

无缓存流水线生产系统中AGV调度问题的研究

汪星恺^{1,2}, 吴维敏^{1,2}, 邢子超^{1,2}, 牛昊一^{1,2}

(1. 浙江大学工业控制技术国家重点实验室, 浙江杭州 310058;

2. 浙江大学控制科学与工程学院, 浙江杭州 310058)

摘 要: 针对无缓存流水线生产系统中的AGV(automated guided vehicle)调度问题提出了复合评分的启发式调度算法,降低了系统中AGV执行任务的间隔等候时间. 首先,通过建立数学模型,对目前主流的AGV规模估计方法做出了改进. 其次,提出了一种新的基于复合评分禁忌搜索的AGV前瞻调度算法. 不同于目前主流的以空驶距离为优化目标的调度算法,该算法能够以最小化工件延误时间、最小化AGV空驶距离等多目标来统筹调度AGV. 最后,为验证所提算法的有效性,从多个角度与已有算法进行了实验对比. 仿真实验结果表明,相较于其他算法,提出的前瞻调度算法能够更有效地解决无缓存流水线生产系统这一新场景的AGV调度问题. 实车实验也证明了该算法在实际生产中的有效性.

关键词: AGV调度算法;AGV规模确定算法;禁忌搜索算法;流水线生产系统

中图分类号: TP 278 文献标志码: A 文章编号: 1005-3026(2024)01-0058-11

Research on the AGV Scheduling Problem in the No-buffer Assembly Line

WANG Xing-kai^{1,2}, WU Wei-min^{1,2}, XING Zi-chao^{1,2}, NIU Hao-yi^{1,2}

(1. State Key Laboratory of Industrial Control Technology, Zhejiang University, Hangzhou 310058, China;

2. College of Control Science and Engineering, Zhejiang University, Hangzhou 310058, China. Corresponding author: WU Wei-min, E-mail: wmwu@iipc.zju.edu.cn)

Abstract: To solve the AGV (automated guided vehicle) scheduling problem in the no-buffer assembly line, a heuristic scheduling algorithm base on composite score is proposed, reducing the waiting time of AGVs when execute tasks in the system. Firstly, the classical AGV fleet size determination algorithm based on the mathematical model is improved. Then, an improved tabu search-based AGV look-ahead scheduling algorithm (LSA) is proposed, which has a composite score strategy. Different from the most reported algorithms, LSA takes parts delay time, AGV empty travel distance and other objects into account. Finally, the proposed algorithms are compared with other reported algorithms. The simulation experiments prove that LSA has better performance than other algorithms. The robotic experiments prove that the proposed algorithm works well in actual production scenarios.

Key words: AGV scheduling algorithm; AGV fleet sizing algorithm; tabu search algorithm; assembly line

在自动化程度较高的流水线生产系统中,自动导引车(automated guided vehicle, AGV)常被用于生产系统的物料转运以增加系统的柔性. 因流水线生产系统中运输任务的高频性和高精度性的特征,AGV的任务调度问题一直是国内外学者想要解决的难题^[1-3]. 目前,随着机械臂等工业

机器人的大规模运用,流水线生产系统中每道工序的耗费时间也越来越可控,为节省场地资源,在空调、摩托等大型产品制造流水线上,已有部分工厂减少甚至取消了用于平衡工人加工时间不确定而设置的缓存区^[4-5]. 这类新型的无缓存流水线生产系统的加工节奏更加紧凑,对AGV调

度算法的要求也更高,目前国内外鲜有相关的研究.

流水线生产系统中的 AGV 调度问题由于运行场地狭小、运输频率高、任务效率要求高等特点,一直是国内外学者关注的热点.目前,国内外 AGV 调度问题研究主要包含三个方面,分别为 AGV 规模确定问题、AGV 路径规划问题、AGV 任务分配问题.其中,AGV 规模确定问题和 AGV 任务分配问题常被一起研究^[6-7].本文将针对无缓存流水线生产系统的 AGV 调度问题,从 AGV 规模确定和 AGV 任务分配两个方面进行研究.

在 AGV 规模确定问题方面,国内外主流的方式是采用任务需求时间与可提供服务时间的比值进行估算. Rajotia 等^[8]在经典的 AGV 规模估测算法基础上,提出了降低空驶距离的整数规划模型,并据此提出了新的 AGV 规模确定算法. Liu 等^[9]针对单设备的最大作业效率问题采用了 Petri 网进行建模,并据此提出了 AGV 规模确定算法及对应的调度算法. Punceth 等^[10]采用了仿真的方式,对流水线生产系统中的 AGV 规模进行估测. Vivaldini 等^[11]结合 AGV 任务分配和路径规划对系统所需 AGV 数量进行了预测.在新出现的无缓存流水线生产系统这一特殊场景中,系统所需的 AGV 规模还会受到运输任务并发需求的影响,目前鲜有学者在此场景下对 AGV 规模估算问题进行研究.

在 AGV 任务分配问题方面,国内外大多数学者选择结合系统的整体效率来设计 AGV 调度算法. Rahman 等^[12]以装配系统总延误时间最短为目标,设计了元启发式算法,同时将 AGV 调度问题和装配线平衡问题同时考虑在算法中. Essam 等^[13]基于离散事件模拟构建了流水线生产系统中 AGV 和工序设备同时调度的模型. Lin 等^[14]提出了一种基于仿真的局部搜索算法,并将该算法与遗传算法结合,得到的混合算法被用于解决 AGV 和工序设备同时调度问题.伍乐等^[15]提出了一种混合变邻域搜索的差分进化算法来解决流水线生产系统中的 AGV 与机器同时调度问题. Zhou 等^[16]针对汽车工业中的混流装配线的 AGV 调度问题,提出了基于深度 Q 网络和非支配排序的超启发式算法来解决流水线平衡和 AGV 调度问题.

1 问题描述和建模

1.1 无缓存流水线生产系统中的 AGV 运输问题

本文研究无缓存流水线生产系统中的 AGV

任务调度问题,其典型场景如图 1 所示.该场景可以描述为:在由 n 个工序设备构成的流水线生产系统中,共有 m 个工件需要进行加工,工件的每个工序仅需要进行一次加工,且每个工序的加工时间固定.当工件的某道工序完成加工后,由 AGV 搬运至后续工序的设备上进行下一道工序,直至所有工序加工完成.在该问题中,AGV 运输系统对于产品的生产效率的影响主要是通过产品的搬运时间来间接影响.基于上述背景,本文所研究问题的重点为如何调度 AGV,使得工件在工序设备之间的运输时间最短.

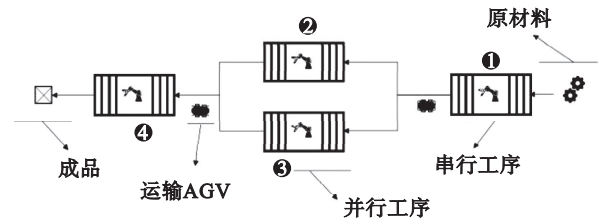


图 1 流水线生产加工示意图

Fig. 1 Processing diagram in assembly line

基于本文的研究重点,为不失一般性,给出下述假设:

假设 1:所研究的流水线生产系统所有产品加工工序固定且在生产开始前已知.

假设 2:为使得生产效率最大化,工厂一般会设置不同数量的设备来完成各道工序,从而平衡各道工序所花费的时间.因此,假设所研究的生产系统中,所有工序的平均耗时都相同,以 T_M 表示.

假设 3:不同工艺产品的混合式生产调度是另一个复杂的问题,并非本文的研究重点.因此,本文假设所研究的流水线生产系统中,同一时刻仅有一种工艺的产品正在加工.

假设 4:无缓存流水线生产系统主要运用于受空间限制的空调生产线、摩托生产线等大型产品生产系统.基于现实生产对于安全运输的要求,本文假设所研究的流水线生产系统中,AGV 必须要等工件的后序设备空闲时才可开始取货.

1.2 数学模型

本文所研究的流水线生产系统的 AGV 调度问题优化目标为调度 AGV 使得产品的运输时间最短.

本文所使用的数学符号及其含义如表 1 所示.

针对所研究的问题,本文给出以下数学模型:

表 1 数学符号
Table 1 Mathematical symbols

数学符号	说明
T	运输任务集合, 大小为 $t, T = \{T_i i = 1, 2, \dots, t\}$
R	AGV 集合, 大小为 $r, R = \{R_i i = 1, 2, \dots, r\}$
M	设备集合, 大小为 $m, M = \{M_i i = 1, 2, \dots, m\}$
F_i	任务 T_i 运输总花费时间
X_{ij}	任务分配决策变量
t_{ij}^0	AGV 执行任务的空驶时间
t_{ij}^1	AGV 执行任务的载货时间
t^L	AGV 载货动作时间
t^U	AGV 卸货动作时间
t_i^W	任务等候后序设备空闲的时间
t_i^S	任务取货完成时间
t_i^F	任务所运送货物的运抵当前设备时间
$M_i \rightarrow M_j$	加工先后顺序, 先由 M_i 加工后由 M_j 加工
$d(M_i, M_j)$	设备 M_i 到设备 M_j 的距离
$d(R_i, M_j)$	AGV R_i 到设备 M_j 的距离

$$\min \sum_{i=1}^t F_i, \quad (1)$$

$$\text{s.t. } F_i = t^L + t^U + t_i^W + t_i^1 + \sum_{j=1}^r X_{ij} t_{ij}^0, \quad (2)$$

$$\sum_{j=1}^r X_{ij} = 1, \quad (3)$$

$$t_{ij}^0 = d(R_j, M_i) / v^0, \quad (4)$$

$$t_i^1 = d(M_i, M_k) / v^1, M_i \rightarrow M_k, \quad (5)$$

$$t_i^W = t_i^S - (t_k^S + t^L), \quad (6)$$

$$t_i^S \geq t_i^F + t^M, \quad (7)$$

$$t_i^S \geq t_k^S + t^L, \quad (8)$$

$$X_{ij} \in \{0, 1\}. \quad (9)$$

式(2)计算了AGV执行一个运输任务所花费的时间;式(3)保证了一个任务只可以被分配给一个AGV;式(4)和式(5)计算了AGV执行任务花费的空载时间和重载时间;式(6)计算了设备先于后序设备加工完成时所要等候的时间;式(7)表示货物在运达工序设备后,必须要等候到加工完成才可开始运输;式(8)保证了工件运输任务开始时,后序设备的工件一定已被取走;式(9)规定了决策变量的取值范围.

2 改进的无缓存流水线AGV规模确定算法

目前主流的AGV规模估算算法是采用任务

总需耗时与AGV总体可用时间比值的估算方式. 根据Rajotia等的研究^[10],对于一般的AGV运输场景,所需的AGV数量应满足AGV在单位时间内完成单位时间内产生的任务,即应当满足式(10).

$$r_1 \geq \left\lceil \frac{T_c + T_{wb} + T_e}{T_a} \right\rceil. \quad (10)$$

其中: r_1 为通过式(10)估算的AGV数量; T_a 为该时间段内AGV可提供服务的时间,即考虑的单位时间段; T_c 为所有任务的载货运行时间; T_{wb} 为AGV等候时间; T_e 为所有任务的空驶时间.

此外,在无缓存流水线生产系统中,由于各工序的加工节奏比较统一,还需要考虑设备同时加工完成时所产生的工件运输任务并发需求. 在本文所研究的无缓存流水线生产系统中,影响产品生产效率的关键因素是工件的搬运耗时. 这包括工件等候AGV取货的时间和AGV搬运工件的时间,后者由于仅与前后工序设备的距离和AGV性能有关,调度可优化空间较小. 因此,在本文所研究的场景中,当任务并发出现时,AGV的数量还应当满足工件等候取货的时间尽可能短,即需要满足式(11).

$$\min (t_k^F + d(R_j, M_i) / v^0 - t_i^S). \quad (11)$$

其中: t_k^F 表示任务 T_k 的完成时刻; T_k 表示 R_j 执行任务 T_i 前的一个任务, R_j 为执行该任务的AGV.

当工件等候取货时间为最小值0时,AGV数量应满足式(12).

$$t_k^F + d(R_j, M_i) / v^0 \leq t_i^S. \quad (12)$$

记根据式(12)估算得到的AGV数量为 r_2 . 根据式(10),式(12),本文将对无缓存流水线生产系统所需的AGV规模进行估计,即实际AGV所需数量应当满足式(13).

$$r = \max \{r_1, r_2\}. \quad (13)$$

2.1 基于传统方法预估AGV数量

根据式(2),当AGV能够在后序设备空闲时即开始运输,则无需等候时间. 此时,式(2)可以转换为式(14).

$$F_i = t_i^1 + t^U + \sum_{R_j \in R} X_{ij} t_{ij}^0. \quad (14)$$

以一个工件在流水线生产系统中花费的全部时间为单位时间,即共计 m 个加工工序的加工时间及其运输时间. 考虑到实际作业时,所有工序设备应当都有工件在加工. 因此,整条产线中,这段时间中一共有 $m \times m$ 个运输任务. 则式(10)中的各时间可转换为式(15)至式(17).

$$T_c + T_{wb} = m \sum_{i=1}^m (t_i^1 + t^L + t^U), \quad (15)$$

$$T_c = m \sum_{i=1}^m \sum_{R_j} X_{ij} t_{ij}^0, \quad (16)$$

$$T_a = \sum_{i=1}^m (F_i + T^M). \quad (17)$$

式中,仅有 t_{ij}^0 和 t_i^1 为未知量,需要估算.

式(15)包括了 AGV 运行时间和 AGV 因拥堵等原因的等候时间,因此需要采用实际运行数据进行计算. 本文采取了仿真数据采集的方式进行估算 t_i^1 的数值.

对于式(16)中的 AGV 空驶时间 t_{ij}^0 , 本文研究的无缓存流水线生产系统中,前后工序设备的运输要求严格满足式(8). 因此,执行 M_i 处设备的运输任务的 AGV 一定是其后续 r_1 个 AGV 其中之一,当 $i > m - r_1$ 时,则从流水生产线头部调度 AGV,如图 2 所示. 因此, t_{ij}^0 可根据式(18)计算.

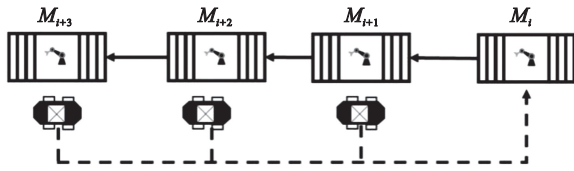


图 2 AGV 作业调度

Fig. 2 AGV scheduling in assembly line

$$\left. \begin{aligned} t_{ij}^0 &= \frac{\sum_{k=1}^r \eta(M_{i+k}, M_i) \times d(M_{i+k}, M_i)}{v^0}, i+k \leq m; \\ t_{ij}^0 &= \frac{\sum_{k=1}^r \eta(M_{i+k-m}, M_i) \times d(M_{i+k-m}, M_i)}{v^0}, i+k > m. \end{aligned} \right\} \quad (18)$$

其中, $\eta(M_{i+k}, M_i)$ 表示由执行 M_{i+k} 任务的 AGV 执行 M_i 处任务的概率,该数据可由仿真采集.

2.2 考虑任务并发的 AGV 数量估算方法

上文提到,执行 M_i 设备处任务的 AGV,其上一个任务一定是其后续 r_2 个设备处的任务之一. 因此,式(12)可以转换为下述不等式组.

$$\left. \begin{aligned} t_{i+k}^F + d(M_{i+k}, M_i)/v^0 &\leq t_i^S, i+k \leq m; \\ t_{i+k-m}^F + d(M_{i+k-m}, M_i)/v^0 &\leq t_i^S, i+k > m. \end{aligned} \right\} \quad (19)$$

其中: k 取值为 $1 \leq k \leq r_2$; t_i^S 表示 M_i 设备的第二个工件的运输任务开始时间.

根据 t_i^F 的定义,可以得到 t_i^F 的计算方式:

$$t_i^F = t_i^S + F_i. \quad (20)$$

因此,不等式组(19)可转换为不等式组:

$$\left. \begin{aligned} F_{i+k} + \frac{d(M_{i+k}, M_i)}{v^0} &\leq t_i^S - t_{i+k}^S, i+k \leq m; \\ F_{i+k-m} + \frac{d(M_{i+k-m}, M_i)}{v^0} &\leq t_i^S - t_{i+k-m}^S, i+k > m. \end{aligned} \right\} \quad (21)$$

首先,需要推导 t_i^S 和 t_{i+k}^S 之间的关系,而根据式(8),可以将工件的开始运输时间转换为不等式(22):

$$t_i^S - t_{i+k}^S \geq k \times t^L. \quad (22)$$

当 AGV 数量足够,且 AGV 在工件需要运输时即刻开始运输,即不存在工件等候搬运时间,不等式(22)应当取等号,变为等式(23):

$$t_i^S - t_{i+k}^S = k \times t^L. \quad (23)$$

接下来推导 t_i^S 和 t_{i+k-m}^S 之间的关系. 设备 M_i 第二个任务开始运输的时间应当为设备 M_{i-1} 第一个运输完成时间加上工件在设备 M_i 上的加工时间. 即 M_i 的第二个工件运达时间与工件加工时间之和,如式(24)所示.

$$t_i^S = t_{i-1}^F + T^M = t_{i-1}^S + F_{i-1} + T^M. \quad (24)$$

由于 $k \leq r$, 则 $i > i+k-m$. 因此 t_{i-1}^S 与 t_{i+k-m}^S 的关系可按式(23)计算. 因此,式(24)可转换为式(25):

$$t_i^S = t_{i+k-m}^S - (m-k-1) \times t^L + F_i + T^M. \quad (25)$$

根据上述推导的式(23)和式(25),可将不等式组(21)转换为不等式(26)和式(27):

$$F_{i+k} + d(M_{i+k}, M_i)/v^0 \leq k \times t^L, i+k \leq m, \quad (26)$$

$$\left. \begin{aligned} F_{i+k-m} + \frac{d(M_{i+k-m}, M_i)}{v^0} &\leq F_i + T^M - \\ &(m-k-1) \times t^L, i+k > m. \end{aligned} \right\} \quad (27)$$

式中, F_i 可以通过式(14)和仿真采集的数据求得. 除 k 外其他变量均为已知量,而 k 需要满足 $1 \leq k \leq r_2$ 的取值范围. 因此,可以设计算法 1(见表 2)来搜寻满足上述不等式(26)和式(27)的取值范围.

3 基于禁忌搜索的前瞻调度算法

记 Δ_m 为有 m 道工序的生产线的工件加工耗时,则可以通过式(28)计算该值.

$$\Delta_m = \sum_{i=1}^m (t^M + F_i). \quad (28)$$

根据式(28),在设备加工时间恒定时,完成一件工件加工的耗时主要取决于工件的运输时间. 然而,如果等设备加工完成之后才开始分配运输任务,此时工件等候 AGV 的时间 t_{ij}^0 则一定无法避免. 针对这个问题,本文提出了基于禁忌搜索的 AGV 前瞻调度算法(look-ahead scheduling

表2 考虑工件等候时间的AGV规模确定算法
Table 2 AGV fleet size determination algorithm

```

算法1 AGV规模确定算法
Step1:输入仿真采集的AGV运行数据
Step2:初始化  $r_2 = 1$ .
Step3:开始遍历搜索,具体流程如下:
for:  $r_2 = 1$  to  $m$  do
    初始化满足标志为 flag = true
    for:  $k = 1$  to  $r_2$ 
        if:  $i + k < m$ 
            计算不等式(26), 并赋值给 flag
        else
            计算不等式(27), 并赋值给 flag
        end if
        if: flag = false
             $r_2 = r_2 + 1$ 
            break for
        end if
    end for
    if: flag = true
        break for
    end if
end for
    
```

algorithm, LSA). 本算法主要通过在工序设备尚未加工完成时,提前将AGV调度至相应位置等候运输来降低甚至消除运输任务的空驶时间,运输任务的两个阶段见图3.

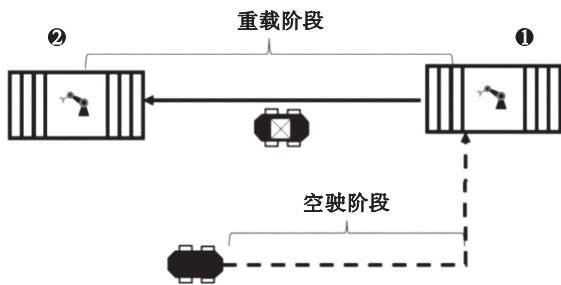


图3 运输任务的两个阶段
Fig. 3 Two steps of a transportation task

定义1 载货任务,包含重载阶段和空驶阶段或仅包含重载阶段的任务(如AGV已执行过前瞻任务),以 T^1 表示,当工件加工完成时才生成.

定义2 前瞻任务,仅包含空驶阶段的任务,以 T^0 表示,一旦货物运送到工序设备处即生成,相对应的载货任务生成之后则取消.

本文研究的AGV调度属于典型的NP-hard问题,每次参与计算的任务共有 m 个,参与计算的AGV有 r 个. 因此可行的解共有 r^m 种排列. 考

虑到AGV调度算法对实时性的要求,精确求解算法难以在较短的时间内获取可行解. 本文采用禁忌搜索算法对问题进行求解. 在传统禁忌搜索算法的基础上,本文结合前瞻调度模型,提出了多类型任务的复合评分搜索以加速算法的搜索速度. 该算法的主要设计内容如下:

解构成:禁忌搜索算法的解构成设计如图4所示. 解由两部分构成,即表示运输任务执行顺序的第一行和表示所分配AGV的第二行. 解以 S 表示, $S = \{(T_i, R_j) | T_i \in T^0 \cup T^1, R_j \in R\}$.

图4所示的解就表示共有三台AGV被分配了运输任务,1号AGV执行的运输任务为3号、4号设备处的运输任务,2号AGV执行的运输任务为1号设备处的运输任务,3号AGV执行的运输任务为2号、5号、6号设备处的任务.

设备编号	3	2	1	5	4	6
AGV编号	1	3	2	3	1	3

图4 禁忌搜索算法解构建
Fig. 4 Solution structure of tabu search algorithm

本文设计的改进禁忌搜索算法采用解评分与重复度评分的复合评分方式. 解评分反映调度方案的优良程度,重复度评分则为该调度方案在搜索中的重复次数. 二者构成的复合评分作为禁忌搜索算法搜索方向的评分依据.

解评分:算法的解评分按照式(29)计算,评分越低的解表示越优秀.

$$\rho(S) = \sum_{i=1}^m (Z_i \times \rho_i^0 + (1 - Z_i) \times \rho_i^1), \quad (29)$$

$$\rho_i^0(S) = \alpha(T^M + t_i^A - t_{cur}) + \sum_{R_j \in R} X_{ij} d(M^S(T_i^0), M^E(T_{R_j}^1)), \quad (30)$$

$$\rho_i^1(S) = \sum_{R_j \in R} (X_{ij} d(M^S(T_i^1), M^E(T_{R_j}^1)) + |X_{ij} - Y_{ij}| \Delta) - 2\Delta. \quad (31)$$

其中: $\rho(S)$ 表示解的总评分; $\rho_i^0(S)$ 表示解中第 i 个分配如果是前瞻任务的评分; $\rho_i^1(S)$ 表示解中第 i 个分配如果是载货任务的评分; Y_{ij} 表示 R_j 是否执行过载货任务 T_i^1 对应的前瞻任务; Z_i 表示工件是否已经加工完成; Δ 表示极大值; $M^S(T_i)$, $M^E(T_i)$ 分别表示任务的取货设备编号和目的的设备编号; $T_{R_j}^1$ 表示 R_j 的上一个任务; α 表示时间权重; α 表示前序任务执行时间权重; t_i^A 表示设备开始加工货物的时间; t_{cur} 表示计算的时刻.

对于前瞻任务,评分由两部分构成,即AGV

上一个任务完成位置与当前任务取货位置的距离和当前设备剩余加工时间延误数值. 对于载货任务,评分由三部分构成,即 AGV 距离取货位置的距离、是否已有 AGV 执行了前瞻任务来此等候、极大值.

总的来说,通过解评分的设计,载货任务优先级高于前瞻任务(由评分中最后一项极大值保证). 对于载货任务而言,已被分配了对应的前瞻任务的 AGV 优先分配该载货任务. 对于前瞻任务而言,剩余加工时间更短的优先级更高.

复合评分:为了避免搜索大量重复解,LSA 选择解时,并不直接采用解评分,而是采用包含已考察解惩罚的复合评分的方式,用式(32)计算.

$$\omega(S) = \rho(S) + \delta \times \sum_{T_i \in T^0} X_{ij} C(T_i, R_j). \quad (32)$$

其中: ω 为复合评分的最终分值; δ 表示已考虑调派方式的惩罚值权重; $C(T_i, R_j)$ 为禁忌表,表示“任务-AGV”出现的次数.

邻域生成:本文设计的改进禁忌搜索算法有两种生成邻域新解的方式.

第一种通过依次将任务分配给 AGV 变为任意其他 AGV 来产生新解,如图 5 所示,每次邻域生成会生成 $|T| \times (r-1)$ 个解.

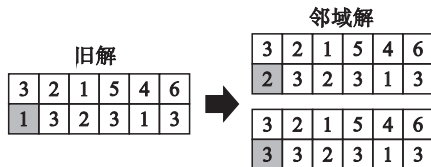


图 5 改派 AGV 生成邻域

Fig. 5 Neighbor solution generation by reallocating AGV

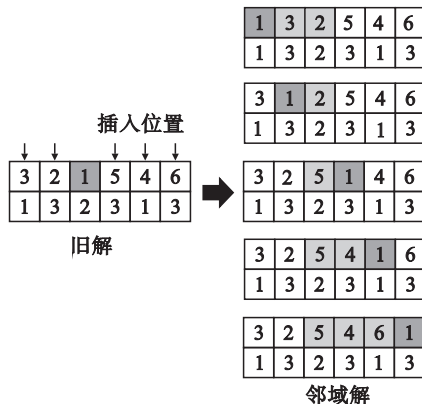


图 6 插入执行任务生成邻域

Fig. 6 Neighbor solution generation by inserting tasks

第二种方式是通过将运输任务的执行顺序插入至任意一其他位置来产生,如图 6 所示,每次生成 $|T| \times (|T|-1)$ 个邻域. 因此,每次迭代生成的邻域解共有 $|T| \times (|T|-1) + |T| \times (r-1)$ 个.

LSA 的整体设计如图 7 所示.

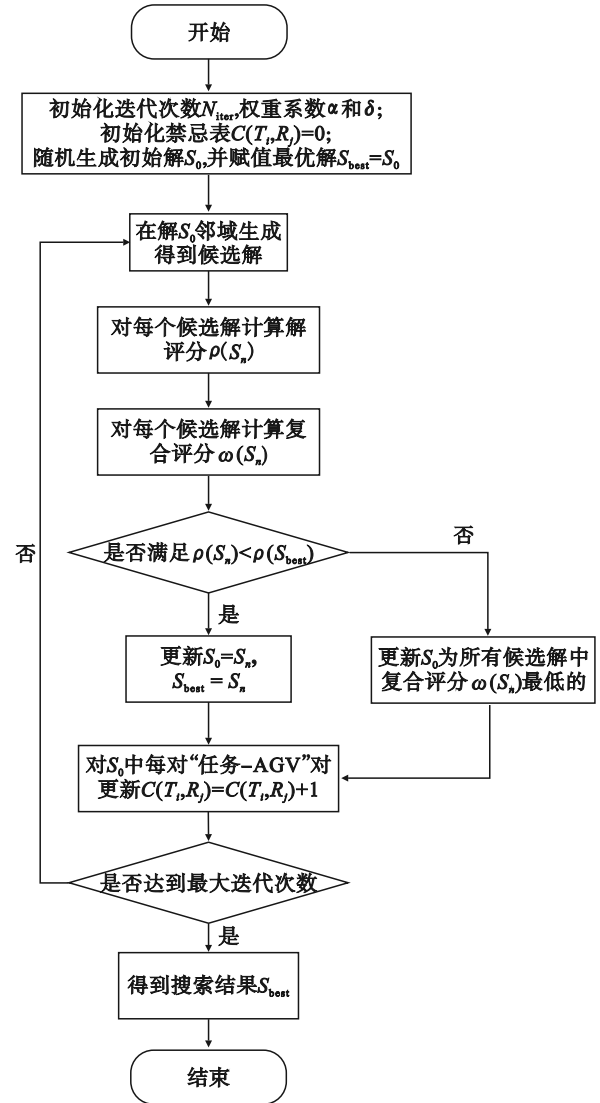


图 7 基于禁忌搜索的前瞻调度算法流程图

Fig. 7 Flow chart of look-ahead scheduling algorithm based on tabu search

4 实 验

4.1 工件加工总耗时下限估算

根据式(28),产品的加工时长主要取决于工序设备的加工时间以及工件的运输时间. 本文所提出的基于禁忌搜索的前瞻调度算法对工件的等候搬运时间进行了优化. 据此可以估算本文所研究系统的工件加工耗时下限.

式(2)中, t^L 与 t^U 为固定数值. $t_i^W \geq 0, \sum_{j=1}^r X_{ij} t_{ij}^0 \geq 0$.

而 t_i^l 为重载搬运时间, 当 AGV 无避让时, 即是最小值. 因此, F_i 的最小值为

$$F_i = t^L + t^U + t_i^l. \quad (33)$$

本文所研究的场景中, 产品的加工总耗时下限可以由式(34)估算.

$$\Delta_m \geq m \times (T^M + t^L + t^U) + \sum_{i=1}^m t_i^l. \quad (34)$$

在式(34)中, t_i^l 由 AGV 实际无避让运行测试得到. 计算的工件加工耗时下限在本文的实验部分与本文所提出的算法得到的调度结果进行对比.

4.2 数值实验: 搜索性能对比

本文研究的问题是无缓存流水线生产系统中的 AGV 动态调度问题. 搜索算法的效率对于 AGV 调度效果有着非常关键的影响. 为证明本文所采用的改进禁忌搜索算法能够很好地适应本文所研究的场景, 该算法在 AGV 调度问题的标准数据集上与目前表现最好的算法进行了对比. 对比实验从算法的搜索结果和搜索时间两个角度进行了比较, 结果如表 3 所示.

每组实验共进行 10 次, 结果取平均值. 从实验结果来看, 本文所提出的算法在搜索结果上与目前最优的搜索算法还存在微弱的差距. 在共计 40 组标准案例中, 本文提出的算法仅在 11 组实验中表现优于 Yan 等^[9]提出的算法. 然而从搜索时间的角度来看, 本文所提出的搜索算法占有绝对优势.

结合两个指标来看, 本文所提出的改进的禁忌搜索算法能够以更短的时间搜索到近似最优的结果. 因此, 在 AGV 动态调度问题的背景下, 该算法更适合本文提出的前瞻调度模型.

4.3 仿真实验 1: 规模确定算法对比

本研究基于目前主流的 AGV 规模确定算法设计了适用于无缓存流水线生产系统的 AGV 规模确定算法 (fleet size determination algorithm, FSDA). 在仿真实验 1 中, 该算法与其他学者提出的规模确定算法进行了比较. 在本文所研究的流水线生产系统中, 影响 AGV 作业效率最大的两个因素为设备的加工时间和工序设备的数量规模. 因此本文的仿真实验从设备加工时间和数量规模设计了 6 组不同规模的实验. 每组实验按照“ $T^M - m$ ”表示, 即“12-18”表示设备加工时间为 12 min, 产线共有 18 台工序设备.

加工时间共分成 4, 8, 12min 三种加工时长, 产线 AGV 数量分成 9 台和 18 台两种. 6 组仿真实验结果如表 4 所示.

表 3 标准数据集实验结果
Table 3 Experiment results on benchmark

案例	搜索时间/s		搜索结果		已发表最优值
	文献[8]	LSA	文献[8]	LSA	
EX11	18.6	2.5	96	98.8	96
EX21	28.7	4.2	103.2	105.9	100
EX31	33.9	5.9	103.2	104.8	99
EX41	47.6	4.2	115.4	116.8	112
EX51	19.4	3.7	87.7	94.6	87
EX61	45.5	8.1	120.7	123.4	118
EX71	61.4	11.9	118.4	118.2	111
EX81	57.5	5.2	161	161.1	161
EX91	34.9	2.9	117.1	122.7	116
EX101	65.4	7.9	150.8	160.9	146
EX12	18.0	0.4	82	81.9	82
EX22	30.1	4.1	77.8	80.6	76
EX32	34.2	4.6	85	86.3	85
EX42	47.2	5.0	90.9	92.5	87
EX52	18.3	3.8	69	72.8	69
EX62	44.9	4.3	99	101.9	98
EX72	62.6	7.7	83.9	81.1	79
EX82	58.4	6.9	151	151.4	151
EX92	32.3	6.4	102.6	104.7	102
EX102	66.8	8.7	136.9	138.4	135
EX13	18.9	4.0	84	84.7	84
EX23	28.2	4.2	86	86.0	86
EX33	37.3	6.1	86.3	86.6	86
EX43	49.0	5.9	93	92.7	89
EX53	21.1	1.9	74.6	76.6	74
EX63	53.6	6.9	103.9	104.9	103
EX73	72.9	9.1	88.3	90.2	83
EX83	65.2	5.0	153	153.5	153
EX93	43.7	5.3	106.1	105.3	105
EX103	74.5	6.6	141.2	140.0	137
EX14	17.3	0.7	103.2	107.8	103
EX24	29.2	5.8	113.8	116.9	108
EX34	35.9	4.2	114.2	113.2	111
EX44	48.5	7.6	130.1	129.2	121
EX54	17.7	3.4	96.8	98.6	96
EX64	44.0	6.1	125.9	122.9	120
EX74	63.2	7.9	133.6	138.3	126
EX84	54.9	6.4	163	163.4	163
EX94	32.4	3.3	123.3	123.0	120
EX104	65.7	8.8	164.3	161.9	157

Vivaldini 等^[11]提出的 AGV 规模确定算法是基于 AGV 执行任务所需时间和实际所需任务执行时间比值来估算. 从表 4 (LSA+FSDA 表示 AGV 规模确定算法, NN+TS 表示 AGV 调度算

表 4 实验 1 结果
Table 4 Results of experiment 1

算法	案例	AGV 数量	单任务耗时/s	工件耗时/min	与理论差距/%	加工耗时下限/min
LSA+FSDA	4-9	6	96.4	57.9	3.4	56.0
NN+TS		2	127.2	73.2	30.7	
LSA+FSDA	8-9	6	95.8	99.2	3.3	96.0
NN+TS		2	127.0	103.8	8.1	
LSA+FSDA	12-9	6	95.8	139.2	2.4	136.0
NN+TS		2	135.4	144.8	6.5	
LSA+FSDA	4-18	15	92.1	111.3	2.1	109.0
NN+TS		3	141.2	202.6	85.9	
LSA+FSDA	8-18	9	95.9	197.5	6.8	185.0
NN+TS		2	127.7	308.0	66.5	
LSA+FSDA	12-18	6	96.7	267.7	2.6	261.0
NN+TS		2	125.8	315.4	20.8	

法)可以看到, Vivaldini 等^[11]提出的算法并没有考虑到无缓存流水线生产系统的任务并发的需求,从而低估了场景所需的 AGV 数量.而本文提出的 AGV 规模确定算法能够获得更少的整体加工时间,能精确地计算实际所需 AGV 规模.

从仿真实验结果来看,场景所需 AGV 数量越多,则本文提出的算法得到的效果与 Vivaldini 等^[11]提出算法相差越大.如案例“4-9”对比案例“4-18”,工序设备的增多则意味着需要更多的 AGV.案例“4-9”中,LSA+FSDA 和 NN+TS 得到的结果与加工耗时下限的差距分别为 3.4% 和 30.7%.而在案例“4-18”中,二者分别为 2.1% 和 85.9%.随着所需 AGV 数量的增加,两种算法效果的差距变大了.同样地,按照相同的工序设备数量,不同的加工时间,可以得到案例“4-9”,“8-9”,“12-9”和“4-18”,“8-18”,“12-18”两组对比实验.从实验结果可以看到,不论是哪一组实验,当所需 AGV 数量变多时(加工时间变短),本文所提出的算法得到的效果与 Vivaldini 等^[11]提出算法效果差距都在变大.

为进一步验证本文所提出的算法对于 AGV 数量的估计是精准的,以“12-18”案例为例进行了使用不同 AGV 数量观察系统加工情况的实验.图 8 显示了使用不同数量的 AGV 时,工件在生产线上耗费的时间.由于仿真实验为冷启动,即初始状态时,所有设备都为空闲,依次添加工件,因此工件的耗时从开始耗时较短增加到平稳状态,最终效果以平稳状态为准.

根据表 2 的实验结果,对于“12-18”案例,最优 AGV 数量为 6 台.从图 8 中可以看到,AGV 数量少于 6 台时,每增加一台 AGV,工件的耗时变

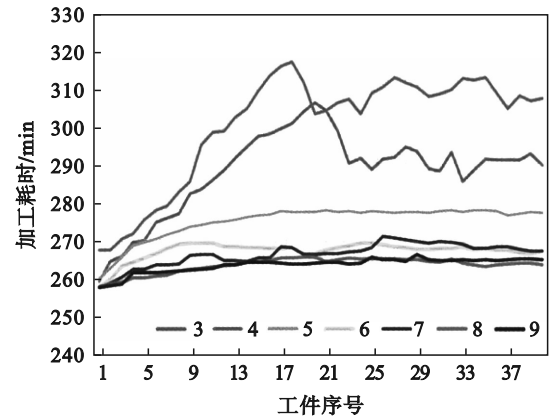


图 8 “12-18”案例不同车数加工时间

Fig. 8 Makespan of case “12-18” with different AGVs

化非常明显,但超过 6 台之后,即使再增加 AGV 数量,工件的耗时变化也不明显.由此可以验证本文提出的 AGV 规模确定算法估测的最优 AGV 数量非常准确.

图 9 和图 10 显示了“12-18”案例中,LSA 算法的调度下,工序设备在车辆不足(4 台 AGV)和 AGV 足够(6 台 AGV)时,共计 18 个设备的作业甘特图.从图 9 和图 10 的对比可以看到,随着冷启动状态所有设备依次开启作业,所产生的 AGV 任务需求也随之增加.而在测试前半部分时,AGV 尚可满足较少的运输需求,工件等候运输时间较短.随着所有设备开始工作,AGV 便无法满足所有工件的运输需求,从而导致很多工件延误时间较长,如图 9 中虚线框内所示.而如果 AGV 数量足够,应当是冷启动完成,所有设备开始作业之后,工件的延误时间仍然维持在较短的水平,如图 10 所示.因此,从设备作业的角度看,本文所提出的 AGV 规模估算方法能够准确估计使得整条生产线稳定高效生产所需的 AGV 数量.

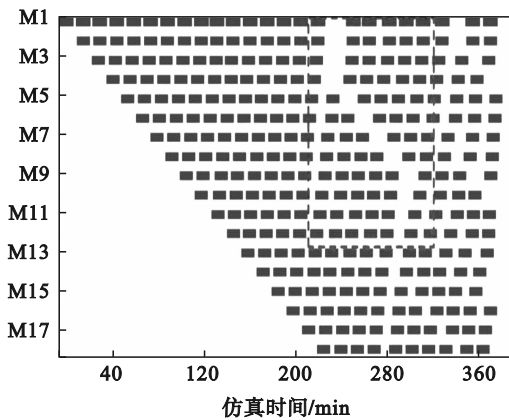


图9 “12-18”案例4车甘特图

Fig. 9 Gantt chart of case “12-18” with 4 AGVs

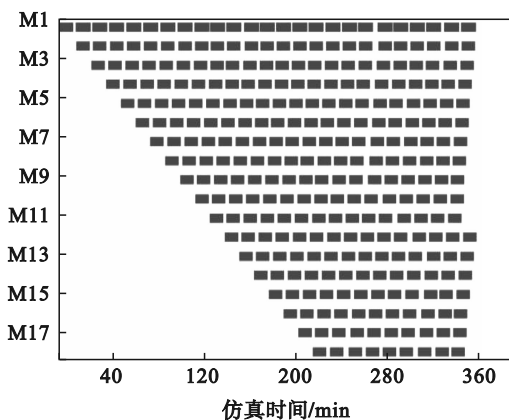


图10 “12-18”案例6车甘特图

Fig. 10 Gantt chart of case “12-18” with 6 AGVs

4.4 仿真实验2:AGV调度算法对比

通过仿真实验1的对比,可以看到本文提出的规模确定算法能够更精准地对无缓存流水线生产系统所需的AGV数量进行估计.在仿真实验2中,本文所提出的前瞻调度算法将在不同规模的场景下与其他学者提出的算法进行对比^[11,19].

通过设置不同规模的AGV数量和不同时长的加工时间,本实验一共进行了10组对比实验. AGV数量分别设置为2~6,加工时长分别设置为8~16min.两种算法在相同的条件下进行实验对比.

实验2验证本文所提出的前瞻调度算法的效率.从多种场景下的实验结果来看,当AGV数量不够时,本文提出的LSA算法并不具有显著优势.而当AGV数量充足时,LSA算法能发挥最大效果,如表5所示.

从工件搬运延误的角度来看,当AGV都足够时,本文提出的LSA算法能够通过避免AGV空驶时间来降低工件延误时间.图11显示了在

表5 实验2结果

Table 5 Results of experiment 2

$r \times T^M$	单任务耗时/s			产品加工耗时/min		
	TS	IG	LSA	TS	IG	LSA
6×8	137.7	236.7	115.1	209.8	221.1	190.3
6×10	113.9	211.9	107.0	253.2	249.1	226.5
6×12	112.0	205.6	96.7	280.7	281.0	267.7
6×14	105.6	212.7	94.5	319.8	320.5	308.9
6×16	92.8	215.1	86.1	356.7	358.5	324.9
2×12	81.6	166.1	91.8	382.7	320.6	387.8
3×12	92.5	189.5	100.8	301.0	320.1	322.7
4×12	104.8	206.5	101.8	290.9	291.3	290.9
5×12	113.5	223.5	103.4	281.4	290.5	270.6

“12-18”案例中,采用足够的AGV数量(6台AGV)时,LSA算法(图11中每个机器编号的第1行)和Vivaldini等^[11]所提算法(图中每个机器编号的第2行)、Zou等^[19]所提算法(图中每个机器编号的第3行)的设备加工甘特图.

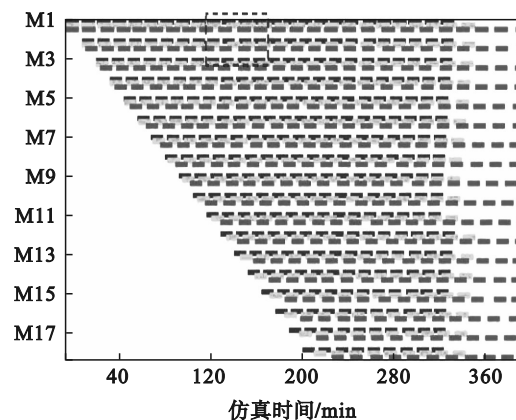


图11 “12-18”案例6车甘特图(对比)

Fig. 11 Gantt chart of case “12-18” with 6 AGVs (comparison)

从图11中可以看到,本文提出的LSA算法的调度结果相较于其他算法能够有更短的设备空闲时间,即设备两次加工之间的空闲时间更短,如图11中框线内所示.通过对比在相同AGV数量情况下,整条生产线中每个工序设备的作业情况,可以明显看到,本文所提出的前瞻调度在提高作业效率、降低工件搬运延误时间上具有一定优势.

4.5 实车实验

为进一步验证本文提出算法的有效性,本文进行了一组实车实验.本次实验的系统共计18个工序设备,即 $m=18$,每个工序固定的加工时间为 $T^M=15$ min,车辆的性能参数如表6所示.

对于本场景,采用FSDA算法得到的最小

表 6 AGV 性能参数
Table 6 AGV performance parameters

参数名称	参数值
满载加速度/($\text{m}\cdot\text{s}^{-2}$)	0.3
空载加速度/($\text{m}\cdot\text{s}^{-2}$)	0.4
满载最大速度/($\text{m}\cdot\text{s}^{-1}$)	0.8
空载最大速度/($\text{m}\cdot\text{s}^{-1}$)	1
装货时间/s	35
卸货时间/s	15

AGV 数量为 4 辆,采用 LSA 算法实车实验得到的结果如表 7 所示.表中呈现的为平均加工时间,即 Δ_m/m .实验共计耗时两天,表 7 所得结果为稳定后的加工耗时.

通过实际数据计算,根据式(34)计算的工件耗时下限为 967 s.可以看到,实验得到的加工耗时与耗时下限的差距在 0.2%~12.9%之间,且平均差距仅为 3.6%.在仿真实验中,本文提出算法得到的所有实验结果与耗时下限差距在 2.1%~6.8%之间,与实车实验得到的结果基本吻合,这也证明本文提出的基于改进禁忌搜索的前瞻调度算法能够有效地在实际生产中调度 AGV,提升系统生产效率.

表 7 实车测试结果
Table 7 Results of AGV experiments

工件序号	实际耗时/s	差距/%
1	1 001	3.5
2	979	1.2
3	988	2.2
4	1 025	6.0
5	969	0.2
6	1 035	7.1
7	980	1.4
8	1 011	4.6
16	978	1.2
17	973	0.6
18	1 029	6.4
19	1 092	12.9
20	989	2.3
21	1 015	5.0
22	977	1.1
23	985	1.8

5 结 语

针对无缓存流水线生产系统中的 AGV 调度问题,本文从 AGV 规模确定和 AGV 任务分配两

个方面进行了研究.首先,为了更精准地估算系统所需 AGV 数量,本文设计了适用于无缓存流水线生产系统的 AGV 规模确定算法.该算法基于目前主流的 AGV 规模确定算法,结合了无缓存流水线生产系统的任务高并发特性,从任务运输时间和工件等候时间两个角度对系统所需的 AGV 数量进行估算.其次,为了更快速、高效地解决 AGV 任务分配问题,本文设计了前瞻调度策略并改进了传统的禁忌搜索算法.该算法由解评分和重复度构成,相较于克服目前大多数动态调度算法,该算法在控制工件等候的同时,满足了动态调度的实时性要求.

通过仿真实验对比,本文提出的考虑任务并发的规模确定算法能够更精准地估算无缓存流水线生产系统所需的 AGV 数量.通过与系统加工耗时下限对比,本文提出的算法在实际所需 AGV 数量越多时效果越好.通过与其他调度算法对比,本文提出的基于改进禁忌搜索的前瞻调度算法能够更有效减少工件的搬运延误,提高系统的加工效率.实车实验与仿真结果的一致性进一步证明了本文所提算法具有实际应用价值.本文所提算法已实际应用于部分无缓存流水线制造系统,给生产系统的效率带来了极大的提升.

本文的研究暂时没有考虑 AGV 的避让问题.在未来的研究中,将在 AGV 规模确定算法中考虑 AGV 的避让问题,以达到更精准预估 AGV 规模的目的.

参考文献:

- [1] Zhou B H, Shen C Y. Multi-objective optimization of material delivery for mixed model assembly lines with energy consideration [J]. *Journal of Cleaner Production*, 2018, 192: 293-305.
- [2] Fazlollahabari H. Scheduling of multiple autonomous guided vehicles for an assembly line using minimum cost network flow [J]. *Journal of Optimization in Industrial Engineering*, 2018, 11(1): 185-193.
- [3] Zhang L X, Hu Y G, Guan Y. Research on hybrid-load AGV dispatching problem for mixed-model automobile assembly line [J]. *Procedia CIRP*, 2019, 81: 1059-1064.
- [4] Wang Y Q. The evaluation of buffer effects on bowl versus inverted bowl assembly line arrangements through a simulation model [J]. *International Journal of Modelling and Simulation*, 1995, 15(2): 60-64.
- [5] Buxey G M, Sadjadi D. Simulation studies of conveyor-paced assembly lines with buffer capacity [J]. *International Journal of Production Research*, 1976, 14(5): 607-624.
- [6] Aziez I, Côté J F, Coelho L C. Fleet sizing and routing of healthcare automated guided vehicles [J]. *Transportation Research Part E: Logistics and Transportation Review*, 2022, 161: 102679.

(下转第 75 页)