

文章编号: 1001-0920(2011)04-0501-06

基于遗传粒子群混合算法的供应链调度优化

刘小华, 林 杰

(同济大学 经济与管理学院, 上海 200092)

摘要: 针对大规模定制模式下的供应链调度优化问题, 首先将供应链调度优化问题进行数学规划建模, 并通过一些定义将模型映射为有向图, 以便于智能搜索算法的应用; 然后, 结合遗传算法全局搜索能力强与粒子群算法收敛速度快的特点, 进行优势互补, 构造了一种混合算法; 最后, 运用混合算法对供应链优化调度问题模型进行求解. 通过与其他算法进行比较, 所得结果表明混合算法有着更好的整体性能.

关键词: 调度优化; 混合算法; 遗传算法; 粒子群算法

中图分类号: F273

文献标识码: A

Scheduling optimization in supply chain based on GA-PSO hybrid algorithm

LIU Xiao-hua, LIN Jie

(College of Economics and Management, Tongji University, Shanghai 200092, China. Correspondent: LIU Xiao-hua, E-mail: focus2077@sina.com)

Abstract: For scheduling optimization problem in mass customization supply chain, a mathematical programming model is proposed, which is corresponded with a digraph via several definitions in order to use intelligent algorithm to solve it. A hybrid algorithm is proposed to optimize the objective function, which takes both advantages of genetic algorithm and particle swarm algorithm, so this hybrid algorithm integrated global searching ability with high convergence speed. Compared with the results of other algorithms, the simulation results show that the proposed algorithm is an efficient method for solving the scheduling optimization problem.

Key words: scheduling optimization; hybrid algorithm; genetic algorithm; particle swarm algorithm

1 引 言

大规模定制(MC)是一种集企业、客户、供应商和环境于一体, 在系统化思想指导下, 用整体优化的思想, 充分利用企业已有的各种资源, 根据客户的个性化需求, 以大批量生产的低成本、高质量和高效率提供定制产品和服务的生产方式^[1]. 在MC模式下, 供应链企业之间的协作对于整个供应链系统的有效运行非常关键, 其中供应链调度是协作的主要内容. 如何做好供应链调度的优化已成为当前的一个紧迫问题. 大规模定制模式下, 部分生产任务是由用户订单驱动的, 核心制造商只有在接到订单后才能明确需求, 对定制件供应商进行选择 and 调度, 并组织生产. 需求的随机性以及各协作生产单元的随机生产能力, 决

定了其供应链调度是一个典型的随机、动态过程^[2].

供应链生产调度是NP-hard问题, 对多个订单进行综合调度时, 各订单子任务之间的工艺约束关系复杂, 相对于一些独立任务的生产调度问题, 其难度增加了许多. 在MC模式下, 解决该问题不仅要求得到好的结果, 而且需要在较短时间内解决, 只有成为准实时调度, 才能较有效地应对复杂多变的随机动态因素. 事实证明, 随着调度问题规模的扩大, 采用最优化等传统方法解决此类问题的难度急剧增大. 近年来, 智能搜索算法因其优良的表现, 已成为当前的研究热点^[3]. 应用于调度问题的智能搜索方法包括模拟退火、禁忌搜索、遗传算法、蚁群算法和粒子群算法等. 大量的研究表明, 这些算法对于调度问题的解决有着

收稿日期: 2010-01-12; 修回日期: 2010-03-18.

基金项目: 国家863计划项目(2007AA04Z151); 国家自然科学基金重点项目(70531020); 新世纪优秀人才支持计划项目(NCET-06-0377).

作者简介: 刘小华(1978-), 男, 博士生, 从事供应链管理和系统建模仿真等研究; 林杰(1967-), 男, 教授, 博士生导师, 从事决策支持系统和系统建模仿真等研究.

不错的效果,但是往往存在各自的局限性.如粒子群算法使用简单、收敛速度快,但容易早熟,陷入局部最优^[4];遗传算法全局搜索能力强,但搜索速度慢^[5].可见,两者有明显的互补性.

2 供应链调度数学规划模型

本文以某核心制造商协调多个供应商共同完成若干定制任务的情况为例进行研究.核心制造商需要将时间域值 T_0 内接到的订单进行分类、合并,并对企业内部及合作供应商进行调度.调度过程包括供应商选择、各企业内任务排序及各子任务开工时间确定.设核心制造商需要对 M 个订单进行调度,每个订单都可以分解为 N 个子任务,且每个订单的最迟交货期为 $l_i (i = 1, 2, \dots, M)$. 定义子任务约束矩阵 K^i 为存储订单 i 各子任务之间时间约束关系的 N_i 阶方阵,即

$$K^i = (K_{p,q}^i) = \begin{cases} 1, & \text{若订单 } i \text{ 中 } q \text{ 为 } p \text{ 的前置任务;} \\ 0, & \text{其他.} \end{cases}$$

定义方案描述矩阵为

$$C = (c_{j,m}^i) =$$

$$(\langle \text{Executor, sTime, oTime, Cost, cStock} \rangle_{j,m}^i).$$

其中: $c_{j,m}^i$ 表示完成第 i 个订单第 j 个子任务的第 m 种方案; Executor 表示 $c_{j,m}^i$ 的执行人; sTime 表示 $c_{j,m}^i$ 的开工时间; oTime 表示 $c_{j,m}^i$ 的完工时间; Cost 表示 $c_{j,m}^i$ 的加工成本; cStock 表示 $c_{j,m}^i$ 由于提前完成所需支付的单位时间在制品库存成本. 于是,核心制造商的优化调度模型为

$$\begin{aligned} \min E = & \omega_1 \left\{ \sum_{i=1}^M \sum_{j=1}^N \text{Cost}_{j,m_{ij}}^i + \sum_{i=1}^M \sum_{p=1}^N \sum_{q=1}^N k_{p,q} \times \right. \\ & \left. \text{cStock}_{p,m_{ip}}^i (\text{sTime}_{q,m_{iq}}^i - \text{oTime}_{p,m_{ip}}^i) \right\} + \\ & \omega_2 \max_{i=1}^M (\text{oTime}_{N,m_{iN}}^i) + \\ & P \sum_{i=1}^M \max(\text{oTime}_{N,m_{iN}}^i - l_i, 0). \end{aligned} \quad (1)$$

s.t.

$$\text{sTime}_{j',m_{ij}}^i \geq \text{oTime}_{j,m_{ij}}^i, \text{ 若 } k_{jj'} = 1; \quad (2)$$

$$\text{sTime}_{j,m}^i \geq \text{oTime}_{j',m}^i \text{ 或 } \text{sTime}_{j,m}^i \geq \text{oTime}_{j',m}^i. \quad (3)$$

其中: ω_1 和 ω_2 为时间和成本的权重因子; P 为交货期延迟惩罚因子; m_{ij} 表示订单 i 的第 j 个子任务最终选定的方案编号; $\text{oTime}_{N,m}^i$ 表示订单 i 最后一个子任务的完工时间. 调度模型的第 1 部分为成本评价因子; 第 2 部分为时间评价因子; 第 3 部分为订单延期惩罚项. 式 (2) 为各子任务之间的时间约束; 式 (3) 为企业资源约束.

3 遗传粒子群混合算法

粒子群优化 (PSO) 采用群体进化和适应度函数

评价优化结果,主要模拟鸟群飞行觅食行为,通过鸟群的集体协作达到寻优目的^[6].将鸟群中的每只鸟看作 D 维搜索空间中一个无体积的粒子,代表问题的一个可行解,具有位置和速度两个属性,以粒子位置坐标对应的适应度函数值确定粒子的“优劣”程度.每个粒子利用自身的历史最优位置和整个粒子群的全局最优解提供的信息,在解空间内不断飞行,寻找满足条件的最优解^[7].

遗传算法是由美国 Holland 教授于 1969 年提出,后经 DeJong 和 Goldberg 等人归纳总结所形成的一类模拟进化算法^[5].遗传算法是模拟生物在自然环境中的遗传和进化过程而形成的一种自适应全局优化概率搜索算法,其生物学原理是自然界生物的遗传和进化过程^[8].在遗传算法中,一个解向量被称为一个染色体(或称个体),解向量中的分量称为基因.遗传算法的思想:由若干个体组成一个群体,该群体不断地进行选择、交叉和变异操作,并且每次都按照“优胜劣汰”的规则将优秀个体的基因遗传到下一代,最终在群体中得到一个优良的个体,即问题的近似最优解.基本 GA 算法包括选择算子、交叉算子和变异算子 3 个遗传算子.

粒子群算法使用简单,收敛速度快,但是容易早熟,陷入局部最优;而遗传算法全局搜索能力强,但搜索速度慢.可以看出这两种算法之间有着很强的互补性.结合供应链调度问题的特点,将这两种基于群概念的算法结合起来对供应链调度优化问题进行求解,可得到一个整体性能更优的混合算法 (GP).其思路是先利用粒子群算法收敛速度快的特点进行前一阶段的优化,得到一定进化程度的初始种群;然后由遗传算法进行后一阶段的优化,避免陷入局部最优.该算法的优化框架如图 1 所示.

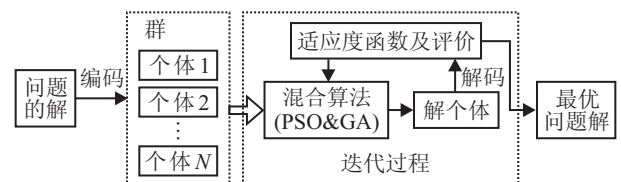


图 1 供应链调度优化的流程框架

首先,根据问题的编码规则,随机生成若干个体编码,组成一个种群;然后该种群依据混合算法进行进化,进化过程产生的新个体将被解码,得到对应的目标函数值;评价机制根据这些目标函数值,对新种群中的个体进行比较和评价,可得到当前的最优解;同时,每一次对种群的评价结果又为算法的进化过程提供了方向.上述过程不断迭代,最终可得到一个最优个体.

4 供应链调度问题的混合算法求解

4.1 映射及编码

将随机搜索算法用于生产调度优化, 一个重要的问题是在个体表达和问题的解之间找到合适的映射, 这样才能将个体的搜索空间限定在问题的解空间内, 并在此空间进行最优解的搜索以达到问题求解的目的. 为能够使用智能搜索算法求解, 有必要先将模型映射到图中. 定义图 G 为表示供应链调度优化问题的有向图, 即 $G = (N, A)$. 其中: $N = (n_{i,j,m})$ 是图中节点的集合, 每个节点对应方案描述矩阵 $C_i(1, 2, \dots, M)$ 中的一个元素; A 是图中弧的集合, 表示节点间的关系, 即规则库中所定义的工序约束关系. 对于任意节点 $N = (n_{i,j,m})$, 由它出发的弧指向所有满足 $i' \neq i$ (或 $i' \neq i, k_{j',j}^i$) 的 $n_{i',j',m}$.

另定义虚拟起始点 sPoint 为图中一个仅与子任务约束矩阵 $K_i(1, 2, \dots, M)$ 中全为“0”的列所对应的子任务节点相连的节点, 虚拟起始点不占用任何操作时间和操作成本.

通过上述定义, 供应链调度优化问题便转化为有约束的最短路径问题. 这样, 将由虚拟起点出发遍历各个节点的有效路径作为一个可行解, 通过编码对应于遗传算法的一条染色体, 同时对应于粒子群算法中的粒子, 使得编码更为简单便利.

4.2 算法参数设计

算法的参数对于算法的性能有着较大的影响, 因此如何进行参数设置是一项重要内容. 混合算法由粒子群算法和遗传算法两部分结合而成, 下面分别对这两部分进行参数分析设计, 所得结果见表1.

表1 GA-PSO 混合算法中的参数设计

PSO 部分		GA 部分	
群体规模 K	20	种群大小 K	20
最大迭代数 MG-PSO	100	最大迭代数 MG-GA	300
惯性权重 W	1.6	交叉概率 P_c	0.7
加速度常数 C_1, C_2	2.0	变异概率 P_m	0.05

4.2.1 粒子群算法参数设计

人们已开发了多种 PSO 模型, 本文使用的是全局优化模型^[7], 具体形式如下:

$$V_{id} = WV_{id} + C_1 \text{Rand}() (p_{id} - X_{id}) + C_2 \text{rand}() (p_{gd} - X_{id}), \quad (4)$$

$$X_{id} = X_{id} + V_{id}. \quad (5)$$

其中: V_{id} 是微粒 i 的飞行速度; X_{id} 是微粒 i 的位置; p_{id} 是微粒 i 所经历的最好位置 (p_{best}); p_{gd} 是群体中所有微粒所经历过的最好位置 (g_{best}); W 是惯性权重, 负责调整微粒群的全局搜索和局部探索能力; C_1 和 C_2 是加速度常数, 表示将微粒拉向 p_{best} 和 g_{best}

的随机项权重; $\text{Rand}()$, $\text{rand}()$ 是两个产生于 $[0, 1]$ 范围内的随机数.

下面结合供应链调度优化问题, 对模型中的相关参数进行设计.

1) 惯性权重 W , 加速度常数 C_1 和 C_2

惯性权重 W 的意义是使微粒保持运动惯性, 使其有扩展搜索空间的趋势, 有能力探索新的区域^[9]. 本文中的惯性权重 W 取为 $1.5 \sim 2.0$. 加速度常数 C_1 和 C_2 用于调整 PSO 空间的“弹性”, 低加速度值允许微粒被拉回前在远离目标的区域徘徊, 而高加速度值则导致微粒突然冲向或冲过目标区域. 根据以往经验, 取 $C_1 = C_2 = 2.0$.

2) 最大速度 V_{max} 和最大位置 X_{max}

速度 V_{id} 和位置 X_{id} 的绝对值有可能很大, 使得微粒有可能一下子飞出问题空间. 因此, 应将 V_{id} 和 X_{id} 的值限定在一定范围内, 即设定最大速度 V_{max} 和最大位置 X_{max} . V_{max} 作为一种约束来控制微粒群的全局搜索能力, 较大的值有利于全局搜索, 而较小的值有利于局部探索. 在优化调度问题中, V_{max} 被设定为 M (任务数量), 因为速度可为负值, 所以 V_{id} 是 $[-(M-1), M-1]$ 之间的值. X_{max} 设定为 M , 因为 X_{id} 代表任务 ID 号, 它只能为正值, 所以 X_{id} 是区间 $[0, M-1]$ 内的值.

3) 群体规模 K 和最大迭代次数 MG-PSO

该参数的设计主要是依据问题的规模, 本问题取群体规模 K 为 20, 最大迭代次数为 100.

4.2.2 遗传算法参数设计

为保持两算法接口, 将种群大小设计为与 PSO 的群体规模一致. 以下结合供应链优化调度问题, 对各算子的设计进行分析.

1) 选择算子

选择算子的设计采用比例选择和最优保存的混合策略. 比例选择方法对于一个种群大小为 K , 个体适应度为 F_i 的种群, 个体 i 被选中的概率为

$$P_{is} = F_i^{-1} / \sum_{i=1}^K F_i^{-1}.$$

该方法能保证具有最小适应度的个体被选中的概率最大. 在新种群生成之后, 应采用最优保存策略, 即比较出新旧种群中的最小适应度个体, 并用该个体取代新种群中的最大适应度个体.

比例选择方法能以概率选择的方式保证种群的质量和多样性, 而最优保存策略则是保证遗传算法收敛的一个重要条件.

2) 交叉算子

交叉算子的设计和实现与所研究的问题密切相关

关,一般要求既不要太多地破坏个体编码串中表示优良性状的优良模式,又要有效地产生出一些较好的新个体模式.交叉算子的设计包括两方面内容:确定交叉点的位置以及基因交换的方法.

本文采用基本的单点交叉来实现这一算子.其方法为:首先随机产生一个交叉位置;然后两个父染色体以交叉位置为界,互相交换交叉位置前段的染色体,从而生成两个新的子染色体.交叉操作的实施与否由交叉概率 P_c 决定,一般的取值范围是 $0.4\sim 0.9$,本文实验取为 0.7 .

3) 变异算子

变异算子的设计包括两方面内容:确定变异点的位置以及变异位置处的基因替换方法.在此,设计变异算法如下:首先随机产生两个变异位置;然后交换这两个变异位置上的代码值.变异操作由变异概率 P_m 决定是否进行,一般取为 $0.001\sim 0.1$,本文实验取为 0.05 .

4.3 混合算法步骤

GP混合算法步骤如下:

1) 初始化各参数.随机初始化粒子种群,将每个粒子的位置值代入目标函数,得到对应的适应值.

2) 评价适应度函数,记录粒子本身最优解 p_{best} 和整个种群目前找到的最优解 g_{best} .

3) 根据式(4)和(5)循环优化更新粒子自己的速度与位置,达到最大迭代次数后,输出初始优化种群.

4) 按照适应度值的大小,选择要交叉的个体,并以交叉概率 P_c 进行遗传算法的交叉操作,将两个父代个体的部分结构加以替换重组,生成新个体.

5) 以 P_m 为变异概率进行变异操作,辅助产生新个体加入子代种群中.

6) 若满足迭代条件,则停止,选择最佳个体作为遗传粒子群算法的结果;否则,转第4)步.

4.4 非法解的修正

在GA-PSO混合算法的计算过程中,由于算法的计算机理可能会导致非法解的出现,即新生成的个体不符合问题的编码规则,这样的个体无法被解码成问题的最终解,必须对其进行修正.GA-PSO混合算法中存在两种非法解情况:1)在PSO优化过程中,经过式(4)计算之后,微粒个体位置可能会出现小数,比如 7.12 ,这不符合问题的编码定义,即微粒个体的位置矢量代表子任务编号,应为整数.对此类非法解,其修正方法为:将小数值取整到与之最接近的整数,使之满足合法解的基本要求.2)交叉运算后有可能出现新个体中某个任务号的个数大于或小于设定个数,这也不符合个体的编码规则.GA与PSO两个算法过

程都不会改变任务号的总数目,因此,若有某个任务号大于设定值,则必有其他任务号小于设定值.此类非法解的修正方式为:依次对个体中任务号的数目进行检查,若大于设定值,则随机删除多余的个数,并记录相应位置为空白位置;然后,对小于设定个数的任务号,随机挑选空白位置进行添加.

5 仿真结果及分析

为简化计算过程,便于调度结果的优劣及稳定性比较分析,针对如图2所示的MC供应链系统,抽取其中订单数据,在各次仿真过程保持不变.选择6个具有相同分解结构的订单作为实验数据.订单均分解为6个子任务.其中:子任务 P_1, P_3, P_6 由核心制造商自行完成;子任务 P_2, P_4, P_5 为外购件,由相应的供应商完成.这里 P_2 有两个供应商 S_1, S_2 可选; P_4 为 S_3, S_4 和 S_5 ; P_5 则为 S_6 和 S_7 .不同的生产厂商完成同一任务的加工时间及加工成本是不同的.

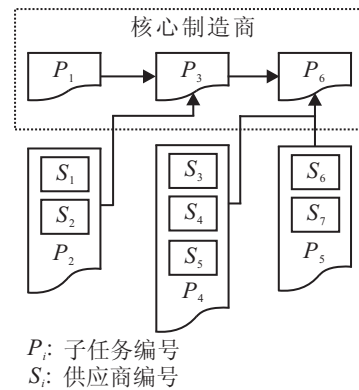


图2 订单分解结构示意图

表2为各企业完成订单各子任务所需的加工时间及加工成本,括号外数据为时间,括号内的数据为加工成本.另外,每单位时间的库存成本为在制品生产成本的 0.5% .

分别运用标准遗传算法、粒子群算法和本文的混合算法求解该供应链调度优化问题,算法收敛曲线如图3所示.各进行50次的仿真计算,最优结果在混合算法的某次仿真第379代得到(如图4所示),并统计得出表3的实验结果.

通过对比可以看出,3种算法的最优值、最劣值、平均值从小到大依次为混合算法、遗传算法、粒子群算法,搜索成功率从小到大依次为粒子群算法、遗传算法、混合算法,这反映出混合算法的全局搜索能力最强,遗传算法其次,粒子群算法最弱.平均成功搜索收敛代数从小到大依次为粒子群算法、混合算法、遗传算法,这反映出粒子群算法收敛速度最快,混合算法其次,遗传算法最慢.从总体结果上看,混合算法有着更好的优化性能.

表2 订单子任务加工时间及成本表

子任务	P_1	P_2		P_3	P_4			P_5		P_6	交货期
		S_1	S_2		S_3	S_4	S_5	S_6	S_7		
订单1	3 (123.67)	3 (13.26)	5 (14.17)	1 (21.54)	2 (7.42)	2 (5.81)	3 (6.95)	3 (29.61)	5 (34.81)	2 (30.83)	40
订单2	6 (247.34)	9 (28.01)	8 (32.83)	2 (43.08)	2 (12.13)	3 (11.44)	4 (14.17)	6 (72.99)	7 (58.40)	4 (61.66)	60
订单3	6 (247.34)	5 (26.62)	5 (33.46)	2 (43.08)	4 (12.89)	5 (11.57)	4 (15.47)	10 (62.40)	14 (59.20)	4 (61.66)	75
订单4	9 (371.01)	12 (39.23)	10 (43.28)	3 (64.62)	5 (19.65)	4 (17.20)	5 (21.23)	9 (100.15)	10 (91.56)	6 (92.49)	75
订单5	9 (371.01)	8 (39.69)	10 (50.64)	3 (64.62)	4 (19.21)	9 (17.75)	8 (23.38)	13 (108.39)	9 (88.45)	6 (92.49)	60
订单6	18 (742.02)	16 (79.39)	20 (89.94)	6 (129.94)	11 (44.16)	17 (41.63)	8 (45.79)	3 (219.18)	15 (165.58)	12 (184.98)	50

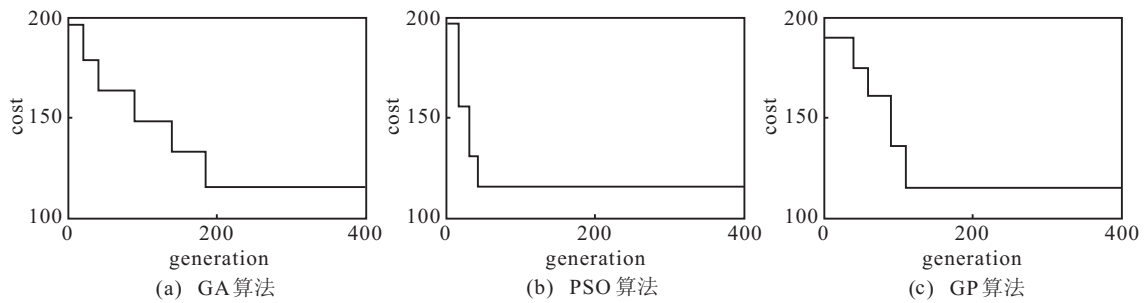


图3 3种算法的收敛曲线

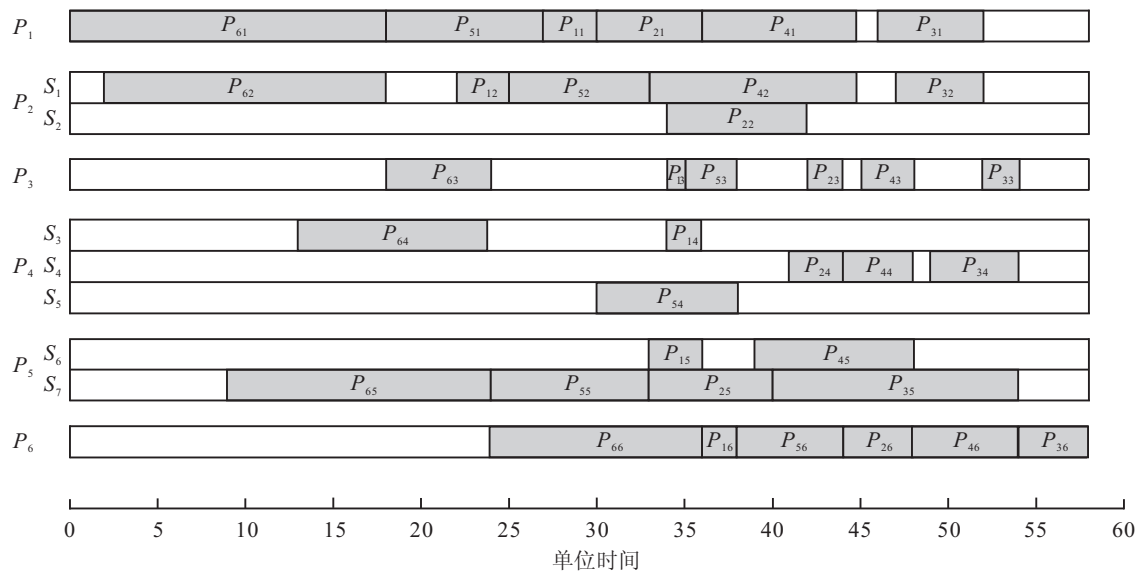


图4 优化调度方案

表3 实验结果比较

算法	最优值	最劣值	平均值	搜索成功率/%	平均搜索收敛代数	平均用时/min
GA	116.036	192.356	141.781	15.3	190.12	8.03
PSO	116.955	197.862	146.293	13	62.53	4.11
GP	115.182	185.658	133.765	59	127.68	5.27

为了进一步检验本文算法的性能, 采用8个Lawrence基准测试例子^[12], 再次对3种算法进行测试, 并加入瓶颈转移算法(SB)^[13]进行比较, 结果见表4. 可以看出, 本文混合算法对不同规模问题的求解有着良好的稳定性, 寻优性能也强于其他算法.

表 4 Lawrence 测试例子的算法比较

算例(规模)	已知最优解	GA		PSO		SB		GP	
		Best	Dev/ %	Best	Dev/ %	Best	Dev/ %	Best	Dev/ %
La01(10×5)	666	678	1.80	684	2.70	666	0.00	666	0.00
La06(15×5)	926	926	0.00	930	0.43	926	0.00	926	0.00
La11(20×5)	1222	1277	4.50	1253	2.54	1222	0.00	1222	0.00
La16(10×10)	945	956	1.15	983	4.02	978	3.49	956	1.16
La21(15×10)	1048	1088	3.82	1139	8.68	1084	3.44	1067	1.81
La26(20×10)	1218	1271	4.14	1311	7.64	1224	0.49	1218	0.00
La31(30×10)	1784	1813	1.63	1857	4.09	1784	0.00	1795	0.62
La36(15×15)	1268	1291	1.81	1369	7.97	1305	2.92	1291	1.81

6 结 论

本文针对大规模定制模式下的生产调度优化问题,利用粒子群算法和遗传算法的互补性,设计了一种将两者结合的混合算法,通过与粒子群算法和遗传算法的求解结果进行比较,验证了混合算法具有更优的性能,具备了粒子群算法收敛速度快和遗传算法全局搜索能力强的优势,同时在很大程度上克服了二者的局限性.但是,本文采用的数学规划模型有待于进一步的扩展和完善.算法参数对于求解结果的影响较大,相关参数也有待于进一步的优化.

参考文献(References)

- [1] Ashton J E, Cook F X. Time to reform job shop manufacturing[J]. Harvard Business Review, 1989, 7(3): 106-111.
- [2] 姚建明, 蒲云. 基于动态生产能力约束的MC模式下供应链调度优化[J]. 系统工程, 2005, 23(2): 25-30.
(Yao J M, Pu Y. Scheduling optimization of supply chain in mass customization based on dynamic production ability restriction[J]. Systems Engineering, 2005, 23(2): 25-30.)
- [3] 钱晓龙, 唐立新, 刘文新. 动态调度的研究方法综述[J]. 控制与决策, 2001, 16(2): 141-145.
(Qian X L, Tang L X, Liu W X. Dynamic scheduling: A survey of research methods[J]. Control and Decision, 2001, 16(2): 141-145.)
- [4] Lotfi K G, Sherif A M, Ashraf O N. A particle swarm-based genetic algorithm for scheduling in an agile environment[J]. Computers & Industrial Engineering, 2008, 55(3): 707-720.
- [5] 葛继科, 邱玉辉, 吴春明, 等. 遗传算法研究综述[J]. 计算机应用研究, 2008, 25(10): 2911-2916.
(Ge J K, Qiu Y H, Wu C M, et al. Summary of genetic algorithms research[J]. Application Research of Computers, 2008, 25(10): 2911-2916.)
- [6] 曹平, 陈盼, 刘世华. 改进的粒子群算法在旅行商问题中的应用[J]. 计算机工程, 2008, 34(11): 217-218.
(Cao P, Chen P, Liu S H. Application of improved particle swarm optimization in TSP[J]. Computer Engineering, 2008, 34(11): 217-218.)
- [7] 张秀芬, 张树有. 基于粒子群算法的产品拆卸序列规划方法[J]. 计算机集成制造系统, 2009, 15(3): 508-514.
(Zhang X F, Zhang S Y. Product disassembly sequence planning based on particle swarm optimization algorithm[J]. Computer Integrated Manufacturing Systems, 2009, 15(3): 508-514.)
- [8] Chen Jen-shiang, Pan Jason Chao-hsien, Lin Chien-min. A hybrid genetic algorithm for the re-entrant flow-shop scheduling problem[J]. Expert Systems with Applications, 2008, 34(1): 570-577.
- [9] 王俊伟, 汪定伟. 粒子群算法中惯性权重的实验与分析[J]. 系统工程学报, 2005, 20(2): 194-198.
(Wang J W, Wang D W. Experiments and analysis on inertia weight in particle swarm optimization[J]. J of Systems Engineering, 2005, 20(2): 194-198.)
- [10] Shi X H, Liang Y C, Lee H P, et al. Particle swarm optimization-based algorithms for TSP and generalized TSP[J]. Information Processing Letters, 2007, 103(5): 169-176.
- [11] 王国新, 宁汝新, 王爱民. 基于仿真的生产调度优化技术研究[J]. 计算机集成制造系统, 2007, 13(7): 1419-1427.
(Wang G X, Ning R X, Wang A M. Production optimal scheduling technology based on simulation[J]. Computer Integrated Manufacturing Systems, 2007, 13(7): 1419-1427.)
- [12] Lawrence S. Resource constrained project scheduling: An experimental investigation of heuristic scheduling techniques[D]. Pittsburghers: School of Computer Science, Carnegie Mellon University, 1984.
- [13] Adams J, Balas E, Zawack D. The shifting bottleneck procedure for job shop scheduling[J]. Management Science, 1988, 34(5): 391-401.