

文章编号:1001-9081(2007)04-0936-03

基于蚁群粒子群算法求解多目标柔性调度问题

张维存^{1,2}, 郑丕谔¹, 吴晓丹²

(1. 天津大学 管理学院, 天津 300072; 2. 河北工业大学 管理学院, 天津 300130)

(zweicun@sina.com)

摘要: 通过分析多目标柔性作业车间调度问题中各目标的相互关系, 提出一种主、从递阶结构的蚁群粒子群求解算法。算法中, 主级为蚁群算法, 在选择工件加工路径过程中实现设备总负荷和关键设备负荷最小化的目标; 从级为粒子群算法, 在主级工艺路径约束下的设备排产中实现工件流通时间最小化的目标。然后, 以设备负荷和工序加工时间为启发式信息设计蚂蚁在工序可用设备间转移概率; 基于粒子向量优先权值的大小关系设计解码方法实现设备上的工序排产。最后, 通过仿真和比较实验, 验证了该算法的有效性。

关键词: 蚁群算法; 粒子群算法; 柔性作业; 车间调度; 多目标优化

中图分类号: TP18 **文献标识码:** A

Ant colony and particle swarm optimization algorithm-based solution to multi-objective flexible job-shop scheduling problems

ZHANG Wei-cun^{1,2}, ZHENG Pi-e¹, WU Xiao-dan²

(1. School of Management, Tianjin University, Tianjin 300072, China;

2. School of Management, Hebei University of Technology, Tianjin 300130, China)

Abstract: A hybrid of ant colony and particle swarm optimization algorithms was proposed to solve the multi-objective flexible job-shop scheduling problem based on the analysis of objectives and their relationship. The hybrid was formulated in a form of hierarchical structure. The ant colony algorithm was performed at the master level to minimize the total load and bottleneck load through selecting job-processing route, while the particle swarm optimization algorithm was carried out at the slave level to minimize the makespan through scheduling the operations with machines without violating the result from the master level. The transfer probabilities of ant between machines were designed by using heuristic information of processing time and machine load. The decoding method of particle vector was well designed in order to sequence operations of every machine based on the size relations of element priority values. The simulation and results from comparison with other algorithms demonstrate the effectiveness of the proposed algorithm.

Key words: ant colony algorithm; Particle Swarm Optimization (PSO); flexible job; job-shop scheduling; multi-objective optimization

0 引言

具有路径柔性的作业车间调度问题(Flexible Job-Shop Scheduling Problem, FJSSP)考虑工序的多台可选加工设备及设备间的依赖关系, 能够动态选择加工路线及加工属性, 这类问题是经典 JSSP 的扩展, 比一般的 JSSP 更具复杂性。目前, 具有多目标的 FJSSP 由于更贴近实际生产需求而引起了人们的关注^[1,2]。采用诸如数学规划法、Petri 网等方法解决此类问题, 其计算时间会随着调度问题规模扩大呈现指数增长现象。因此, 优化此类问题需要研究新的方法。

蚁群算法(Ant Colony Optimization, ACO)是一种全局优化启发式算法^[3]。该算法具有正反馈作用, 可利用最优解信息和启发式信息增强全局求解能力。而粒子群算法(Particle Swarm Optimization, PSO)^[4]具有个体数目少、计算简单、鲁棒性好等优点。基于蚁群算法和粒子群算法, 本文提出一种解决多目标 FJSSP 的新方法。实验表明, 该具有较好的求解能

力和较强的实用性。

1 多目标 FJSSP 的数学模型

1.1 问题描述

假定加工系统有 M 台设备和 N 个工件, 每个工件包含一道或多道工序, 工序顺序是预先确定的, 每道工序可以在多台不同设备上加工, 加工时间随设备性能不同而变化。调度目标是选择最佳的工序加工设备, 并确定每台设备上工件的最佳加工顺序, 使各工件的流通时间、关键设备负载和设备总负载最小。

1.2 数学模型

N 为工件数量; M 为设备数量; Ω 为所有设备集合; Ω_{eg} 为工件 e ($e = 1, \dots, N$) 的第 g ($g = 1, \dots, J_e$) 道工序可选设备集, $\Omega_{eg} \subseteq \Omega$; J_e 为工件 e 需加工的工序数; t_{egk} 为工件 e 的第 g 道工序在机器 k ($k \in \Omega_{eg}$) 的加工时间; S_{egk} 为工件 e 的第 g 道工序在机器 k 上的开始时间; E_{egk} 为工件 e 的第 g 道工序在机

收稿日期: 2006-10-27; 修订日期: 2006-12-16 基金项目: 河北省自然科学基金资助项目(F2006000090)

作者简介: 张维存(1975-), 男, 河北乐亭人, 博士研究生, 主要研究方向: 工业工程; 郑丕谔(1942-), 男, 福建莆田人, 教授, 博士生导师, 主要研究方向: 管理科学与工程经济; 吴晓丹(1973-), 男, 河北邯郸人, 副教授, 主要研究方向: 智能技术应用。

器 k 上的完工时间; EM_k 为所有工件在设备 k 上的完工时间; W_k 为设备 k 的负载(总加工时间);当工件 e 的第 g 道工序和工件 i 的第 j 道工序在同一台机器上执行且工序 j 紧先于工序 g 时, $R_{jeg} = 1$;否则, $R_{jeg} = 0$ 。当工件 e 的第 g 道工序在机器 k 上执行时, $X_{egk} = 1$;否则, $X_{egk} = 0$ 。

目标函数:

$$\text{Min}(EE) = \max_{k=1, \dots, M} (EM_k) \quad (1)$$

$$\text{Min}(W_M) = \max_{k=1, \dots, M} (W_k) \quad (2)$$

$$\text{Min}(W_T) = \sum_{k=1}^M W_k \quad (3)$$

约束条件:

$$EM_k = \max(E_{egk}), X_{egk} = 1, e = 1, \dots, N, g = 1, \dots, J_e \quad (4)$$

$$S_{egk} - E_{e(g-1)n} \geq 0, X_{egk} = X_{e(g-1)n} = 1 \quad (5)$$

$$S_{egk} - E_{ijk} \geq 0, X_{ijk} = X_{egk} = 1, R_{jeg} = 1 \quad (6)$$

其中, EE, W_M, W_T 分别表示工件流通时间、关键设备负荷和设备总负荷;式(1)、(2)、(3) 分别表示使各工件流通时间、关键设备负荷、设备总负荷最小的优化目标;(4) 表示设备 k 的完工时间取决于在其上加工的所有工件中最后一个工件的完工时间;(5) 表示工件 e 的第 g 道工序必须在第 $g-1$ 道工序完成后才能开始;(6) 表示任一确定时刻,机器 k 不能同时加工任意两个不同的工件,也不能同时加工任意两道不同的工序。这里, e, k 可分别看成以自然数为序的工件与设备的编号(标识)。

2 多目标 FJSSP 的蚁群粒子群算法实现

2.1 蚁群粒子群算法结构设计

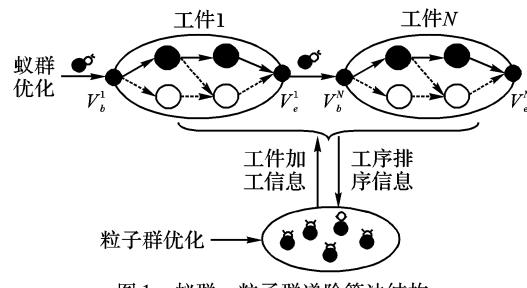


图1 蚁群—粒子群递阶算法结构

目标函数中的三个目标具有一定的联系:1)选用加工时间短的工序加工设备可使设备总负荷最小(目标3);2)在设备总负荷最小条件下,若使设备间负荷尽可能的均匀分配则可使关键设备负荷最小(目标2);3)确定设备上最佳调度方案(工件加工顺序),减少工件加工过程的等待时间可使工件流通时间最小(目标1)。因此,为各工序选出加工时间短的设备并使各设备负荷均衡是解决多目标 FJSSP 的关键。本文采用分解的方法,首先选定工件加工路径,再依此对各设备排产。算法采用蚁群粒子群算法两层递阶结构,如图 1 所示。图中,椭圆代表工件,小圆圈代表某道工序可选设备。蚁群算法中,把工序可选设备看成蚂蚁要游历的节点(障碍物)。蚂蚁依次游历各工件,选出穿过工件各道工序的路径(即选出工序加工设备,如图中黑实线所示)。蚂蚁游历过程实现目标 2 和 3 的优化,具体方法见 2.4 节。当蚂蚁游历完各工件,就将该蚂蚁游历的路径信息(即工艺路线)编码成粒子向量(见 2.2 节),生成初始的粒子群。该粒子群优化实现目标 1 并反馈给蚁群算法,作为下一只蚂蚁游历的依据(详见 2.3 节)。

2.2 蚂蚁地图与粒子群编码设计

由 2.1 节分析,蚂蚁通过各道工序间游历选择工序加工设备,需用到工序设备地图(如图 1 椭圆内小圆圈连成的网络图)。工序设备地图由工件待加工工序及其可选加工设备生成,反映了蚂蚁从某道工序出发下一步可游历节点(工序可选设备)状况。算法中在工件首工序前设虚拟起始工序(如图 1 中 V_b^1),蚂蚁由此进入工件选择第一道工序加工设备;工件末工序后设虚拟终止工序(如图 1 中 V_e^1),蚂蚁由此离开工件,完成工件所有工序加工设备选择。

从级粒子群算法中采用基于工序的编码方式,首先定义

总工序数 $L = \sum_{e=1}^N J_e$,并把一个粒子表示为一个 L 维的向量。 L 维向量的每个分量包含信息:工件号、工序号、设备号、优先权值、开始加工时间、加工时间和加工完成时间等。工件号 e 和工序号 g 为每个分量的标识。这里 L 维向量各分量优先权值取 $[0,1]$ 范围的数,代表该工序被解码(加工)的优先级,也是本算法优化的对象。而其他信息由工件加工信息初始化(如工件号、工序号、设备号、加工时间),或者由算法解码得到(如开始加工时间、加工完成时间)。同时,为解码的方便,将每个粒子对应的 L 维的向量根据工件的最大工序数($Q = \max(J_e), e = 1, \dots, N$) 分成若干段。其中,含所有工件的第 1 道工序加工信息的分量编码成 L 维向量的第 1 段,含所有工件的第 2 道工序加工信息的分量编码成 L 维向量的第 2 段,依此类推。如图 2,第 1 段向量由所有工件的第 1 道工序加工信息组成,而第 $p_j \geq q$ 段向量由 q 的工件的第 q 道工序加工信息组成。



图2 粒子向量分段结构

2.3 算法过程

步骤 1:生成一只蚂蚁 a ,并依工序设备间转移概率逐步选定各工件的工序加工设备。设置蚂蚁个数计数器 $r = r + 1$ 。

步骤 2:粒子群优化,生成蚂蚁 a 游历路径所对应的调度方案及其工件流通时间 EE_a 。

1) 初始化粒子群,即由蚂蚁 a 已游历(选择)工件工序信息初始化粒子分量中的工件号、工序号、设备号和加工时间等,用 $[0,1]$ 随机数初始化优先权值;

2) 对粒子向量解码(方法见 2.5 节),计算适应值函数 $f = \frac{1}{EE}$,根据流通时间最小评价所有粒子,寻找各粒子的个体极值 $pbest$ 和全局极值 $gbest$;

3) 若某个粒子的当前适应值 f 优于其个体极值的适应值 f_{pbest} ,则由该粒子更新其个体极值 $pbest$;若优于全局极值的适应值 f_{gbest} ,则由该粒子更新全局极值 $gbest$;

4) 若连续 10 次优化的全局极值 $gbest$ 无变化,则粒子群优化结束,并作如下处理:(1)由全局极值 $gbest$ 及其适应值更新蚂蚁 a 的调度方案及其 EE_a ;(2)若 $EE_a < EE_{best}^q$,则令 $EE_{best}^q = EE_a$;(3)若 $r < Q^a$,则转入步骤 1;若 $r = Q^a$,转入步骤 3。设 EE_{best}^q 为本次循环的最短流通时间, Q^a 为蚁群规模。否则,转入 5);

5) 对每一个粒子的向量优先权值,参照文献[5]方法计算速度向量 $V[L]$ 和位置向量 $X[L]$,其中 $W \in (0,1), C_1 = C_2 = 2$,即只更新粒子向量优先权值,转入步骤 2);

步骤 3: 更新蚁群的信息素浓度。

1) 若 $EE_{best} > EE'_{best}$, 则 $EE_{best} = EE'_{best}$, 并更新最优调度方案。设 EE_{best} 为最优调度方案的工件流通时间。

2) 按参考文献[6]提出的 MMAS 更新规则对工序设备地图上信息素作更新处理。

3) 令 $q = q + 1$, 若 $q > Q$, 则算法结束; 否则令 $r = 0$, 转入步骤 1。设 q 为循环计数器, Q 为算法循环次数, $q = 1, \dots, Q$ 。

2.4 蚁群转移概率设计

本文采用如下方法设计蚁群工序间的转移概率, 以实现减小工序加工时间, 使设备总负荷最小(目标 3) 和平衡各设备负荷, 使关键设备负荷最小(目标 2) 的两个目标。

$$P_{k_g, k_j}^a(m) = \begin{cases} \frac{[\tau_{k_g, k_j}(m)]^\alpha [\eta_{k_g, k_j}(m)]^\beta}{\sum_{l \in W_{k_j}^a} [\tau_{k_g, k_l}(m)]^\alpha [\eta_{k_g, k_l}(m)]^\beta}, & k_j \in W_{k_j}^a \\ 0, & k_j \notin W_{k_j}^a \end{cases} \quad (7)$$

其中, $P_{k_g, k_j}^a(m)$ 表示蚂蚁 a 游历到 m 工序时选择了设备 k_g , 选择设备 k_j 为紧后工序加工设备的概率; $W_{k_j}^a$ 为紧后工序的可选加工设备集; $\tau_{k_g, k_j}(m)$ 为工序设备 (k_g, k_j) 间的信息素水平; $\eta_{k_g, k_j}(m) = C_k / t_{ejk}$, $X_{ejk} = 1$, 为工序设备 (k_g, k_j) 间的局部启发式信息。对于每一个设备在算法运行初设定一个较大的可承载负荷 C_k 。随着各工序加工设备的选定, C_k 被动态更新(减少)。而 $\eta_{k_g, k_j}(m)$ 使已承载负荷少且加工时间短的设备选中作为紧后工序加工设备的概率大, 进而同时实现式(2)和式(3)两个目标。 α, β 决定了 $\tau_{k_g, k_j}(m)$ 和 $\eta_{k_g, k_j}(m)$ 在转移概率中的重要程度。

2.5 粒子的解码

粒子向量的解码过程是计算粒子向量各分量所代表的工序加工开始、结束加工时间, 并进一步确定该粒子所代表的调度方案流通时间 EE 的过程, 作为计算粒子适应值、评价粒子优劣的依据。解码方法是依次遍历粒子向量的各分量, 若某待解码分量 $< e, g >$ 的 $E_{e(g-1)n}$ 已知, 且优先权值最小, 则由式(5)和式(6)计算 $S_{egk} = \max(E_{e(g-1)n}, E_{ijk})$, $R_{ijeg} = 1$, 同时计算, $E_{egk} = S_{egk} + t_{egk}$ 。重复上述过程, 直到所有分量的 S_{egk} 和 E_{egk} 均已确定为止。

在上述解码过程中, $E_{ijk}, R_{ijeg} = 1$ 应在计算 S_{egk} 时已知。由于算法中解码是按优先权值由小到大的顺序, 实现工序加工的先后顺序, 同时又保证了 E_{ijk} 在计算 S_{egk} ($R_{ijeg} = 1$) 时已经解码(即加工 $< e, g >$ 的设备可用时间已知)。由式(4)依次遍历粒子向量的各位分量, 找出最迟工序完工时间, 即得 $EE = \max(E_{egk})$, $e = 1, \dots, N, g = 1, \dots, J_e, k = 1, \dots, M$; 同时计算 $f = \frac{1}{EE}$ 。

可见, 上述解码方法实质是将 JSSP 中每个设备上工序次序的调度转换成优先权值大小关系的调整, 并由粒子群算法实现这种优先权值大小关系的调整。

2.6 粒子的变异

实验中发现, 标准粒子群优化算法(见文献[5])使粒子个体分量的优先权值向个体极值 $pbest$ 和全局极值 $gbest$ 的优先权值拉近。但算法过程中, 特别是后期各分量优先权值的大小关系却可能僵化, 即陷入局部最优。为防止算法陷入局

部最优, 保持粒子的多样性, 借鉴遗传算法中个体变异从而实现种群多样性的思想, 以一个较小的变异概率对各分量的优先权值变异。即用 $[0, 1]$ 范围的随机数代替原粒子个体分量的优先权值, 以打破优先权值僵化的大小关系。

3 应用实例

本文用两个实例(8×8)和(10×10)来说明所提出算法的有效性。两个实例的原始数据参见文献[1], 其中, 实例(8×8)是部分柔性的 FJSSP, 而实例(10×10)是完全柔性的 FJSSP。蚁群粒子群算法的参数设置: 种群规模均为 10, 蚁群算法中挥发率, 最大、最小信息素浓度分别是 0.6、0.5、0.08, $\alpha = 1, \beta = 10$; 粒子群算法中变异率 0.2; 进化代数 200。表 1 和表 2 分别列出了本文算法运行 10 次对该问题的求解结果与文献[1]结果的比较情况。可见, 本算法不仅可以得到更优的调度结果, 而且可以提供多种调度方案以满足实际的生产需求。

表 1 问题(8×8)算法结果与比较

比较项目	EE	W_T	W_M	调度方案数
本算法结果	15	75	12	24
	16	73	13	118
文献[1]结果	16	75	14	

表 2 问题(10×10)算法结果与比较

比较项目	EE	W_T	W_M	调度方案数
本算法结果	8	42	5	17
	7	42	6	89
文献[1]结果	7	45	6	

算法中参数设置对运行结果的影响很大。其中, 蚁群算法中最大、最小信息素浓度, 挥发率, 特别是 α, β 的相对取值(参见公式 7)对工件最优加工路径选择; 粒子群算法中变异率对工件最优加工序列选择的影响很大, 并进一步影响算法求得更多、更优的调度方法, 甚至使算法陷入局部最优。表 3 是在保持种群规模、进化代数等参数如上设置, 而 α, β 及粒子群变异率取不同值时对实例(10×10)的优化结果。当然, 增大种群规模和进化代数对算法寻优也有利, 但如其他基于种群的优化算法一样会增加算法的运行时间, 而且对运行结果的影响远不及 α, β 及粒子群变异率显著, 这里就不再赘述。

表 3 不同参数下的优化结果

参数设置	α	β	变异率	优化结果			
				EE	W_T	W_M	方案数
1	1	5	0.2	8	41	7	31
				8	42	6	17
2	1	20	0.2	8	41	7	58
				8	42	6	10
3	1	10	0.0	8	42	5	8
				7	42	6	45
4	1	10	0.4	8	42	5	14
				7	42	6	51
5	1	10	0.6	8	42	5	9
				7	42	6	38

4 结语

提出了一种解决多目标 FJSSP 的主、从蚁群粒子群算法。(下转第 941 页)

2 仿真算例

本文使用两个算例来验证 WPLS 的性能。

第一个算例使用如下动态系统产生观测数据进行回归运算: $x_1(k+1) = g(k) + u(k)$, 其中 $u(k) = \sin(2\pi k/25)$, $g(k) = \frac{x_5(k)}{x_6(k)}x_2(k)x_3(k)$, $x_2(k) = x_1(k-1)$, $x_3(k) = x_2(k-1)$, $x_4(k) = 1 + x_1(k)^2 + x_2(k)^2 + x_3(k)^2$, $x_5(k) = x_1(k) + 2.5$, $x_6(k)$ 、 $x_7(k)$ 为 $[-1, 1]$ 之间的随机变量。 $[x_1(k), \dots, x_7(k), g(k)]$ 构成 1 组观测数据。随机抽取 2500 个数据作为回归建模使用, 均匀抽取 1250 个数据作为验证模型使用。本文分别使用 PLS 和 WPLS 算法对数据建立回归模型:

使用 PLS 建立的回归模型:

$$\begin{aligned} g_1(k) = & 0.3233x_1(k) + 0.0687x_3(k) + \\ & 0.0005x_6(k) - 0.0013x_7(k) \end{aligned}$$

使用 WPLS 建立的回归模型:

$$\begin{aligned} g_2(k) = & 0.23456x_1(k) + 0.012932x_3(k) - \\ & 0.00017x_6(k) - 0.005013x_7(k) \end{aligned}$$

$g_1(k)$ 的预测误差均方差为 4.785×10^{-4} , $g_2(k)$ 的预测误差均方差为 2.381×10^{-4} 。

第二个算例使用一组公路附近空气污染研究的数据建立回归模型。该数据由挪威公路管理局 (Norwegian Public Roads Administration) 发布, 可以在 <http://www.kernel-machines.org> 获得。数据包含 7 维输入, 1 维输出: y 为空气颗粒浓度; x_1 为每小时汽车数量; x_2 为地面 2m 处的气温; x_3 为地面 2m 处的风速; x_4 为地面 2m 与 25m 处的气温差别; x_5 为风向角度; x_6 为 1 天中的日照时间(h); x_7 为距离 2001 年 10 月 1 日的时间(d)。首先将数据进行标准化处理, 使其均值为 0, 方差为 1。然后使用前 400 组数据进行建模, 后 100 组数据进行验证。为了进行比较, 本文分别使用 PLS, WPLS 和 SVM 进行回归建模。其中 SVM 使用的径向基核函数为: $K(x, y) = \exp(-0.5 \times (\text{norm}(x-y)/4.2)^2)$, 惩罚系数 $C = 15.4$, 不敏感系数 $\epsilon = 0.08$ 。三种方法预测误差的均方值如表 1 所示。WPLS 回归结果为:

$$\begin{aligned} y = & 0.4051x_1 - 0.1965x_3 + 0.0604x_4 - 0.0084x_5 - \\ & 0.0875x_6 + 0.0329x_7 + 3.2871 \end{aligned}$$

其中 x_i 为标准化处理以后的变量, $i = 1, 2, 3, 4, 5, 6, 7$ 。

从两个数值算例都可以看出, WPLS 可以有效地提高 PLS

的推广性。与 SVM 相比, WPLS 的预测效果相差不大, 但是 WPLS 的回归结果具有可解释性, 这是 SVM 所不具备的。

表 1 三种方法的预测误差

算法名称	PLS	WPLS	SVM
预测误差均方值	0.5413	0.5103	0.4493

3 结语

PLS 是一种多元统计数据分析方法, 它可以有效地对高维数据空间进行降维处理, 较好地克服变量多重相关性。由于 PLS 是基于 ERM 准则的学习方法, 因此泛化能力较差。为了在保留 PLS 优点的基础上改善其泛化能力, 本文提出了基于结构风险最小化的 WPLS。通过提出训练样本权值, 使用支持向量机理论计算 WPLS 中主元与输出之间的线性回归模型, 实现了 SRM 准则。WPLS 与 SVM 的预测效果相差不大, 但是 WPLS 的回归结果具有可解释性。

参考文献:

- VAPNIK VN. 统计学习理论的本质 [M]. 张学工, 译. 北京: 清华大学出版社, 2000.
- VAPNIK VN. Statistical learning theory [M]. New York, 1998.
- WOLD S, RUHE H, WOLD H, et al. The collinearity problem in linear regression. The partial least squares (PLS) approach to generalized inverse [J]. SIAM Journal of Scientific and Statistical Computations, 1984, 5: 735–743.
- 王华忠, 俞金寿. 基于混合 SVR-PLS 方法的丙烯腈收率软测量建模 [J]. 控制与决策, 2005, 5(20): 549–552.
- 成忠, 陈德钊. 模糊偏最小二乘及其在药物构效关系中的应用 [J]. 浙江大学学报(工学版), 2005, 39(10): 1613–1617.
- CHIANG J-H, HAO P-Y. Support Vector Learning Mechanism for Fuzzy Rule-Based Modeling: A New Approach [J]. IEEE Transactions on Fuzzy Systems, 2004, 12(1): 1–12.
- 王惠文. 偏最小二乘回归方法及其应用 [M]. 北京: 国防工业出版社, 1999.
- 李建民, 张钹, 林福宗. 支持向量机的训练算法 [J]. 清华大学学报(自然科学版), 2003, 43(1): 120–124.
- SMOLA AJ, SCHOLKOPF B. A Tutorial on Support Vector Regression [J]. Statistics and Computing, 2004, 14(3): 199–222.
- CHANG MW, LIN C J. Leave-One-Out Bounds for Support Vector Regression Model Selection [J]. Neural Computation, 2005, 17: 1188–1222.

(上接第 938 页)

主级中, 将多目标 FJSSP 中设备总负荷和关键设备负荷最小的优化通过蚁群转移概率设计为蚁群优化过程; 而从级中, 流通时间最小化通过粒子编码和解码方法设计为粒子群优化过程。仿真实验和比较表明, 该算法具有良好的求解性能和应用前景。

参考文献:

- KACEM I, HANNADI S, BORME P. Approach by localization and multi-objective evolutionary optimization for flexible job-shop scheduling problems [J]. IEEE Trans on Systems, Man, and Cybernetics, Part C, 2002, 32(1): 1–13.
- KACEM I, HAMMADI S, BORME 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(3–5): 245–276.
- DORIGO M, MANIEZZO V, COLORNI A. The Ant System: Optimization by a colony of cooperating agents [J]. IEEE Transactions on Systems, Man, and Cybernetics, Part B, 1996, 26(1): 29–41.
- KENNEDY J, EBERHART RC. Particle swarm optimization [A]. Proceedings of IEEE International Conference on Neural Networks, IV[C]. Piscataway, NJ: IEEE Service Center, 1995. 1942–1948.
- EBERHART RC, SHI Y. Particle swarm optimization: developments, applications and resources [A]. Proceedings of Congress on Evolutionary Computation 2001[C]. Piscataway, NJ: IEEE Press, 2001. 81–86.
- STUTZLE T, HOOS HH. The MAX-MIN ant system and local search for the traveling salesman problem [A]. Proceedings of ICEC'97 – 1997 IEEE 4th International Conference on Evolutionary Computation [C]. IEEE Press, Piscataway, NJ, 1997. 308–313.