

基于云模型的 PSO 算法求解差异工件单机批调度问题^①

刘娟 陈华平 (中国科学技术大学 管理学院 安徽 合肥 230026)

摘要: 为了提高粒子群(PSO)算法的性能,提出一种基于云模型理论的改进 PSO 算法,并应用于差异工件单机批调度问题的求解。首先根据粒子的适应值把种群划分为三个子群,提出一种随机的位置和速度更新方法,来有效平衡算法的局部和全局搜索;然后引入基于云模型理论的自适应参数策略,不同的子群采用不同的惯性权重生成方法,提高种群的多样性和算法的收敛速度。实验比较结果验证了该算法的全局搜索性能。

关键词: 云模型;粒子群优化;批调度;自适应参数调整;惯性权重

Particle Swarm Optimization Based on Cloud Model for Scheduling Single Batch-Processing Machine with Non-Identical Job Sizes

LIU Juan, CHEN Hua-Ping

(College of Management, University of Science and Technology of China, Hefei 230027, China)

Abstract: An adaptive Particle Swarm Optimization based on cloud model theory is proposed to improve its capability and applied to minimizing the makespan of a single batch-processing machine with non-identical job sizes. The particles are first divided into three groups based on the fitness of the particle to propose a new method for updating location and velocity. Then an adaptive strategy for varying parameters of PSO based on cloud model theory is introduced and different groups adopted different inertia weight generating methods, which does not only improve the convergence speed, but also maintain the diversity of the population. The global search performance of this adaptive algorithm is validated by the results of the comparative experiments.

Keywords: cloud model; particle swarm optimization; batch-processing; adaptive varying parameters; inertia weight parameter

1 引言

云模型是李德毅院士提出的一种定性定量转换模型^[1,2]。该模型在统计学和模糊数学的基础上,统一刻画了不确定性语言值和精确数值之间的随机性和模糊性,实现了定性语言值和定量数值之间的转换。目前基于云模型的各种云技术已经成功应用于智能控制、数据挖掘、模糊评测、进化算法等多个领域。

差异工件单机批调度(Single Batch-processing

Machine with Non-identical Job Sizes, SBMN)问题是一类 NP-hard 组合优化问题,兼具古典调度和批调度的性质,是生产调度领域中的一个新的研究方向。该问题在求解具体的调度目标时,考虑到工件尺寸与机器容量的共同约束。SBMN 问题在现实工业生产中应用广泛,例如电路测试、港口货物装卸、半导体芯片的预烧等等。SBMN问题被证明是 NP-Hard 问题^[3],与经典调度问题相比,其可行解空间增大,问题的复

① 基金项目:国家自然科学基金(70671096);国家杰出青年基金(B类)(70629002)

收稿时间:2009-05-19

杂性更高。近年来,以模拟退火(SA)^[4]、遗传算法(GA)^[5]等为代表的智能优化算法在 SBMN 问题领域得到了广泛的应用。

PSO(Particle Swarm Optimization)算法是一种基于群体智能理论的新兴进化计算技术^[6]。PSO 算法采用实数求解,并且需要调整的参数较少,易于实现,目前被成功应用于神经网络训练、函数优化和模型预测控制等领域^[7]。邵浩等^[8]将 PSO 和动态规划规则相结合,成功解决了具有动态到达时间的批调度问题。然而,PSO 算法也存在易于陷入局部最优,出现早熟收敛的问题^[9]。许多研究都集中于参数惯性权重的改进上,如 Shi^[10]等提出了随着迭代的进行线性地减小惯性权重的策略,但这使算法在求解复杂问题的后期易于发生早熟收敛;刘建华等^[11]提出一种能使其惯性权重随不同粒子与最优粒子距离动态变化的算法,但算法的鲁棒性不强。本文将云模型引入 PSO,利用云模型同时具有随机性和稳定倾向性的特性,通过保持种群多样性和加快收敛速度来改善 PSO 的性能。

首先根据粒子的适应度把种群划分为三个子群,提出一种随机的位置和速度更新方法,来有效控制算法的局部和全局搜索;此外,引入基于云模型理论的自适应参数策略,构造出一种的改进 PSO 算法,并应用于 SBMN 问题的求解。通过实验结果发现,该算法不仅提高了 PSO 的收敛速度和解的精度,并在一定程度上解决了 PSO 的早熟收敛问题。

2 差异工件单机批调度问题模型

在一个工件数为 n 的 SBMN 问题中,工件集合记为 $J = \{1, 2, \dots, n\}$,其中工件 i 的加工时间和尺寸分别为 p_i 和 s_i ,机器的容量为 B ,加工时的批次为 b_j ($j = 1, 2, \dots, k$),批 b_j 的处理时间记为 T_j 。加工约束为:①中的工件分批加工,每一批工件的总尺寸不大于 B ; ②批的加工不允许中断, T_j 等于 b_j 中最后完成工件的加工时间; ③制造跨度为所有批的加工时间总和。

根据上述约束条件,得到优化制造跨度的 SBMN 的问题数学模型如下:

$$\text{Minim } c_{\max} = \sum_{j=1}^k T_j \quad (1)$$

$$\text{s.t. } \sum_{j=1}^k x_{ij} = 1, i = 1, \dots, n \quad (2)$$

$$\sum_{i=1}^n s_i x_{ij} \leq B, j = 1, \dots, k \quad (3)$$

$$T_j \leq p_i x_{ij}, i = 1, \dots, n; j = 1, \dots, k \quad (4)$$

$$x_{ij} \in \{0, 1\}, i = 1, \dots, n; j = 1, \dots, k \quad (5)$$

$$\left\lceil \sum_{i=1}^n s_i / B \right\rceil \leq k \leq n, k \in Z^+ \quad (6)$$

式(1)表明模型的目标是最小化问题的制造度;式(2)限定每个工件 i 只能分配到一个批 j 中;式(3)和式(4)分别是机器容量和批加工时间约束;式(5)说明若工件 i 被安排在批 j 中,则决策变量 x_{ij} 等于 1,否则等于 0;式(6)给出了问题的最小批数,其中 k 为总批数, n 为最大批数。

3 云模型原理

定义 1. 设 U 是一个用数值表示的定量论域, C 是 U 上的定性概念,若定量值 $x \in U$ 是定性概念 C 的一次随机实现, x 对 C 的确定度 $\mu(x) \in [0, 1]$ 是有稳定倾向的随机数。 $\mu: U \rightarrow [0, 1], x \in U, x \rightarrow \mu(x)$ 则 x 在论域 U 上的分布称为云,记为 $C(x, \mu)$ 。每一个 x 称为一个云滴^[1]。

根据该定义可以看出,云是用语言值表示的某个定性概念与定量表示之间的不确定性转换模型。云模型所表达的概念的整体特性可以用 $C(Ex, En, He)$ 表征, Ex (Expected value)是云滴在论域空间分布的期望, En (Entropy)代表定性概念的可度量粒度, He (Hyper entropy)是熵的不确定性度量,由熵的随机性和模糊性共同决定。它将模糊性和随机性完全集成到一起,作为知识表示的基础。

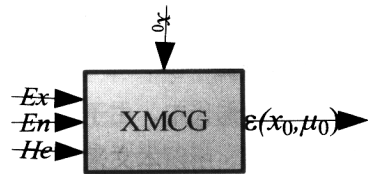


图 1 X 条件隶属云发生器

定义 2. X 条件隶属云发生器(X-Membership Clouds Generator, XMCG)是给定隶属云的三个数字特征(Ex, En, He)和论域 U 上的特定的值 x_0 , 产生云滴 $\varepsilon(x_0, \mu_0)$ ^[12]。

4 基于云模型的粒子群算法

4.1 基本粒子群算法

PSO 算法将问题空间的寻优过程看作鸟之间的协作捕食过程。群中的每个粒子记忆并追随当前的最优粒子在解空间中进行搜索。设在 D 维空间中有一个粒子数为 N 的粒子群, 记第 i 个粒子的位置向量和速度向量分别 $x_i=(x_{i1}, x_{i2}, \dots, x_{iD})$ 和 $v_i=(v_{i1}, v_{i2}, \dots, v_{iD})$, $i=1, 2, \dots, N$, 其中 x_{ij} 和 v_{ij} 分别表示粒子 i 在第 j 维的位置和速度。迭代寻优过程中, 每个粒子根据如下公式来更新自己的速度和位置^[6]:

$$v_i(t+1) = \omega v_i(t) + c_1 r_1 [x(pb_i(t)) - x_i(t)] + c_2 r_2 [x(gb_i(t)) - x_i(t)] \quad (7)$$

$$v_i(t+1) = \begin{cases} v_{\max} & v_i(t+1) > v_{\max} \\ v_{\min} & v_i(t+1) < v_{\min} \\ v_i(t+1) & \text{其他} \end{cases} \quad (8)$$

$$x_i(t+1) = x_i(t) + v_i(t+1)$$

其中, ω 为惯性因子; c_1 和 c_2 为加速因子, r_1 和 r_2 为 0 到 1 之间的随机数; $pb_i(t)$ 和 $gb_i(t)$ 分别表示在 t 时刻第 i 个粒子的局部极值以及群体发现的全局最优值。

4.2 粒子的编码及初始化

对于 n 个工件的调度问题, 使用 n 维向量表示一个粒子。第 i 个微粒的位置向量 $x_i=(x_{i1}, x_{i2}, \dots, x_{in})$, 其中 x_{ij} ($j \in \{1, \dots, n\}$) 对应工件 j 的优先值, 每个 x_i 代表一个可行的工件序列。速度向量为 $v_i=(v_{i1}, v_{i2}, \dots, v_{in})$, 每一维的速度 v_{ij} 用来更新 x_i 中的对应维度 x_{ij} 。

由于传统的随机粒子生成方法在实际问题中难以在有限的迭代次数内获得高质量的解, 本文采用 LPT (Largest Processing Time first) 方法生成初始种群。粒子 i 对应的工件序列中, 将 p_j 按递减的顺序排列得到一个工件序 π_i , $\pi_i(l)$ 表示 π_i 中的第 l 个工件。

4.3 粒子适应度的计算

在初始化及每次迭代中, 由于粒子 i 获得的工件序列并不是问题的解, 所以本文采用 BF(Best Fit)^[13] 规则对工件序列进行分批, 计算分批后的制造跨度作为 i 的适应度 $f(i)$ 。

例如, 在某工件数为 10 的 SBMN 问题中, 粒子 i : $x_i=(6, 4, 2, 10, 3, 8, 9, 1, 7, 5)$; $\pi_i=(9, 6, 4, 3, 10, 5, 2, 7, 6, 1)$; $s_i=(7, 9, 1, 3, 2, 4, 5, 1, 2, 4)$ 。若 $B=10$, 那么根据 BF 规则, 分批结果如表 1。可知批数 $k=4$, 粒子 i 的适应度 $f(i)=9+6+10+2=27$ 。

表 1 10 工件 SBMN 问题某粒子 i 的分批结果

b_i	批中工件	批尺寸	加工时间
b_1	6, 10	10	9
b_2	4, 2	10	6
b_3	3, 8, 1, 7	9	10
b_4	9, 5	9	2

4.4 改进的粒子位置和速度更新方法

基本 PSO 算法求解离散问题时, 不能保证算法收敛于一个局部极值, 结果是所有粒子仅收敛于当前种群所能找到的最优值^[14]。为了克服 PSO 的过早收敛的缺点, 本文采用一些改进措施来改善其搜索能力。

首先将微粒群 N 划分为三个不同的子群: $N=N_1 \cup N_2 \cup N_3$, 其中 $N_1=\{i | f(i)=f(gb), i \in N\}$; $N_2=\{i | f(i)=f(gw), i \in N\}$; $N_3=\{i | i \in N_1, i \in N_2, i \in N\}$ 。其中 gw 表示当前种群中最差的个体。然后, 本文在基本 PSO 的位置和速度的更新过程中做了如下修改:

$$v_i(t+1) = \omega v_i(t) + c_1 r_1 [x(pb_i(t)) - x_i(t)] + c_2 r_2 [x(gb_{ir}(t)) - x_i(t)] \quad (9)$$

其中, $gb_{ir}(t) \in \text{rand}(N_1(t))$, 表明是从当前时刻下 $N_1(t)$ 中随机选出的最优个体。每个粒子 i 根据公式 (9) 计算新的速度后, 再根据公式 (8) 更新当前位置。可以看出, 这种随机的迭代方式可以扩大粒子的飞行范围, 从而避免早熟现象的发生。

4.5 云自适应调整策略

传统的线性递减策略^[10]往往不能反映实际优化搜索过程。本文对不同的子群分别采用不同的惯性权重生成策略, 具体的 ω 的生成规则为:

① $i \in N_1$, 是群体中最优的粒子, 采用较小的惯性权重加快全局收敛的速度。 $\omega = 0.2$ 。

② $i \in N_2$, 是群体中普通的粒子, 按照 X 条件隶属云发生器非线性动态调整 i 的惯性权重。自适应的粒子惯性权重 ω 生成新算法如下:

$$Ex = f_{gb}$$

$$En = (f_{gb} - C^*) / 3$$

$$He = En / (\frac{1}{n_1} \sum_{i=1}^{n_1} f_i - C^*)$$

$$En' = \text{normrnd}(En, He)$$

$$\omega = 1.0 - 0.2 * e^{-\frac{(f_i - Ex)^2}{2(En')^2}}$$

其中, C^* 是根据 Uzsoy^[6] 提出的下界计算方法计算得到。根据正态云的“ $3En$ ”规则^[2], 可以忽略 $[Ex-3En, Ex+3En]$ 区间之外的定量值对定性概念的贡献, He

决定云滴的离散程度，太小的 He 会丧失“随机性”而太大的 He 会丧失“稳定倾向性”，为此本文做了如上的参数选择。

③ $i \in N_3$ ，是群体中最差的粒子，采用较大的惯性权重，进行全局寻优。 $\omega = 1.0$ 。

4.6 算法描述

综上所述，可以得到应用于 SBMN 问题的基于云模型的 PSO 算法(CPSO)如下：

Step1.令 $t=0$ 。初始化 N 和所有参数。对每个粒子 i ：

Step1.1.初始化 $x_i(0)$, $v_i(0)$;

Step1.2.根据位置向量确定的工件序列进行 BF 分批,计算适应度值 $f(i)$;

Step1.3.初始化个体极值,即 $x(pb_i(0))=x_i(0)$, $f(pb_i(0))=f(i)$;

Step1.4.根据粒子的适应度划分子群 $N=N_1(0) \cup N_2(0) \cup N_3(0)$; 不同的子群采用不同的惯性权重生成策略,如 4.5 节所述。

Step2. $t=t+1$ 。根据对应的惯性权重更新各子群的速度和位置。对每个子群中的粒子 i ：

Step 2.1.利用全局极值和个体极值,根据公式(9)、(8)更新粒子的速度 $v_i(t)$ 和位置 $x_i(t)$;

Step 2.2.根据更新后位置向量 $x_i(t)$,确定 i 对应的工件 i 序列并进行 BF 分批,计算其适应度 $f(i)$;

Step 2.3.如果 $f(i)<f(pb_i(t-1))$,则 $f(pb_{ti}(t))=f(i)$;

Step 2.4.根据粒子的适应度更新 $N_1(t)$ 、 $N_2(t)$ 、 $N_3(t)$ 。

Step 3. 判断是否满足终止条件。若满足,则输出最优解,算法结束;否则转入 Step 2。

5 仿真实验

本文采用 Melouk^[4]等提出的方法随机方法获得算例,算例的规模按工件数量 n 、加工时间 p_j 和工件尺寸 s_j 三个标准进行划分。按工件数分类可以得到 J_1 、 J_2 、 J_3 、 J_4 类问题,其工件数依次为 10、20、50 和 100;根据 p_j 可分为 p_1 、 p_2 类问题,区间分别为 $[1,10]$ 和 $[1,20]$;按 s_j 分,有 s_1 、 s_2 、 s_3 三类问题,区间依次是 $[1,10]$ 、 $[2,4]$ 和 $[4,8]$ 。每一类的实例记为 J_{ip_ks} ($i=1,2,3,4;j=1,2;k=1,2,3$)。例如 $J_{1p_1s_1}$ 表示 10 个

工件、加工时间和尺寸均服从 $[1,10]$ 的离散均匀分布的实例。在所有实例中,假设机器容量都为 10。

表 2 SBMN 问题 24 种算例的仿真结果比

Size	Best Cmax			Average Cmax		
	GA	PSO	CPSO	GA	PSO	CPSO
$J_{1p_1s_1}$	30	30	30	30	30	30
$J_{1p_1s_2}$	20	20	18	20	20	18.6
$J_{1p_1s_3}$	52	52	50	52	52	50.2
$J_{1p_2s_1}$	46	46	45	46	46	45.9
$J_{1p_2s_2}$	34	34	33	34	34	33.8
$J_{1p_2s_3}$	86	86	82	86	86	83.2
$J_{2p_1s_1}$	71	69	62	71	69	62.4
$J_{2p_1s_2}$	41	38	30	42.2	39.7	31.0
$J_{2p_1s_3}$	72	70	62	72.3	72	63.0
$J_{2p_2s_1}$	107	104	93	108.3	106.4	94.2
$J_{2p_2s_2}$	72	67	50	72.1	71.9	51.1
$J_{2p_2s_3}$	149	145	126	149	145.2	127.3
$J_{3p_1s_1}$	165	164	143	165	161.2	145.7
$J_{3p_1s_2}$	93	88	74	95.4	87.3	74.8
$J_{3p_1s_3}$	178	169	129	178	171.1	130.0
$J_{3p_2s_1}$	321	314	262	322.9	318.5	262.9
$J_{3p_2s_2}$	201	193	182	204.5	200	182.9
$J_{3p_2s_3}$	378	371	339	378.9	376.4	340.1
$J_{4p_1s_1}$	276	267	211	281	275.5	212.8
$J_{4p_1s_2}$	193	186	143	193.7	190.6	143.6
$J_{4p_1s_3}$	387	369	317	389.7	381.7	317.8
$J_{4p_2s_1}$	607	572	524	611.8	588.6	524.5
$J_{4p_2s_2}$	344	320	306	345.4	330.2	307.2
$J_{4p_2s_3}$	804	769	687	832.3	786.4	688.1

表 3 SBMN 问题不同规模问题的平均运行时间比

Size	Average time(s)		
	GA	PSO	CPSO
J_1	2.66	1.21	1.21
J_2	5.18	3.49	3.36
J_3	20.41	14.26	13.4
J_4	76.45	58.79	55.49

本文对各个不同规模的算例进行实验,并与 GA 算法以及 PSO 算法进行了对比实验,其中 GA 算法为 Damodaran,P[5]提出的带有精英保留策略的改进算法,GA 参数为:种群规模 $N=50$,交叉概率 $P_c=0.9$,变异概率 $P_m=0.1$; PSO 和 CPSO 算法采用相同的粒子编码方式,具有如下相同的参数:种群规模 $N=50$ 、加速因子 $c_1=c_2=2$, $v_{max}=5.0$, $v_{min}=-5.0$,但 PSO 算法采用 shi Y[10]提出的线性惯性因子, $\omega_0=1.0$,

$\omega_0=0.2$, 且利用标准公式(7)和(8)进行迭代; CPSO即为本文提出的基于云模型的改进 PSO 算法。随机产生了一组实例作为仿真实例, 三个算法分别独立运行 100 次, 运行结果如表 2 和表 3 所示。从表 2 中 100 次运行中的最好结果和平均结果行对比可见, CPSO 算法的两种指标均优于 GA 和 PSO, 随着问题规模的增大, CPSO 算法的解与 GA 和 PSO 的差距更为明显, 显示了很好的稳定性。

另外, 在运行时间上, 从表 3 可以看出, CPSO 算法在四种不同规模的算例上的求解时间上均小于 GA 和 PSO, 相对传统的算法有着更快的收敛速度。面对大规模的问题, 尤其是 J₃、J₄ 类问题, CPSO 算法具有更加明显的优势, 不仅在制造跨度的数值上有明显的改进, 而且运行速度提高更快。说明在 SBMN 问题复杂度提高时, CPSO 算法具有很好的鲁棒性。

6 总结

本文基于云模型理论, 提出一种改进的粒子群算法 CPSO, 并应用于 SBMN 问题的求解。仿真实验结果表明, CPSO 算法一定程度上改善了 PSO 容易出现早熟现象的问题, 并证明了其在求解 SBMN 问题时是有效的, 不仅在近似解的质量和运行时间上有显著的改进, 而且在求解大规模 SBMN 问题时具备很好的鲁棒性。今后将致力于对多机问题的深入研究, 以及将云模型和优化技术结合应用到更加复杂的组合优化问题求解中。

参考文献

- 李德毅, 孟海军, 史雪梅. 隶属云和隶属云发生器. 计算机研究与发展, 1995, 32(6): 15-20.
- 刘常昱, 李德毅, 潘莉莉. 基于云模型的不确定性知识表示. 计算机工程与应用, 2004, 40(2): 32-35.
- Uzsoy R. Scheduling a single batch processing machine with non identical job sizes. International Journal of Production Research, 1994, 32: 1615-1635.
- Melouk S, Damodaran P, Chang PY. Minimizing makespan for single machine batch processing with non-identical job sizes using simulated annealing. International Journal of Production Economics, 2004, 87: 141-147.
- Damodaran P, Kumar MP, Srihari K. Minimizing makespan on a batch-processing machine with non-identical job sizes using genetic algorithms. International Journal of Production Economics, 2006, 103(2): 882-891.
- Kennedy J, Eberhart RC. Particle swarm optimization. Proceedings IEEE International Conference on Neural Networks V. Piscataway, NJ: IEEE Service Center, 1995. 1942-1948.
- Eberhart RC, Kennedy J. A new optimizer using particle swarm theory. Institute of Electrical and Electronics Engineers, 1995, 10: 39-43.
- 邵浩, 陈华平, 等. 优化差异工件单机批调度问题的混合微粒群算法. 系统工程, 2008, 26(12): 98-102.
- Van den Bergh F, Engelbrecht AP. A study of particle swarm optimization: particle trajectories. Information Sciences, 2006, 176(8): 937-971.
- Shi Y, Eberhart RC. A Modified Particle Swarm Optimizer. Proc. of the IEEE Conference on Evolutionary Computation. Piscataway, NJ: IEEE Press, 1998. 69-73.
- 刘建华, 樊晓平, 瞿志华. 一种惯性权重动态调整的新型粒子群法. 计算机工程与应用, 2007, 43(7): 68-70.
- Zhu YF, Dai CH, Chen WR, et al. Adaptive probabilities of crossover and mutation in genetic algorithm based on cloud generators. Journal of Computational Information Systems, 2005, 1(4): 671-678.
- Dupont L, Jolai GF. Minimizing makespan on a single batch processing machine with non-identical job sizes. European Journal of Automation System, 1998, 32: 431-440.
- Van den Bergh F. An analysis of particle swarm optimizers. [Ph.D. Dissertation]. South Africa: Department of Computer Science, University of Pretoria, 2002.