

考虑切换时间的机器人装配线平衡优化方法*

周炳海[†], 康雪云

(同济大学机械与能源工程学院, 上海 201804)

摘要:考虑装配过程中的切换时间,以装配线的工位节拍最小为优化目标,建立了机器人装配线平衡问题的数学模型.在此基础上,将自适应策略引入模拟退火遗传算法框架中,构建了自适应模拟退火遗传算法.以轿车制造车身车间左前门内板总成的焊接装配为例,分别对比了标准遗传算法、模拟退火遗传算法及自适应模拟退火遗传算法在解决该问题方面的性能,结果表明自适应模拟退火遗传算法在解的性能和算法收敛性两个方面表现出了较大的优势.

关键词:机器人;装配线平衡;切换时间;模拟退火;自适应

中图分类号:TP29

文献标志码:A

Optimal Method of Robotic Assemble Line Balancing Considering Changeover Time

ZHOU Binghai[†], KANG Xueyun

(School of Mechanical Engineering, Tongji University, Shanghai 201804, China)

Abstract: Considering the changeover time in the process of assembly production, a mathematical model with an objective function to minimize the cycle time of robotic assembly line was developed. On the basis of the model, an adaptive strategy was combined into the frame of simulated annealing genetic algorithm, and an Adaptive Simulated Annealing Genetic Algorithm (ASAGA) was then proposed. Finally, an example of the welding assembly for the inner assembly of the left front door in the Body Shop was provided to illustrate greater advantages in the quality of the solution and the algorithm convergence, compared with the SGA (Standard Genetic Algorithm) and SAGA (Simulated Annealing Genetic Algorithm) in solving this problem.

Key words: robot; assembly line balancing; changeover time; simulated annealing; adapt

随着国家战略性新兴产业的提出,机器人技术和产业得到了迅速发展.作为实现制造业自动化的关键,机器人被广泛应用于自动化装配线,由其产生的机器人装配线平衡问题(Robotic Assembly Line Balancing Problem, RALBP)得到了学术界和工业界的密切关注.

为了有效地解决 RALBP,国内外学者们进行了不少的研究工作.文献[1]和[2]分别采用了带局部优化的遗传算法和粒子群算法来最小化工位节拍;文献[3]构造了一种新的染色体交叉和突变方法,并采用具有局部搜索的遗传算法来求解该类问题;文献[4]在粒子群算法的基础上,融入了布谷鸟

* 收稿日期:2016-07-06

基金项目:国家自然科学基金资助项目(71471135,61273035), National Natural Science Foundation of China(71471135,61273035)

作者简介:周炳海(1965—),男,浙江浦江人,同济大学教授,博士

[†] 通讯联系人, E-mail: bhzhou@tongji.edu.cn

搜索机制,表现出了较好的性能;文献[5]和[6]分别研究了该问题的 U 型装配线和双边装配线的数学模型;文献[7]和[8]分别提出了同时最小化工位节拍和能源消耗的单边、双边 RALBP 的双目标模型;文献[9]提出了同时最小化工位节拍,机器人 Setup 时间和机器人成本的多目标模型,并给出了该问题的混合整数规划数学模型;文献[10]采用多目标规划方法来最小化工位节拍、工位数量以及机器人成本.然而,以上研究均集中于装配任务本身,并未考虑装配过程中任务之间的切换对工位节拍的影响.

在现代随机混流柔性装配模式下,共线装配的同类、甚至不同类产品之间的工具、工装夹具等的切换时间占工位节拍比例较大;而且就同一产品的装配而言,不同装配任务之间也会存在切换时间.因此,在解决 RALBP 时,考虑切换时间显得尤为重要.

1 问题描述

产品的装配过程是在满足给定的任务优先关系的前提下,通过相应的设备,将产品的所有装配任务 $i(i=1,2,\dots,I)$ 在一系列的工位 $s(s=1,2,\dots,S)$ 上装配出产品的过程.

在 RALBP 中,首先要合理地将所有装配任务分配到相应工位;其次,由于不同类型的机器人 $r(r=1,2,\dots,R)$ 完成同一任务的时间不同,并且切换时间也不尽相同,所以需要为每个工位选择最合适的机器人;最后,在确保任务优先关系的前提下,使工位任务序列内的任务之间的切换次数最少.

为了有效地刻画研究的问题域,做如下基本假设:1)各装配任务不可再分,产品的任务优先图和任务之间的切换关系事先已知;2)每项任务的装配时间取决于该工位的机器人,且机器人执行每项任务的时间确定;3)每个工位分配一台机器人,但不同类型的机器人数量没有限制;4)任务和机器人的分配没有限制;5)物料搬运及其他 setup 时间忽略或者包含在任务时间中,但切换的时间不能忽略;6)不同任务之间的切换时间相等,且为一个定值;7)装配线是只装配一种产品的直型装配线.为方便描述,现定义符号如下:

- i —任务序号 $i(i=1,2,\dots,I)$;
- j —任务优先图中紧随任务 i 的任务;

- k —工位任务环中紧随任务 i 的任务;
- s —工位序号 $s(s=1,2,\dots,S)$;
- r —机器人种类序号 $r(r=1,2,\dots,R)$;
- t_{ir} —机器人 r 完成任务 i 的工艺时间;
- c_r —机器人 r 或夹具单次切换时间;
- τ_s^* —工位 s 在任务环内的切换次数;
- τ_s —工位 s 的最小切换次数;
- x_{is} —表示任务 i 分配到工位 s 为 1,反之为 0;
- y_{rs} —表示机器人 r 分配到工位 s 为 1,反之为 0;
- z_{ij} —表示任务 i 和 j 之间有切换为 1,反之为 0;
- w_{ij} —表示任务 i 和 j 有优先关系为 1,反之为 0;
- $\text{pre}(i)$ —任务 i 的直接前序的任务集合;
- $\text{tcs}(i)$ —任务环内任务 i 的直接后续任务集合.

为了进一步分析问题,做以下定义并给出相关定理如下.

定义 1 在保证任务优先关系的基础上,以切换次数最小为原则,将分配到工位 s 的任务进行排序,其最小切换次数记为 τ_s ,再将排序后第一个和最后一个任务相连接,构成工位 s 的任务环,如图 1 所示.则工位 s 在任务环内的切换次数 τ_s^* 为:

$$\tau_s^* = \sum_{i=1}^I \sum_{k \in \text{tcs}(i)} x_{is} x_{ks} z_{ik} \tag{1}$$

特殊地,单、双任务环分别如图 2 和图 3 所示.

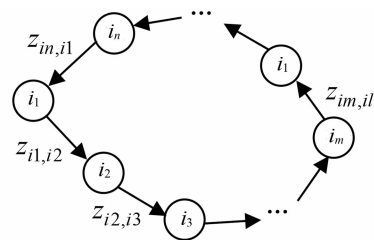


图 1 任务环

Fig.1 The loop of assembly tasks

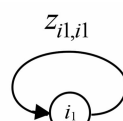


图 2 单任务环

Fig.2 Single task loop

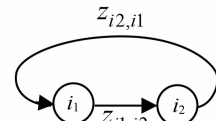


图 3 双任务环

Fig.3 Double tasks loop

定理 1 如果 $z_{i,j}=0, z_{j,k}=0$, 那么 $z_{i,k}=0$. 即任务之间的切换具有传递性.

证 如果任务 i 和 j 之间不存在切换, 并且任务 j 和 k 之间也不存在切换, 那么, 显然任务 i 和任务 k 之间也不存在切换, 即有 $z_{i,k}=0$.

证毕.

定理 2 假设工位 s 的任务环内元素有序集合是 $\Omega = \{i_1, i_2, \dots, i_n\}$, $\Omega_1, \Omega_2, \dots, \Omega_m$ 分别表示集合 Ω 中无切换的作业集合, k_1, k_2, \dots, k_m 分别为其中的任意元素, 它们组成新的集合 $\Omega' = \{k_1, k_2, \dots, k_m\}$, 即有 $\sum_{i=1}^m \text{card}(\Omega_i) = n$, $\text{card}(\Omega_i)$ 为集合 Ω_i 里元素个数, 那么工位 s 的切换次数有:

1) 如果 $\sum_{i=1}^m \omega_{kl,ki} = m - 1 (\forall k_l \in \Omega')$, 那么 $\tau_s = \tau_s^*$;

2) 如果 $\sum_{i=1}^m \omega_{kl,ki} < m - 1 (\forall k_l \in \Omega')$, 那么:

$$\tau_s = \begin{cases} \tau_s^* - 1 & z_{kl,ki} = 1 \quad ki \in \text{tcs}(l) \\ \tau_s^* & z_{kl,ki} \in 0 \quad ki \in \text{tcs}(l) \end{cases} \quad (2)$$

证 1) 如果 $\sum_{i=1}^m \omega_{kl,ki} = m - 1 (\forall k_l \in \Omega')$, 即工位 s 任务环内的任意两相邻任务之间都存在优先关系, 那么显然有 $\tau_s = \tau_s^*$;

2) 由定理 1 知, k_1, k_2, \dots, k_m 可代表集合 $\Omega_1, \Omega_2, \dots, \Omega_m$ 间的切换关系, 如果 $\sum_{i=1}^m \omega_{kl,ki} < m - 1 (\forall k_l \in \Omega')$, 即至少存在一个作业 k_l 与其后续作业 k_i 没有优先关系, 当 $z_{kl,ki} = 0 (\forall k_l \in \Omega')$ 时, 即所有任务之间都没有切换时, 显然有 $\tau_s = \tau_s^*$, 但当 $z_{kl,ki} = 1 (\forall k_l \in \Omega')$ 时, 即 k_l 与其后续作业 k_i 存在切换时, 那么在前一个节拍中机器人先执行任务 k_1, k_2, \dots, k_l , 然后执行任务 k_l, k_{l+1}, \dots, k_m , 而在接下来一个节拍内则可先执行任务 k_l, k_{l+1}, \dots, k_m , 再执行任务 k_1, k_2, \dots, k_l , 从而减少一次切换时间, 即有 $\tau_s = \tau_s^* - 1$, 同理, 如果有多个任务存在切换也有 $\tau_s = \tau_s^* - 1$.

证毕.

定理 3 工位节拍下界可表达为:

$$C_{LB} = \max_{1 \leq r \leq R} \left[\max \left\{ \min_{1 \leq r \leq R} \{t_{ir}\}, \frac{\sum_{i=1}^I \min_{1 \leq r \leq R} \{t_{ir}\}}{S} \right\} + \frac{\tau_0 \min_{1 \leq r \leq R} \{C_r\}}{S} \right] \quad (3)$$

证 工位的节拍等于瓶颈工位的总时间, 即 $C = \max_{1 \leq s \leq S} T(s)$, 其中 $T(s)$ 为工位 s 的总时间; 即有 $T(s) = AT(s) + OT(s)$, 其中, $AT(s)$ 为机器人执行装配任务的时间, $OT(s)$ 为机器人执行切换任务或者夹具切换的时间; 而装配时间必须大于等于执行最大时间装配任务速度最快的机器人的装配时

间和每项任务执行最快的机器人执行时间的平均时间之间的最大值^[11], 即

$$AT(s) \geq \max \left[\min_{1 \leq r \leq R} \{t_{ir}\}, \frac{\sum_{i=1}^I \min_{1 \leq r \leq R} \{t_{ir}\}}{S} \right]$$

假设 $C_{at} = AT(s)/S$, 即 C_{at} 为只考虑机器人执行装配任务时的节拍下界; 在满足优先关系的原则下, 找出最少切换次数 τ_0 的序列, 以 C_{at} 为节拍下界分割序列成 S 段, 每段中的任务分别代表被分配到相应工位的元素, 得到最小切换次数 τ_s , 切换时间应大于等于最快执行切换的机器人执行最小所有切换次数的总时间, 即 $OT(s) \geq \tau_0 \min_{1 \leq r \leq R} \{c_r\}/S$; 所以, 工位节拍下界为 $AT(s)$ 和 $OT(s)$ 两者的下界之和.

证毕.

根据定义和定理, 对所研究的考虑切换时间的 RALBP 建模如下:

$$\min \left\{ \max_{0 \leq s \leq S} \left\{ \sum_{i=1}^I \sum_{r=1}^R (x_{is} t_{ir} + \tau_s c_r) y_{rs} \right\} \right\} \quad (4)$$

约束条件:

$$\sum_{s=1}^S s x_{is} - \sum_{s=1}^S s x_{js} \leq 0 \quad \forall i, j \in \text{pre}(j) \quad (5)$$

$$\sum_{s=1}^S x_{is} = 1 \quad \forall i \quad (6)$$

$$\sum_{r=1}^R y_{rs} = 1 \quad \forall s \quad (7)$$

$$x_{is} \in \{0, 1\}, i = 1, 2, \dots, I; s = 1, 2, \dots, S \quad (8)$$

$$y_{rs} \in \{0, 1\}, r = 1, 2, \dots, R; s = 1, 2, \dots, S \quad (9)$$

$$z_{ik} \in \{0, 1\}, i = 1, 2, \dots, I; k = 1, 2, \dots, K \quad (10)$$

目标函数(4)最小化工位节拍; 约束条件参考文献[3], 不等式(5)保证各项任务之间的优先关系; 等式(6)表示每个工位都必须被分配到工位; 等式(7)表示每个工位只能被分配到一台机器人; 约束(8)~(10)是 0-1 变量.

2 算法构建

ALBP 是一个典型的 NP-hard 组合优化问题, 问题的复杂度随着任务数的增加呈几何级数增长, 不存在精确求得最优解的多项式时间算法^[12]. 而作为 ALBP 的子问题, RALBP 更具复杂性.

为此, 本文提出了一种有竞争性的自适应模拟退火遗传算法 (ASAGA). 该算法以遗传算法为主体, 融入模拟退火机制, 从而极大地克服了遗传算法局部搜索能力较弱的缺点; 为了进一步改善遗传

算法早熟收敛以及收敛性能差的缺点,将自适应策略引入算法框架中,有效缓解遗传算法的退化现象,提高收敛速度,并克服遗传算法的早熟收敛和收敛性能差的缺点,大大地提高了遗传算法的运行效率^[13-14].

算法的总体流程图如图4所示.

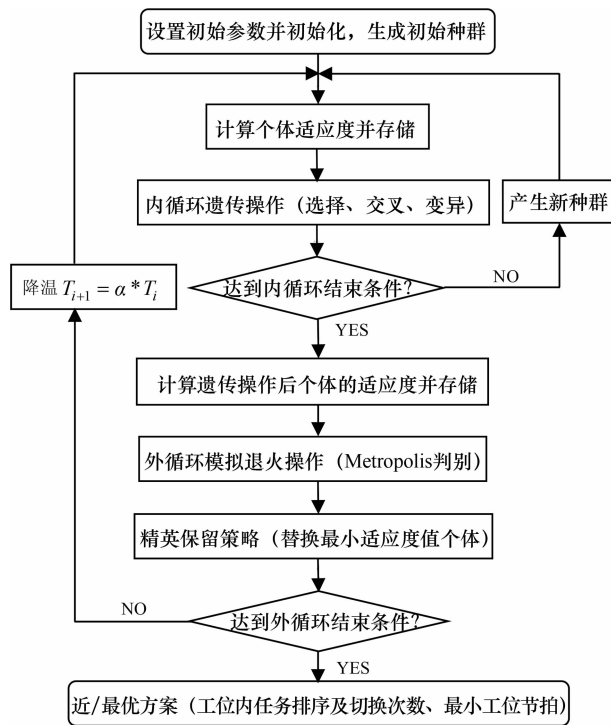


图4 算法总体流程

Fig.4 The flow chart of ASAGA

算法的具体步骤如下:

步骤1 设置初始参数,包括种群大小 Num_Pop,最大遗传代数 Num_Gene,退火初始温度 T_0 ,温度冷却系数 α ,退火迭代次数 MaxIter.

步骤1.1 编码

本文采用基于基因序列的编码方式.首先,按作业元素分派至工位的先后顺序,将作业排成一列,每个作业对应一个基因位;其次,对基因序列进行分割,将每个作业按照加工顺序分配给工位;最后,根据节拍最小原则为工位选择最合适的机器人.

步骤1.2 译码

在实际装配中,线体的节拍增量不是一个连续的值,而是以一个作业元素的时间值作为增量.基于此,参考文献^[15]的步骤进行译码操作.

步骤1.3 初始种群的产生

本文在满足优先关系的前提下,采用如下的步骤产生初始种群:

步骤1.3.1 首先随机选取一个无前序作业的

元素作为染色体的第一个基因.

步骤1.3.2 挑选无前序作业或者前序作业已被分配到的作业元素放入备选集合中.然后在备选集合中随机地选择一个编入染色体基因位.

步骤1.3.3 更新备选集合.如此循环,直到所有作业元素都编入染色体,产生一个染色体.

步骤2 适应度计算

对于RALBP,不仅要求工位节拍最小,而且还应该考虑机器人装配线的负载是否均衡,因此,构造适应度函数如下:

$$F = \omega_1 F_1 + \omega_2 F_2 \quad (11)$$

式中:

$$F_1 = \frac{\sum_{i=1}^I \sum_{r=1}^R (x_{is} t_{ir} + \tau_s c_r) y_{rs}}{m * ST_{\max}} \quad (12)$$

$$F_2 = \frac{1}{\sqrt{\frac{\sum_{s=1}^S (ST_{\max} - ST_s)}{S} + 0.01}} \quad (13)$$

其中 F_1 表示按工位节拍最小的适应度, F_2 表示按工位分配相对均匀的适应度, ST_{\max} 为工位最大时间, ST_s 为第 s 工位的时间.

步骤3 遗传操作

步骤3.1 选择

采用轮盘选择策略进行个体选择,同时采用最优保留策略.每个染色体的选择概率正比于适应度,对于适应度为 F_s 的染色体,其选择概率为:

$$P_s = F_s / \sum_{s=1}^S F_s \quad (14)$$

步骤3.2 交叉

在遗传操作中,交叉概率 P_c 的选取是影响算法行为和性能的关键因素. P_c 过大或过小都会影响收敛速度和过程.因此,对交叉概率 P_c 做如下设置:

$$P_c = \begin{cases} P_{c1} - \frac{(P_{c1} - P_{c2})(f_c - f_{\text{avg}})}{f_{\max} - f_{\text{avg}}} \\ P_{c1} \end{cases} \quad (15)$$

式中: f_{\max} 和 f_{avg} 分别为群体中的最大和平均适应度值, f_c 为要交叉的两个个体中较大的适应度值, P_{c1} 和 P_{c2} 为交叉相关系数,分别取值为 $P_{c1} = 0.90$, $P_{c2} = 0.20$.从选择操作产生的交配库中随机选取两条染色体,依据自适应概率 P_c 采用两点交叉进行交叉操作(见图5),相同地可得到第二条后代染色体.

步骤3.3 变异

同样地,采用该策略对 P_m 进行自适应调整:

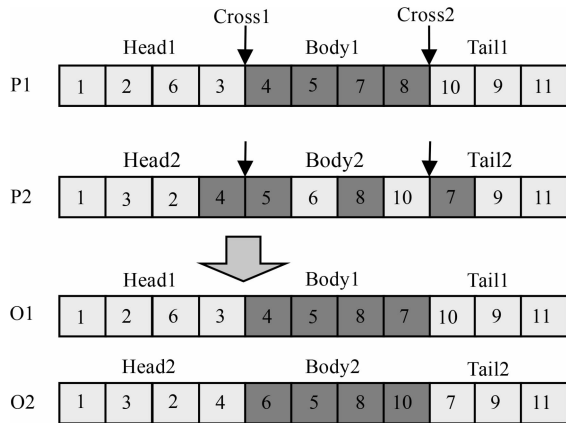


图 5 染色体交叉示意

Fig.5 Illustration of crossover

$$P_m = \begin{cases} P_{m1} - \frac{(P_{m1} - P_{m2})(f_m - f_{avg})}{f_{max} - f_{avg}} \\ P_{m1} \end{cases} \quad (16)$$

式中: P_{m1} 和 P_{m2} 为变异相关系数, 分别取值为 $P_{m1} = 0.15$, $P_{m2} = 0.05$. 任意选择一个个体, 依据概率 P_m 进行突变, 得到突变后个体.

步骤 4 模拟退火操作

步骤 4.1 令内循环次数 $gene = 0$, 计算遗传操作后种群的适应度值, 并分别采用 Metropolis 判别准则进行模拟退火操作. Metropolis 判别准则: 状态 F 变为状态 F' 的接受概率:

$$p = \begin{cases} 1 & f(F') \geq f(F) \\ e^{-\frac{(f(F) - f(F'))}{T_i}} & f(F') < f(F) \end{cases} \quad (17)$$

式中: T_i 为当前控制温度; F 和 F' 分别为遗传操作前、后的适应度值.

步骤 4.2 计算退火后种群的适应度, 并将群体中适应度最小的个体替换为遗传操作前适应度最大的个体, 以保证算法的收敛性.

步骤 4.3 如果当前退火代数 $Iter < MaxIter$, 则按式(18)的策略对温度进行降温, 并返回步骤 3, 否则结束整个优化过程.

$$T_{i+1} = \alpha * T_i \quad (18)$$

式中: T_{i+1} 为当前控制温度; T_i 为上一代温度, 取 $T_1 = 100$; α 为冷却系数, 取 $\alpha = 0.90$.

3 实例分析

某轿车车身车间中左前门内板任务优先关系如图 6 所示, 焊接工艺流程如图 7 所示, 机器人对各作业的装配时间见表 1, 各任务间的切换关系及机器人的切换时间见表 2 和表 3. 机器人焊接装配

线的工位数 $S = 6$, 共有 $R = 4$ 种类型的机器人可供选择.

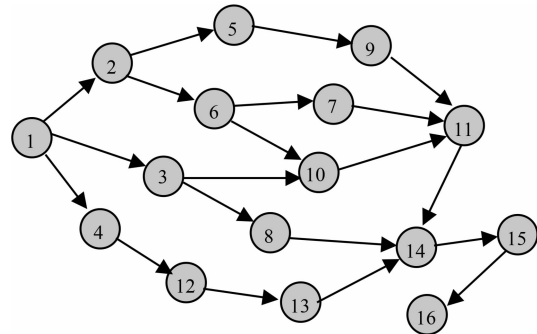


图 6 任务优先图

Fig.6 Precedence diagram of the tasks

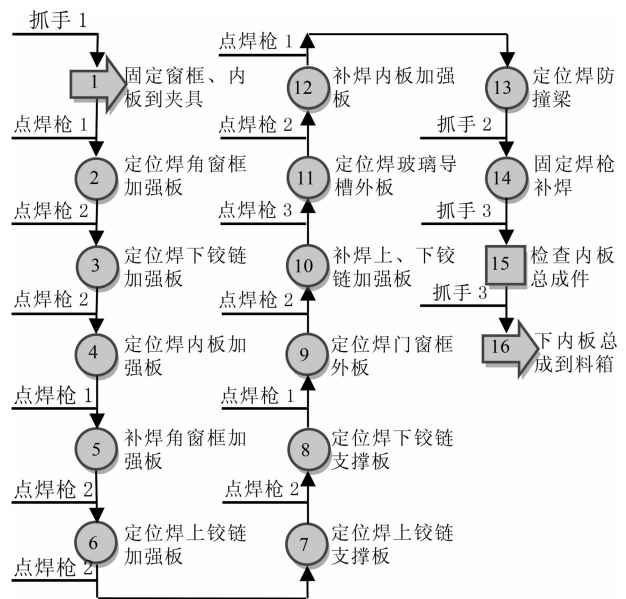


图 7 左前门内板总成工艺流程图

Fig.7 Process flow chart of the inner assembly of left front door

本文采用 Matlab R2014a 编程, 运用 ASAGA 算法来平衡车身车间中左前门内板总成焊接装配的 RALBP, 在主频为 2.13 GHz, 内存为 2 GB 的 PC 机上运行, 并做了如下几种分析.

3.1 评价指标

为有效评价算法的性能, 提出以解的相对偏差 Q 作为算法的评价指标:

$$Q = \frac{V(H, T, Y) - C_{LB}}{C_{LB}} \quad (19)$$

其中 $V(H, T, Y)$ 表示用算法 H 在迭代次数为 T 时对实例 Y 的求解结果, C_{LB} 表示定理 3 提到的工位节拍下界. Q 能够真实地反映算法的收敛性能, 其值越小, 表明算法的收敛性能越好.

表 1 机器人装配时间表

Tab.1 Table of assembly time by robots

工序	Type A	Type B	Type C	Type D	AVG.
1	20	14	16	18	17.00
2	30	24	26	25	26.25
3	21	15	17	18	17.75
4	25	23	19	23	22.50
5	21	24	28	21	23.50
6	30	24	26	25	26.25
7	30	27	26	34	29.25
8	30	27	26	34	29.25
9	65	74	60	74	68.25
10	35	37	37	32	35.25
11	56	46	46	58	51.50
12	37	39	38	42	39.00
13	35	41	33	38	36.75
14	30	45	30	40	36.25
15	18	30	20	25	23.25
16	16	10	12	16	13.50

表 2 任务切换关系表

Tab.2 Table of the changeover relationships of tasks

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
2	X	0	X	X	0	X	X	X	0	X	X	X	0	X	X	X
3	X	X	0	0	X	0	0	0	X	0	X	0	X	X	X	X
4	X	X	0	0	X	0	0	0	X	0	X	0	X	X	X	X
5	X	0	X	X	0	X	X	X	0	X	X	X	0	X	X	X
6	X	X	0	X	X	0	0	0	X	0	X	0	X	X	X	X
7	X	X	0	X	X	0	0	0	X	0	X	0	X	X	X	X
8	X	X	0	0	X	0	0	0	0	0	X	0	X	X	X	X
9	X	0	X	X	0	X	X	0	0	X	X	X	0	X	X	X
10	X	X	0	X	X	0	0	0	X	0	X	X	X	X	X	X
11	X	X	X	X	X	X	X	X	X	0	X	X	X	X	X	X
12	X	X	0	0	X	0	0	0	X	X	X	0	X	X	X	X
13	X	0	X	X	0	X	X	X	0	X	X	X	0	X	X	X
14	X	X	X	X	X	X	X	X	X	X	X	X	0	X	X	X
15	X	X	X	X	X	X	X	X	X	X	X	X	X	0	0	0
16	X	X	X	X	X	X	X	X	X	X	X	X	X	0	0	0

表 3 不同类型机器人切换时间表

Tab.3 Table of changeover time by the different robots

机器人类型 r	A	B	C	D
X	20	14	16	18

本例中,首先,根据优先图和切换关系找到序列 1→3→8→4→12→2→5→9→13→6→7→10→11→14→15→16,得到最小切换次数 $\tau_0 = 6$,根据 $C_{at} = 72.5$,然后进行序列分割,得到如图 8 所示的切换结果,即最少切换次数 $\tau_s = 4$.即工位下界 $C_{LB} = 81.8$.

3.2 参数分析

为了确定较佳的内层迭代次数,先设置外层循

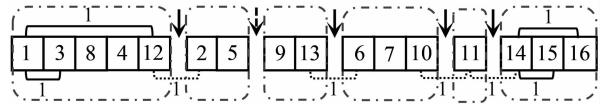


图 8 最少切换次数分析

Fig.8 Analysis of the minimum changeover times

环次数为 1,再根据作业分配的约束条件,产生 20 个初始种群进行试验,并将自适应模拟退火遗传算法(ASAGA)与标准遗传算法(SGA)、模拟退火遗传算法(SAGA)进行对比,得到内层迭代次数对最大适应度和求解时间的影响分别如图 9 和 10 所示。

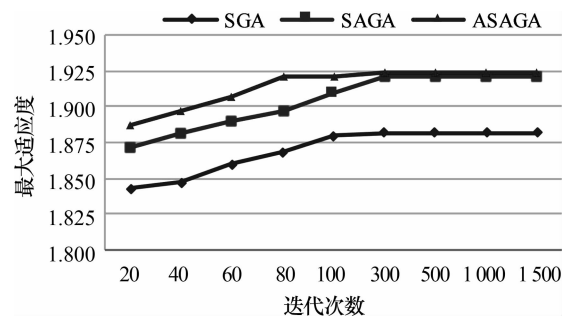


图 9 迭代次数对最大适应度值的影响

Fig.9 The effect of number of iterations to maximum fitness

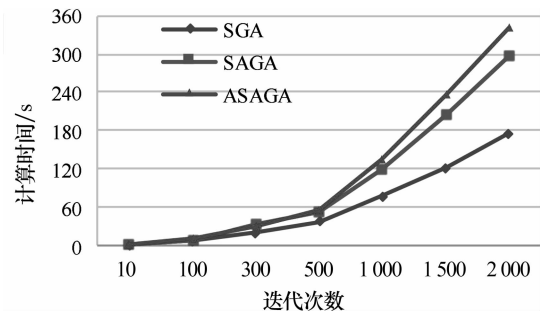


图 10 迭代次数对计算时间的影响

Fig.10 The effect of number of iterations to computation time

由图 9,图 10 可知,当迭代次数达到 300 到 1 500 时,3 种算法的最大适应度值都趋于稳定,同时可以看出 ASAGA 的收敛速度更快,当迭代次数达到 80 次时已经找到了较稳定的最大适应度值.综合考虑时间成本和最大适应度值,本文认为在该试验中,ASAGA 的内层迭代次数定为 80,SAGA 的迭代次数定为 120,SGA 的内层迭代次数定为 300 是合理的。

确定内层迭代次数后,按照上述方式再次进行试验,得到外层迭代次数对 Q 和时间的的影响如表 4 所示.当迭代次数在 30 到 50 时,两种算法都趋于稳定,融入自适应机制的 ASAGA 算法在外循环系数

为 10 次时,解达到了近优值,此时 $Q=16.14\%$,而 SAGA 在外循环系数为 30 次时才达到近优解.因此,ASAGA 相比于 SAGA 表现出了较好的收敛性.

表 4 外层迭代次数对解的相对偏差 Q 和求解时间 T 的影响

Tab.4 The effect on the relative deviation Q & time T by number of outer iterations

G	SAGA			ASAGA		
	C/s	Q	T /ms	C/s	Q	T /ms
1	129	0.577 0	8 012	120	0.467 0	7 764
2	121	0.479 2	15 212	116	0.418 1	14 389
4	118	0.442 5	30 583	111	0.356 9	30 085
6	117	0.430 3	47 605	104	0.271 2	44 588
8	113	0.381 4	62 955	97	0.185 8	56 745
10	104	0.271 4	79 194	95	0.161 4	71 127
30	95	0.161 4	288 455	95	0.161 4	223 508
50	95	0.161 4	484 203	95	0.161 4	362 076

3.3 结果分析

根据以上分析,在运用 ASAGA, SAGA 以及 SGA 算法求解决该问题时,就解的性能而言,ASAGA 和 SAGA 的最/近优值均为 95 s,解的相对偏差为 16.14%,总切换次数为 7 次,而 SGA 的最/近优值为 104 s,解的相对偏差为 27.12%,总切换次数为 12 次,即增加模拟退火机制可以减少装配任务间的切换次数,减少解的相对偏差,提高解的性能;就算法的收敛速度而言,ASAGA 在内、外循环次数分别为 80, 10 次时取得最/近优值,耗时 71.127 s,均小于 SAGA 在内、外循环次数分别为 120, 30 时取得最/近优值时所耗时的 288.455 s 以及 SGA 循环次数为 300 次解趋于平衡时所耗时的 174.905 s,显然,ASAGA 的收敛速度均大于 SAGA 和 SGA.可见,融入模拟退火机制和自适应策略不仅可以减少切换次数,增强解的性能,而且还极大地提高算法的收敛速度.

由表 5 和图 11 可知,工位 1 中,由于任务 1, 2, 5 之间存在优先关系,所以机器人先上件到夹具后,再切换焊枪焊接,完成后需要换回抓手,等待进行下一个节拍,因此有两次切换时间;而在工位 2 中,由于产品结构和焊点位置的不同,使得任务 9 和任务 4 之间需要换枪,且它们之间没有优先关系,因此,可以在前一个节拍先执行任务 9,再执行任务 4,而在下一个节拍先执行任务 4,再执行任务 9,从而节省一次换枪时间;同理,工位 5 中,任务 11 和 13 之间没有优先关系,可节省一次机器人的换枪时间;工位 6 中,由于任务 14, 15 和 16 之间有优先关系,机器人在固定焊枪焊接完成后,换抓手进行检

查和下件,结束后需换回执行固定焊枪焊接时所用的抓手,因此,该过程中有两次切换时间.

表 5 实例求解结果

Tab.5 The result of the example by ASAGA

工位	作业元素	机器人	τ_s	工位时间			空闲时间/s
				AT/s	OT/s	C/s	
1	1,2,5	B	2	62	28	90	5
2	9,4	C	1	79	16	95	0
3	6,3,7,8	B	0	93	0	93	2
4	10,12	B	1	76	14	90	5
5	11,13	C	1	81	16	95	0
6	14,15,16	C	2	62	32	94	1

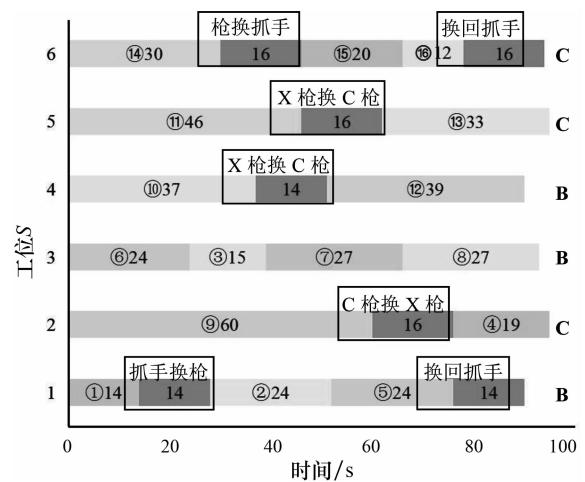


图 11 装配任务切换示意

Fig.11 Illustration of changeover of assembly tasks

4 结论

1)考虑了装配过程中任务间的切换关系,以最小化装配线的工位节拍为目标,提出了相关的定义、定理,建立了该类问题的数学模型.

2)将模拟退火机制和自适应策略引入遗传算法的框架中,构造了自适应模拟退火遗传算法,该算法能够在解的质量和算法的收敛速度两方面取得令人满意的结果,丰富了解决 RALBP 的理论方法.

3)在平衡机器人装配线过程中,考虑装配任务间的切换关系,可以减小装配线的工位节拍,提高不同装配任务间设备的共用性,充分节省资源,降低生产成本,为企业带来更大的经济效益.

参考文献

[1] LEVITIN G, RUBINOVITZ J, SHNITS B. A genetic algorithm for robotic assembly balancing[J]. European Journal of Operational Research, 2006,168:811-825.

- [2] NILAKANTAN J Mukund, PONNAMBALAM S G. An efficient PSO for type-II robotic assembly line balancing problem [C]/Automation Science and Engineering(CASE), 2012 IEEE International Conference on. IEEE, 2012; 600—605.
- [3] GAO Jie, SUN Linyan, WANG Lihua, *et al.* An efficient approach for type-II robotic assembly line balancing problems [J]. Computers & Industrial Engineering, 2009, 56(3): 1065—1080.
- [4] NILAKANTAN J Mukund, PONNAMBALAM S G, JAWAHAR N, *et al.* Bioinspired search algorithms to solve robotic assembly line balancing problems[J]. Neural Computer & Application, 2015, 26; 1379—1393.
- [5] NILAKANTAN J Mukund, PONNAMBALAM S G. Robotic U-shaped assembly line balancing using particle swarm optimization [J]. Engineering Optimization, 2016, 48(2): 231—252.
- [6] AGHAJANI M, GHODSI R, JAVADI B. Balancing of robotic mixed-model two-sided assembly line with robot setup times [J]. The International Journal of Advanced Manufacturing Technology, 2014, 74; 1005—1016.
- [7] NILAKANTAN J Mukund, HUANG George Q, PONNAMBALAM S G. An investigation on minimizing cycle time and total energy consumption in robotic assembly line systems[J]. Journal of Cleaner Production, 2015, 90; 311—325.
- [8] LI Zixiang, TANG Qihua, ZHANG Liping. Minimizing energy consumption and cycle time in two-sided robotic assemble line systems using restarted simulated annealing algorithm[J]. Journal of Cleaner Production, 2016, 135; 508—522.
- [9] YOOSEFELAH A, AMINNAYERI M, MOSADEGH H, *et al.* Type II robotic assembly line balancing problem: An evolution strategies algorithm for a multi-objective model [J]. Journal of Manufacturing Systems, 2012, 31; 139—151.
- [10] ÇIL Z A, METE S, AĞPAK K. A goal programming approach for robotic assembly line balancing problem [J]. IFAC-Papers on Line, 2016, 49(12); 938—942.
- [11] SCHOLL A, BECKER C. State-of-the-art exact and heuristic solution procedures for simple assembly line balancing[J]. European Journal of Operational Research, 2006, 168; 666—693.
- [12] KILINCCI O. A petri net-based heuristic for simple assembly line balancing problem of type 2 [J]. International Journal of Advanced Manufacturing Technology, 2010, 46; 329—338.
- [13] 黄英杰, 姚锡凡, 古耀达. 基于熵的混合粒子群算法在柔性调度中的应用[J]. 湖南大学学报: 自然科学版, 2012, 39(3): 48—52.
HUANG Yingjie, YAO Xifan, GU Yaoda. Hybrid particle swarm optimization based on entropy for flexible job shop scheduling problems[J]. Journal of Hunan University; Natural Sciences, 2012, 39(3); 48—52. (In Chinese)
- [14] 曹一家, 曹丽华, 李勇, 等. 改进的自适应多目标粒子群算法 [J]. 湖南大学学报: 自然科学版, 2014, 41(10); 84—90.
CAO Yijia, CAO Lihua, LI Yong, *et al.* Improved adaptive multiobjective particle swarm algorithm[J]. Journal of Hunan University; Natural Sciences, 2014, 41(10); 84—90. (In Chinese)
- [15] 陈星宇. 基于改进遗传算法的装配生产线平衡技术研究[D]. 上海: 上海交通大学机械与动力工程学院, 2011; 40—41.
CHEN Xingyu. Study on assembly line production balance technique base on the improved genetic algorithm[D]. Shanghai; School of Mechanical Engineering, Shanghai Jiaotong University, 2011; 40—41. (In Chinese)