

基于混合蝙蝠算法的多目标柔性作业车间调度问题

姚 妮, 李红婵

(郑州轻工业学院 计算机与通信工程学院, 河南 郑州 450002)

摘 要: 该文对基本蝙蝠算法进行了改进, 并将其与变邻域搜索算法相结合提出了一种新的混合蝙蝠算法 (Hybrid bat algorithm, HBA). 算法中设计了个体位置向量的表示方法以及位置向量与调度解间的转换机制. 此外, 给出了种群初始化方法和两种邻域结构, 并引入变邻域搜索算法增强算法搜索能力. 最后, 通过对基准算例仿真验证了所提算法的有效性.

关键词: 柔性作业车间调度; 多目标优化; 蝙蝠算法; 变邻域搜索算法

中图分类号: TH165

文献标识码: A

文章编号: 1000-7180(2017)03-0025-05

Hybrid Bat Algorithm for Multi-objective Flexible Job Shop Scheduling Problem

YAO Ni, LI Hong-Chan

(School of Computer and Communication Engineering, Zhengzhou University of
Light Industry, Zhengzhou 450002, China)

Abstract: In this paper, the multi-objective flexible job shop scheduling problem (FJSP) and the basic bat algorithm are introduced at first. Secondly, the basic bat algorithm is improved and combined with the variable neighborhood search algorithm to obtain a new hybrid bat algorithm (HBA). In HBA, an individual position vector representation method and the conversion mechanism between the individual position vector and the scheduling solution are designed. In addition, a population initialization scheme and two neighborhood structures are given, and a variable neighborhood algorithm is introduced to enhance the searching ability. Finally, benchmark instances are simulated to demonstrate the effectiveness of the proposed algorithm.

Key words: flexible job shop scheduling; multi-objective optimization; bat algorithm; variable neighborhood search algorithm

1 引言

作业车间调度问题 (Job shop scheduling problem, JSP) 是一种复杂的离散组合优化问题, 具有很强的理论价值和实际应用背景. 因此, 它一直是制造领域学者们热切关注的问题. 柔性作业车间调度问题 (Flexible job shop scheduling problem, FJSP) 是 JSP 问题的延伸形式, 其更多地考虑了实际的生产状况, 减小了工序在加工过程中对机器选择的限制, 但同时又大大增加了问题求解的难度, 已被证明属

于 NP 难问题^[1].

蝙蝠算法 (Bat algorithm, BA) 是一种受自然界启发的新型智能优化算法, 它模拟蝙蝠利用回声定位原理进行捕食的行为来对优化问题进行求解^[2]. 目前利用蝙蝠算法求解生产车间调度问题的研究还很少, 并且这些研究大都针对的是流水作业车间^[3-5], 还未有关于多目标柔性作业车间调度问题的研究. 因此, 本文在基本蝙蝠算法的基础上进行改进, 使改进后的算法能够更好地适用于多目标 FJSP 问题的求解.

2 多目标 FJSP 问题

本文所研究的多目标 FJSP 问题可描述为:某车间内包含 n 个工件和 m 台机器,不同工件间相互独立且优先级相同,完成每个工件的加工需一道或多道操作,且同一工件的工序间存在固定的加工顺序.各工件具有柔性加工路径,即每道工序可在若干台机器上加工,加工时间则由其所在机器决定.优化目标是为工序分配机器,并排列机器上工序的加工顺序,实现系统三个性能指标的优化,即最大完工时间、机器总负载以及所有机器负载中的最大值.对于此系统,具备几个假设条件:初始零时刻所有工件和机器都是可用的;各机器在同一时刻只允许加工一道工序;工序加工一旦开始不允许被中断;机器加工工序前所需的调整时间忽略不计.对此,建立的数学模型为

$$\min C_{\max} = \min \max_{1 \leq i \leq n} (C_{ij}) \quad (1)$$

$$\min W_T = \min \left(\sum_{k=1}^m W_k \right) \quad (2)$$

$$\min W_{\max} = \min \max_{1 \leq k \leq m} (W_k) \quad (3)$$

$$\begin{aligned} \text{s. t. } C_{ij} &= S_{ij} + \sum_{k=1}^m x_{ijk} p_{ijk} \\ (i &= 1, 2, \dots, n; j = 1, 2, \dots, J_i) \end{aligned} \quad (4)$$

$$\begin{aligned} S_{i(j+1)} - C_{ij} &\geq 0 \\ (i &= 1, 2, \dots, n; j = 1, 2, \dots, J_i - 1) \end{aligned} \quad (5)$$

$$\begin{aligned} S_{i'j'} + \alpha(1 - y_{iji'j'k}) &\geq C_{ij} \\ i, i' &= 1, 2, \dots, n; j, j' = 1, 2, \dots, J_i; \\ k &= 1, 2, \dots, m \end{aligned} \quad (6)$$

$$\begin{aligned} S_{ij} + \alpha y_{iji'j'k} &\geq C_{i'j'} \\ i, i' &= 1, 2, \dots, n; j, j' = 1, 2, \dots, J_i; \\ k &= 1, 2, \dots, m \end{aligned} \quad (7)$$

$$W_k = \sum_{i=1}^n \sum_{j=1}^{J_i} p_{ijk} x_{ijk} \quad (8)$$

$$\sum_{k=1}^m x_{ijk} = 1, i = 1, 2, \dots, n; j = 1, 2, \dots, J_i \quad (9)$$

$$\begin{aligned} x_{ijk} &\in \{0, 1\}, i = 1, 2, \dots, n; j = 1, 2, \dots, J_i; k \\ &= 1, 2, \dots, m \end{aligned} \quad (10)$$

$$\begin{aligned} y_{iji'j'k} &\in \{0, 1\}, i, i' = 1, 2, \dots, n; j, j' = 1, 2, \\ &\dots, J_i; k = 1, 2, \dots, m \end{aligned} \quad (11)$$

模型中各符号含义为, n 为车间内工件数; m 为车间内机器数; J_i 为工件 i 包含的工序数; O_{ij} 为工件 i 的第 j 道工序; p_{ijk} 为工序 O_{ij} 在机器 k 上的加工时间; S_{ij} 为 O_{ij} 的开始加工时间; C_{ij} 为 O_{ij} 的完成时间; $C_{\max}$ 为所有工序最大完工时间; W_k 为机器 k 的负

载; $W_{\max}$ 为所有机器最大负载; W_T 为所有机器总负载; α 为一个很大的数; x_{ijk} 为 $0 \sim 1$ 变量,若工序 O_{ij} 在机器 k 上加工则 $x_{ijk} = 1$,否则 $x_{ijk} = 0$; $y_{iji'j'k}$ 为 $0 \sim 1$ 变量,若机器 k 上工序 O_{ij} 先于工序 $O_{i'j'}$ 加工则 $y_{iji'j'k} = 1$,否则 $y_{iji'j'k} = 0$.

式(1)~(3)表示车间内的三个调度性能;式(4)表示工序加工过程的不可中断性;式(5)表示各工序间的加工顺序;式(6)和式(7)表示在每台机器同时只可加工一道工序;式(8)表示机器负载;式(9)表示工序所在的机器确定后,一直保持到其完工;式(10)和式(11)表示 $0 \sim 1$ 变量.

在求解多目标 FJSP 问题时,线性加权法是一种很常用的方法.本文同样采用该方法描述优化目标: $\min f = \omega_1 C_{\max} + \omega_2 W_T + \omega_3 W_{\max}$,其中 $\omega_1 + \omega_2 + \omega_3 = 1$.

3 基本蝙蝠算法

基本蝙蝠算法是通过模拟蝙蝠觅食时发出的脉冲频率、响度和频度的变化而设计.算法中每个个体代表一只蝙蝠,运行初期以较大的脉冲响度和较低的频度进行搜索,搜索过程中若发现目标则增加脉冲频度 r_i ,减小脉冲响度 A_i ,从而逐渐逼近优化目标.基本蝙蝠算法的实施步骤如下^[4]:

步骤 1 设置算法参数和终止条件,并进行种群初始化.

步骤 2 评价每个个体,找到当前最优个体位置向量 x^* .

步骤 3 更新个体位置及速度向量,式(12)中 fr_i 为个体 i 的脉冲频率, β 为 $[0, 1]$ 间服从均匀分布的随机数,式(13)和(14)中 x_i^t 和 v_i^t 为第 t 次迭代个体 i 的位置和速度向量.

$$fr_i = fr_{\min} + (fr_{\max} - fr_{\min}) \times \beta \quad (12)$$

$$v_i^t = v_i^{t-1} + (x_i^{t-1} - x^*) fr_i \quad (13)$$

$$x_i^t = x_i^{t-1} + v_i^t \quad (14)$$

步骤 4 对于每个个体,任意产生一个 $[0, 1]$ 间服从均匀分布的随机数 rand ,若 $\text{rand} > r_i$,则对当前最优个体位置进行局部搜索构造一个新解.

步骤 5 同样对于每个个体,任意产生一个 $[0, 1]$ 间服从均匀分布的随机数 rand ,若随机数 $\text{rand} < A_i$ 且新产生的解优于当前最优解,则接受新解,并更新脉冲频度 r_i 和响度 A_i .

步骤 6 更新当前最优个体位置向量 x^* .若达到终止条件,转到步骤 7;否则,转到步骤 3.

步骤 7 算法结束.

4 混合蝙蝠算法

4.1 调度解的表示

根据柔性作业车间调度问题的特点,FJSP 问题中包含两个子问题:机器分配问题和工序排序问题.因此,一个调度解由两部分构成,每部分的长度均为车间内工序的总数 $q = \sum_{i=1}^n J_i$. 图 1 为调度解的示例图,图中共包含 3 个工件,每个工件包含两道工序.前半段为工序排序方案,各元素值为工序编号,同一工件不同工序具有相同的编号,例如第 2 个“1”表示工件 1 的第 2 道工序 O_{12} ;后半段为机器分配方案,元素值为工序所在的机器的编号,按照图 1 中指定的顺序存储.

O_{21}	O_{11}	O_{31}	O_{22}	O_{12}	O_{32}	O_{11}	O_{12}	O_{21}	O_{22}	O_{31}	O_{32}
2	1	3	2	1	3	1	2	1	2	4	3

图 1 调度解的表示

4.2 个体位置向量表示与转换

蝙蝠算法通过不断地更新蝙蝠个体的位置和速度对优化目标进行搜索.假设存在一个 l 维实数型个体位置向量 $x = \{x(1),x(2),\cdots,x(l)\}$,其中各元素取值范围均为 $[-\delta,\delta]$. 根据调度解的表示,每个位置向量也包含两部分,前半段对应工序排序方案,后半段对应机器分配方案,整个个体位置向量的长度与调度解的长度相同,即 $l = 2q$. 图 2 为个体位置向量的示例图,示例中包含 3 个工件,每个工件包含两道工序,前半段和后半段的数值分别按照图 2 中指定的顺序进行保存.

O_{11}	O_{12}	O_{21}	O_{22}	O_{31}	O_{32}	O_{11}	O_{12}	O_{21}	O_{22}	O_{31}	O_{32}
-0.3	1.5	-1.9	1.3	2.6	-0.8	2.2	2.7	-0.7	1.1	-0.6	1.4

图 2 个体位置向量

基本蝙蝠算法中每个个体位置元素均为连续值,而柔性作业车间调度问题则属于离散组合优化问题.因此,采用蝙蝠算法求解 FJSP 问题时,首先要解决的是如何实现解空间与问题空间之间的映射关系,即设计一种转换机制,实现个体位置向量与调度方案间的转换.本文设计的转换机制分为以下两部分.

(1)工序排序部分:首先将工序名称转换成工序编号;然后根据个体位置向量中元素数值的大小进行升序排列,并找到各数值对应的工序编号;由此得到的工序序列即为所得的工序排序方案,如图 3 所示.

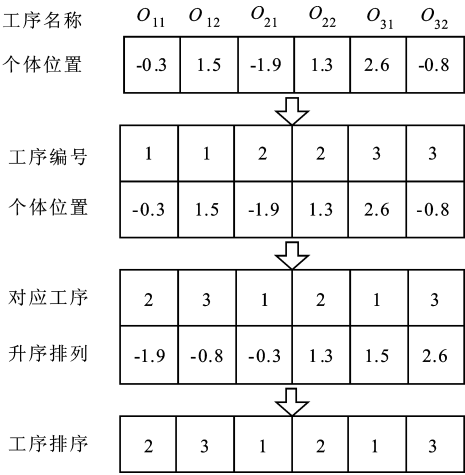


图 3 个体位置与工序排序间转换

(2) 机器分配部分:按照文献[6]中的转换方式,如式(15)所示,将个体位置向量后半部分的元素值转换成机器编号.式中, $z(h)$ 为个体位置向量中第 h 个元素对应的工序的可选机器的总数, $u(h) \in [1,z(h)]$ 为转换后工序对应的机器在工序可选机器集合中的序号,即工序可选机器集合中第 $u(h)$ 个元素就是得到的机器的编号.

$$u(h) = \text{round}(\frac{x(h) + \delta}{2\delta}(z(h) - 1) + 1) \\ \frac{l}{2} + 1 \leq h \leq l \tag{15}$$

4.3 初始调度解生成与转换

初始调度解的好坏在一定程度上影响着算法的性能.一个好的初始调度解有利于改善算法最终解的质量,并加快算法的收敛速度.根据 FJSP 问题包含的两个子问题,种群初始化分两阶段进行,即机器分配阶段和工序排序阶段.

对于机器分配阶段,采用随机生成法、全局搜索法和局部搜索法^[7],按照一定比例(20%、60%和 20%)生成初始种群.后两种方法通过确定初始机器分配方案,平衡机器的工作负载,提高机器使用率,同时还考虑了工序的最大完工时间.对于工序排序阶段,采用基于搜索的方法^[8].根据已生成的机器分配方案,随机产生若干道工序排序方案,并评估各自的目标值.选择最优的一个作为初始排序方案,结合前后两部分方案得到一个初始调度解.

根据蝙蝠算法的搜索机制,初始调度解生成后还需转换成个体位置向量才能进行下一步搜索.该转换机制同样分为两部分.

(1) 工序排序部分:首先在 $[-\delta,\delta]$ 间随机产生 $\frac{l}{2}$ 个随机数,将随机数向量按升序排列,并与初始

排序方案一一对应;然后根据工序编号确定个体位置向量中各元素的值,转换过程如图 4 所示.

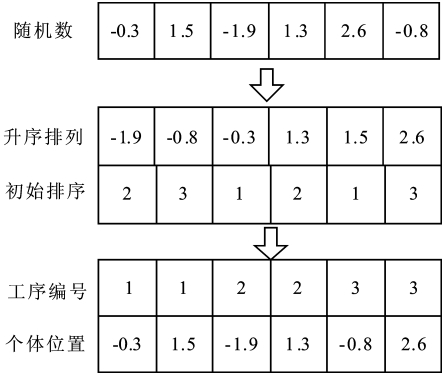


图 4 初始排序与个体位置间的转换

(2) 机器分配部分:按照文献[17]中的转换方式,如式(16)所示,若工序可选机器数为 1,即 $z(h) = 1$ 时, $x(h)$ 取 $[-\delta, \delta]$ 内任一数值.

$$x(h) = \begin{cases} \frac{2\delta}{z(h)-1}(u(h)-1)-\delta, & z(h) \neq 1 \\ x(h) \in [-\delta, \delta], & z(h) = 1 \end{cases} \quad (16)$$

4.4 变邻域搜索

4.4.1 邻域结构

对于 FJSP 问题,邻域搜索有利于增强算法解的质量.本文采用以下两种邻域结构,分别作用于个体位置向量前后两部分.

(1) 邻域结构 N_1 :在个体位置向量的前半段中任选两个对应于不同工件工序的两个元素,互换两元素的位置.

(2) 邻域结构 N_2 :在个体位置向量的后半段任选一个元素,其对应工序的可选机器数应大于 1,将该工序分配至加工时间最短的可选机器上,然后将式(16)生成的数值替代原值.

4.4.2 局部搜索

作为变邻域搜索算法中的关键组成部分,局域搜索是在当前解的邻域内搜索局域最优解.根据上述邻域结构 N_1 和 N_2 ,设计一种基于阈值接受的局部搜索方法.具体步骤如下:

步骤 1 算法初始化,获取初始解 s ,设置阈值 $dr > 0$,终止条件 $\rho_{\max}$.

步骤 2 设置 $\rho \leftarrow 0, u \leftarrow 1$.

步骤 3 while $\rho < \rho_{\max}$
if $(u = 1)$ then $s' \in N_1(s)$; else if $(u = 0)$ then $s' \in N_2(s)$ end if

if $(f(s') - f(s) \leq dr)$ then $s \leftarrow s'$; else $u \leftarrow$

$|u - 1|$ end if

$\rho \leftarrow \rho + 1$

end while

步骤 4 设置局部最优解 $s' \leftarrow s$.

4.5 脉冲频度和响度的更新

采用文献[5]中脉冲频度和响度的更新方法,如式(17)和式(18)所示.其中 t 表示迭代次数, $t_{\max}$ 表示最大迭代次数, f_i 表示个体 i 的目标值, $f_{\max}$ 和 $f_{\min}$ 分别表示种群中个体的最大和最小目标值.式(17)与 Sigmoid 函数具有相似的曲线形式[5],这样可在算法早期以较大概率搜索较优个体,加快收敛速度;而在算法后期由于脉冲频度较大,可以确保算法的多样性.式(18)中个体 i 的响度与其自身目标值有关,较优的个体拥有较低的响度值.

$$r_i^t = (1 + \exp(-\frac{10}{t_{\max}} \times (t - \frac{t_{\max}}{2}) + r_i^0))^{-1} \quad (17)$$

$$A_i = \frac{f_i - f_{\min}}{f_{\max} - f_{\min}} \quad (18)$$

4.6 HBA 算法流程

HBA 算法具体流程如下:

步骤 1 初始化种群,设置算法参数和终止条件 $t_{\max}$.

步骤 2 评价每个个体,并找出当前最优个体位置向量.

步骤 3 根据式(12)~(14)更新蝙蝠个体的位置和速度向量.

步骤 4 对于每个个体,若 $\text{rand} > r_i$,执行 4.4.2 节中局部搜索操作,产生新的个体.

步骤 5 评价每一个新个体,若优于当前最优解且满足 $\text{rand} < A_i$,则接受新解,然后根据式(17)和式(18)更新 r_i 和 A_i .

步骤 6 更新当前最优个体位置向量,然后对其执行变邻域搜索.

步骤 7 检查终止条件.若满足,转到步骤 8,否则转到步骤 3.

步骤 8 算法结束.

5 仿真实验

本文针对多目标柔性作业车间调度问题对算法进行仿真.采用 Fortran 和 MATLAB 语言混合编程,并在 WinXP 操作系统下内存 4 GB 的 Intel Core (TM) i5-2400 CPU 3.10 GHz 的计算机上运行.仿真过程中各参数如表 1 所示.

表 1 仿真参数设置情况

种群大小	$t_{\max}$	$\rho_{\max}$	w_1	w_2	w_3
80	200	10	0.5	0.3	0.2

采用由 Kacem 等^[9]设计的四个基准算例验证本文所提算法的可行性. 为了进一步测试算法,将

表 2 对比结果

算例	$n \times m$	目标	AL+CGA ^[9]		PSO+SA ^[10]		PSO+TS		HBA	
算例 1	4×5	$C_{\max}$	16	—	—	—	11	—	11	12
		W_T	34	—	—	—	32	—	32	32
		$W_{\max}$	10	—	—	—	10	—	10	8
算例 2	8×8	$C_{\max}$	15	16	15	16	14	15	16	15
		W_T	79	75	75	73	77	75	73	75
		$W_{\max}$	13	13	12	13	12	12	13	12
算例 3	10×10	$C_{\max}$	7	—	7	—	7	—	8	8
		W_T	45	—	44	—	43	—	43	42
		$W_{\max}$	5	—	6	—	6	—	5	6
算例 4	15×10	$C_{\max}$	23	24	12	—	11	—	12	12
		W_T	95	91	91	—	93	—	92	91
		$W_{\max}$	11	11	11	—	11	—	11	12

实验结果表明,对于算例 1,HBA 算法获得的结果优于 AL+CGA,并能够获得与 PSO+TS 算法相同甚至更多的解;对于算例 2,HBA 算法可获得与 PSO+SA 算法完全相同的结果;对于算例 3,HBA 算法能够获得比 AL+CGA 和 PSO+SA 更优的结果,并且获得比其他算法更多的解,为实际调度提供更多的选择;对于算例 4,HBA 算法结果非常接近 PSO+SA 算法,并能获得解的个数多于 PSO+SA 算法和 PSO+TS 算法. 因此,本文算法求解多目标柔性作业车间调度问题具有一定的有效性.

6 结束语

本文将蝙蝠算法和变邻域搜索算法相结合,提出了一种混合蝙蝠算法用于求解多目标柔性作业车间调度问题. 根据问题的特点,设计了一种个体位置向量的表示方法,并给出了一种个体位置向量与调度解之间的转换机制,实现离散组合优化问题的连续编码. 为了提高算法的性能,设计了一种种群初始化方法,并在算法中引入变邻域搜索算法对当前最优个体进行局部搜索,以增强算法局部搜索能力. 通过基准实例验证了本文算法在求解多目标柔性作业车间调度问题方面具有一定的有效性. 蝙蝠算法是一种新兴的群智能优化算法,目前将它运用于解决车间调度问题的研究还处于初步阶段,今后还需进

HBA 算法与其他三种现有文献中提出的算法进行比较,这三种算法分别为 AL+CGA^[9]、PSO+SA^[10]和 PSO+TS,对比结果如表 2 所示,其中“—”表示文献中没有给出相应的实验数据.

一步探索,将其用于其他更复杂的车间调度问题.

参考文献:

[1] Brandimarte P. Routing and scheduling in a flexible job shop by tabu search [J]. Annals of Operations Research, 1993, 41(1):157-183

[2] Yang X S. A new metaheuristic bat-inspired algorithm [C]//Nature inspired cooperative strategies for optimization (NICSO 2010). Germany: Springer, Berlin Heidelberg, 2010:65-74.

[3] Marichelvam M K, Prabakaran T, Yang X S, et al. Solving hybrid flow shop scheduling problems using bat algorithm[J]. International Journal of Logistics Economics and Globalisation, 2013, 5(1): 15-29.

[4] 马邦雄,叶春明. 基于蝙蝠退火算法的无等待流水线调度问题研究[J]. 数学理论与应用, 2014, 34(1): 92-101.

[5] Luo Q, Zhou Y, Xie J, et al. Discrete bat algorithm for optimal problem of permutation flow shop scheduling[J]. The Scientific World Journal, 2014,25(1):1-15.

[6] Yuan Y, Xu H, Yang J. A hybrid harmony search algorithm for the flexible job shop scheduling problem [J]. Applied Soft Computing, 2013, 13 (7): 3259-3272.

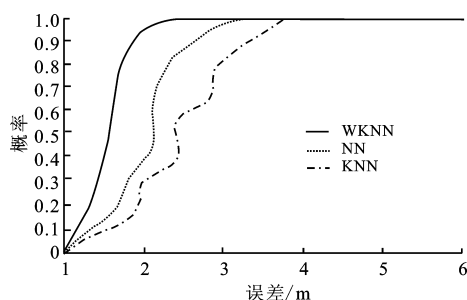


图 8 WKNN/KNN/NN 三种算法定位精度对比

6 结束语

本文针对传统 Wifi 指纹定位算法存在的三个突出的问题展开研究. 提出成新的指纹算法 WKNN, 并将新的指纹匹配算法 WKNN 应用到文本设计的 Wifi 指纹定位系统中. 为了验证算法改进的有效性, 我们在真实的场景中进行了定位实验, 实验结果表明: 本文提出的改进算法在定位精度和稳定性方面较传统 Wifi 指纹定位算法有大幅度的提高. 对于推动室内定位技术的发展具有重要的意义.

参考文献:

- [1] Alonso J M, Ocana M, Hernandez N, et al. Enhanced WiFi localization system based on soft computing techniques to deal with small-scale variations in wireless sensors[J]. Applied Soft Computing, 2011, 11(8): 4677-4691.
- [2] Galvan Tejada I, Sandoval E I, Brena R. Wifi bluetooth based combined positioning algorithm[J]. proceed Engineering, 2012, 7(4): 101-108.

- [3] Zandbergen P A. Accuracy of iPhone locations: A comparison of assisted GPS, WiFi and cellular positioning[J]. Transaction in GIS, 2009, 13(S1): 5-25.
- [4] Christos Laoudias, Robert Piche, Christos G. Device self-calibration in location systems using signal strength histograms[J]. Journal of Location Based Services, 2013, 7(3): 165-181.
- [5] Yang Z, Zhou Z, Liu Y. From RSSI to CSI: Indoor localization via channel response[J]. ACM Computing Surveys, 2013, 46(2): 1-25.
- [6] Fang S H, Liu T N, Lee K C. A novel algorithm for multipath fingerprinting in Indoor WLAN environments[J]. IEEE Transactions on Wireless Communications, 2008, 7(9): 3579-3588.
- [7] Huaiyu Li, Xiuwan Chen, Guifei Jing, et al. An indoor continuous positioning algorithm on the move by fusing sensors and WiFi on smart phones[J]. Sensor, 2015, 15(12): 31244-31267.
- [8] Wang J, Hu A, Liu C, et al. A floor-map-aided WiFi/pseudo-odometry integration algorithm for an indoor positioning system [J]. Sensors, 2015, 15(4): 7096-7124.

作者简介:

汪伦杰 女, (1992-), 硕士研究生. 研究方向为数字媒体设计. E-mail: 1530305469@qq.com.

廖兴宇 男, (1987-), 博士研究生. 研究方向为大数据与数据挖掘.

潘伟杰 男, (1983-), 博士, 副教授, 硕士生导师. 研究方向为文化遗产数字化、数字化艺术与设计、云计算与安全.

吕健 男, (1983-), 博士. 研究方向为创新设计、工业设计等.

(上接第 29 页)

- [7] 张国辉, 高亮, 李培根, 等. 改进遗传算法求解柔性作业车间调度问题[J]. 机械工程学报, 2009, 45(7): 145-151.
- [8] Demir Y, İsleyen S K. An effective genetic algorithm for flexible job-shop scheduling with overlapping in operations[J]. International Journal of Production Research, 2014, 52(13): 3905-3921.
- [9] Kacem I, Hammadi S, Borne P. Pareto-optimality approach for flexible job-shop scheduling problems: hybridization of evolutionary algorithms and fuzzy logic [J]. Mathematics and computers in simulation, 2002, 60(3): 245-276.

- [10] Xia W, Wu Z. An effective hybrid optimization approach for multi-objective flexible job-shop scheduling problems[J]. Computers & Industrial Engineering, 2005, 48(2): 409-425.

作者简介:

姚妮 女, (1978-), 土家族, 硕士, 实验师. 研究方向为智能信息处理、计算智能及其应用.

E-Mail: zzulijsjyn@163.com.

李红婵 女, (1983-), 硕士, 讲师. 研究方向为智能信息处理、计算智能及其应用.