一种基于数据挖掘的制造业工厂设备布局方法

宋中山,陈雯颖,孙 翀,帖 军
(中南民族大学 计算机科学学院,武汉 430074)

摘 要 针对经典的求解单行直线型布局算法中需要大量参数、要求设备等概率使用的限制,提出了一种基于数据挖掘的制造业工厂设备布局方法 FMDM. FMDM 采用数据挖掘 Apriori 算法对已有的生产调度计划或柔性作业车间调度问题的调度解进行挖掘,根据贪心方法在频繁项的基础上获得的初步布局方案,给出了将候选方案进行筛选得到最终方案的算法 CACULATE_EDIT_DISTANCE. 实验结果表明:该方法可对无参数的初建车间进行有效的初步布局,不限制设备的使用概率,能实现多工件共享设备,多工件并发生产,且 FMDM 结果作为经典算法的输入可提高经典算法的收敛速度.

关键词 数据挖掘,Apriori 算法,贪心算法,直线型布局
中图分类号 TP301 **文献标识码** A **文章编号** 1672-4321(2017)04-0106-06

Facility Layout Method for Manufacturing Based on Data Mining

Song Zhongshan, *Chen Wenying*, *Sun Chong*, *Tie Jun*
(College of Computer Science, South-Central University for Nationalities, Wuhan 430074, China)

Abstract The algorithm for the design of a straight-line layout has been widely employed to solve facility layout problems. However, this algorithm requires sufficient parameters and demands equal probability of machines being used. This paper accordingly proposes an approach based on data mining to manufacturing facility layout, named as FMDM. FMDM firstly adopts the Apriori Algorithm for mining the existing production scheduling plan. Then a greedy algorithm is applied to the frequent item set, resulting in layout plans. Lastly, an algorithm is provided to select the best plan from these layout options. Experimental evidence shows that FMDM can be applied to a newly-built workshop's facility layout planning without parameters and regardless of the probability of machines' use. This approach helps to achieve multiple jobs on the shared device and concurrent production. In addition, the rate of convergence can be increased through inputting the result of FMDM to traditional algorithms.

Keywords data mining, Apriori algorithm, greedy algorithm, single straight line layout

设备布局问题是制造系统常见问题之一,与企业的生产率和生产成本密切相关. 产品从原材料进厂到出厂的整个生产周期中处于加工检验的时间仅占整个生产周期的 5% ~10%,而处于停滞和搬运的时间占 90% ~95%;从费用来看,总经营费用的 20% ~50% 是物料搬运费,这严重影响了企业的生产效率. 合理的设施布局可以使制品过程顺畅流通,缩短产品在工位与工位之间的运送时间,使物料搬运费用减少 10% ~30%,节约企业成本,缩短生产周期,使生产消耗最小化,生产效益最大化[1]. 所以机器设备布局问题成为制造业亟待解决的问题.

在遗传算法被提出后,陆续出现了蚁群算法、模拟退火算法、禁忌搜索算法等各种算法在设备布局问题上的研究与应用[2-4]. 文献[5]根据车间的设备布局参数及物流信息,运用遗传算法优化目标函数求解布局方案,文献[6]将原车间作为初始种群,采

收稿日期 2017-09-15
作者简介 宋中山(1963-),男,副教授,研究方向:数据挖掘和计算机网络,E-mail: songzs@scuec.edu.cn
基金项目 国家科技支撑计划项目子课题(2015BAD29B01);中央高校基本科研业务费专项资金资助项目(CZY16002)

用遗传算法进行优化设计,通过选择、交叉、变异等操作得到更合理的车间布局方案. 这些算法虽然可以很好的求解布局最优解,但是模型要求提供的参数却不易获得,特别在车间建成初期. 本文提出一种基于数据挖掘的制造业工厂设备布局方法(FMDM, Facility Layout Method for Manufacturing Based on Data Mining),在已有的生产调度计划或柔性作业车间调度问题(FJSP)的调度解上调用数据挖掘算法[7],不需要各种设备参数,帮助工厂对建设中的车间进行设备布局规划. FMDM 克服了经典算法基于设备等概率使用的限制,考虑了实际生产中多工件共享设备,多工件并发的情况. 另一方面,考虑到经典求解算法对初始解的依赖性比较大,如禁忌搜索算法,不合理的初始解不容易找到较优的布局,而好的初始解可以极大地提高算法效率. 本文将 FMDM 的结果作为初始解来运行禁忌搜索算法,可以提高经典算法收敛速度.

1　问题描述

在设备布局问题中,单行设备布局问题是比较特殊的一种布局形式,其物料传输时间短且费用低. 单行布局将所有的设备按照一定的要求安排在一行内,满足产品加工需求,采用搬运物料总距离最短作为衡量该布局模型的优化目标. 单行直线型布局可描述为 n 台设备$\{M_1, M_2, \cdots, M_n\}$ 按照一定的方向沿一条直线放置的线性排列,如图 1 所示.

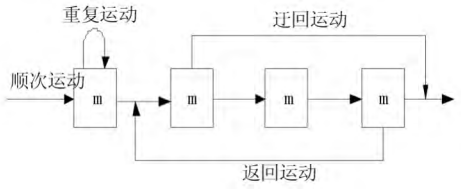

图 1　单行线型布局的物流运动方式

Fig. 1　Logistics movement of Single Straight line layout

本文在研究中做以下假设和抽象:

(1) 所有设备由其中心点代表,忽略其具体细节形状,忽略其长和宽的长度.

(2) 所有设备摆放时与车间的长度方向平行,代表机器的中心点在一条水平直线上.

(3) 设备之间的物流路径平行于车间长度方向,且设备等距离摆放,任意两个相邻的机器之间的距离记为单位长度 1.

令 $M = \{M_1, M_2, \cdots, M_n\}$ 为待布局的设备集,$j = \{j_1, j_2, \cdots, j_m\}$ 为待加工工件集,$J = \{J_1, J_2, \cdots, J_m\}$ 为工件加工工序集合,J_i 表示工件 j_i 的加工序列.

在单行线型布局中常见的物流运动方式有以下 4 种: 重复运动(在一台设备上重复加工);顺次运动(按照加工顺序依次在设备上加工);迂回运动(跳过其中一台或几台设备进行加工);返回运动(先在后面的设备上加工再在前面的设备上加工).

n 台设备间物流方向上的编辑距离矩阵 D 为:

$$D = \begin{vmatrix} D_{11} & D_{12} & \cdots & D_{1n} \\ D_{21} & D_{22} & \cdots & D_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ D_{n1} & D_{n2} & \cdots & D_{nn} \end{vmatrix},$$

$$d = \sum_{\{i,j \in PQ\}} D_{ij} \quad .$$

d 指工件按加工工序产生的编辑距离,其中 PQ 是指工件加工过程中所经过的设备组.

本文的研究目标可概括为: 找到在加工过程中,工件物流运动成本最低的设备布局序列. FMDM 根据工件加工工序计算候选设备序列的编辑距离 d,使用此编辑距离作为该设备布局序列对应的搬运物料总距离,算法的目标函数即转换为找到使得 d 最短的设备序列,编辑距离越小设备序列越合理.

2　解决方案

FMDM 算法利用 FJSP 的调度结果或已有的生产调度计划,通过数据挖掘 Apriori 算法产生设备布局序列,对初建车间或计划中的车间进行设备布局规划[8]. FMDM 的主要步骤包括以下内容: (1) 清理柔性工厂工作调度数据并生成事务数据库; (2) 使用 Apriori 算法挖掘事务数据库中的频繁项; (3) 调用 *GREEDY_SET_COVER* 生成初步候选布局方案,调用 *CACULATE_EDIT_DISTANCE* 算法选择最终直线设备布局方案.

2.1　清理数据并生成事务数据库

算法的数据对象是 FJSP 调度结果集或已有的生产调度计划,首先对生产调度结果进行清理,将无效的和不完整的数据删除,保证后续算法的准确性.

将生产调度结果作为一个事务数据库,每条柔性作业调度结果或生产调度计划元组转换为事务数据表中的若干元组. 转换方法如下. 首先读取一条调度结果或生产调度计划,获得当前调度结果集中的工件集合及对应的工序码和设备码. 然后取出当前

工序集合中的一个工序,重复下列动作直至当前工序集为空:为此工序新建一个事务元组并分配一个唯一的事务ID号,接着根据此工序和调度结果集中的设备码获取当前工件的设备序列项集,并将该序列项集写入事务项属性. 当所有的调度结果都按上述过程处理完毕后,输出事务数据库. 经过上述处理后事务数据表最终为二维表,包含事务ID属性和事务项属性.

FJSP结果的一个元组如图2所示.

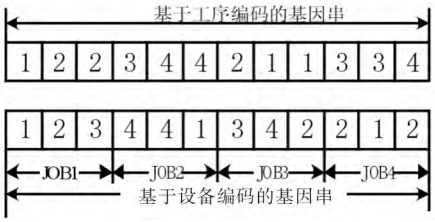

图2 柔性作业调度元组

Fig. 2 Tuples of flexible job-shop scheduling

在图2中,基于设备编码的基因串显示设备集为 $M=\{M_1, M_2, M_3, M_4\}$,待加工工件集 $j=\{j_1, j_2, j_3, j_4\}$,对应的工件加工工序集为 $J=\{J_1, J_2, J_3, J_4\}$,其中 $J_1=M_1,M_2,M_3$,$J_2=M_4,M_4,M_1$,$J_3=M_3,M_4,M_2$,$J_4=M_2, M_1,M_2$. 根据上述步骤生成事物数据库 D,如表1所示.

表1 事务数据库D

Tab. 1 Transactional database D

ID	事务项
1	M_1, M_2, M_3
2	M_4, M_1
3	M_3, M_4, M_2
4	M_1, M_2

2.2 使用Apriori算法挖掘频繁项

对2.1中产生的事务数据库调用Apriori算法,挖掘事务数据库中的频繁项[8],Apriori算法对以工件为ID,以加工设备序列为事务项的事务数据库进行频繁项挖掘,其结果包含了加工设备之间的关联,而这些关联以工件的加工序列为依据,以此建立加工序列和设备布局之间的联系. 此步骤利用已有的知识和经验对设备布局的求解过程进行优化和简化,充分考虑了工件加工顺序对设备布局的影响,由此得出的设备布局方案也将有利于工件的加工过程. 计算过程如下.

步骤1:顺序扫描事务数据库,根据最小支持度获得频繁一项集.

步骤2:产生 $k+1$ 项集. 将前 $k-1$ 项相同的 k 项集两两连接生产候选的 $k+1$ 项集,生成完所有的候选项后,以频繁 $k+1$ 项集的任意规模为 k 的子集必须在频繁 k 项集中为剪枝原则进行剪枝. 顺序扫描事务数据库,检查 $k+1$ 项候选集中的每个候选项,删除支持度小于最小支持度的候选项.

步骤3:重复步骤2直至不再产生新的频繁项集.

令最小支持度为2,表1中事务数据库 D,如图3所示,产生的频繁一项集有 $\{\{M_1\},\{M_2\},\{M_3\},\{M_4\}\}$,频繁2项集有 $\{\{M_1,M_2\},\{M_2,M_3\}\}$.

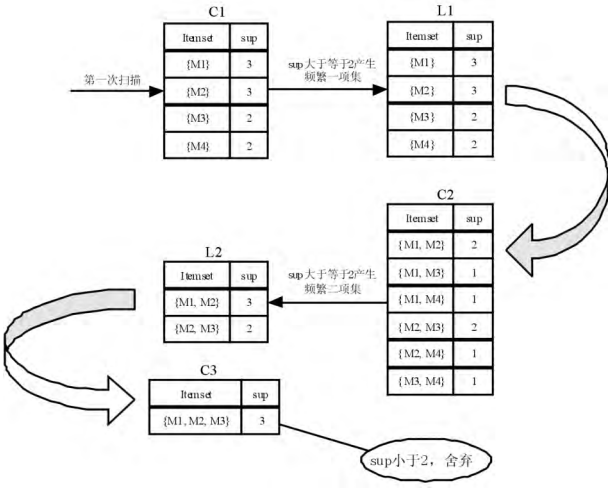

图3 AP算法求解D中频繁项集过程

Fig. 3 Process of AP algorithm in database D

2.3 根据频繁项生成直线设备布局方案

由于在上一步产生的频繁项集中并不一定存在某个频繁项是等于设备集的,因此需要在频繁项集合中找到能覆盖设备集的最小频繁项集合. FMDM算法在此步中运用集合覆盖的贪心算法对频繁项集进行选择,调用 $GREEDY\_SET\_COVER(M,F)$ 形成初步候选布局方案. 此步骤的目标是找到能覆盖所有设备的频繁项集,且被选频繁项集的个数尽量小. 贪心策略为:每次选择能覆盖最多尚未被覆盖设备元素的频繁项集.

示例如下,在上一步产生的频繁项集 $F=\{\{M_1\},\{M_2\},\{M_3\},\{M_4\},\{M_1,M_2\},\{M_2,M_3\}\}$ 中进行选择,调用 $GREEDY\_SET\_COVER(M,F)$ 找到最小规模的子集 $X\subseteq F$ 可以覆盖集合 M.

$GREEDY\_SET\_COVER(M,F)$

输入:设备集 M,D 中的频繁项集 F

输出:F 的子集 X

1. $U = M$
2. $X = \emptyset$
3. while $U \neq \emptyset$
4. select an $Q \in F$ that maximizes $|Q \cap U|$
5. $U = U - Q$
6. $X = X \cup \{Q\}$
7. End while
8. Return X

算法 $GREEDY\_SET\_COVER(M,F)$ 的时间复杂度为 $O(|F|\min(|M|,|F|))$.

按照 $GREEDY\_SET\_COVER(M,F)$ 算法的描述,图 2 所示调度结果集在其频繁项集 F 中找到的覆盖集合设备集 M 的子集 X 为$\{\{M_1, M_2\}, \{M_3\}, \{M_4\}\}$. X 即为初步的候选布局方案.

在初步候选方案 X 中的设备仍是无序的,下面对 X 中每个频繁项内的设备以及 X 中各频繁项之间进行排序,以此确定每个设备的位置形成最终的布局方案. 首先调整频繁项内设备顺序,调整方案为将每个频繁项内的设备序列按其在事务数据库中出现频度非降序排序. 例如,X 中的频繁项$\{M_1, M_2\}$,由于 M_1 与 M_2 的出现频度相同,可随机排列,在此例中取$\{M_1, M_2\}$ 这一顺序. X 中剩余两个频繁项均为一项集,无需调整.

下一步调整布局方案 X 中各频繁项之间的顺序. 根据 X 中的频繁项集,随机产生一个频繁项集序列,并利用事务数据库计算该序列的编辑距离值,具体方法如下.

$CACULATE\_EDIT\_DISTANCE(X, J, M)$

输入: 集合 X,工件加工工序集 J,设备集 M

输出: 设备序列 A 和对应工件加工工序集 J 的编辑距离 s

1. $t = X.length$
2. $m = J.length, n = M.length$
3. let $A[1..n]$ be an new array
4. for $i = 1$ to t
5. swap $X[i]$ with $X[\text{RANDOM}(i,t)]$
6. End for
7. put the index of frequent items of X in A
8. return A
9. let $D[1:n][1:n]$ be an new matrix
10. for $i = 1$ to n
11. for $j = 1$ to n
12. $D[A[i]][A[j]] = |j - i|$
13. End for
14. End for
15. let $S = [1..m]$ be an new array
16. for $i = 1$ to m
17. $S[i] = 0$
18. If $|Ji| = 1$ then
19. $S[i] = 0$
20. End if
21. Else if $|Ji| > 1$ then
22. for $k = 1$ to $|Ji| - 1$
23. $S[i] = S[i] + C[Ji,k][Ji,k+1]$
24. End for
25. End if
26. End for
27. $s = 0$
28. for $j = 1$ to m
29. $s = s + S[j]$
30. End for
31. return s

反复多次调用 $CACULATE\_EDIT\_DISTANCE(X,J,M)$,选择编辑距离 s 最小的输出数组 A 即为所得.

m 为待加工工序数,n 为待布局的设备数量,$CACULATE\_EDIT\_DISTANCE(X,J,M)$ 整体的时间复杂度为 $O(max(m,n) \times n)$.

下面举例说明如何调整频繁项集之间的布局顺序. 调用 $CACULATE\_EDIT\_DISTANCE$ 算法,调整布局方案 X 中包含的 3 个频繁项集之间的顺序,产生随机序列,以及根据加工工序计算这些随机序列的编辑距离值.

(1) $\{M_1, M_2\}$,$\{M_3\}$,$\{M_4\}$,产生的设备序列为$\{M_1, M_2, M_3, M\|_4\}$,每个工序步骤的编辑距离为$\{0,1,1\}$,$\{0,3\}$,$\{0,1,2\}$,$\{0,1\}$,此设备序列的编辑距离为 9;

(2) $\{M_1, M_2\}$,$\{M_4\}$,$\{M_3\}$,产生的设备序列为$\{M_1, M_2, M_4, M\|_3\}$,每个工序步骤的编辑距离为$\{0,1,2\}$,$\{0,2\}$,$\{0,1,1\}$,$\{0,1\}$,此设备序列的编辑距离为 8;

(3) $\{M_3\}$,$\{M_1, M_2\}$,$\{M_4\}$,产生的设备序列为$\{M_3, M_1, M_2, M\|_4\}$,每个工序步骤的编辑距离为$\{0,1,2\}$,$\{0,2\}$,$\{0,3,1\}$,$\{0,1\}$,此设备序列的编辑距离为 10;

(4) $\{M_3\}$,$\{M_4\}$,$\{M_1, M_2\}$,产生的设备序列为$\{M_3, M_4, M_1, M_2\}$,每个工序步骤的编辑距离为$\{0,1,3\}$,$\{0,1\}$,$\{0,1,2\}$,$\{0,1\}$,此设备序列的编

辑距离为9;

(5)$\{M_4\}$,$\{M_3\}$,$\{M_1, M_2\}$,产生的设备序列为$\{M_4, M_3, M_1, M\|_2\}$,每个工序步骤的编辑距离为{0,1,2},{0,2},{0,1,3},{0,1},此设备序列的编辑距离为10;

(6)$\{M_4\}$,$\{M_1, M_2\}$,$\{M_3\}$,产生的设备序列为$\{M_4, M_1, M_2, M_3\}$,每个工序步骤的编辑距离为{0,1,1},{0,1},{0,3,2},{0,1},此设备序列的编辑距离为9.

通过比较每个设备序列的编辑距离,距离最小的是方案2)为8,即最优的设备序列为$\{M_1, M_2, M_4, M_3\}$.

3 测试结果

算法FMDM对求解过程中涉及的参数要求较低,而大部分的设备布局问题求解算法都对设备参数和过程中的物流参数要求极高,这是FMDM明显优于其他算法的地方,上述实例说明了FMDM的可行性.更重要的是,FMDM算法的结果可以作为经典算法的初始解起到提高算法收敛速度,减少计算时间的作用.以下实验将以FMDM的结果作为初始解来进行禁忌搜索算法,与经典算法进行比较.

利用Python编写相关程序,设最小支持度分别为2,3,3,3,4,4,4,4,使用来自单行设备布局研究算例库的算例,在Intel(R) Core(TM) i5-2400 CPU3.10GHZ,内存8G的PC上进行测试,FMDM作为禁忌搜索初始解与其他经典算法进行比较,结果如表2和图4所示.从图5和图6中可看出当设备的规模为中小型时,FMDM作为禁忌搜索初始解的算法比其他算法的收敛速度明显提高,缩短了计算时间.

表2 算例求解对比结果

Tab.2 Comparision result of different algorithms

算例名称	机器台数	运行时间/s			
		Love and Wang[9]	Amaral[10]	MIP3[11]	FMDM+禁忌搜索
P4	4	0.03	0.02	0.02	0.01
S8	8	6.04	0.89	1.22	0.01
S9	9	18.68	3.88	3.37	0.021
S9H	9	325.7	36.53	3.27	0.023
S10	10	40.55	8.34	7.92	0.033
S11	11	555.78	29.5	19.19	0.045
LW11	11	1317.88	45.78	17.85	0.049
P15	15	22912.3	2357.9	1267.9	0.054

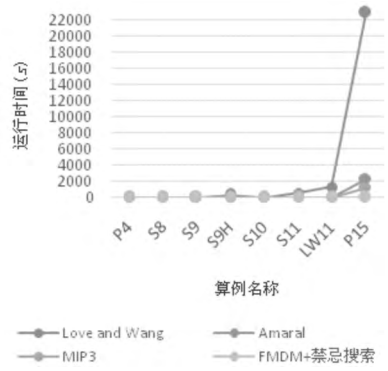

图4 算例求解对比结果

Fig.4 Comparision result of different algorithms

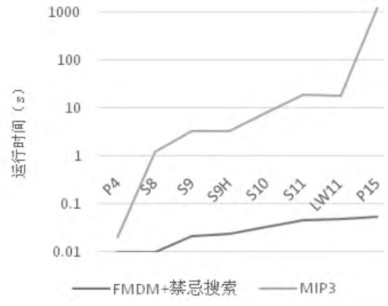

图5 FMDM+禁忌搜索与MIPS对比结果

Fig.5 Comparision result of FMDM+Tabu search and MIPS

图6 FMDM+禁忌搜索与Amaral对比结果

Fig.6 Comparision result of FMDM+Tabu search and Amaral

4 结语

基于数据挖掘的制造业工厂设备布局方法FMDM,通过清理柔性工厂工作调度结果并生成事物数据库,使用Apriori算法挖掘事务数据库中的频繁项,根据频繁项使用*GREEDY_SET_COVER*方法生成直线设备布局方案,利用事务数据库调用*CACULATE_EDIT_DISTANCE*计算该方案的编辑距离值,反复多次选择最优方案.FMDM克服了经典的求解单行直线型布局过程中需要提供各种参数的问题,基于设备等概率使用的限制,考虑了实际生产中

多工件共享设备,多工件并发的情况,实现车间在建设或规划初期对设备进行布局. FMDM 算法结果作为经典算法如禁忌搜索算法的初始解可提高算法的收敛速度.

参 考 文 献

[1] Marcello Braglia, Simone Zanoni, Lucio Zavanella. Layout design in dynamic environments: Strategies and quantitative indices [J]. International Journal of Production Research, 2003, 41(5): 995-1016.

[2] 梁勤欧, 周晓艳. 基于免疫遗传算法的设备布局问题研究 [J]. 武汉理工大学学报(信息与管理工程版), 2011, 33(04): 643-646 +655.

[3] 左兴权, 王春露, 赵新超. 一种结合多目标免疫算法和线性规划的双行设备布局方法 [J]. 自动化学报, 2015, 41(3): 528-540.

[4] 郑永前, 项德海. 基于单向环形方式的制造单元布局方法 [J]. 计算机集成制造系统, 2013, 19(06): 1224-1231.

[5] 邱胜海, 陈曙鼎, 王云霞, 等. 遗传算法在车间设施布局优化中的应用 [J]. 机械设计与制造工程, 2017, 46(2): 80-83.

[6] 杨国俊, 陈　健, 孙思蒙, 等. 基于遗传算法的车间布局优化研究 [J]. 机械研究与应用, 2016, 29(1): 12-14.

[7] 刘　琼, 张超勇, 饶运清, 等. 改进遗传算法解决柔性作业车间调度问题 [J]. 工业工程与管理, 2009, 14(2): 59-66.

[8] Han Jia-wei, Kamber M. 数据挖掘概念与技术 [M]. 范　明, 孟小峰, 译. 北京: 机械工业出版社, 2001: 158-166.

[9] LOVE R F, WONG J Y. On solving a one-dimensional space allocation problem with integer programming [J]. Information Processing and Operations Research (INFOR), 1976, 14(2): 139-143.

[10] AMARAL R S. On the exact solution of a facility layout problem [J]. European Journal of Operational Research, 2006, 173(2): 508-518.

[11] AMARAL R S. An exact approach to the one-dimensional facility layout problem [J]. Operations Research, 2008, 56(4): 1026-1033.