

基于改进降顶算法的非可靠生产线最优缓存分配

王会琦¹, 薛晓明¹, 吕王勇²

(1. 四川大学 数学学院, 成都 610065; 2. 四川师范大学 数学与软件科学学院, 成都 610068)

摘要: 缓存分配是生产系统设计中一个重要的优化问题. 基于非可靠连续生产线递推分解方法及其统计特性, 提出在给定缓冲配制条件下的蒙特卡洛生产线吞吐量仿真估算方法, 相比于传统的吞吐量估算模型该方法能更准确地描述各种生产场景. 通过构造具有记忆性的禁忌集改进了传统降顶算法, 并将其应用于实际缓存最优分配方案搜索中. 仿真结果表明, 对于各种规模的平衡生产线和非平衡生产线, 改进降顶算法都可以快速有效地搜索到最优解.

关键词: 非可靠生产线; 缓存分配; 蒙特卡罗仿真; 禁忌集; 降顶算法

中图分类号: TP202

文献标志码: A

Improved degraded ceiling algorithm for optimal buffer allocation in unreliable production lines

WANG Hui-qi¹, XUE Xiao-ming¹, LV Wang-yong²

(1. College of Mathematics, Sichuan University, Chengdu 610065, China; 2. College of Mathematics and Software Science, Sichuan Normal University, Chengdu 610068, China. Correspondent: LV Wang-yong, E-mail: lvwangyong@163.com)

Abstract: Buffer allocation is a major optimization problem in manufacturing systems design. The recursive decomposition model and its statistic characteristics are studied in the unreliable successive product firstly, and the Monte-Carlo simulation method is used to estimate the throughput of product line in the condition of fixed buffer allocation. Comparing with the conventional estimation method, it is easy to depict the production conditions. Then, the tabu set based improved degraded ceiling algorithm is proposed to solve the optional buffer allocation problem. Because of the tabu set with memorability, the search efficient of the algorithm can be increased greatly. Finally, the computer simulation results show that the proposed algorithm can search for the optimum solution effectively and fleetly.

Key words: unreliable product line; buffer allocation; Monte-Carlo simulation; tabu set; degraded ceiling algorithm

0 引言

生产线由一系列互相连接的工位和缓存构成, 如果每个工位的操作节拍相同, 则称为平衡生产线. 对于非平衡生产线, 每个工位单次操作时间不同而容易产生工位的空闲和堵塞. 最优缓存分配(BAP)正是平衡生产线和非平衡生产线设计中一个重要的研究问题, 即假定生产线可分配的总缓存容量固定, 如何将这些缓存空间分配到每个工位之间使得整条生产线的吞吐量(生产效率)达到最大. 近年来, 该问题已成为生产制造系统设计方面的一个研究热点^[1-10]. 早在1967年Buzacott就提出了基于Markov链模型的最优缓存分配问题^[11], 但传统的研究基本都是考虑确定性

生产线系统, 如Hillier等^[12-13]. 随后, Spinellis等^[14-16]提出了基于模拟退火算法的最优缓存分配, 并证明了其在大型确定性生产线系统中的有效性. 虽然模拟退火算法在BAP问题中得到一定的应用, 但作为一种基于局部搜索的元启发式方法在解决组合优化问题中的一个普遍缺陷时要设置一系列不确定性参数, 使得算法很难具有通用性, 任何一个参数设置的偏差都会直接影响到搜索效率或使得搜索陷入局部最优解^[17-18].

Nahas于2006年提出利用降顶算法在较大型生产线缓存分配中搜索最优解^[5], 相比模拟退火法, 因其只有单个参数, 更容易控制整个搜索过程. 但随着

收稿日期: 2013-06-09; 修回日期: 2013-11-24.

基金项目: 国家自然科学基金青年科学基金项目(11301360); 四川省教育厅重点基金项目(12ZA137); 高等学校博士学科点专项科研基金项目(20120181120089).

作者简介: 王会琦(1980—), 男, 讲师, 博士, 从事随机优化、随机信号处理等研究; 薛晓明(1990—), 男, 硕士生, 从事优化控制的研究.

生产线规模(即工位数或总缓存容量)的进一步增加, 由于想要搜索到全局最优解而提升门限值后, 解空间的布点与之前门限值下的布点相互独立, 使得之前已被验证的非最优解仍在本门限下重复验证, 造成搜索效率降低.

1 缓存分配问题

考虑整条生产线有 N 个工位(见图 1), 记为 M_1, M_2, \dots, M_N , 第 i 个工位对一件产品的操作时间服从参数为 μ_i 的负指数分布, 即期望操作时间为 μ_i 个最小时间单位; 每个工位间都有一个用于临时存放产品的缓存空间, 容量为 B_1, B_2, \dots, B_{N-1} . 每件待加工产品都从第 1 个工位流入, 然后有序地经过后面所有工位和缓存, 最后从第 N 个工位流出. 假设第 1 个工位不会出现空闲状态, 第 N 个工位不会出现堵塞状态. 实际操作中, 由于每个工位上设备的连续工作必然会产生故障, 这里假设第 i 工位的连续正常工作时间服从参数为 q_i 的几何分布, 即正常工作时期望为 $1/q_i$ 个最小时间单位; 故障后修复时间服从参数为 p_i 的几何分布, 即修复时期望为 $1/p_i$ 个最小时间单位.



图 1 连续生产线结构

最优缓存分配问题就是在 N 个工位之间各自设置多大的缓存空间, 使得整条生产线在实际运行过程中拥有最大的平均吞吐量(工作效率). 这里假设整条生产线上可分配的总缓存空间 K 有限. 上述带约束的最优化问题可以用如下数学模型表示:

$$B_{opt} = \arg \max_{\sum_{i=1}^{N-1} B_i = K, B_i \in \mathbb{N}} f(\mathbf{B}), \quad (1)$$

其中: $\mathbf{B} = [B_1, B_2, \dots, B_{N-1}]$ 为缓存分配向量; B_{opt} 为最优缓存分配向量; $f(\mathbf{B})$ 为按照 \mathbf{B} 分配缓存时整条生产线的平均吞吐量; \mathbb{N} 为自然数集. 最优化问题(1)可以表述为在约束条件 $\sum_{i=1}^{N-1} B_i = K, B_i \in \mathbb{N}$ 下搜索使得 $f(\mathbf{B})$ 值最大的缓存分配向量, 记为 B_{opt} . 这里需要注意的是:

1) 分配方案数. 想要获得使 $f(\mathbf{B})$ 值最大时所对应的 \mathbf{B} , 最直接的方法是考虑所有 \mathbf{B} 的可能情况, 即考虑所有 K 个缓存空间在 $N-1$ 个工位之间进行分配的所有可能. 但由简单的概率论知识可知, 分配方案数可以表示为如下与 K 和 N 的关系:

$$\begin{bmatrix} K + N - 2 \\ N - 2 \end{bmatrix} = \frac{(K + 1)(K + 2) \cdots (K + N - 2)}{(N - 2)!}. \quad (2)$$

可见, 当 K 或 N 较大时, 通过全局搜索来获得最优缓存分配方案几乎不可能, 必须借助高效的搜索算法.

2) 平均吞吐量. 生产线平均吞吐量 $f(\mathbf{B})$ 由工位个数 N 、每个工位的平均工作效率 $\mu_1, \mu_2, \dots, \mu_N$ 及其分布类型、工位间缓存分配向量 $\mathbf{B} = [B_1, B_2, \dots, B_{N-1}]$ 所决定, 目前主要用文献[19]中提出的基于排队论模型的分解法来估算, 通过计算每个工位的闲置概率和堵塞概率, 建立生产线平衡状态方程后迭代求解. 下面基于分解模型采用蒙特卡罗模拟的吞吐量估算方法.

2 吞吐量估算

蒙特卡罗模拟被广泛应用于各工程领域, 尤其是在生产线仿真中, 通过使用离散事件仿真实际操作系统的方法更为有效, 即使对于连续的操作系统, 也只需要通过调整相邻离散事件的时间间隔便可达到期望效果. 只要保证模拟输入数据的准确性和离散事件数量的充分性, 根据仿真结果很容易估算出生产线的各种系统参数和性能. 此外, 蒙特卡罗模拟法估算生产线平均吞吐量需要已知每个工位的相关参数(如工作效率和分布类型等)和每个工位之间的缓存容量. 为了模拟更符合实际生产线工作场景, 假设每个工位的设备会以一定的概率发生故障, 并且维修时间和连续正常工作时间服从几何分布. 下面采用递归分解方式构建仿真, 通过改变模型中相关参数的设置使仿真场景较容易地近似逼近实际生产线场景.

根据实际生产流程, 将有 N 个工位 M_1, \dots, M_N , $N-1$ 个缓存的生产线按图 2 递归分解成等价的 $N-1$ 个易通过计算机高速模拟的简单自相似模型.

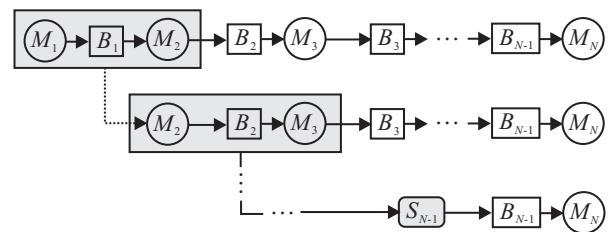


图 2 生产线的递归分解

将第 1 个工位 M_1 记为初始综合工位 S_0 , 所有产品都按照一定的分布从这个工位流入生产线. 通过图 1 的递推方式可以依次得到 $N-1$ 个 $S_n \rightarrow B_{n+1} \rightarrow M_{n+2} (n = 0, 1, \dots, N-2)$ 的简单结构, 其中 S_n 是由 $S_{n-1} \rightarrow B_n \rightarrow M_{n+1}$ 构成的第 n 个综合工位. 通过定义最小仿真时间间隔, 可以按照要求产生一批到达间隔服从负指数分布的模拟产品流入综合工位 S_0 , 每件产品依次经过每个工位和缓存, 在每个工位上的停留时间也是服从事先从真实生产线统计得到或满足生产线设计人员所预期的分布类型, 运用递归方

法,直到产品最后从 $S_{N-2} \rightarrow B_{N-1} \rightarrow M_N$ 结构的工位 M_N 中输出. 根据这个过程中所记录的信息,可以统计生产线的各种指标和性能,如在给定时间 T 内,计数得到离线产品数量 N_c ,这里定义该生产线的平均吞吐量为

$$f(\mathbf{B}) = \frac{N_c(\mathbf{B})}{T}, \quad (3)$$

其中离线产品数 N_c 由每个工位的统计特性、缓存分配向量 \mathbf{B} 、模拟时间 T 、连续正常工作时间及其分布修复时间及其分布等决定. 针对最优缓存 \mathbf{B} 分配问题中的其他参数可按生产线设计要求预先确定,把 N_c 只看成是缓存分配向量 \mathbf{B} 的函数关系形式. 另外,模拟时间 T 的大小一般应能保证离线产品数量达到 100 件以上. 可见,本文的目标是寻找一种缓存分配方式 \mathbf{B} ,使得该目标函数 $f(\mathbf{B})$ 值达到最大.

3 改进降顶算法

降顶法最早由 Nahas 提出并应用于生产线缓存分配问题中,该方法已被证明对于较大型生产线有效,收敛到全局最优解的速度较快^[5]. 对于更大型生产线,虽相比传统启发式算法(如模拟退火算法等)有一定优势,但其收敛速度仍不够理想.

传统降顶法首先需要确定初始缓存分配向量 $\mathbf{B}^{(0)}$ 和目标函数的初始门限值

$$L_0 = f(\mathbf{B}^{(0)}). \quad (4)$$

每次获得新有效缓存分配向量 $\mathbf{B}^{(i)}$ 时,门限值更新为

$$L_i = L_{i-1} + \Delta L. \quad (5)$$

其中: $\Delta L = \frac{\bar{L} - f(\mathbf{B}^{(0)})}{N_{\max}}$ 为等速门限增加值, $\bar{L} = \min\{\mu_1, \mu_2, \dots, \mu_N\}$ 为预先给定的生产线吞吐量的一个上界, N_{\max} 为最大迭代搜索次数. 通过随机选取缓存分配向量 \mathbf{B}^* 作为待选解代入目标函数,其结果 $f(\mathbf{B}^*)$ 作为输出,将本次输出 $f(\mathbf{B}^*)$ 与第 $i-1$ 次的门限值输出 $f(\mathbf{B}^{(i-1)})$ 作比较,如果满足

$$f(\mathbf{B}^*) \geq \min\{f(\mathbf{B}^{(i-1)}), L_{i-1}\}, \quad (6)$$

则将 \mathbf{B}^* 记为第 i 次有效缓存分配向量 $\mathbf{B}^{(i)}$; 否则,随机修正 \mathbf{B}^* 并重新计算其输出,直到其标记为有效输出 $\mathbf{B}^{(i)}$.

显然,上述搜索算法存在一个主要缺陷,若每次待选解 \mathbf{B}^* 不满足式(6)要求,则会作为无效解直接舍弃,而在下一次随机选取待选解时很可能再一次遇上此无效解,造成试验时间的浪费,使得求解效率降低,尤其表现在大型生产线中. 针对此问题,利用无效解构造禁忌集 Θ ,判断随机选取的缓存分配向量 \mathbf{B}^* 是否属于禁忌集,只对不属于 Θ 的待选解计算输出,并在 \mathbf{B}^* 被判断为无效缓存分配向量时用作更新 Θ . 显然,搜索效率将随着禁忌集的不断扩大而提高.

禁忌集设计的关键是以何种有效方式来记录舍弃点 \mathbf{B}^* ,如果只是简单的逐点记录,则随机待选解落入禁忌集的概率近似为零,不能达到提高搜索效率的目的. 所以基于有限覆盖原理,记录以舍弃点 \mathbf{B}^* 为中心、 c 为拒绝半径的领域 $\mathfrak{S}(\mathbf{B}^*, c)$ 更新禁忌集

$$\Theta_i = \Theta_{i-1} \cup \mathfrak{S}(\mathbf{B}^*, c). \quad (7)$$

若随机产生的新待选解落入禁忌集,则不使用该解进入试验,而是重新随机产生待选解直到没有落入禁忌集才进入正常试验步骤,计算其所对应的目标函数值. 整个改进算法流程如下.

Step 1: 参数初始化. 初始化门限增加值 $\Delta L = \frac{1}{N_{\max}}(\bar{L} - f(\mathbf{B}^{(0)}))$, 迭代次数 N_{\max} ; 初始化缓存分配向量 $\mathbf{B}^{(0)}$: $\mathbf{B}^{(0)} \leftarrow \lfloor K/(N-1) \rfloor$, 将剩余缓存 $m = K - (N-1) \times \lfloor K/(N-1) \rfloor$ 分配给 $\mathbf{B}^{(0)}$ 中最中间的 m 个分量; 初始化门限值 $L_0 = f(\mathbf{B}^{(0)})$; 初始化禁忌集参数: 拒绝域半径 $c = 1$, 临时半径 $k = 1$, 初始禁忌集 $\Theta_0 = \emptyset$.

Step 2: 根据 $\mathbf{B}^{(i-1)}$ 随机构造待选解 \mathbf{B}^* : 在向量 $\mathbf{B}^{(i-1)}$ 中随机选择两个满足大于等于 c 的分量 B_r 和 B_s , $B_r \leftarrow B_r - c$, $B_s \leftarrow B_s + c$, 新向量记为 \mathbf{B}^* .

Step 3: 判断 \mathbf{B}^* 是否属于 Θ_{i-1} , 如果属于,则按 Step 2 重新随机构造 \mathbf{B}^* , 否则,转 Step 4.

Step 4: 根据获得的 \mathbf{B}^* 缓存分配方式,利用蒙特卡罗仿真估算目标函数 $f(\mathbf{B}^*)$.

Step 5: 如果 $f(\mathbf{B}^*) \geq \min\{f(\mathbf{B}^{(i-1)}), L_{i-1}\}$,则记 \mathbf{B}^* 为有效缓存分配向量 $\mathbf{B}^{(i)}$,更新门限值 $L_i = L_{i-1} + \Delta L$, 否则,转 Step 6.

Step 6: 更新禁忌集 Θ_i : $\Theta_i = \Theta_{i-1} \cup \mathfrak{S}(\mathbf{B}^*, c)$, $c = \lfloor n(B_r + B_s)/2K \rfloor$.

Step 7: 令 $n = n + 1$, 转 Step 6.

4 数值实验

数值实验中,假设每个工位连续正常工作时间 and 出现故障后修复时间都服从几何分布,其均值即平均正常工作时间 (MTTF) 和平均修复时间 (MTTR) 作为已知参数给定. 即 $MTTR = 1/p_i$ 表示第 i 个工位发生故障所需的修复时间服从参数为 p_i 的几何分布; $MTTF = 1/q_i$ 表示第 i 个工位连续正常工作时间服从参数为 q_i 的几何分布. 分别考虑小型、中型、大型生产线中最优缓存分配问题,并与传统方法进行比较. 模拟过程通过 Matlab 实现.

考虑 5 个工位的小型平衡生产线,其中可以用于分配的总缓存容量 $K = 31$,其余参数设置见表 1. 将基于改进降顶算法 (IDC) 的最优缓存分配与传统降顶算法 (DC)^[5]、Ho 等提出的 Gradient Technique (GT) 算法^[20]、Diomidis 提出的模拟退火法 (SA)^[14] 进行比较,

仿真结果见表 2. 4 种算法所得到的最优缓存分配方案有所差异, 这主要是由生产线吞吐量估算方法或定义形式不同造成的. 其中 SA 算法参数设置如下: 初始温度为 1000, 温度下降速度为 0.98, 外循环最大迭代次数为 1000, 内循环最大迭代次数为 500, 方差为 0.0005.

表 1 小型生产线仿真参数设置

工位	1	2	3	4	5
μ_i	1	1	1	1	1
MTTR	11	19	12	7	7
MTTF	20	167	22	22	26

表 2 小型生产线中各最优缓存分配算法仿真结果

算法	吞吐量	最优缓存分配方案				CPU 时间/s
		B_1	B_2	B_3	B_4	
GT	0.5144	5	11	8	7	14.24
SA	0.5156	7	9	8	7	12.74
DC	0.5184	10	9	6	6	4.18
IDC	0.5184	10	9	6	6	3.39

为了比较 IDC 算法与传统 DC 算法的搜索效率, 仿真中记录各搜索算法所占用的 CPU 时间以及各自搜索过程中生产线平均吞吐量随迭代次数之间的关系见图 3. 从图 3 可以看出, IDC 算法只经过 18 次迭代后即可搜索到最优解, 而传统 DC 算法需要 80 次左右的迭代. 所以, 改进算法通过构造禁忌集并实时更新可以有效减少搜索次数, 提高搜索效率.

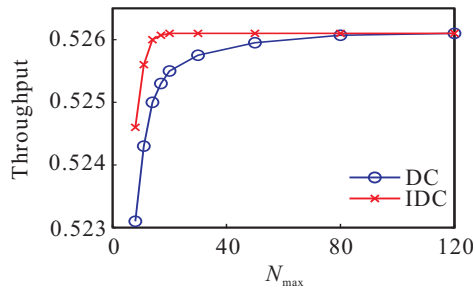


图 3 小型生产线中 IDC 算法与 DC 算法搜索效率比较

考虑 10 个工位的中型平衡生产线, 可用于分配的总缓存容量 $K = 270$, 其余参数设置见表 3. IDC 算法、DC 算法和 SA 算法仿真结果与小型平衡生产线类似 (见表 4). 此外, IDC 算法与 DC 算法的搜索效率比较见图 4. 可见, IDC 算法只用了 70 次迭代即搜索到最优解, 而传统 DC 算法在 600 次迭代后仍未达到最优解. 此外, 从各自运行所占用的 CPU 时间也可以看出, IDC 算法具有较高的搜索效率, 并且对于大型生产线它的搜索效率优势更为明显.

表 3 中型生产线仿真参数设置

工位	1	2	3	4	5	6	7	8	9	10
μ_i	1	1	1	1	1	1	1	1	1	1
MTTR	7	7	5	10	9	14	5	8	10	10
MTTF	20	30	22	22	25	40	23	30	45	20

表 4 中型生产线中各最优缓存分配算法仿真结果

算法	吞吐量	最优缓存分配方案									CPU 时间/s
		B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8	B_9	
SA	0.4421	24	31	32	32	32	33	32	30	24	582.37
DC	0.4692	31	30	37	32	28	26	29	28	29	169.43
IDC	0.4692	31	30	37	32	28	26	29	28	29	76.95

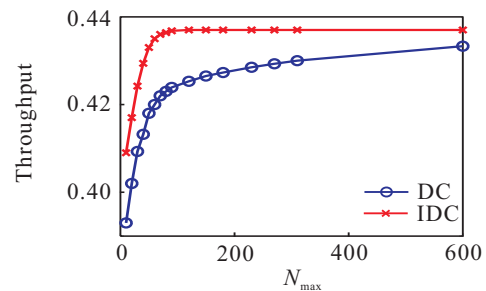


图 4 中型生产线中 IDC 算法与 DC 算法搜索效率比较

对于 20 个工位的大型生产线, 设置每个工位的 $\mu_i = 1, p_i = 0.2, q_i = 0.2 (i = 1, 2, \dots, 20)$ 以及总缓存容量 $K = 100$. IDC 算法与 DC 算法的搜索效率比较见图 5, IDC 经过 600 次左右迭代搜索到最优解, 而 DC 算法经过 2000 次迭代后平均吞吐量仍与最优值有一定差距.

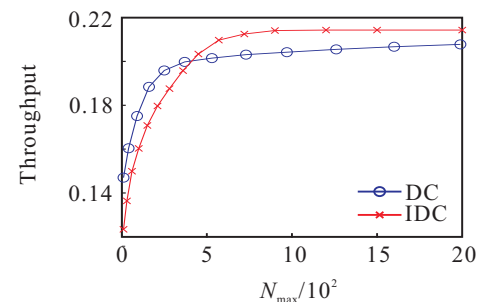


图 5 大型生产线中 IDC 算法与 DC 算法搜索效率比较

5 结 论

针对连续非可靠平衡生产线设计中最优缓存分配问题, 本文给出了基于递归分解模型的新型改进降顶算法, 即通过构造具有记忆性的禁忌集及其更新策略来提高最优解搜索效率. 最后的计算机仿真表明, 此算法不仅可以有效解决小型、中型和大型生产线中缓存分配问题, 而且相比传统降顶算法, 搜索效率提高明显.

参考文献(References)

- [1] Dolgui A, Ereemeev A, Kolokolov A, et al. A genetic algorithm for the allocation of buffer storage capacities in a production line with unreliable machines[J]. J of Mathematical Modeling and Algorithms, 2002, 1(2): 89-104.
- [2] Diamantidis A C, Papadopoulos C T. A dynamic programming algorithm for the buffer allocation problem

- in homogeneous asymptotically reliable serial production lines[J]. *Mathematical Problems in Engineering*, 2004(3): 209-223.
- [3] Vidalis M, Papadopoulos C, Heavey C. On the workload and phaseload allocation problems of short reliable production lines with finite buffers[J]. *Computers & Industrial Engineering*, 2005, 48(4): 825-837.
- [4] Allon G, Kroese D P, Raviv T, et al. Application of the cross-entropy method to the buffer allocation problem in a simulation-based environment[J]. *Annals of Operations Research*, 2005, 134(1): 137-151.
- [5] Nahas N, Ait-Kadi D, Nourelfath M. A new approach for buffer allocation in unreliable production lines[J]. *Int J of Production Economics*, 2006, 103(2): 873-881.
- [6] Dolgui A, Ereemeev A, Sigaev V. HBBA: Hybrid algorithm for buffer allocation in tandem production lines[J]. *J of Intelligent Manufacturing*, 2007, 18(3): 411-420.
- [7] Nahas N, Nourelfath M, Ait-Kadi D. Ant colonies for structure optimization in a failure prone series—Parallel production system[J]. *J of Quality in Maintenance Engineering*, 2008, 14(1): 7-33.
- [8] Shi C, Gershwin S. An efficient buffer design algorithm for production line profit maximization[J]. *Int J of Production Economics*, 2009, 122(2): 725-740.
- [9] Massim Y, Yalaouib F, Amodeob L, et al. Efficient combined immune-decomposition algorithm for optimal buffer allocation in production lines for throughput and profit maximization[J]. *Computers & Operations Research*, 2010, 37(4): 611-620.
- [10] Leyla Demir, Semra Tunali, et al. A tabu search approach for buffer allocation in production lines with unreliable machines[J]. *Engineering Optimization*, 2011, 43(2): 213-231.
- [11] Buzacott J A. Automatic transfer lines with buffer stocks[J]. *The Int J of Production Research*, 1967, 5(3): 183-200.
- [12] Hillier F S, So K C. The effect of the coefficient of variation of operation times on the allocation of storage space in production line system[J]. *IIE Transactions*, 1991, 23(2): 198-206.
- [13] Hillier F S, So K C, Boling R W. Notes: Toward characterizing the optimal allocation of storage space in production line systems with variable processing times[J]. *Management Science*, 1993, 39(1): 126-133.
- [14] Spinellis D, Papadopoulos C T. A simulated annealing approach for buffer allocation in reliable production lines[J]. *Annals of Operations Research*, 2000, 93(1/2/3/4): 373-384.
- [15] Spinellis D, Papadopoulos C T. Stochastic algorithms for buffer allocation in reliable production lines[J]. *Mathematical Problems in Engineering*, 2000(5): 441-458.
- [16] Spinellis D, Papadopoulos C T, Macgregor Smith J. Large production line optimization using simulated annealing[J]. *Int J of Production Research*, 2000, 38(3): 509-541.
- [17] Kirkpatrick S, Gelatt Jr C D, Vecchi M P. Optimization by simulated annealing[J]. *Science*, 1983, 220(4598): 671-679.
- [18] Van Laarhoven P J M, Aarts E H L. *Simulated annealing: Theory and applications*[M]. The Netherlands, 1987: 671-679.
- [19] Dallery Y, Frein Y. On decomposition methods for tandem queueing networks with blocking[J]. *Operations Research*, 1993, 41(2): 386-399.
- [20] Ho Y C, Eyster M A, Chien T T. A gradient technique for general buffer storage design in a production line[J]. *Int J of Production Research*, 1979, 17(6): 557-580.

(责任编辑: 孙艺红)