

文章编号:1005-0523(2018)03-0062-07

循环甩挂运输网络下的集装箱牵引车调度优化

黄勇,乙永松,邱琦

(华东交通大学交通运输与物流学院,江西 南昌 330013)

摘要:基于集装箱甩挂运输循环式网络,建立甩挂运输牵引车调度优化模型,并设计基于启发式规则的改进遗传算法对该模型进行求解。运用算例验证了模型和算法的有效性和可行性,通过对比发现,设计的结合启发式规则改进的遗传算法解的质量优于模拟退火算法,牵引车完成所有任务的效率提高了5.54%,而且收敛速度很快,得出的解质量较高,对甩挂运输实际调度作业具有一定的参考价值。

关键词:甩挂运输;循环式网络;调度优化;遗传算法

中图分类号:U4-9

文献标志码:A

循环甩挂运输作业模式是指在循环运输网络上的装卸工作需求点配属特定数目的挂车,当牵引车抵达后,放下挂斗,换上新的挂车后继续行驶。循环甩挂运输运作模式实质是将牵引车作为循环调配的目标,把挂车作为车辆在循环网络上需运输的货品,循环调度来组织甩挂运输作业。国外学者对有关甩挂运输牵引车路径优化的研究起步较早,成果相对较多,Chao I-Ming较早的提出甩挂运输路径优化模型并使用禁忌搜索算法进行求解^[1]。Derigs等设计了一种包括邻域搜索、启发式策略等的混合算法,并求解了具有时间窗约束条件的甩挂运输调度优化问题^[2]。Kris Braekers提出了非对称的带时间窗的多车辆旅行商问题(AM-TSPTW),并设计了两阶段混合确定性退火和禁忌搜索混合算法来求解^[3]。Seyedmehdi等在传统的TTRP中加入随机需求和时间窗,并引入多重交叉、变异、局部搜索的启发式算法求解^[4]。对比国外、国内对甩挂运输调度优化建模和采用优化算法求解的研究以及在特定甩挂运输网络下的牵引车调度优化研究还不充分,杨珍花等在空挂任务不确定情形下建立甩挂运输调度优化模型并提出了两种启发式算法探索空挂调运的合理方案^[5]。杨光敏基于甩挂运输轴辐式网络建立甩挂运输调度优化模型,设计了启发式模拟退火算法求解,验证了其有效性^[6]。曾毅,朱旭生针对旅行商问题提出了一种新的混合和声搜索算法,为维持混合算法的多样性给出新的记忆库更新策略,并通过测试验证了算法的有效性^[7]。

基于以上研究基础,研究了循环甩挂运输网络下牵引车调度优化问题,建立了牵引车调度优化模型,并运用改进的遗传算法对模型进行求解,探究模型与算法的有效性和实用性。

1 问题描述

牵引车从甩挂中心出发抵达装卸作业需求点后甩下所拖挂的挂车,置换新的挂车行驶,到达另一装卸作业需求点或甩挂中心后甩下挂车换上其他挂车连续行驶,如此循环往复形成一个循环甩挂运输网络。

在循环甩挂运输中,任务集包括取箱作业任务和送箱作业任务,送箱作业任务的空箱来自甩挂中心或者客户点。文中取箱作业任务默认为取重箱作业任务,不考虑空箱调度。为便于建模,把每个工作的起始点和到达点组合到一处,路径模式如图1所示。

收稿日期:2017-10-30

作者简介:黄勇(1959—),男,教授,硕士生导师,研究方向为物流与供应链管理。

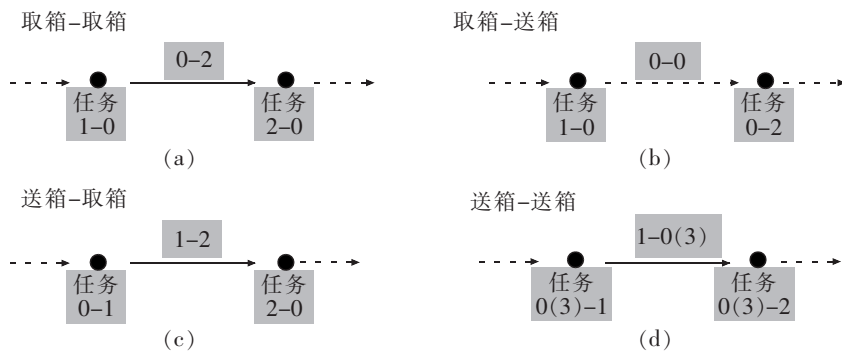


图 1 车辆路径模式图

Fig.1 Vehicles routing

图 1 的任务 1-0 代表从装卸需求点 1 到甩挂中心的取箱作业; 0-2 代表牵引车进行完取箱作业 1-0 后去执行下一个工作 2-0 走过的路线, 即从甩挂中心到装卸作业需求点 2 所走过的路径; 任务 2-0 代表从装卸作业需求点 2 到甩挂中心的取箱作业; 0-0 代表牵引车从任务点到任务点走过的虚拟路线。牵引车走过的线路如图 2 所示。

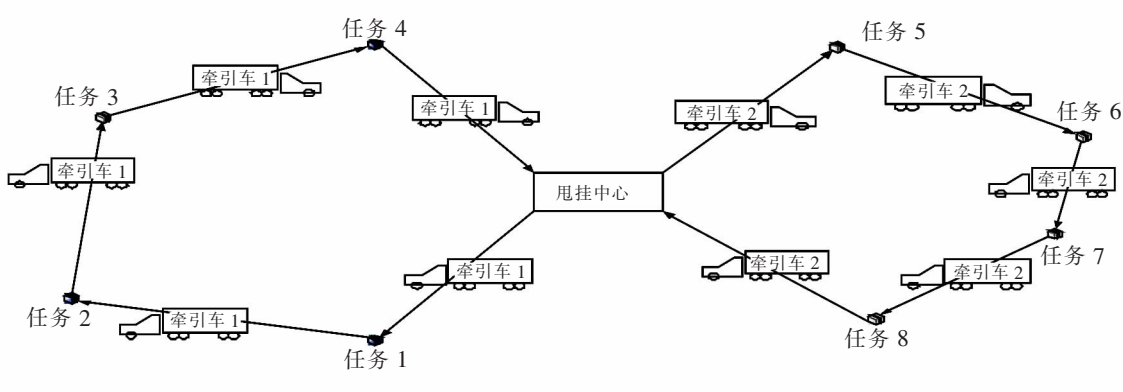


图 2 牵引车行驶路线

Fig.2 The truck driving route

从图 2 可以看出, 循环甩挂运输网络下的牵引车调度在于工作节点是由起始点和终到点组合成一点形成, 执行该任务的时间为牵引车从起始点到终到点的所用时间, 且牵引车的行驶线路中有一定概率包含虚拟路线的存在。假设在调度之前所有任务都已确定, 不考虑送箱作业任务将会派生出来的取箱作业任务, 每个任务的时间窗也已给出。

2 牵引车调度优化模型

2.1 假设条件

根据目前物流园区运输实际操作情况以及建模需要, 做出如下假设: ① 牵引车和挂车的标准统一; ② 牵引车每回只能拖挂一台车; ③ 牵引车拖挂空挂车和重挂车的速率一样; ④ 牵引车挂上拖车和甩下拖车的时间忽略; ⑤ 所有任务在优化之前都已确定。

2.2 牵引车调度优化模型

目标函数:

$$\min \sum_{k=1}^K \sum_{j=0}^M C_1 x_{0jk} + \sum_{k=1}^K \sum_{i=0}^M \sum_{j=0}^M (w_i + f_i + t_{ij}) C_2 x_{ijk} \tag{1}$$

约束条件:

$$\sum_{k=1}^K \sum_{j=0}^M x_{0jk} \leq K \quad (2)$$

$$\sum_{j=1}^M x_{0jk} = \sum_{i=1}^M x_{i0k} \leq 1 \quad k=1, 2, \dots, K \quad (3)$$

$$\sum_{k=1}^K \sum_{i=0}^M x_{ihk} = \sum_{k=1}^K \sum_{j=0}^M x_{hjk} = 1 \quad h=1, 2, \dots, M \quad (4)$$

$$\sum_{i=0}^M \sum_{j=0}^M x_{ijk} (t_{ij} + w_i + f_i) \leq R_k \quad k=1, 2, \dots, K \quad (5)$$

$$\sum_{k=1}^K \sum_{i=0}^M x_{ijk} (A_i + f_i + t_{ij}) \leq A_j \quad j=1, 2, \dots, M \quad (6)$$

$$ET_i \leq A_i \leq LT_i \quad i=1, 2, \dots, M \quad (7)$$

$$x_{ijk} \in \{0, 1\} \quad i=1, 2, \dots, M \quad j=1, 2, \dots, M \quad k=1, 2, \dots, K \quad (8)$$

式中： M 为任务数量； K 为牵引车数量； C_1 为单辆牵引车的使用固定成本； C_2 为牵引车单位时间行驶成本； t_{ij} 为牵引车从作业*i*的终到点到作业*j*的起始点所需时间； f_i 为执行作业*i*需要的时间，即牵引车从作业*i*的起始点到终到点需要的时间； w_i 为由于牵引车提早抵达作业*i*的起点而产生的等待时间； A_i 为牵引车到达任务*i*起点的时刻； R_k 为牵引车*k*的工作时间； $[ET_i, LT_i]$ 为任务*i*的时间窗。

$$x_{ijk} = \begin{cases} 1 & \text{采用牵引车 } k \text{ 执行完任务 } i \text{ 立刻执行任务 } j \\ 0 & \text{否则} \end{cases}$$

其中，式(1)为目标函数，表示牵引车调度所需固定成本加可变成本，且 $C_1 > C_2$ ，以确保减少使用牵引车数目的优先级高于牵引车所用时间的优先级；式(2)为牵引车可调度的数量不大于牵引车总和；式(3)为牵引车从甩挂点出发到完成作业后回到甩挂中心；式(4)为每项作业仅被一辆牵引车执行1次；式(5)代表每辆牵引车走过的时间小于等于它的作业时间；式(6)为作业被完成的先后次序约束；式(7)为作业起始时间要求在时间窗的范畴之内；式(8)为决策变量。

3 基于启发式规则的改进遗传算法求解

利用启发式规则求解牵引车需求数量上限，设任务集 $M = \{1, 2, \dots, n\}$ ，牵引车集 $K = \{1, 2, \dots, m\}$ ，过程如 Step 1，设未完成任务集 $L = M$ ，任务 $i=1 (i \in M)$ ，牵引车 $j=1 (j \in K)$ ，进入 Step 2；Step 2，将未完成任务集 L 按照时间窗自小到大排序，进入 Step 3；Step 3，根据时间窗判断任务*i*是否能添加到第*j*辆牵引车的任务集中。若满足，则 $L=L/i$ ，进入 Step 4；否则，令 $j=j+1$ ，重复 Step 3；Step 4，判断是否为空集，若是，停止计算，输出*j*；否则令 $i=i+1, j=1 (j \in K)$ ，转到 Step 3。然后用改进的遗传算法继续求解。

3.1 编码方案设计

文中的遗传算法中每个染色体包含牵引车需求数量和牵引车行驶路线2个方面，所以设计了每个染色体含有2个子串构成，子串1有 $|P|$ 个基因位（ $|P|$ 为牵引车需求数量上限），采用二进制编码，随机产生 $|P|$ 个0,1变量，每个基因位的取值为1代表使用该牵引车，反之不使用；子串2的 $|J|$ 个基因位代表任务数量，采用浮点数编码，随机产生 $|J|$ 个介于(1, $k+1$)之间的相互不等价的浮点数， K 为子串1中基因位取值为1的个数，浮点数的整数部分为牵引车编码，被相同牵引车车作业的任务所对应的浮点数整数位相同，小数部分按照升序排列表示为牵引车完成任务的顺序。

假定某优化问题中，采用启发式规则得到的牵引车数上限为3辆，对其进行编号，有8个任务需要完成，按照如下方式编码：

1) 随机产生3个0-1整数编码A1和8个介于(1, $\text{sum}(A_1)+1$)之间的浮点数构成一个染色体，如1-1-1-1.42-3.25-3.76-2.73-2.43-1.89-3.57-2.21，表示车辆需求为3，后面的浮点数位置与任务点对应。

2) 将后8位浮点数按从小到大进行分组排序:(1.42,1.89),(2.21,2.43,2.73),(3.25,3.57,3.76)。

3) 根据分组排序结果可知,编码方案表示执行完8个任务需3辆牵引车,每辆牵引车完成作业的先后次序为(0表示甩挂中心):车次1:0-1-6-0;车次2:0-8-5-4-0;车次3:0-2-7-3-0。

3.2 适应度函数

文中利用 ranking 函数对每个染色体的适应度值进行运算。ranking 函数把染色体的目标函数值按从小到大的次序排列,并得到染色体适应度值 FitV。假设某个种群的10个染色体目标函数值为

$$\text{Objv}=[10,9,8,7,6,1,2,3,4,5]^T$$

适应度函数计算出的值为[2.000 0,1.777 8,1.555 6,1.333 3,1.111 1,0,0.222 2,0.444 4,0,666 7,0.888 9]^T

3.3 遗传算子设计

1) 选择算子。采用轮盘赌选择算法。种群数量为 n , 种群中的每个染色体都有一定概率被选中, 其被选的几率为 $nf(X_i)/\sum f(X_i)$ 。通过这种算法得到的染色体就变成下一代种群的父代。这样虽然将目标函数值高的染色体选中的概率增大, 但也许并没有选中最优染色体。所以在选择算子时, 可以将此次进化得到的最劣染色体用上一代的精英染色体来替换, 然后利用随机产生的新群体(种群规模为原种群规模的10%)代替最劣的10%, 用来担保每代中都保留着精英染色体使之来做交叉、变异等计算, 达到丰富种群的目的。

2) 交叉算子。用轮盘赌选择算子选择出 M 个染色体, 其中 M 为种群规模, 把这些染色体交配成 $[M/2]$ 对染色体组, 其中 $[x]$ 代表小于等于 x 的最大整数。计算种群中染色体的平均适应度函数值 \bar{f} , 并与每组的染色体的 f_i 相比较, 分别计算出每组染色体的自适应交叉概率:

如果 $f \geq \bar{f}$, 则 $pc = \lambda_1 \frac{f_{\max} - f}{f_{\max} - \bar{f}}$; 如果 $f < \bar{f}$, 则 $pc = \lambda_2$; $0 < \lambda_1, \lambda_2 < 1$, pc 为自适应交叉概率, f_{\max} 为精英染色体的

适应度值, f 为染色体组中较优染色体的适应度值; 取 $\lambda_1 = 0.7, \lambda_2 = 0.9$ 。当 $f \geq \bar{f}$ 时, 为了使精英染色体的基因结构不被损坏, 所以使其交叉概率变小, $pc = \lambda_1 \frac{f_{\max} - f}{f_{\max} - \bar{f}}$; 当 $f < \bar{f}$ 时, 为了使劣质染色体的基因结构发生改变, 因

此让交叉概率变大, $pc = \lambda_2$ 。确定了交叉概率后, 则按照概率进行多点交叉操作。

3) 变异算子。采用自适应变异概率对染色体进行变异操作, 根据染色体的适应度函数值规定该染色体的变异概率, 变异概率的得出方法如下:

如果 $f \geq \bar{f}$, 则 $p_m = \lambda_3 \frac{f_{\max} - f}{f_{\max} - \bar{f}}$; 如果 $f < \bar{f}$, 则 $p_m = \lambda_4$; $0 < \lambda_3, \lambda_4 < 1$, p_m 为自适应变异概率, f_{\max} 为种群中最优染色

体的适应度值, f 为变异染色体的适应度值; 取 $\lambda_3 = 0.005, \lambda_4 = 0.01$ 。当 $f \geq \bar{f}$ 时, 其变异概率变小是为了不损坏优秀染色体的基因结构, $p_m = \lambda_3 \frac{f_{\max} - f}{f_{\max} - \bar{f}}$; 当 $f < \bar{f}$ 时, 为了改变劣质染色体的基因结构故增大其变异概率,

$p_m = \lambda_4$ 。确定了变异概率后, 对染色体编码的某个基因位变换后得到新的染色体。

4 算例分析

4.1 数值参数模拟

假设把某地物流园区的堆场 P_0 作为甩挂中心, 周边存在20个客户点, 分别为 $P_1, P_2, P_3, P_4, P_5, P_6, P_7, P_8, P_9, P_{10}, P_{11}, P_{12}, P_{13}, P_{14}, P_{15}, P_{16}, P_{17}, P_{18}, P_{19}, P_{20}$, 为了保证数值与实际的拟合程度, 使用 matlab 中的 triu 函数来产生甩挂中心到客户点之间的行驶时间, 通过 triu 函数生成一个上三角矩阵, 然后进行转置得到一个下三角矩阵, 结果如表1, 这里假设牵引车匀速行驶, 1个时间单元等于10 min。

表1 甩挂中心及客户点位置
Tab.1 The dropping and pulling center and customers locations

	P_0	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}	P_{11}	P_{12}	P_{13}	P_{14}	P_{15}	P_{16}	P_{17}	P_{18}	P_{19}	P_{20}
P_0	0	4	5	5	4	5	1	4	10	7	4	5	6	8	10	6	7	1	6	6	10
P_1	4	0	4	3	5	4	5	7	1	10	9	7	10	7	1	4	3	4	7	8	9
P_2	5	4	0	1	2	9	2	1	10	8	5	3	7	9	4	9	5	1	4	6	5
P_3	5	3	1	0	3	4	8	9	1	4	5	2	2	4	7	6	2	2	2	1	8
P_4	4	5	2	3	0	9	4	9	7	7	3	9	4	8	3	5	3	8	5	5	5
P_5	5	4	9	4	9	0	9	8	8	3	2	8	1	9	3	9	3	8	4	7	10
P_6	1	5	2	8	4	9	0	9	6	3	4	6	10	4	8	4	4	9	8	6	10
P_7	4	7	1	9	9	8	9	0	9	7	8	2	5	6	7	5	2	6	8	4	9
P_8	10	1	10	1	7	8	6	9	0	6	8	2	10	10	6	10	4	1	7	10	4
P_9	7	10	8	4	7	3	3	7	6	0	7	1	9	6	7	1	2	10	2	9	5
P_{10}	4	9	5	5	3	2	4	8	8	7	0	5	5	4	1	10	9	9	1	9	3
P_{11}	5	7	3	2	9	8	6	2	2	1	5	0	9	7	4	2	1	3	6	4	8
P_{12}	6	10	7	2	4	1	10	5	10	9	5	9	0	4	5	7	10	6	4	6	9
P_{13}	8	7	9	4	8	9	4	6	10	6	4	7	4	0	3	6	4	10	10	9	10
P_{14}	10	1	4	7	3	3	8	7	6	7	1	4	5	3	0	7	1	8	10	10	6
P_{15}	6	4	9	6	5	9	4	5	10	1	10	2	7	6	7	0	4	9	3	7	6
P_{16}	7	3	5	2	3	3	4	2	4	2	9	1	10	4	1	4	0	5	9	3	2
P_{17}	1	4	1	2	8	8	9	6	1	10	9	3	6	10	8	9	5	0	9	7	9
P_{18}	6	7	4	2	5	4	8	8	7	2	1	6	4	10	10	3	9	9	0	1	5
P_{19}	6	8	6	1	5	7	6	4	10	9	9	4	6	9	10	7	3	7	1	0	3
P_{20}	10	9	5	8	5	10	10	9	4	5	3	8	9	10	6	6	2	9	5	3	0

客户点所包括的取箱作业数和送箱作业数为 $[0, a]$ 之间的随机整数,利用 randint 函数随机产生。表2为随机产生的任务数。取 $a=2$ 。

表2 客户点任务量
Tab.2 Task size of customers

客户点序号	取箱作业任务	送箱作业任务	客户点序号	取箱作业任务	送箱作业任务
P_1	X_1	X_2	P_{11}	—	X_{21}
P_2	—	X_3	P_{12}	X_{22}	—
P_3	X_4	—	P_{13}	X_{23}	—
P_4	X_5, X_6	X_7	P_{14}	X_{24}	X_{25}
P_5	X_8	X_9, X_{10}	P_{15}	X_{26}	X_{27}
P_6	X_{11}	—	P_{16}	X_{28}	X_{29}, X_{30}
P_7	—	X_{12}	P_{17}	X_{31}	—
P_8	X_{13}, X_{14}	X_{15}, X_{16}	P_{18}	—	X_{32}
P_9	X_{17}	—	P_{19}	X_{33}, X_{34}	X_{35}
P_{10}	X_{18}, X_{19}	X_{20}	P_{20}	X_{36}	—

注:“—”表示任务量为0。

根据时间窗的特性以及数值实验的有意义性,使用 `unifrnd` 函数随机产生每项作业的开始执行时间, `unifrnd` 函数生成一个服从均匀分布的函数,然后用 `round` 函数进行取整,把每项作业的开始时间加上一个固定的时间得到了作业的结束时间 $t_0(t_0 \neq 0)$,取 $t_0=100$,开始与结束时间共同组成任务的时间窗。表3为所有作业随机产生的时间窗。

表3 各任务对应的时间窗
Tab.3 The task time windows

任务编号	时间窗/min	任务编号	时间窗/min	任务编号	时间窗/min	任务编号	时间窗/min
X_1	[64,164]	X_{10}	[346,446]	X_{19}	[355,455]	X_{28}	[21,121]
X_2	[108,208]	X_{11}	[234,334]	X_{20}	[250,350]	X_{29}	[285,385]
X_3	[291,391]	X_{12}	[352,452]	X_{21}	[42,142]	X_{30}	[72,172]
X_4	[121,221]	X_{13}	[212,312]	X_{22}	[303,403]	X_{31}	[192,292]
X_5	[174,274]	X_{14}	[143,243]	X_{23}	[33,133]	X_{32}	[5,105]
X_6	[160,260]	X_{15}	[278,378]	X_{24}	[336,436]	X_{33}	[270,370]
X_7	[36,136]	X_{16}	[362,462]	X_{25}	[202,302]	X_{34}	[109,209]
X_8	[326,426]	X_{17}	[304,404]	X_{26}	[10,110]	X_{35}	[96,196]
X_9	[231,331]	X_{18}	[180,280]	X_{27}	[267,367]	X_{36}	[57,157]

4.2 参数及其求解结果分析

启发式规则得出牵引车的最大需求数量为4辆。为确保减少使用牵引车数目的优先级高于牵引车运行时间的优先级,设定 $C_1=10\ 000$, $C_2=1$ 。经多次调试,算法参数设定种群规模为 $NIND=200$,最大迭代次数为 $MAXGEN=1\ 000$,自适应交叉概率 $\lambda_1=0.7$, $\lambda_2=0.9$,变异概率 $\lambda_3=0.005$, $\lambda_4=0.01$ 。求得最优解为需求的牵引车数量为4辆,目标函数值为43 160,车辆行驶路径如表4所示。程序运行时间为156 s,求解速度很快且比较稳定。

表4 牵引车执行任务顺序
Tab.4 Orders of tractors' tasks

牵引车号	执行任务顺序
1	$X_7 \rightarrow X_{36} \rightarrow X_2 \rightarrow X_5 \rightarrow X_9 \rightarrow X_{15} \rightarrow X_3 \rightarrow X_{17} \rightarrow X_{12}$
2	$X_{32} \rightarrow X_1 \rightarrow X_{30} \rightarrow X_4 \rightarrow X_{18} \rightarrow X_{13} \rightarrow X_{27} \rightarrow X_8 \rightarrow X_{16}$
3	$X_{26} \rightarrow X_{23} \rightarrow X_{35} \rightarrow X_6 \rightarrow X_{25} \rightarrow X_{33} \rightarrow X_{22} \rightarrow X_{10} \rightarrow X_{19}$
4	$X_{28} \rightarrow X_{21} \rightarrow X_{34} \rightarrow X_{14} \rightarrow X_{31} \rightarrow X_{11} \rightarrow X_{20} \rightarrow X_{29} \rightarrow X_{24}$

计算数据说明,文中提出的算法能对模型求解,而且求解速度较快。利用模拟退火算法求解该调度优化模型,设置初始温度 $T_0=1\ 000$,内循环次数 $L=500-T/4$ (T 为当前温度),终止温度 $T_{end}=1$,降温速率 $\alpha=0.8$,得到的最优解为需要4辆牵引车,目标函数值为45 690。对比两种算法的结果,文中设计的算法得出的解的质量优于模拟退火算法求解策略,牵引车完成所有任务的时间效率提高了5.54%。在求解速度上,由于模拟退火算法解矩阵的规模过大导致程序运行时间过长,且每次在解矩阵时交换两个位置的数值,易产生较多的无效解,相比而言,文中所述算法收敛速率较快。

5 结语

对于牵引车在装卸货物中所浪费的等候时间,通过甩挂运输过程都可以被大大减少,使得牵引车的工作效率得到很大提高,同时对物流运输过程中所产生的成本也得到降低。在集装箱甩挂运输循环式网络模式下,通过整合起始点和任务点,重新确定甩挂运输牵引车的路径模式,设计基于启发式规则的遗传算法对牵引车调度优化模型求解,获得了比较理想的结果。通过算例分析和比较,发现模型与算法具有较高的实用性,能够有效提升牵引车的作业效率,对企业提高甩挂运输作业组织效率提供指导。

参考文献:

- [1] CHAO I-MING. A tabu search method for the truck and trailer routing problem[J]. *Computers & Operations Research*, 2002(29): 33-51.
- [2] DERIGS ULRICH, PULLMANN MARKUS, VOGEL ULRICH. Truck and trailer routing-problems, heuristics and computational experience[J]. *Computers & Operations Research*, 2013, 40(2): 536-546.
- [3] KRIS BRAEKERS, AN CARIS, GERRIT K. Janssens. Bi-objective optimization of drayage operations in the service area of inter-modal terminals[J]. *Elsevier Transportation Research Part E*, 2014(65): 50-69.
- [4] SEYEDMEHDI MIRMOHAMMADSADEGHI, SHAMSUDDIN AHMED. Memetic heuristic approach for solving truck and trailer routing problems with stochastic demands and time windows[J]. *Networks & Spatial Economics*, 2015(15): 1093-1115.
- [5] 杨珍花, 杨光敏, 徐奇. 空挂任务不确定情形下的甩挂运输调度优化[J]. *交通运输工程学报*, 2016(5): 103-111.
- [6] 杨光敏, 罗志勇, 杨珍花. 轴辐式网络结构下的集装箱甩挂运输调度优化[J]. *大连海事大学学报*, 2016, 42(3): 63-69.
- [7] 曾毅, 朱旭生. 基于混合和声搜索算法求解旅行商问题[J]. *华东交通大学学报*, 2016, 33(6): 131-136.
- [8] RUIYOU ZHANG, WON YOUNG YUN, HERBERT KOPFER. Multi-size container transportation by truck: modeling and optimization[J]. *Flexible Services & Manufacturing Journal*, 2015, 27(2-3): 403-430.
- [9] OLIVIER REGNIER-COUDERT, JOHN MECALL, MAYOWA AYODELE, et al. Truck and trailer scheduling in a real world, dynamic and heterogeneous context[J]. *Transportation Research Part E*, 2016(6): 93-98.
- [10] 胡志华. 集装箱码头间互拖的集卡甩挂运输调度问题[J]. *重庆交通大学学报: 自然科学版*, 2013, 32(2): 313-0317.

Scheduling Optimization of Container Tractor Transportation in Circulation Transportation Network

Huang Yong, Yi Yongsong, Qiu Qi

(School of Transportation and Logistics East China Jiaotong University, Nanchang 330013, China)

Abstract: Based on the circulation transportation network, this study establishes the model of tractor optimal operation and designs an improved genetic algorithm according to the heuristic rules to solve. Numerical examples illustrate the effectiveness and feasibility of the proposed model and algorithm. From the comparison, we know that the improved genetic algorithm based on heuristic rules is better than the solving strategy of simulated annealing algorithm with the efficiency of the tractor to complete all tasks increased by 5.54%. The simulation results show that the proposed method possesses high convergence speed and higher quality solutions, which may provide significant guidance for the practical application of scheduling of semi-trailer swap transportation.

Key words: semi-trailer swap transportation; the circulation transportation network; scheduling optimization; genetic algorithm