

动态评价免疫微粒群算法在 Job-shop 调度中的应用

常桂娟^{1,2}, 张纪会¹

CHANG Gui-juan^{1,2}, ZHANG Ji-hui¹

1. 青岛大学 复杂性科学研究所, 山东 青岛 266071

2. 莱阳农学院 理学院, 山东 青岛 266109

1. Complexity Science Institute of Qingdao University, Qingdao, Shandong 266071, China

2. The College of Science of LaiYang Agricultural University, Qingdao, Shandong 266109, China

E-mail: lucycgj@163.com

CHANG Gui-juan, ZHANG Ji-hui. Dynamic evaluated immune Particle Swarm Optimization for Job-shop scheduling. Computer Engineering and Applications, 2007, 43(24): 189-191.

Abstract: Traditional Particle Swarm Optimization(PSO) has some limitation to solve the combinatorial optimization problems. An Improved Particle Swarm Optimization(IPSO) by improving the iterative formula is proposed after analyzing the optimization mechanism of the PSO. In IPSO, to update the particles, the crossover idea of genetic algorithm is utilized by particles to exchange information. To keep excellent particle in the course of evolution, the optimization operator of acceleration is proposed and utilized. Particles are evaluated dynamically by immune algorithm in the course of evolution in order to avoid getting into the local search. The experimental results show that JSP Can be solved by IPSO effectively. The rationality of IPSO is validated.

Key words: Particle Swarm Optimization; immunity; Job-shop scheduling

摘要:传统粒子群优化算法在解决组合优化问题上具有一定的局限性,通过分析其优化机理,对迭代公式加以改进,提出了改进微粒群算法。算法中,利用遗传算法的交叉思想来完成粒子间的信息交换,以期达到粒子更新。粒子进化过程中,为保留群体中的优秀粒子,使用了加速度这一优化算子。为避免粒子陷入局部搜索,迭代过程中使用免疫算法来动态评价微粒群体。通过大量实验仿真,算法可以有效求解作业车间调度问题,验证了算法的合理性。

关键词:微粒群优化;免疫;作业车间调度

文章编号:1002-8331(2007)24-0189-03 **文献标识码:**A **中图分类号:**TP18

微粒群优化算法(Particle Swarm Optimization Algorithm, PSO)是一种基于群体智能理论的优化算法,由 Kennedy 和 Eberhart 博士在 1995 年^[1]提出的,该算法通过模拟鸟类群体调整自身飞行速度和飞行方向,将所有个体移动到适应度好的环境中,从而抽象出一种可以求解具有复杂解空间性质问题的优化算法。同时在进化过程中该算法保留位置与速度上的信息,由于其概念和参数调整简单而且容易编程实现,它既保持传统进化算法深刻的群体智慧背景,同时又有自己许多良好的优化性能。该算法收敛速度快,尤其是在进化初期,运算简单,易于实现,没有遗传算法的编解码和交叉、变异等复杂运算。因此,PSO 算法一经提出,立刻引起进化计算领域学者们的广泛关注。但是,该算法是根据全体粒子和自身的搜索经验向着最优解的方向“飞行”,在进化后期收敛速度明显变慢,并且当算法收敛到一定精度时,就无法继续优化,因此算法最终所能达到的精度较差。

自 PSO 算法提出,在这短短的几年时间里出现了大量的

研究成果,发展十分迅速,由于其应用领域广泛而成为学术界研究的一个新的热点。目前 PSO 算法已经应用于神经网络、多目标优化、模糊控制系统等领域,但将 PSO 算法应用于车间调度问题领域的还不成熟。目前已有学者利用 PSO 来解决调度问题^[2,3],这需要对 PSO 算法本身或编码进行一系列的改进。夏蔚军^[4]等提出了将微粒群算法和模拟退火算法相结合的混合启发式算法,通过实验仿真证明了该算法的有效性。

本文提出了一种改进的微粒群优化算法,该算法在进化过程中利用遗传算法的交叉思想。使得加以改进的微粒群优化算法成功应用于离散问题 JSP 中。鉴于微粒群算法在收敛到一定精度时无法继续优化,引进了免疫优化算法,结合免疫系统的免疫信息处理机制给出了基于动态评价免疫微粒群算法。这种算法结合了微粒群优化算法具有的全局寻优能力,实现简单和免疫算法的免疫信息处理机制,从而有效避免了微粒群优化算法易陷入局部最优的缺点,提高了进化后期算法的收敛速度和精度。仿真结果也证明了该混合算法的可行性和有效性。

基金项目:国家自然科学基金(the National Natural Science Foundation of China under Grant No.70671057);教育部博士点基金(No.20051065002);青岛市自然科学基金(the Natural Science Foundation of Qingdao City of China under Grant No.03-2-jz-19)。

作者简介:常桂娟(1976-),女,讲师,博士研究生,主要从事生产调度、遗传算法、微粒群算法等研究;张纪会(1969-),男,博士生导师,博士后,教授,主要从事复杂适应系统理论、物流与供应链管理、智能调度等方面的研究。

1 微粒群^[4]算法描述

在 n 维空间中有 N 个微粒, 每个微粒的坐标为 $X_i=(X_{i1}, X_{i2}, \dots, X_{in})$, 并具有与优化目标函数相关的适应度 Fit_i , 同时每个微粒具有各自的速度 $V_i=(V_{i1}, V_{i2}, \dots, V_{in})$ 。对于微粒所经历过的历史最好位置记为 $P_i=(P_{i1}, P_{i2}, \dots, P_{in})$, 也称为 p_{best} 。群体所有微粒经历过的最好位置表示为 $P_g=(P_{g1}, P_{g2}, \dots, P_{gn})$, 也称为 g_{best} 。微粒群算法描述如下:

$$V_{id}(t+1)=WV_{id}(t)+c_1r_1(P_{id}(t)-X_{id}(t))+c_2r_2(P_{gd}(t)-X_{id}(t)) \quad (1)$$

$$X_{id}(t+1)=X_{id}(t)+V_{id}(t+1) \quad (2)$$

$$\begin{cases} V_{id}=V_{max}, & \text{if } V_{id}>V_{max} \\ V_{id}=-V_{max}, & \text{if } V_{id}<-V_{max} \end{cases}$$

利用上述两式计算第 $t+1$ 代第 i 维的速度和位置。式中下标 d 表示微粒的维度, i 指第 i 个微粒, W 是惯性权值, c_1, c_2 都是正的常数, 称为加速系数 (acceleration coefficient), r_1, r_2 是两个在 $[0, 1]$ 范围内变化的随机数。 V_{max} 是常数, 限制了速度的最大值。当 V_{max} 较大时, 微粒的飞行速度大, 有利于全局搜索, 但有可能飞过最优解; 若较小, 微粒可以在较小的区域内进行局部精细搜索, 但易陷入局部最优。式(1)中, 等式右侧第一部分为微粒上一代的速度, 它使微粒在搜索空间中有扩张的趋势, 从而使算法具有全局搜索的能力; 第二部分为“认知”部分, 是微粒吸取自身经验知识的过程; 第三部分为“社会”部分, 是微粒学习其它粒子经验的过程, 表现了微粒间信息的共享与社会协作。

2 改进微粒群算法(IPSO)

该算法在 PSO 算法的基础上, 对迭代算法加以改进, 引入了加速度的概念。在算法进行到第 $t+1$ 代时, 通过每个微粒的历史最好位置 $P_i(t)$ 与全局微粒的最好位置 $P_g(t)$ 交叉, 得到了第 $t+1$ 代的一个加速度。具体算法如下:

$$V_i(t+1)'=P_i(t)\Theta P_g(t) \quad (3)$$

$$V_i(t+1)=V_i(t)\Theta V_i(t+1)' \quad (4)$$

$$X_i(t+1)=X_i(t)\Theta V_i(t+1) \quad (5)$$

其中 Θ 的意思如下所述: 假设给出两个微粒 $x=\{6 \ 7 \ 3 \ 9 \ 4 \ 1 \ 5 \ 8 \ 2\}; y=\{3 \ 5 \ 2 \ 7 \ 4 \ 9 \ 1 \ 8 \ 6\}$ 。要求 $x\Theta y$, 则从微粒 x 中随机产生两个位置, 不妨设为 3 和 7。然后将微粒 x 中位置 3 和 7 之间的元素放在子代的相应位置, x 中余下位置元素按照在 y 中的先后顺序放在子代的余下位置, 就得到了 $x\Theta y=\{2 \ 7 \ 3 \ 9 \ 4 \ 1 \ 5 \ 8 \ 6\}$ 。式(3)保留了个体最优解和群体最优解的优良信息, 通过式(4)计算得到第代的速度, 该速度值既可以通过有效继承上代微粒速度信息, 防止群体陷入局部最优, 又可以继承群体中的优良信息, 有助于加快微粒搜索进程。这种进化算法能够极大程度地保留微粒群中的有益信息, 从而可以减少进化代数, 仿真实验也证明了这一点。

3 Job-shop 问题的描述

JSP 问题可简单描述为有 n 个工件在 m 台机器上加工, 每个工件包含 m 个工序。加工过程中要满足以下几条:

- (1) 每个时刻每台机器只能加工一个工件, 且每个工件只能被一台机器加工。
- (2) 工序一旦开始, 中途不能被打断。
- (3) 每个工件在同一台机器上最多加工一次。
- (4) 每个工件必须按照工艺路线加工。

所谓调度, 就是指每个工件设定在每台机器上的加工时间, 以使得某些性能指标达到最优。本文性能指标定为总工期 (makespan) 最短。

4 编码方法

成功应用微粒群优化算法的关键就是: 如何将一个机器调度编码为搜索空间的一个解? 设计一种合理的编码方法可有效降低由于约束条件的限制而给问题带来的难度, 本文采用了一种基于工序的一种新的染色体编码方法。染色体的长度为 $n \times m$, 染色体中每个基因的取值范围为 1 到 $n \times m$, 并且各基因值不能重复。在这里每个基因值代表该工序优先调度的可能性, 基因值越小越优先调度此工序。

例如, 对于三个工件三个机器的 JSP 调度问题, 若一条染色体为 $\{3, 5, 1, 7, 4, 9, 8, 6, 2\}$, 则三个工件具有的优先调度权集合分别为 $\{3, 5, 1\}, \{7, 4, 9\}, \{8, 6, 2\}$, 然后将每个集合的基因值从小到大重新排列得到 $\{1, 3, 5\}, \{4, 7, 9\}, \{2, 6, 8\}$, 从左到右每个集合依次对应着第一个工件的 3 道工序, 第二个工件的 3 道工序, 第三个工件的 3 道工序的优先调度权值。由此我们可以得到加工队列为 $O_{11}, O_{31}, O_{12}, O_{13}, O_{32}, O_{22}, O_{33}, O_{23}$ 。这里 O_j 表示第 i 个工件的第 j 道工序。将此加工队列按照尽早开工的原则进行调度即可得到目标函数 makespan。

5 免疫算法

免疫算法充分利用每代最优个体的信息, 立足于已有的进化算法, 吸取其中有益的思想, 在进化的过程中把随机搜索和确定性的变化结合起来, 减少了随机因素对算法本身的影响, 能较好地克服不成熟收敛。通过一个抗原与多个抗体比较, 找出适应度最佳的抗体, 并利用遗传算法促进抗体群体能够得到进化。

如果以抗体的亲和力评价为标准, 当群体中的某个抗体占据了相当的规模, 而又不是最优解时就极易导致过早收敛。为此, 当有些抗体的规模达到一定程度后, 就要对其进行限制, 以防过早收敛。本算法采用抗体浓度来抑制这种抗体, 并且以信息熵作为度量亲和力的指标, 以抗体的期望繁殖率作为评价抗体的标准。

抗体的亲和力: 用于表明两抗体之间的相似度。如果有 N 个抗体, 每个抗体由 n 位基因组成, 每位可供选择的字母表中共有 s 个字母: k_1, k_2, \dots, k_s , 则这 N 个抗体的平均信息熵为 $H(N)=\frac{1}{n} \sum_{j=1}^n H_j(N)$ 。式中: $H_j(N)=\sum_{i=1}^s (-P_{ij} \log_2 P_{ij})$, $H_j(N)$ 为 N 个抗体中第 j 位基因的信息熵, P_{ij} : N 个抗体中第 j 位是字母 k_i 的概率。如果在位置 j 上所有抗体的等位基因都相同, 那么 $H_j(N)=0$ 。抗体与抗原之间的亲和力为 $ax_v=\frac{1}{1+opt_v}$, $v \in N$ 其中 opt_v 为 makespan。抗体与抗体之间的亲和力为, 式中 v, w 为抗体群中的两个抗体。

抗体的促进和抑制: 对解群体中的个体进行评价时, 采用抗体的浓度和期望繁殖率作为标准。根据群体中抗体的亲和力, 由下面两式可以计算得到抗体的浓度和期望繁殖率。抗体的浓度 $R_v=\frac{1}{N} \sum_{w=1}^N K_{vw}, v, w \in N$ 。 $K_{vw}=\begin{cases} 1 & ay_{vw} \geq \delta \\ 0 & \text{其它} \end{cases}$, 其中 δ 为一个预先给定的阈值。抗体的期望繁殖率 $E_v=\frac{ax_v}{R_v}, v \in N$ 。因此, 与抗

表1 DEI-PSO 算法求解 FT06 问题仿真结果

算法名称	1	2	3	4	5	6	7	8	9	10	最好解	平均解	最优解出现的次数	最差解
DEI-PSO 算法	55	55	55	59	59	58	55	58	59	56	55	56.9	5	59

原亲和力的抗体和低浓度的抗体生存机率较大,将会得到促进,而高浓度的抗体会受到抑制,体现了控制机制的多样性。

6 动态评价免疫微粒群算法流程(记为 DEI-PSO)

步骤1 识别抗原,设定参数

通过对问题及解的特性进行分析和了解,进行微粒编码。本文采用基于优先权值的编码方法。设定粒子群体规模 N ,最大迭代次数 \maxgen ,免疫算法中阈值 δ 。

步骤2 产生粒子群

因为初始粒子的优劣对优化效果具有很大的影响,为了提高种群的搜索效率,避免盲目搜索,初始种群中应该包括部分质量较高的解。为保证初始粒子的有效性,引入种子粒子群的概念,产生 $N+M$ 个粒子。

步骤3 评价粒子

计算每个粒子(抗体)和抗原的亲合力,计算出每个粒子的浓度,进而求出粒子的期望繁殖率,将粒子的期望繁殖率按降序排列取出前 N 个粒子作为初始粒子群。计算每个粒子的目标函数值,为不遗失每代最好粒子,这里采用保优策略,即用上代最好粒子替换本代最差粒子转入步骤4。

步骤4 利用改进微粒群(IPSO)算法优化微粒群

对 N 个粒子进行 IPSO 优化算法得到新一代的微粒群的位置及速度。

步骤5 动态评价调整微粒群

为保证新种群的多样性,去掉其中的相同粒子,并由随机产生的新微粒替换它,为丰富种群,再随机产生 M 个微粒加入到新种群当中,转入步骤6。

步骤6 判断满足的终止条件

满足最大迭代次数则输出最优目标函数值并终止;否则,返回步骤3。

7 仿真结果

本文采用随机产生初始微粒群的方法,优化目标为最短加工完成时间,采用最大进化代数作为终止准则。下面选择了不同维数的标准算例来说明新算法的有效性。本文用 MATLAB 语言编程,对于上述设计的算子,本文对 LA 类问题中的若干问题进行了计算机仿真。微粒群规模设为 30,进化代数为 300 代。先以 FT06 问题为例来说明其收敛性:仿真 10 次结果如下:

对于较大规模的算例,本文算法同样表现出了良好的收敛性能。保持参数不变,分别用本文算法及基于动态评价免疫的遗传算法(DEI-GA)对 LA 类问题中几个不同规模的算例进行了计算机仿真。仿真结果显示本文算法明显优于 DEI-GA 算法。都得到了其最优解,而 DEI-GA 算法只对其中几个算例得到最优解。对其中的 LA05、LA10、LA14 问题,本文算法表现出了强收敛性,基本每次仿真均可收敛到最优解。对其它算例,若扩大种群规模和进化代数,仿真结果还会得到进一步改善。下表记录了对这些问题仿真 10 次的实验数据。

图 1 给出了对 LA05、LA10、LA14 问题的仿真图,横坐标表示迭代次数,纵坐标表示最优解。由图可见,DEI-PSO 算法

表2 DEI-PSO 算法和 DEI-GA 算法求解 LA 类若干问题仿真结果

编号	算例(规模)	最优解	最好解		平均值
			DEI-PSO/DEI-GA	DEI-PSO/DEI-GA	DEI-PSO/DEI-GA
1	LA01(10×5)	666	666/678	680/707.9	680/707.9
2	LA02(10×5)	655	655/689	689/723	689/723
3	LA03(10×5)	597	597/635	631/655	631/655
4	LA04(10×5)	590	590/638	620/641	620/641
5	LA05(10×5)	593	593/593	593/598	593/598
6	LA06(15×5)	926	926/930	943.4/956.5	943.4/956.5
7	LA07(15×5)	890	890/946	930.3/996.2	930.3/996.2
8	LA08(15×5)	863	863/923	883.5/957.4	883.5/957.4
9	LA09(15×5)	951	951/951	967.6/991.5	967.6/991.5
10	LA10(15×5)	958	958/958	958.5/967.3	958.5/967.3
11	LA11(20×5)	1222	1 222/1 277	1 248.8/1 312.8	1 248.8/1 312.8
12	LA12(20×5)	1039	1 039/1 079	1 059.8/1 120.7	1 059.8/1 120.7
13	LA13(20×5)	1150	1 150/1 198	1 171.5/1 237.7	1 171.5/1 237.7
14	LA14(20×5)	1292	1 292/1 292	1 292/1 299	1 292/1 299

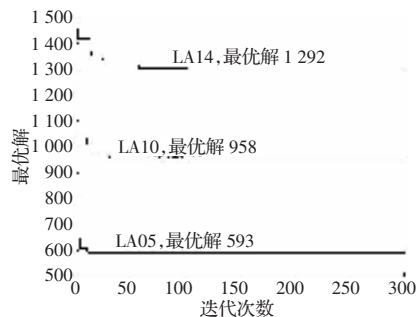


图1 DEI-PSO 算法求解 LA05、LA10、LA14 问题的仿真图

对这类问题表现了良好的收敛性能及收敛速度。三次实验,均仿真一次,每次都以很快的速度收敛到了最优解。

8 结束语

本文提出了一种新的基于动态评价免疫算法的改进微粒群算法,并将其应用于 Job-shop 调度问题。通过改变传统微粒群算法的迭代公式,引进加速度这一算子,极大程度的保留了微粒群体中的优良信息,加快了群体的收敛速度。通过对不同维数的标准算例进行计算机仿真,实验结果令人满意,也表明本文提出算法的有效性。将这种改进微粒群算法应用于其它调度问题是作者下一步的研究方向。(收稿日期:2007年4月)

参考文献:

- [1] Kennedy J, Eberhart R. Particle swarm optimization[C]//IEEE International Conference on Neural Networks, Perth, Australia, 1995: 1942-1948.
- [2] 彭传勇,高亮,邵新宇,等.求解作业车间调度问题的广义粒子群优化算法[J].计算机集成制造系统,2006,12(6):911-917.
- [3] 夏蔚军,吴智铭,张伟,等.微粒群优化在 Job-shop 调度中的应用[J].上海交通大学学报,2005,39(3):381-385.
- [4] 张燕,汪福,吴启迪.随机微粒群优化算法[J].计算机工程,2006,32(16):9-10.