径 (i, j) 上有集卡通过,否则 $F_{ij} = 0$. 在动态调度中,集卡既要运输进口集装箱,也要运输出口集装箱.故对进口集装箱和出口集装箱进行一一配对.当 $n_1 \neq n_2$ 时,则用 0 补齐使得 $n'_1 = n'_2$,避免无相应集装箱可配对.当某集装箱与 0 配对时,说明此集装箱经过路径 $L_1 S_1$ 或 $S_2 L_2$,故下列模型中,假设 $n_1 = n_2$. 当进口集装箱 i_{1k} 和出口集装箱 i_{2j} 配对,则表示集装箱 i_{1k}, i_{2j} 为由同一辆集卡运输的前后序任务.

集卡的总运输距离为:

$$\min \left\{ \sum_{j \in L_1} \sum_{i \in S_1} t_{ij} F_{i_{1j}} v_1 + \sum_{j \in L_2} \sum_{i \in L_1} t_{ij} F_{i_{1j}} v_2 + \sum_{j \in S_2} \sum_{i \in L_2} t_{ij} F_{i_{1j}} v_1 + \sum_{j \in S_1} \sum_{i \in S_2} t_{ij} F_{i_{1j}} v_2 + \sum_{j \in S_1} \sum_{i \in L_1} t_{ij} F_{i_{1j}} v_2 + \sum_{j \in S_2} \sum_{i \in L_2} t_{ij} F_{i_{1j}} v_2 \right\}. \quad (1)$$

建立集卡空驶距离最小的改进优化模型:

$$\min \left(\sum_{j \in L_2} \sum_{i \in L_1} t_{ij} F_{i_{1j}} + \sum_{j \in S_1} \sum_{i \in S_2} t_{ij} F_{i_{1j}} + \sum_{j \in S_1} \sum_{i \in L_1} t_{ij} F_{i_{1j}} + \sum_{j \in S_2} \sum_{i \in L_2} t_{ij} F_{i_{1j}} \right). \quad (2)$$

s. t.

$$\sum_{j \in L_1} \sum_{i \in S_1} F_{l_{ij}} = \sum_{j \in L_2} \sum_{i \in L_1} F_{l_{ij}} + \sum_{j \in S_1} \sum_{i \in L_1} F_{l_{ij}} \quad (3)$$

$$\sum_{j \in S_2} \sum_{i \in L_2} F_{l_{ij}} = \sum_{j \in S_1} \sum_{i \in S_2} F_{l_{ij}} + \sum_{j \in L_2} \sum_{i \in S_2} F_{l_{ij}} \quad (4)$$

$$\sum_{j \in L_2} F_{l_{ij}} + \sum_{j \in S_1} F_{l_{ij}} = 1 \quad \forall i \in L_1. \quad (5)$$

$$\sum_{i \in L_1} F_{l_{ij}} + \sum_{i \in S_2} F_{l_{ij}} = 1 \quad \forall j \in L_2. \quad (6)$$

式(3)、(4)表示进出堆场的集卡数相等,式(5)、(6)表示集装箱两两搭配,且只被服务一次.

因为进口集装箱和出口集装箱已经过配对,故若进口集装箱的运输顺序确定,则出口集装箱的运输顺序也随之确定.实际操作中,此时应考虑出口集装箱的翻箱问题.这里暂不考虑翻箱.

为求出具体的分配方案,设 $x_{iv} = 1$ 表示集装箱 i 由集卡 v 运输,否则 $x_{iv} = 0$. 设 $x_{jv} = 1$ 表示当集装箱 i 和集装箱 j 均由集卡 v 运输,且 i 是 j 的前序任务,否则 $x_{jv} = 0$. K 表示集卡集. T_i^1, T_j^2 表示进口集装箱 i 和出口集装箱 j 的完成时间. T^1, T^2 表示最后一个进口集装箱和最后一个出口集装箱的完成时间. 从整个作业面考虑集卡效率,建立总作业时间模型为:

$$\min(\max(T^1, T^2)). \quad (7)$$

s. t.

$$\sum_{v \in K} x_{iv} = 1 \quad \forall i \in N_1. \quad (8)$$

$$\sum_{v \in K} x_{jv} = 1 \quad \forall j \in N_2. \quad (9)$$

$$\sum_{j \in N_1} x_{jv} \leq 1 \quad \forall v \in K, \forall i \in N_1. \quad (10)$$

$$\sum_{i \in N_1} x_{jv} \leq 1 \quad \forall v \in K, \forall j \in N_1. \quad (11)$$

$$\sum_{j \in N_2} x_{jv} \leq 1 \quad \forall v \in K, \forall i \in N_2. \quad (12)$$

$$\sum_{i \in N_2} x_{jv} \leq 1 \quad \forall v \in K, \forall j \in N_2. \quad (13)$$

$$M(1-x_{jv}) + T_j^2 - t_{jL_2, S_2} - 2t \geq T_i^1 \quad \forall i \in N_1, \forall j \in N_2, \forall v \in K, \text{如果 } i \text{ 是 } j \text{ 的前序任务.} \quad (14)$$

$$M(1-x_{jv}) + T_i^1 - 2t - t_{iS_1, L_1} \geq T_j^2 \quad \forall i \in N_1, \forall j \in N_2, \forall v \in K, \text{如果 } i \text{ 是 } j \text{ 的后序任务.} \quad (15)$$

$$M(1-x_{jv}) + T_j^1 - t_{jS_1, L_1} - 2t \geq T_i^1 \quad \forall i, j \in N_1, \forall v \in K, \text{如果 } i \text{ 是 } j \text{ 的前序任务.} \quad (16)$$

$$M(1-x_{jv}) + T_j^2 - t_{jL_2, S_2} - 2t \geq T_i^2 \quad \forall i, j \in N_2, \forall v \in K, \text{如果 } i \text{ 是 } j \text{ 的后序任务.} \quad (17)$$

$$\sum_{v \in K} x_{jv} \leq F_{ij} \quad \forall i \in N_1, \forall j \in N_2. \quad (18)$$

$$T^i = T_{n_i}^i, \quad i = 1, 2. \quad (19)$$

$$\text{当 } x_{jv} = 1 \text{ 且 } t_{iL_1L_2} - t_{iL_1S_1} - t_{iS_1L_1} \geq 0 \text{ 时 } x_{jv} = 0. \quad (20)$$

$$\text{当 } x_{i(j+1)v} = 1 \text{ 且 } t_{jS_2S_1} - t_{jS_2L_2} - t_{jL_2S_2} \geq 0 \text{ 时 } x_{i(j+1)v} = 0. \quad (21)$$

$$\text{当 } T_j^2 + t_{jS_2S_1} - T_i^1 - t_{iL_1S_1} \geq t \text{ 时 } x_{ikv} = 0. \quad (22)$$

$$\text{当 } T_i^1 + t_{iL_1L_2} - T_j^2 - t_{jS_2L_2} \geq t \text{ 时 } x_{k(j+1)v} = 0. \quad (23)$$

式(8)、(9)表示每个集装箱只分配给 1 辆集卡运输,式(10)、(11)表示每个集装箱最多有 1 个后序任务,式(12)、(13)表示每个集装箱最多有 1 个前序任务,式(14)~(17)表示作业完成时间关系,式(18)表示由于空驶的存在,使得所有集装箱经过路径的次数不大于集卡经过路径的次数,式(19)表示最后一个进(出)口集装箱的完成时间就是整个船舶进(出)口任务的完成时间. 式(20)表示,当集卡 u 到达 L_1 时,如果 L_2 处集卡出现等待,且预计 u 在 L_2 处的等待时间大于 u 经 L_1S_1, S_1L_1 运输一个进口集装箱的时间,则 u 经 L_1S_1 到 S_1 运输一个进口集装箱. 式(21)表示,当集卡 u 到达 S_2 时,如果 S_1 处集卡出现等待,且预计 u 在 S_1 处的等待时间大于 u 经 S_2L_2, L_2S_2 运输一个出口集装箱的时间,则 u 经 S_2L_2 到 L_2 运输一个出口集装箱. 式(22)表示,当从某时刻起,预计最先从 S_2 出发的集卡 u 到 S_1 的时间与最先从 L_1 出发的 v 经 L_1S_1 到

S_1 的时间差大于 t 时, v 直接经 L_1S_1 到 S_1 运输进口集装箱. 式(23)表示, 当从某时刻起, 预计最先从 L_1 出发的集卡 u 到 L_2 的时间与最先从 S_2 出发的 v 经 S_2L_2 到 L_2 的时间差大于 t 时, v 直接经 S_2L_2 到 L_2 运输出口集装箱.

此模型与文献[5]相比, 利用了船舶 S_1 和堆存点 L_1 、船舶 S_2 和堆存点 L_2 之间的路径, 增加了有助于说明运输情况的约束条件(式(18)、(20)~(23))进行求解, 降低了模型的计算量. 特别是当 $n_1 \neq n_2$ 时, 由于文献[5]以外围路径为集卡运输路线, 则如果集卡从船 1 运输进口集装箱到堆场 1 后无运输出口集装箱的任务, 集卡依然沿外围路径经堆场 2 行驶至船 2 或船 1; 如果集卡从堆场 2 运输出口集装箱到船 2 后无运输进口集装箱的任务, 集卡依然沿外围路径经船 1 堆场 1 行驶至堆场 2. 本文模型利用码头面积大、堆场间道路(用于龙门吊移动等)宽的特点, 引入道路 L_1S_1 、 S_2L_2 使集卡作业时灵活安排路径, 即: 如果集卡从船 1 运输进口集装箱到堆场 1 后无运输出口集装箱的任务, 则集卡经 L_1S_1 直接回到船 1; 如果集卡从堆场 2 运输出口集装箱到船 2 后无运输进口集装箱的任务, 则集卡经 S_2L_2 直接到堆场 2, 从而减少集卡的空驶距离, 减少总作业时间.

2 引入 N6 邻域法的混合遗传算法求解

2.1 N6 邻域

由于本文提出的改进的遗传算法使用了邻域结构, 故有必要对邻域结构进行概述. 邻域结构是通过给定解一个小扰动而得到一组新的邻域解的一种机制^[9], 通过采用一个“动作”把给定的解修正为一个邻域解. 车间作业邻域的理解分析可参考文献[10].

作业调度问题(job shop scheduling problem, JSP)的 $N1$ 邻域结构由 Van、Aarts 和 Lenstra 提出^[11]. $N1$ 邻域的特性是方便应用于路径重新链接. 在 $N1$ 邻域中, “动作”被定义为同一机器上需处理的关键路径上的 2 个连续操作进行的交换. 由文献[10]知, 这种邻域具有 2 个重要属性: 给定一个可行解, 则交换 2 个相邻的关键操作不会产生一个不可行解; 置换非关键操作不能改善目标函数, 甚至有可能产生不可行解.

$N5$ 和 $N6$ 邻域是由邻域 $N1$ 逐渐推广而来的. 为了解释 $N5$ 和 $N6$ 邻域, 则需定义“块”的概念. 简单地说, “块”是关键路径的最大子序列, 它包含同一台机器上处理的操作. 为了定义 $N5$ 邻域, 只考虑单个关键路径. $N5$ 邻域是指除了第一个块和最后一个块以外, 互换其他每个块中的最前 2 个和最后 2 个操作. 第一个块中, 只互换最后 2 个操作, 最后一个块中只互换最前 2 个操作^[12]. $N5$ 邻域见图 1.

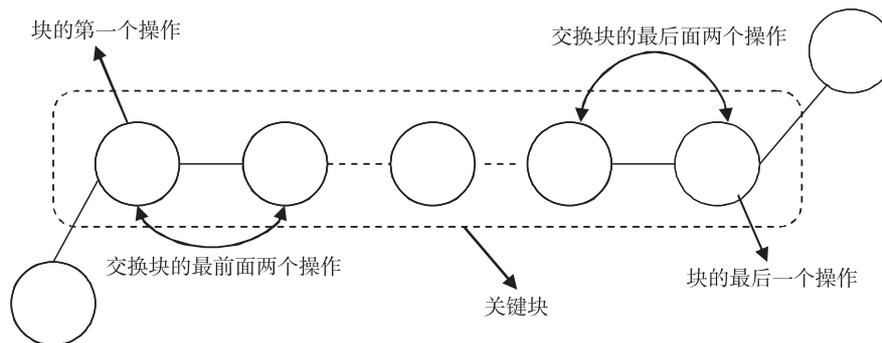


图 1 $N5$ 邻域的过程

Fig.1 The outline of the $N5$ neighborhood

同理, $N6$ 邻域的定义中也用到“块”的概念, 且只考虑单个关键路径. $N6$ 邻域是指关键块内的操作可以恰好移动到第一个操作的前面, 或者恰好移动到最后一个操作后的后面. $N6$ 邻域的说明见图 2^[13]. $N6$ 邻域的实验结果表明它在深度搜索中很有效^[13].

由于本文描述的集卡调度问题可描述为集卡如何运输集装箱序列, 因此将所有集装箱看作是所有待调度的作业, 将所有由同一集卡运输的集装箱看作是在同一机器上进行操作的作业, 将同一集装箱不能由 2 辆集卡运输看作是一作业不能在 2 台机器上进行操作, 将具有较好适应度函数值的有效集卡序列看作是关键路径, 将对较好适应度函数值贡献最大的子序列看作是优质序列, 则通过合理的编码和合理的交叉、变异等操作就能把 $N6$ 邻域和遗传算法结合起来. 遗传算法的具体编码和相应操作下文介绍.

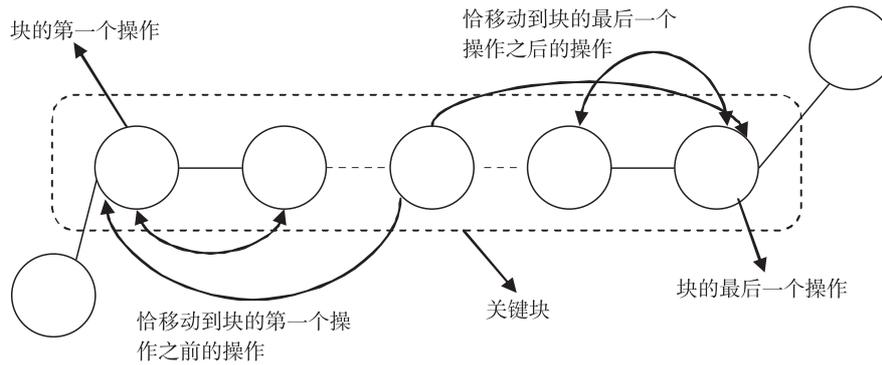


图2 N6邻域的过程

Fig.2 The outline of the N6 neighborhood

2.2 染色体编码

采用二进制编码. 染色体的长度由集卡和集装箱的数量决定. 例如, 集卡8辆, 运输任务为50个集装箱, 则任意分派运输任务给集卡, 染色体 $010101 \cdots 101 \cdots 000 \cdots 010$ 表示第1个集装箱由集卡2运输, 第2个集装箱由集卡5运输, 第k个集装箱由集卡5运输, 第n个集装箱由集卡8运输, 依此类推. 这种编码使集卡和集装箱序列间的关系明确, 由同一集卡运输的集装箱也容易统计.

2.3 初始种群的产生

初始种群随机产生, 即任意分派运输任务给集卡形成不同的染色体. 由于每辆集卡都有运输任务, 故删除有集卡无任务的染色体序列, 直到有效染色体数为N. 由于这种编码方式使染色体序列较长, 故记C为同一辆集卡运输的集装箱序列, 则由染色体可得不同的集装箱序列, 且任意2个序列内的元素不重合. 由上述染色体的构成知, 集装箱1在序列 C_2 内; 集装箱2和集装箱k都在序列 C_5 内, 且2是k的前序. 由于集卡的编号对运输并无实际影响, 即把集卡5看成集卡2, 把集卡2看成集卡5, 并不影响实际运输, 故任意2个C序列交换后的染色体是有效染色体.

2.4 适应度函数

以集卡空驶时间为适应度函数 $f(x)$, 计算初始种群的适应度值, 并分别计算8个集装箱序列的适应度贡献值. 为方便应用N6邻域法, 这里记适应度贡献值最小的序列为优质序列.

2.5 交叉操作

为提高种群的多样性, 本文设计了两种不同的染色体交叉操作.

(1) 染色体内交叉: ①序列内交叉. 以概率 P_1 交换同一集装箱序列 C_i 内的任意2个集装箱, 则集装箱的前后序改变. ②序列外交叉. 用N6邻域搜索方法以概率 P_2 进行交叉操作, 即以概率 P_2 将优质序列移到染色体最前端或最后端构成新的染色体序列. 文献[10]描述的N邻域的两条性质保证用该交叉方式得到的染色体必然是有效染色体, 因此这种交叉方法减少了有效染色体的检验和无效染色体的修正所花费的时间, 进而简化了程序, 提高了程序效率. 文献[13]也用实验结果说明了结合了N6邻域的禁忌搜索方法在JSP上的有效性.

(2) 染色体间交叉: 以概率 P_3 交换2个染色体的优质序列, 检验新染色体的有效性. 若为无效染色体, 则修正染色体使其有效. 例如: 2个不同染色体对应的集装箱序列X, Y分别为(2, 3, 4)(1, 7, 8)(5, 6, 9), (5, 8)(2, 7, 6, 9)(1, 4, 3), 优质序列分别为 $C_{2x}=(1, 7, 8)$, $C_{2y}=(2, 7, 6, 9)$, 交换优质序列后的新染色体序列 X', Y' 分别为(2, 3, 4)(2, 7, 6, 9)(5, 6, 9), (5, 8)(1, 7, 8)(1, 4, 3), 此时对应的不是有效染色体. 修正序列 X', Y' 使染色体有效, 方法为: 找到 X' 中2的位置, 在未被运输的集装箱中任取一个替换2; 找到6的位置, 在未被运输的集装箱中任取一个替换6; 找到9的位置, 此时无未被运输的集装箱, 则删除非优质序列中的9. 找到 Y' 中1的位置, 在未被运输的集装箱中任取一个替换1; 找到8的位置, 在未被运输的集装箱中任取一个替换8; 此时还有未被运输的集装箱2, 则在考虑集装箱的均衡性后将2插入非优质序列(如图3).

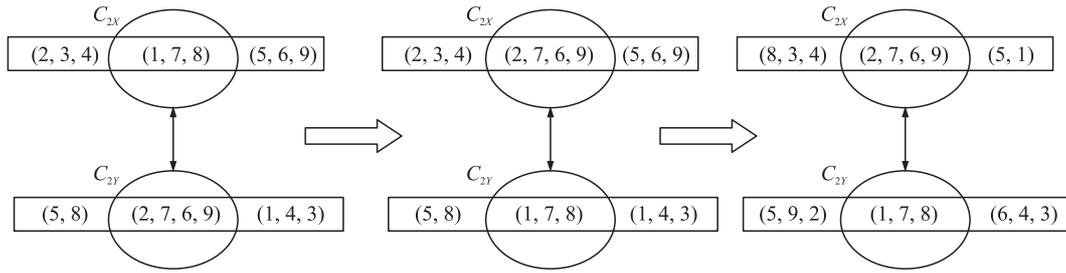


图3 无效染色体序列修正方法

Fig. 3 The modification of invalid chromosome sequence

2.6 变异操作

变异操作是为了保证种群多样性,而为了保证种群的稳定性还需要实时调整变异概率.本文设计2个变异概率进行变异操作.对序列的变异,采用文献[13]的路径重链时产生新解的概率(式(24))作为变异概率:

$$P_{uvr}^{new} = \frac{P_{uvr}^{old}}{P_{uvr}^{old} + (1 - P_{uvr}^{old}) \exp(-\mu(\min_{x \in f_{uv0}} f(x) - \min_{x \in f_{uv1}} f(x)))}. \quad (24)$$

为便于利用式(24),这里设 P_r 为概率向量,每个分量表示序列 C_i 内所有集装箱的前序概率.以序列 C_{2x} 为例,若集装箱1是7的前序的概率 P_{17r} 为0.7,7是8的前序的概率 P_{78r} 为0.4,8是1的前序的概率 P_{81r} 为0.1,则 $P_r = (0.7, 0.4, 0.1)$.若 u 是 v 的前序,记 $q(u, v) = 1$,否则 $q(u, v) = 0$.因为同一染色体在 $q = 0, 1$ 时的适应度函数值不同,故分别记为 f_{uv0}, f_{uv1} .

当对序列进行变异操作时,以概率 $\sum P_{uvr} / |C_i|$ 改变 P_r 最小分量的 u, v 序.此变异概率能保证当 u 是 v 的前序的可能性越小时,以较大概率改变 u, v 的序,进一步增加了多样性.

对染色体的变异概率,这里设计为 $1 - \bar{f}_i(x) / f_i(x)$, $\bar{f}_i(x)$ 是第 i 代的最小适应度函数值.记 $P = 1 - \bar{f}_i(x) / f_i(x)$.则当对染色体进行变异操作时,以概率 P 交换 C_i, C_j 中的集装箱 C_{iu}, C_{ju} .此变异概率能保证当个体适应值较大、个体性能较差时,以较大概率变异.当个体适应值较小、个体性能较好时,以较小概率变异.且具有最小适应度函数值的染色体的变异概率为0,即此染色体不变异,直接进入下一代进行相应操作.故 P 的选择保证了算法的收敛性.

3 算例与分析

用6个算例来检验本文设计的结合了 N_6 邻域的改进遗传算法的有效性,并与标准遗传算法作对比.运算时设2艘船舶间隔300 m,装(卸)车时间为2 min,集卡空载速度为24 km/h,满载速度为18 km/h.2个堆场间隔1200 m,船舶到堆场的平行距离为900 m.求解时采用C语言编程,运行环境为Core3 CPU、主频为2.0 GHz、内存为2 GB的个人计算机.遗传算法的种群大小为200,最大遗传代数数为500.标准遗传算法的交叉概率为0.9,变异概率为0.2.改进混合遗传算法中取交叉概率 $P_1 = P_2 = P_3 = 0.9$,变异概率 P_r 的分量初始值均为0.5.对每个算例计算10次,每次运行10遍,具体算法的结果如表1所示.其中除第一列外,其他列的“/”前后的数据分别是标准遗传算法的结果和改进混合遗传算法的结果.

表1 改进混合遗传算法的结果与比较

Table 1 Comparison of different algorithms for container truck dispatching

算例(集卡数/进口箱数/出口箱数)	总作业时间/min	集卡总行驶距离/km	集卡等待时间/min	集卡空驶距离/km	空驶百分比/%	最优解出现的平均数/次	计算时间/s
8/25/25	531.65/523.25	112.2/109.8	18.65/16.25	44.7/42.3	39.84/38.52	8.92/9.46	0.15/0.04
18/60/60	1244.975/1228.3	260.5/260.5	23.35/21.675	104.4/98.5	39.19/37.81	9.14/9.61	0.47/0.28
10/45/50	999.325/985.93	213.3/208.5	14.825/13.43	85.05/80.25	39.87/38.49	9.17/9.57	0.33/0.21
12/60/50	1151.9/1141.7	247.8/244.2	16.15/14.95	99.3/95.7	40.07/39.19	9.32/9.53	0.45/0.36
8/100/90	2105.87/2079.58	436.2/427.8	41.62/36.33	179.7/171.3	41.20/40.04	9.21/9.66	1.22/0.68
10/500/600	12356.45/12224.14	2583.6/2583.6	175.95/127.64	1132.2/1098.6	43.26/42.52	8.94/9.51	4.93/2.74

从表1可以看出,改进混合遗传算法的计算时间要小于标准遗传算法,这是由于本文设计的遗传算法的编码方式易于对有效染色体和同一集卡的集装箱任务进行统计,有助于对解空间进行区域划分,从而降

低了解的搜索范围,提高了求解速度。 N 邻域的特性进一步加快了求解速度。将 $N6$ 邻域法在染色体优质序列上进行操作,更使找到最优解的计算时间缩短。改进遗传算法得到最优解的平均次数优于标准遗传算法,这是由于本文选用了多种交叉操作和变异操作,增加了种群多样性。且序列间变异的实时调整,染色体变异概率的取法,进一步提高了获得最优解的概率,也说明了该算法的稳定性。在种群大小、遗传代数和交叉概率都相同时,改进遗传算法求得的总作业时间、集卡总行驶距离、集卡空驶距离和集卡空驶百分比都明显优于标准遗传算法,这是由于编码设计、 $N6$ 邻域、优质序列、多种交叉操作和变异操作等都使前者更易获得最优解,而最优解对应的集卡空驶率较低。当集装箱数量较大时,集卡等待时间和空驶率明显增加,这是由于运输任务重时集卡排队严重,此时集卡会自动选择空驶至另一地点进行进(出)口集装箱的运输,致空驶率增加。当进口集装箱和出口集装箱数量差距增大时,集卡空驶比例增加,这与实际相符。当集卡少、集装箱多时,计算时间大幅增加,且两种算法的时间差增大,编码设计和 $N6$ 邻域的优势凸显。当集装箱数量较大时,标准遗传算法获得最优解的平均次数略有波动,说明改进混合遗传算法更具稳定性。

4 结束语

虽然集卡调度的研究非常广泛,很多研究也结合装卸桥、龙门吊等考虑整个集卡作业系统^[1,4,5],但是针对集卡装卸同时进行的动态调度研究甚少。本文结合文献[5],提出了一种基于集卡空驶最小的装卸同时进行的动态优化改进模型。求解时在遗传算法中采用合理的编码方式使集卡和集装箱的运输情况便于统计,统计过程有利于划分解空间区域和应用 $N6$ 邻域法。算法实现中设计了不同的交叉操作和变异操作来保证种群多样性和稳定性,进而保证算法的收敛性。实验结果表明,在不增加基础设施的基础上,采用装卸结合的集卡动态调度能有效提高集卡利用率,提高港口效率。运算时间、最优解等都说明了此改进遗传算法的优越性。而当集卡运输任务较重(数量少、集装箱数量多)时,排队现象严重,集卡空驶率较高,如何按作业面改进模型、如何交叉安排整个码头的集卡运输任务,进一步提高集卡效率,本文尚未考虑。

相关研究比较:文献[1]基于时间窗冲突反馈机制,用遗传算法求解集卡调度方案。它的重点在于考虑提高龙门吊、岸桥和集卡的协作能力,减少船舶停靠时间,从而提高码头效率。文献[2]探讨了铁路集装箱列车卸车时的集卡调度,以龙门吊最大作业时间和集卡等待时间最小为目标。文献[3]的研究基于双40英尺岸桥的集卡调度,以岸桥作业延迟时间最小和集卡空驶时间最小为目标,引入小生境技术和复合交叉等方式改进遗传算法。算法实现时采用自然数编码。当集装箱个数较多时,不仅染色体很长,而且染色体的有效判断也是个问题。文献[4]以装卸桥等待集卡的时间最小为目标的集卡动态调度,用 Q 学习算法优化调度方案。它们从不同角度考虑了集卡调度问题,且都只考虑了装船或卸船。本文从集卡空驶最小的角度建立集卡动态调度模型,且同时考虑了装船和卸船。文献[5]也基于集卡空驶最小给出了装卸同时进行的集卡调度模型,本文与文献[5]的模型主要不同在于增加了集卡行驶路径 S_1L_1 、 L_2S_2 ,增加了集卡动态调整的约束条件,使集卡调度更具灵活性。文献[5]采用两阶段禁忌搜索算法先求得卸船集装箱顺序,再搜索集卡调度方案。由于是在确定卸船顺序后再进行的集卡调度,因此集卡调度的作用范围受到限制。本文将集装箱任意分配给集卡运输,采用遗传算法搜索整个解空间获得集卡最优调度方案。这个方案基于整个作业过程,集装箱卸船顺序与集卡调度方案同时确定。当然,模型和遗传算法还有需要改进的地方,除了道路因素,哪些因素对集卡空驶影响更大、装卸桥和龙门吊的现场作业情况对模型有哪些影响、如何安排集卡交叉作业来改进最优解,这都是以后要考虑的问题。

[参考文献]

- [1] 丁荣涛. 基于协作能力约束的港口集卡调度优化策略[J]. 清华大学学报:自然科学版,2012,52(8):1158-1164.
- [2] 梁剑,王强,安俊英. 铁路集装箱中心站内部集卡调度和箱位配置[J]. 计算机工程,2012,38(8):213-220.
- [3] 尚晶. 面向双40英尺岸桥的码头集卡调度模型与算法[J]. 华中科技大学学报:自然科学版,2010,38(11):84-87.
- [4] 曾庆成,杨忠振. 集装箱码头集卡调度模型与 Q 学习算法[J]. 哈尔滨工程大学学报,2008,29(1):1-4.
- [5] 曾庆成. 集装箱码头装卸作业集成调度模型与方法[D]. 大连:大连海事大学计算机科学与技术学院,2008.
- [6] Pyung H K, Woon S L, Dong W J. Fleet sizing and vehicle routing for container transportation in a static environment[J]. OR

- Spectrum,2004,26(2):193-209.
- [7] Ren Q D E J, Wang Y P. A new hybrid genetic algorithm for job shop scheduling problem[J]. Computers and Operations Research,2012,39(10):2291-2299.
- [8] 何大勇,查建中,姜义东. 遗传算法求解复杂集装箱装载问题方法研究[J]. 软件学报,2001,12(9):1380-1385.
- [9] Glover F, Laguna M. Tabu Search[M]. Dordrecht:Kluwer Academic Publishers,1997.
- [10] Zhang C Y, Li P G, Guan Z L, et al. A tabu search algorithm with a new neighborhood structure for the job shop scheduling problem[J]. Computers and Operations Research,2007,34(11):3229-3242.
- [11] Van Laarhoven P J M, Aarts E H L, Lenstra J K. Job shop scheduling by simulated annealing[J]. Operations Research,1992,40(1):113-125.
- [12] Nowicki E, Smutnicki C. A fast taboo search algorithm for the job shop scheduling problem[J]. Management Science,1996,42(6):797-813.
- [13] Nasiri M M, Kianfar F. A GES/TS algorithm for the job shop scheduling[J]. Computers and Industrial Engineering,2012,62(4):946-952.

[责任编辑:黄敏]