

文章编号:1005-0523(2014)01-0102-06

铁路车站取送车作业图论模型及算法分析

郭垂江^{1,2}, 雷定猷²

(1. 湖南铁路科技职业技术学院 运输管理学院, 湖南 株洲 412000; 2. 中南大学 交通运输工程学院, 湖南 长沙 410075)

摘要:在设定条件下,以作业点(车站)间机车走行时间为权,把铁路车站取送车作业优化问题转化求解哈密尔顿图最短路问题,设计动态规划法和C-W节约改进算法分别进行求解,并举例比较2种算法的优缺点,提出了两种算法的应用范围。动态规划法计算次数少、能得到最优解,且可选择的方案多,适用于N规模较小情况;C-W节约改进算法虽可能只得到满意解,但能显著降低计算复杂度,适用于N规模较大情况。把车站作业点的3种布置形式统一为树枝形,不会影响算法的有效性。

关键词:铁路车站;取送车作业;哈密尔顿图;动态规划法;C-W节约改进算法

中图分类号:U291.2

文献标志码:A

合理安排货车取送车顺序,有助于加速机车车辆周转,压缩铁路货车在装卸车站的停留时间。为合理安排货车取送车顺序,提高机车车辆作业效率,我国相关领域的研究人员对其进行了一定的研究,并取得了阶段性的研究成果^[1-6]。以前研究只针对某一种布置形式进行研究,没有统一的模型及算法,而且计算复杂,所得到的解的可靠性也难以保证。在综合以前研究的基础上,针对作业点的数目,提出了相应的求解方法,以确保计算的准确性与复杂度的有机统一。

1 模型的建立

货场和专用线是铁路车站的货物装卸作业地点,统称为作业点,按布置方式分为放射形、树枝形和混合形3类^[5]。结合铁路现场实际工作情况,设定如下条件:

- 1) 有1台调车机车作业;
- 2) 各作业点待送、待取车数已定;
- 3) 各作业点的距离或调车机车走行时间已知,确定去货场(专用线)的走行时间以货位中心线为准,且送车时间包括对货位时间,取车时间包括收集车辆时间在内。同一走行线往返机车走行时间相等。
- 4) 机车附挂车辆数量的多少对机车在走行线的走行时间影响忽略不计。
- 5) 调机一批作业向各作业点取送车,规定经过各作业点仅1次,最后调机返回车站。

研究目标是如何安排调车机车一批作业的取送车顺序,使其完成作业点的取送车任务所走行的总时间最小。

如图1, V_0 为铁路车站或技术站调车场, $V_1 \sim V_4$ 为铁路车站的装卸作业点, $a \sim g$ 为分路道岔辙岔心与辙

收稿日期:2013-11-20

作者简介:郭垂江(1980—)男,讲师,博士研究生,主要研究方向为运输组织优化。

岔心(或作业点、车站中心线)间的机车的走行时间。

把问题构造成哈密尔顿图 $G=[V,A,C]$,如图1。图中 $V=\{V_0,V_1,\dots,V_n\}$ 表示调机经过的作业点(车站或调车场中心)编号, $A=\{(i,j)|i,j=0,1,\dots,n,i\neq j\}$ 表示铁路走行线集合; t_{ij} 表示机车在对应段 (i,j) 的走行时间,包括调机在 (i,j) 纯走行时间及在 (i,j) 作业点进行摘挂、对货位等作业所需要的时间,可通过写实法查定。将车站 V_0 视作哈密尔顿图的出发顶点,每一个装卸作业点看作一个需要经过一次的哈密尔顿图顶点,如果找到恰好经过各个装卸作业点并回到车站顶点的一条回路,就得到一条哈密尔顿回路。哈密尔顿回路的权值为机车走行路线上的时间权值 t_{ij} 加总。由于作业点之间是互相连通的,此图是一个完备图,同时也是个无向图,所以对于树枝形作业点构成的哈密尔顿图,其可行的哈密尔顿回路多达 $(N-1)!$ 个(N 是哈密尔顿图的顶点个数)。

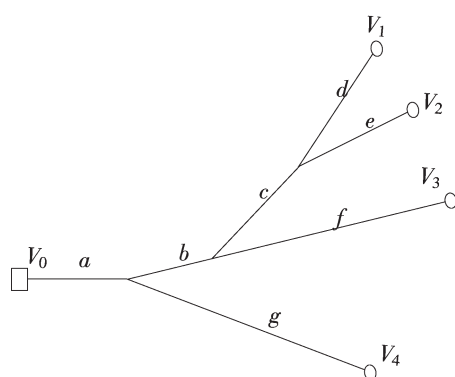


图1 树枝形作业点布置图

Fig.1 Layout of branch-shaped operation sites

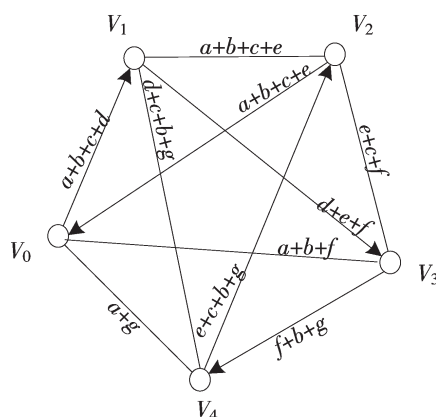


图2 哈密尔顿图

Fig.2 Hamilton graph

3 求解方法

哈密尔顿图最短路回路问题属于组合最优化问题,是典型的 NP 完全问题,当作业点规模较大时,很难得到精确解。本文提出当一批调车作业货物作业点较少时,选择动态规划方法得到最优解;当作业点较多时,利用 $C-W$ 节约改进算法得到满意解或最优解^[7-11]。

3.1 动态规划方法

当作业点 n 较少时,利用动态规划方法求解是比较简便的。把第一个作业点起往下一作业点取送车和返回车站作为作业阶段,因此把整个取送车作业过程划分为 n 个阶段。设 S 表示到达 V_i 作业点之前机车已完成取送作业的作业点(包括车站出发点)集合,则有 $S \subseteq V$ 。因此,可选取 (V_i, S) 作为描述过程的状态变量,状态转移方程为 $s_{k+1} = s_k + V_j$; 决策为机车从作业点选择前往的下一个作业点,允许决策集合 $D_k(s_k) = \{V_j | V_j \in V - (V_i, S)\}$; $P_k(V_i, S)$ 为最优决策函数,它表示从车站 V_0 开始机车经由 k 个装卸作业点的 S 集到 V_i 作业点的最短路线上紧挨着作业点 i 前的作业点;边界条件为 $f_0(V_i, f) = t_{0i}$;最优值函数 $f_k(V_i, S)$ 为从作业点 V_0 开始经由 k 个作业点的 S 集到 V_i 作业点的最小机车走行时间,因此动态规划的递推关系为:

$$f_k(V_i, S) = \min_{j \in S} [f_{k-1}(V_j, S \setminus V_j) + t_{ij}] \quad (k = 1, 2, \dots, n; i = 1, 2, \dots, n; S \subseteq V) \quad (1)$$

采用顺推解法,按机车的作业顺序逐步选择下一个作业点,并推出最优值函数。最后按计算过程的相反顺序逐步确定出每阶段的决策及效益。

3.2 $C-W$ 节约改进算法

3.2.1 算法思想

首先将每个作业点与车站 V_0 相连接,构成线路 $V_0 \rightarrow V_j \rightarrow V_0$ ($j = 1, 2, \dots, n$),这样就得到了一个具有 n

条线路的图(此时尚未形成哈密尔顿回路)。机车按此线路向 n 个点作业所花费的时间总和 z 为:

$$z = 2 \sum_{j=0}^n t_{0j} \quad (2)$$

其中: t_{0j} 为由车站 V_0 到作业点 V_j ($j=1,2,\dots,n$) 的机车走行时间。

若机车直接从作业点 V_i 到 V_j ($i,j \neq 1$), 即不再经过弧 (V_i, V_0) 和 (V_0, V_j) , 直接走弧 (V_i, V_j) 时所引起的走行时间节约值 $t(i,j)$ 可计算如下

$$t(i,j) = 2t_{0i} + 2t_{0j} - (t_{0i} + t_{0j} + t_{ij}) = t_{0i} + t_{0j} - t_{ij} \quad (3)$$

对不同的作业点对 (V_i, V_j) , $t(i,j)$ 越大, 机车通过弧 (V_i, V_j) 时所节约的时间越多, 因而应优先将其安排到走行线路上, 使机车不必再返回车站 V_0 , 而是直接到达下一作业点 V_j 。

3.2.2 算法步骤

可按以下步骤进行求解:

Step1: 将机车出发点 V_0 与其他各作业点连接, 得到 n 条线路 $V_0 \rightarrow V_j \rightarrow V_0$ ($j=1,2,\dots,n$)。

Step2: 对不违背限制条件的所有可作业点对 (V_i, V_j) , 如下计算其节约值 $(V_i, V_j$ 不为出发点 $V_0)$:

$$t(i,j) = t_{0i} + t_{0j} - t_{ij}。$$

Step3: 按 $t(i,j)$ 值由大到小顺序, 逐个考查其端点 V_i 和 V_j , 若满足点 V_i 和点 V_j 不在一条线路上且作业点 V_i 和 V_j 均与车站 V_0 相邻的条件, 就将弧 (V_i, V_j) 插入到线路中, $t(i,j)$ 值相同的点检查顺序没有限制。

Step4: 线路已包含了所有作业点时算法终止。

通过以上各迭代步骤, 使问题的解逐步得到改善, 最后达到满意解(也有可能达到最优解)。

放射形作业点、混合形作业点布置形式都可归结为树枝形作业点布置形式, 如图3, 铁路车站(调车场)中心线到虚拟道岔间的机车走行时间为

$$d = \begin{cases} t_{\text{调车线}}/2 & \text{当线束不在车站同端时} \\ 0 & \text{当线束在车站同端时} \end{cases}$$

其中: $t_{\text{调车线}}$ 为机车在调车线的走行时分。

对树枝形作业点取送车问题讨论所得出的算法同样也适合其它作业点布置形式。

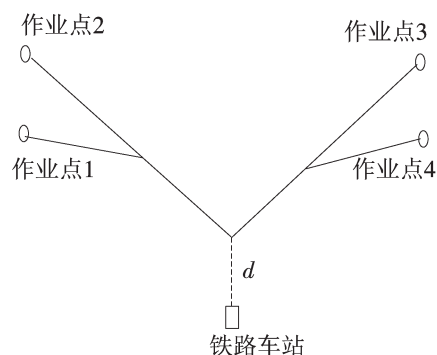


图3 车站货物作业点布置形式统一图

Fig.3 Layout diagram of station goods operation sites

4 算例

某铁路车站一批调车作业需向其取送车的作业点布置示意图如图4, 取送车车数和道岔与道岔(作业点或车站)间机车走行时分已标出, 现确定这批作业效率较高的取送车顺序。

按以上所讨论的方法, 把该铁路车站取送车问题转化为哈密尔顿图问题, 如图5所示。分别应用3.1和3.2节所述方法进行求解。

4.1 应用动态规划求出最优解

由边界条件可知: $f_0(V_1, f) = t_{01} = 33$, $f_0(V_2, f) = t_{02} = 31$, $f_0(V_3, f) = t_{03} = 32$, $f_0(V_4, f) = t_{04} = 22$ 。

当 $k=1$ 时, 即从车站 V_0 开始, 中间经过一个作业点到达作业点 V_j 的最短距离是: $f_1(V_1, \{V_2\}) = f_0(V_2, \varphi) + d_{21} = 47$, $f_1(V_1, \{V_3\}) = 65$, $f_1(V_1, \{V_4\}) = 57$, $f_1(V_2, \{V_1\}) = 49$, $f_1(V_2, \{V_3\}) = 63$, $f_1(V_2, \{V_4\}) = 55$, $f_1(V_3, \{V_1\}) = 66$, $f_1(V_3, \{V_2\}) = 62$, $f_1(V_3, \{V_4\}) = 56$, $f_1(V_4, \{V_1\}) = 68$, $f_1(V_4, \{V_2\}) = 64$, $f_1(V_4, \{V_3\}) = 56$ 。

当 $k=2$ 时, 即从车站 V_0 开始, 中间经过2个作业点到达作业点 V_j 的最短距离是: $f_2(V_1, \{V_2, V_3\}) =$

$\min[f_1(V_2,\{V_3\})+d_{21}f_1(V_3,\{V_2\})+d_{31}]=79$, $f_2(V_1,\{V_2,V_4\})=71$, $f_2(V_1,\{V_3,V_4\})=89$, $f_2(V_2,\{V_1,V_3\})=81$, $f_2(V_2,\{V_1,V_4\})=73$, $f_2(V_2,\{V_3,V_4\})=87$, $f_2(V_3,\{V_1,V_2\})=80$, $f_2(V_3,\{V_1,V_4\})=90$, $f_2(V_3,\{V_2,V_4\})=86$, $f_2(V_4,\{V_1,V_2\})=82$, $f_2(V_4,\{V_1,V_3\})=100$, $f_2(V_4,\{V_2,V_3\})=96$ 。

当 $k=3$ 时,即从车站 V_0 开始,中间经过3个作业点到达作业点 j 的最短距离是: $f_3(V_1,\{V_2,V_3,V_4\})=\min[f_2(V_2,\{V_3,V_4\})+d_{21}f_2(V_3,\{V_2,V_4\})+d_{31}f_2(V_4,\{V_2,V_3\})+d_{41}]=103$, $f_3(V_2,\{V_1,V_3,V_4\})=105$, $f_3(V_2,\{V_1,V_3,V_4\})=105$, $f_3(V_3,\{V_1,V_2,V_4\})=104$, $f_3(V_4,\{V_1,V_2,V_3\})=114$ 。

当 $k=4$ 时,即从车站 V_0 开始,中间经过4个作业点回到车站 V_0 的最短距离是: $f_4(V_0,\{V_1,V_2,V_3,V_4\})=\min[f_3(V_1,\{V_2,V_3,V_4\})+d_{10}f_3(V_2,\{V_1,V_3,V_4\})+d_{20}$, $f_3(V_3,\{V_1,V_2,V_4\})+d_{30}f_3(V_4,\{V_1,V_2,V_3\})+d_{40}]=136$ 。

反方向逐步确定的最佳取送车顺序为: $V_0 \rightarrow V_1 \rightarrow V_2 \rightarrow V_3 \rightarrow V_4 \rightarrow V_0$; $V_0 \rightarrow V_2 \rightarrow V_1 \rightarrow V_3 \rightarrow V_4 \rightarrow V_0$; $V_0 \rightarrow V_3 \rightarrow V_1 \rightarrow V_2 \rightarrow V_4 \rightarrow V_0$; $V_0 \rightarrow V_4 \rightarrow V_1 \rightarrow V_2 \rightarrow V_3 \rightarrow V_0$; $V_0 \rightarrow V_4 \rightarrow V_3 \rightarrow V_2 \rightarrow V_1 \rightarrow V_0$; $V_0 \rightarrow V_4 \rightarrow V_3 \rightarrow V_1 \rightarrow V_2 \rightarrow V_0$;以上取送车顺序的逆序: $V_0 \rightarrow V_4 \rightarrow V_2 \rightarrow V_1 \rightarrow V_3 \rightarrow V_0$; $V_0 \rightarrow V_3 \rightarrow V_2 \rightarrow V_1 \rightarrow V_4 \rightarrow V_0$ 也是最佳顺序。因此,应用动态规划法可得到8种最优方案供决策者选择,其最优函数值为: $z^*=136$ 。

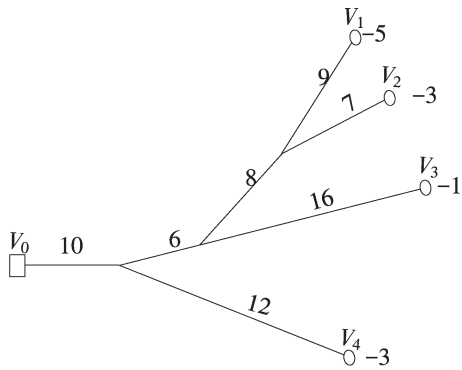


图4 某站装卸作业点布置示意图

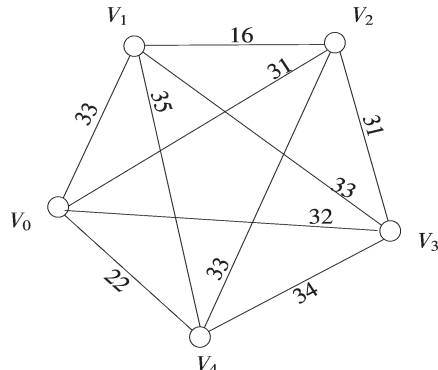


图5 某站装卸作业点哈密尔顿图

Fig. 4 Schematic layout diagram of loading and unloading operation sites for a station

Fig.5 Hamilton graph of loading and unloading operation sites for a station

4.2 应用改进 C - W 节约算法求解

以 V_0 为基点,构成初始机车走行线路图。再用式(3)计算弧 (V_i,V_j) ($V_i,V_j \neq 0$) 插入到线路中时引起的机车走行时间节省值 $t(i,j)$,填入表1中。

表1 插入弧 (V_i,V_j) 的时间节省值

Tab.1 Saving time of inserted arc (V_i,V_j)

序号	弧	$t(i,j)$	序号	弧	$t(i,j)$
1	(V_1,V_2)	48	4	(V_2,V_3)	32
2	(V_1,V_3)	32	5	(V_2,V_4)	20
3	(V_1,V_4)	20	6	(V_3,V_4)	20

按节约值的从大到小的顺序,对每条弧 (V_i,V_j) 加以考查,看能否将其插入到旅行线路中。若能将其插入,就对机车走行线路作相应的改变。当插入弧 (V_1,V_4) 或 (V_3,V_4) 后,线路已包含所有作业点,此时算法终止。用该方法得到的机车走行线路是: $V_0 \rightarrow V_4 \rightarrow V_1 \rightarrow V_2 \rightarrow V_3 \rightarrow V_0$ 。该条线路的总长度为: $z^*=136$ 。同时也可得到了其它较优取送车顺序 $V_0 \rightarrow V_1 \rightarrow V_2 \rightarrow V_3 \rightarrow V_4 \rightarrow V_0$; $V_0 \rightarrow V_4 \rightarrow V_3 \rightarrow V_2 \rightarrow V_1 \rightarrow V_0$; $V_0 \rightarrow V_3 \rightarrow V_2 \rightarrow V_1 \rightarrow V_4 \rightarrow V_0$ 供决策者选择。

4.3 两种计算方法比较

应用两种方法计算案例的情况比较如表2。可以看出应用动态规划方法求解较 $C-W$ 节约改进算法计算次数少、可选择的方案多,而且能得到最优解,它其实是一种隐枚举法。但是随着 N 的增加,应用动态规划法时的计算次数呈指数级增加,达一定程度时会超过人的忍受程度;应用 $C-W$ 节约改进算法虽可能只得到满意解,但计算次数增加速度慢得多。

表2 两种计算方法的优劣比较

Tab. 2 Pros and cons of the calculation methods

特性	动态规划法	$C-W$ 节约改进算法
计算次数	加减56次,比较17次	加减12次,比较6次
可得方案数/个	8	4
解的可靠性	最优解	满意解(也有可能得到最优解)
适应范围	N 规模较小时	N 规模较大时

5 结论

提出了根据一批调车作业涉及的作业点 N 规模的大小选择合适的计算方法,即 N 规模较小时选择动态规划法进行求解,而对于 N 规模较大时选择 $C-W$ 节约改进算法求解。与以往研究相比较,本文采用的方法计算简便、得到最优解的可能性大,且可得到几个最优或满意的取送车方案,为决策者提供了多种选择。

参考文献:

- [1] 雪友诚,涂祖耀,桂卫华,等. 基于遗传蚁群算法的树枝型铁路取送车问题优化[J]. 中南大学学报:自然科学版,2011,48(2):2356-2361.
- [2] 石红国,彭其渊,郭寒英. 树枝型专用线取送车问题的哈密尔顿图解法[J]. 中国铁道科学,2005,26(2):132-135.
- [3] 李海军,朱昌锋. 放射形铁路专用线直达车流取送车问题的单亲遗传算法研究[J]. 铁道科学与工程学报,2011,8(6):114-117.
- [4] 黄向荣. 树枝型专用线取送车的模型及算法研究[J]. 兰州交通大学学报:自然科学版,2007,26(3):51-54.
- [5] 杨运贵,王慈光,薛锋. 树枝形铁路专用线取送车问题的遗传算法研究[J]. 计算机工程与应用,2008,44(12):210-211.
- [6] 王慈光. 运输模型及优化[M]. 北京:中国铁道出版社,2004:8.
- [7] 王更生,俞云新,蔡求元,尹慧. 基于DACS3的改进蚁群算法求解TSP问题[J]. 华东交通大学学报,2010,27(2):57-62.
- [8] 陈冬华. 旅行商问题推广及其混合智能算法[J]. 华东交通大学学报,2011,28(2):102-106.
- [9] HELD M, KARP R M. A dynamic programming approach to sequencing problems, SIAM Journal on Applied Mathematics, 1962,10:196-210.
- [10] 《运筹学》教材编写组. 运筹学[M]. 3版. 北京:清华大学出版社,2005:193-210.
- [11] 杨凌杰,李静. 基于 $C-W$ 节约算法的TSP教学辅助系统研究与设计[J]. 微计算机应用,2008,29(9):22-25.

Model and Algorithm of Wagons' Placing-in and Taking-out in Railway Station

Guo Chuijiang^{1,2}, Lei Dingyou²

(1. Department of Transportation Management, Hunan Vocational College of Railway Technology, Zhuzhou 412000, China; 2. School of Traffic and Transportation Engineering, Central South University, Changsha 410075, China)

Abstract: By taking the locomotive running time between operating points (stations) as weights, the optimization problem of placing-in and taking-out of wagons is transformed into the shortest path problem of Hamilton map under the set conditions, which can be solved by dynamic programming method and the improved Clarke-Wright saving algorithm. This paper generalizes the advantages and disadvantages of the above two algorithms with examples, providing the application scopes simultaneously. Dynamic programming method has characteristics of fewer calculation times, getting the optimal solution, possessing many alternative solutions, and it is applied to the case of small operating scale. Improved Clarke-Wright saving algorithm may get satisfactory solution, but it can significantly reduce the computational complexity, and it is applied to the case of larger scale. The three kind of arrangement forms (radical, branch-shaped, mixed) of loading and unloading points are unified into branch-shaped, which may not affect the validity of the algorithms.

Key words: railway station; placing-in and taking-out operation of wagons; Hamilton graph; dynamic programming method; improved Clarke-Wright saving algorithm