

粒子群算法求解无能力约束生产批量计划问题^①

韩毅, 唐加福, 牟立峰, 潘震东

(东北大学系统工程研究所, 东北大学教育部流程工业综合自动化重点实验室, 沈阳 110004)

摘要:经典的粒子群优化算法是一个在连续的定义域内搜索数值函数极值的有效方法. 目前, 粒子群算法 (particle swarm optimization, PSO) 已经成为优化领域中的一个重要的优化工具, 其应用在很多优化问题中都可以见到. 虽然粒子群算法的应用范围已经十分广泛, 但是关于应用其求解多级生产批量计划问题 (multilevel lot-sizing problem, MLLS) 的文章并不多见. 文章提出结合遗传算法 (genetic algorithm, GA) 变异算子的混合粒子群优化算法 (hybrid particle swarm optimization, HPSO) 求解无能力约束装配结构 MLLS 问题. 通过实验验证了算法的可行性和有效性.

关键词: 粒子群优化算法; 多级生产批量计划问题; 遗传算法; 装配结构; 无能力约束

中图分类号: TP 29 **文献标识码:** A **文章编号:** 1007-9807(2008)05-0033-08

0 引言

物料需求计划 (MRP) 系统是 MRPII/ERP 系统^[1]的基础, 同时也是供应链管理^[2,3]的基础. 动态批量计划问题是 MRP 系统中生产计划的关键问题^[4]. 其主要目标是依据产品的数据结构及各个项目提前期, 根据最终项目的外部需求以及中间项目的关联需求, 在给定的计划时间范围 T 上, 确定全部项目在不同的时间段上的生产数量, 使得在 T 内项目总的调整费用和库存保管费用 (有的问题还考虑生产前的调整准备费用、缺货费用等) 的和最小^[5]. 生产批量计划问题可以分为多种类型. 从生产能力来看, 包括有能力约束和无能力约束批量问题; 从产品的结构角度包括单级结构、线型结构、装配结构和一般网络结构的批量问题; 还有以需求方式角度来划分的固定需求、时变需求、随机需求和模糊需求环境下的批量计划问题等^[6~9].

以往, 人们普遍采用启发式算法来求解

MLLS 问题^[10~13]. 近些年来, 人们开始将遗传算法^[5,14,15]、模拟退火^[16,17]及蚁群算法^[18]等智能优化算法或者结合启发式方法的智能优化方法^[19]来求解 MLLS 问题, 取得了很好的效果. 智能优化算法虽然不一定能够获得最优解, 但是可以花费较小的时间代价来得到近似最优解. PSO 算法是智能优化算法中的一种, 因具有极佳的稳定性和鲁棒性, 目前已被应用到如车辆路径问题^[20]及非线性优化问题^[21]等众多优化领域的研究中. 最近, 有国内学者将 PSO 算法应用于求解单级生产批量计划问题^[22]. 然而, MLLS 问题比单级批量问题要复杂得多, 这是因为多级系统与单级系统不同, 项目之间不再是独立的, 而是相互关联的^[5]. 截至目前, 国内外关于 PSO 算法求解 MLLS 问题的文章并不多见. 有鉴于此, 本文提出结合遗传算法变异算子的 PSO 算法 (HPSO) 来求解无能力约束 MLLS 问题. 实验验证了本文算法的可行性和有效性. 无能力约束 MLLS 问题的发展情况如表 1 所示.

① 收稿日期: 2006-08-28; 修订日期: 2008-07-03.

基金项目: 国家自然科学基金资助项目 (70625001; 70721001); 教育部科技研究重点项目 (104064); 教育部新世纪优秀人才支持计划资助项目 (NCET-04-280).

作者简介: 韩毅 (1979-), 男, 辽宁人, 博士生. Email: arctic_wind@yahoo.cn

表1 无能力约束多级生产批量计划问题的研究列表

Table 1 List of papers on the uncapacitated multilevel lot-sizing problems

作者	年份	产品结构类型	算法
Afentakis et al. [11]	1984	装配结构	拉格朗日松弛和分枝定界结合
Afentakis et al. [12]	1986	一般网络结构	拉格朗日松弛和分枝定界结合
Kuik, et al. [13]	1990	一般网络结构	模拟退火和级到级的启发式算法
唐立新等[5]	1997	装配结构	遗传算法
N. Dellaert et al. [15]	2000	一般网络结构	遗传算法
N. Dellaert et al. [14]	2000	一般网络结构	混合遗传算法
N. Dellaert et al. [19]	2003	一般网络结构	随机多级批量启发式方法
Ou Tang. [16]	2004	线形结构	模拟退火算法
J. Jeunet et al. [23]	2005	一般网络结构	单点随机搜索方法
R. Pitakaso et al. [18]	2006	一般网络结构	蚁群算法

1 生产批量计划问题的数学模型

在批量问题中,有三种经典的产品结构:1) 装配结构 2) 树状结构 3) 一般结构^[14]. 图1给出了三种不同的产品结构示意图.

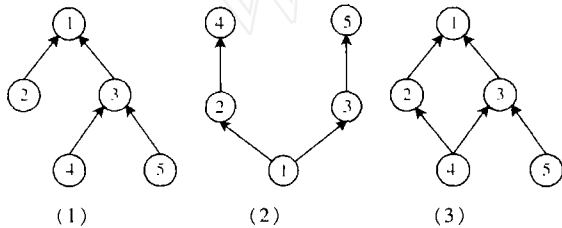


图1 三种不同的产品结构

Fig. 1 Three different product structures

图中每个节点对应一种物料,节点*i*和节点*j*之间的边(*i,j*)表示如果要生产物料*j*需要用到物料*i*.这里 $\Gamma^{-1}(i)$ 和 $\Gamma(i)$ 用来表示节点*i*的直接前驱和直接后继节点的集合.节点*i*的所有(直接和非直接)前驱节点的集合用 $\hat{\Gamma}^{-1}(i)$ 表示;节点*i*的所有后继节点的集合用 $\hat{\Gamma}(i)$ 表示.在图1的最后一个产品结构中物料1是一个完成品.如果用上边的符号来表示节点之间的关系的话,可以得到 $\Gamma^{-1}(1) = \{2, 3\}; \hat{\Gamma}^{-1}(1) = \{2, 3, 4, 5\}; \Gamma(4) = \{2, 3\}; \hat{\Gamma}(4) = \{1, 2, 3\}$ ^[14].文献[5]采用6个定义来说明各种物料之间的关系,其作用与上述方式相同.

本文考虑离散制造业中无能力约束的装配结

构 MLLS 问题. 因为无能力限制,所以提前期为零,并且不允许缺货. 总成本由生产准备费用和库存费用组成且成本参数不随时间变化. 对于问题进行求解时,可以将问题编码成由0或1组成的矩阵A. 矩阵A由公式(1)给出,其中*N*表示物料个数,*T*表示计划周期数. 其中 $a_{i,t}$ 为0表示在周期*t*不生产物料*i*; $a_{i,t}$ 为1表示在周期*t*生产物料*i*. 文献[5]给出了对矩阵A进行解码的公式和得到 $P_{i,t}$ 与 $I_{i,t}$ 的具体方法.

$$A = \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,T} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,T} \\ \vdots & \vdots & & \vdots \\ a_{N,1} & a_{N,2} & \cdots & a_{N,T} \end{pmatrix} \quad (1)$$

问题的数学模型是结合了文献[14]和[16]中的数学模型而得到的

$$\text{Min } Z = \sum_{i=1}^N \sum_{t=1}^T (H_i \times I_{i,t} + K_i \times a_{i,t}) \quad (2)$$

$$\text{s. t. } I_{i,t} = I_{i,t-1} + P_{i,t} - D_{i,t} \quad \forall i, t \quad (3)$$

$$D_{i,t} = \begin{cases} D_i, & \text{如果 } \Gamma(i) = \phi \\ \sum_{j \in \Gamma(i)} C_{i,j} \times P_{j,t} & \text{其它} \end{cases} \quad \forall i, t \quad (4)$$

$$I_{i,t} \geq 0, P_{i,t} \geq 0 \quad \forall i, t \quad (5)$$

$$a_{i,t} \in \{0, 1\} \quad \forall i, t \quad (6)$$

$$P_{i,t} \leq M a_{i,t}, a_{i,t} \in \{0, 1\} \quad \forall i, t \quad (7)$$

$$I_{i,0} = I_{i,T} = 0 \quad \forall i \quad (8)$$

式中:

i ——第 i 种物料;

$C_{i,j}$ ——生产一个单位的物料 j 所需要的物料 i 的数量;

H_i ——一个单位物料 i 的库存保管费用;

K_i ——物料 i 的生产准备费用;

$I_{i,t}$ ——物料 i 在第 t 个时期末的库存量;

$a_{i,t}$ ——二进制决策变量, 表示在第 t 个时期是否需要生产物料 i ;

$D_{i,t}$ ——物料 i 在第 t 个时期的需求量;

$P_{i,t}$ ——在第 t 个时期的生产/补充物料 i 的数量;

M ——非常大的正数

其中, $N, T, M, D_{i,t}, K_i, H_i$ 为已知常量, $a_{i,t}, P_{i,t}, I_{i,t}$ 为决策变量.

目标函数式(2)是对在整个计划区间内所有物料的生产装设和库存费用求和;式(3)表示满足需求的物流平衡方程. 式(4)表示各物料的需求;式(5)表示不允许缺货,且每个时期每种物料的生产量大于等于0;式(6)表示决策变量的取值0或者1;式(7)表示每个时段的生产数量是非负且不允许缺货;式(8)表示初始和结束周期的库存量为0.

2 基本粒子群算法

粒子群算法(PSO)是由 Kennedy 和 Eberhart 等人于 1995 年开发的一种演化计算技术. 其中“群(Swarm)”来源于粒子群,符合 Millonas 在开发应用于人工生命(Artificial Life)的模型时所提出的群体智能的 5 个基本原则;而“粒子(Particle)”则是一个折衷的选择. 因为既需要将群体中的成员描述为没有质量、没有体积,同时也需要描述它的速度和加速状态^[24]. PSO 算法最初是为了图形化地模拟鸟群优美而不可预测的运动. 人们通过对动物社会行为的观察,发现在群体中对信息的社会共享有利于在演化中获得优势^[20],并以此作为开发 PSO 算法的基础. 通过加入近邻的速度匹配,消除不必要的变量,并考虑多维搜索以及根据距离的加速,形成了 PSO 的最初版本. 之后, Shi 等^[25]引入惯性权重 ω 来更好地控制开发(Exploitation)和探索(Exploration),形

成了当前的标准版本.

PSO 基本算法的公式如下:

$$v_{k+1} = \omega \cdot v_k + c_1 \times \text{rand}() \times (p_{best} - p_k) + c_2 \times \text{Rand}() \times (p_{local} - p_k) \quad (9)$$

$$p_{k+1} = p_k + v_{k+1} \quad (10)$$

其中, v_k 是粒子的速度向量, p_k 是当前粒子的位置, p_{best} 是所有粒子所经历过的最好位置, p_{local} 是单个粒子所经历过的自身最好位置, ω 是惯性权重,一般是取(0,1)之间的随机数. c_1, c_2 为群体认知系数,一般取 2. $\text{Rand}()$ 和 $\text{rand}()$ 是两个(0,1)之间的随机数. 粒子在每一维上的速度都被限制在一个最大速度 V_{max} 内,如果某一维上的速度超过 V_{max} ,那么该维上的速度就被限定为 V_{max} .

3 混合粒子群算法

下面讨论如何采用 PSO 算法求解 MLLS 问题. 由于基本的 PSO 算法无法直接对 MLLS 问题进行求解,因此 PSO 算法的基本算法和公式(9)与(10)中的运算规则需要进行重新定义. 定义方式如下:

1) 搜索空间:本文定义搜索空间为所有决策矩阵的集合 $S = \{A_i\}, i \in 1, \dots, N$.

2) 位置:搜索空间是位置的集合,那么每个位置就是一个决策矩阵.

3) 粒子的速度:粒子的速度定义为数对的集合,集合的长度就是速度的长度,等于集合中数对的个数. $v = ((i_k, j_k)), i_k \in \{1, \dots, N\}, j_k \in \{1, \dots, T\}, k \uparrow \|v\|$. 数对的含义是将数对所标示的决策矩阵中的点进行变异操作. 变异的通常定义是:在解中对某一位的变换. 在本文中,变异操作就是将某个解中的某一位 $a_{i,t}$ 从‘1’变到‘0’或者从‘0’变到‘1’. 根据问题的假设和模型的约束条件,变化后的解不一定是可行解,需要进行解的可行化处理. 处理的方法是:一旦某种物料 i 在某个时期的值从‘1’变到‘0’,那么物料 i 的所有前驱节点在该时期的值也要为‘0’;一旦物料 i 在某个时期的值从‘0’变到‘1’,那么物料 i 的所有前驱节点在该时期的值也要为‘1’^[24]. 例如: $v = ((2, 3), (2, 5), (3, 7)), \|v\| = 3$. 就是将决策矩阵

中第2行第3、5列上的点依次进行变异操作,然后对第3行第7列上的点进行变异.

4) 速度“加”速度(\oplus):速度相加 $v_1 \oplus v_2$ 的定义是一个速度和另一个速度进行并集操作. 这样可以得到一个长度被尽量缩小了的速度. 如 $v_1 = ((1,4), (2,3)), v_2 = ((2,4), (2,3))$, 则 $V = v_1 \oplus v_2 = ((1,4), (2,3), (2,4))$.

5) 位置“加”速度(\boxplus):令 A 为一个位置, v 为

$$A = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \xrightarrow{(1,4)} A = \begin{pmatrix} 1 & 1 & 1 & \boxed{0} \\ 1 & 1 & 1 & \boxed{0} \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \xrightarrow{(2,2)} A = \begin{pmatrix} 1 & 1 & 1 & \boxed{0} \\ 1 & \boxed{0} & 1 & \boxed{0} \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \Rightarrow A' = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

6) 位置“减”位置(\boxminus):假设 A_1 和 A_2 是两个位置. A_1 和 A_2 的差为速度 v . 这个速度 v 由一个给定的算法来确定. 该算法的指导思想是通过比较 A_1 和 A_2 的不同点,从而得到一个速度. 为了使速度的长度尽量小,将该速度简化成一个与原来的速度等价的速度,将原速度和简化后的速度应用到 A_2 上,都可以得到 A_1 . 只是应用 v' 时不需要检验解的可行性,而应用 v 时需要根据可行性约束来调整解. 例如:

$$A_1 = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}, A_2 = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

在计算 $A_1 \boxminus A_2$ 时,通过比较 A_1 和 A_2 的不同之处,可以得到一个速度 $v' = ((1,3), (1,4), (2,2), (2,3))$,通过检验 v' 中的数对在 A_2 中的对应位置以及可行性约束条件,可以得到一个“缩小”了的速度 $v, v = ((2,2), (1,3), (1,4))$, v 中的数对按生产周期升序排列.

7) 系数“乘”速度(\cdot):假设 c 是一个实系数并且 v 是速度. 在 c 取不同值时,有不同的情况. 如果 $|c| < 1$, 那么设 $\|cv\|$ 是小于或等于 $c\|v\|$ 的最大整数, 定义 $c \boxdot v = ((i_k, j_k)), k \uparrow \frac{\|cv\|}{\|v\|}$. 令 $c = 0.6, v = ((2,3), (2,5), (3,7))$, 那么, $\|v\| = 3, c\|v\| = 1.8, \|cv\| = 1, c \boxdot v = ((2,3))$; 如果 $|c| > 1$, 那么 v 保持不

速度. 那么 $A' = A + v$ 表示将速度 v 应用到 A 上从而得到另一个位置 A' . 在应用速度 v 的时候,每将一个数对应用到 A 上,就根据“内角”属性对 p 进行

一系列的修改. 例如 $A = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}, v = ((1,4), (2,2))$, 将 v 中的数对依次应用到 P 上:

变. 这样做是为了尽量避免速度中出现重复数对的情况.

综合上述定义,PSO 算法的公式变为

$$v_{k+1} = \omega \cdot v_k \oplus (c_1 * rand()) \cdot (p_{best} \boxminus p_k) \oplus (c_2 * Rand()) \cdot (p_{local} \boxminus p_k) \quad (11)$$

$$p_{k+1} = p_k \boxplus v_{k+1} \quad (12)$$

上述定义的思想来源于文献[26]. 因文献[26]中 PSO 算法求解 TSP 问题的效果不太理想,因此本文提出 HPSO 来求解 MLLS 问题. 为了避免定义复杂的惩罚函数来引导搜索以及避免存储无用的不可行染色体,仅仅考虑可行的染色体^[15]. 公式(11)和(12)中运算符的优先关系并未改变,仍然是先乘除,后加减,括号的优先级最大.

8) HPSO 执行过程:

步骤 1 设定染色体数 n_p , 规定迭代次数 n_{max} , 随机产生 n_p 个初始解和初始速度.

步骤 2 给惯性权重 ω 赋初值 0.9 (随着程序的执行,权重的值由初值逐渐减小至 0.4).

步骤 3 根据当前位置,计算适应值,设定当前的适应值为个体极值 p_{local} , 根据各个粒子的个体极值,得到全局最优值 p_{best} .

步骤 4 根据公式 $\omega = ((Init_Weight - 0.4) * (n_{max} - Current_Iteration)) + 0.4$ 计算当前的惯性权重 ω 的值.

步骤 5 根据公式(11),计算速度 v_i .

步骤 6 将 p_i 与 p_{best} 和 p_{local} 分别做交叉运算得到 P'_i .

步骤7 根据公式(12),将速度 v_i 应用到 p_i 上得到新的 p_i .

步骤8 如果当前局部迭代索引值小于 n_p , 转步骤5.

步骤9 计算各个染色体的适应值,重新更新种群及 p_{best} 和 p_{Local} .

步骤10 如果连续5次迭代,最好解未改变,那么跳过步骤4~9,将种群中各个染色体根据变异概率,进行单点变异.之后,计算各个染色体的适应值,重新更新种群及 p_{best} 和 p_{Local} .转步骤4.

步骤11 如果连续200次迭代后最好解没有被改进,那么终止程序,转步骤13.

步骤12 如果当前迭代次数小于 n_{max} ,转步骤4.

步骤13 输出最好解.

4 数值仿真与分析

为了说明本算法的有效性,本文进行了2类实验.首先,对与文献[16]中具有线型结构的 MLLS 问题进行了求解;随后,对文献[5]中具有装配结构的 MLLS 问题进行了求解.在用 HPSO 求解这2类实验时,有些变量被设为常数. HPSO 的种群是随机产生的,其大小为60. HPSO 中 c_1 和 c_2 的值设为2,惯性权重根据文献[18]中的方法,从0.9逐渐降到0.4. HPSO 中的交叉概率是0.6,变异概率为0.4.算法的最大迭代次数是1000.如果在200次迭代中最好解没有改进或者已经达到最大迭代次数,那么终止程序.实验1的原始数据如表2所示.

表2 实验1的需求数据
Table 2 Demands of 36 periods in experiment 1

月份	1	2	3	4	5	6	7	8	9	10	11	12
需求	45	74	100	87	141	81	22	92	95	107	115	151
月份	13	14	15	16	17	18	19	20	21	22	23	24
需求	67	104	91	65	190	80	27	173	107	47	121	150
月份	25	26	27	28	29	30	31	32	33	34	35	36
需求	51	98	86	38	146	91	117	59	62	106	97	177

在实验1中,物料总数 N 等于9,计划周期数 T 为36.产品的内部需求关系是一一对一的,且第 i 种物料的库存保管费用为 $200 - 15 * i$,生产准备费用为 $100 * i$, i 从1开始直到9.算法中速度的最大值为 $0.02 * N * T$. GA 的交叉和变异率分别是0.6和0.2.本文采用运行时间(26s)作为两算法的比较标准,每种算法重复运行100次.表3给出了两种算法的比较结果,图3显示了两种算

法的成本分布趋势图.从表3中可以看到 HPSO 的平均成本和成本的标准差要比 GA 的对应值小,而且 HPSO 所取得的最好解和最差解比 GA 的最好解和最差解好.从图2中可以清楚的看到 HPSO 的成本函数曲线比 GA 的成本函数曲线分布的更集中.表3和图2说明:HPSO 的结果比 GA 的结果更稳定,HPSO 的性能比 GA 的性能更好.

表3 两种算法对于实验1的求解结果的比较
Table 3 The comparison of results between HPSO algorithm and GA for experiment 1

算法	平均成本	成本标准差	取得的最好解	取得的最差解
HPSO	267 498.65	30 097.57	201 355	368 555
GA	348 408.35	67 054.24	226 220	527 345

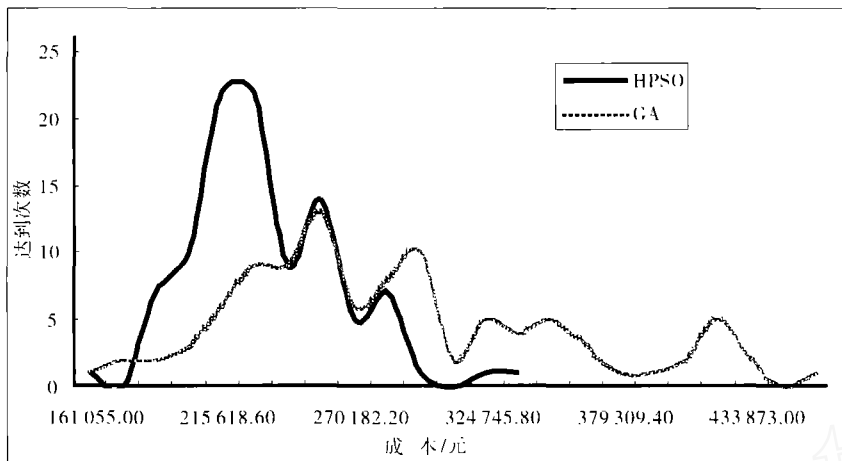


图2 两种算法的成本分布图

Fig. 2 Cost distribution of HPSO algorithm and GA

表4 实验2的参数数据

Table 4 Parameters used in experiment 2

项目号	结构参数		单位费用系数							
	$\Gamma(i)$	$C(i, \Gamma(i))$	库存费用	项目调整费用						
1	0	0	1	130						
2	1	1	2	120						
3	1	3	1	25						
4	2	2	2	30						
5	2	4	3	30						
6	3	2	1	40						
r	1	2	3	4	5	6	7	8	9	10
$D_{i,j}$	32	41	148	36	120	28	32	12	30	10

在实验2中,速度的最大值为 $0.1 \times N \times T$. 实验2的原始数据由表4给出,计算结果由表5给出. 观察表5的结果,可以看到HPSO算法与文献[5]中GA算法对物料6的批量大小的计算结果并不相同. 这是因为原文献将物料3与6之间的内部需求关系计算错误而导致的. 即便如此,因为需要存储的物料是完成品,所以原文献的错误并未影响最优解的值. 表5中,HPSO的最优解与文献[5]中GA的最优解相等,比文献[5]中启发式方法的结果要好得多. 纵观这两个实验,HPSO对MLLS问题的求解是可行的和有效的.

表5 HPSO算法对文献[5]的求解结果

Table 5 Results of HPSO algorithm for one case in literature [5]

项目号	HPSO 的结果					GA 的结果					Heuristic 的结果																			
	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10										
1	257	0	0	0	232	0	0	0	0	0	257	0	0	0	232	0	0	0	0	0	73	0	184	0	148	0	84	0	0	0
2	257	0	0	0	232	0	0	0	0	0	257	0	0	0	232	0	0	0	0	0	73	0	184	0	148	0	84	0	0	0
3	771	0	0	0	696	0	0	0	0	0	771	0	0	0	696	0	0	0	0	0	219	0	552	0	444	0	252	0	0	0
4	514	0	0	0	464	0	0	0	0	0	514	0	0	0	464	0	0	0	0	0	146	0	368	0	296	0	168	0	0	0
5	1 028	0	0	0	928	0	0	0	0	0	1 028	0	0	0	928	0	0	0	0	0	292	0	737	0	592	0	336	0	0	0
6	1 542	0	0	0	1 392	0	0	0	0	0	2 313	0	0	0	2 088	0	0	0	0	0	657	0	1 656	0	1 332	0	756	0	0	0
目标函数值 = 1 493					目标函数值 = 1 493					目标函数值 = 1 707																				

5 结论

本文将 PSO 算法的应用领域进行了扩展,用

其对无能力约束的装配结构 MLLS 问题进行了求解,为生产批量计划问题的研究与应用又提供了一种求解工具. 仿真实验的结果证实了 PSO 算法

求解 MLLS 问题的可行性和有效性. 从问题的通用性角度来讲, 由于编程的复杂性, 本文算法所求解的问题目前还仅限于装配结构的 MLLS 问题.

从算法的角度来看, 随着研究的逐渐深入, 算法的达优能力可以通过引入更加高效的启发式算法而得到进一步的提高.

参考文献:

- [1] 陈伯成, 叶伟雄, 李英杰. 虚企业与 ERP 软件关系研究[J]. 管理科学学报, 2002, 5(5): 62—71.
Chen Bocheng, Ye Weixiong, Li Yingjie. Research on relationship between “virtual enterprise” and ERP software[J]. Journal of Management Sciences in China, 2002, 5(5): 62—71. (in Chinese)
- [2] 刘丽文. 供应链管理思想及其理论和方法的发展过程[J]. 管理科学学报, 2003, 6(2): 81—88.
Liu Liwen. Survey on evolution of SCM theory and methods[J]. Journal of Management Sciences in China, 2003, 6(2): 81—88. (in Chinese)
- [3] 陈安, 刘鲁. 供应链管理问题的研究现状与挑战[J]. 系统工程学报, 2000, 15(2): 179—186.
Chen An, Liu Lu. An overview of supply chain management: An operational perspective[J]. Journal of Systems Engineering, 2000, 15(2): 179—186. (in Chinese)
- [4] 傅学芳. 基于遗传算法的生产批量优化问题[J]. 应用数学与计算数学学报, 2001, 15(2): 58—64.
Fu Xuefang. Optimization of the manufacturing resources planning based on genetic algorithms[J]. Communication on Applied Mathematics and Computation, 2001, 15(2): 58—64. (in Chinese)
- [5] 唐立新, 杨自厚, 王梦光. CIMS 下 MRP-II 多级生产批量计划问题的研究[J]. 控制理论与应用, 1997, 14(3): 376—382.
Tang Lixin, Yang Zihou, Wang Mengguang. Research for lot-sizing problem of MRP-II in CIMS[J]. Control Theory and Applications, 1997, 14(3): 376—382. (in Chinese)
- [6] 谢金星, 姜启源, 邢文训, 等. 能力受限的批量问题的数学模型与算法新进展[J]. 运筹学杂志, 1996, 15: 1—12.
Xie Jinxing, Jiang Qiyuan, Xing Wenxun, et al. New developments of mathematical models and algorithms of uncapacitated lot sizing and scheduling problems[J]. Journal of Operations Research, 1996, 15: 1—12. (in Chinese)
- [7] Karimi B, Fatemi Ghomi S M T, Wilson J M. The capacitated lot sizing problem: A review of models and algorithms[J]. The International Journal of Management Science, 2003, 31: 365—378.
- [8] 唐加福, 董颖, 刘士新, 等. 模糊需求和模糊能力约束的集约生产计划[J]. 控制理论与应用, 2003, 20(6): 894—902.
Tang Jiafu, Dong Ying, Liu Shixin, et al. Aggregate production planning with fuzzy demands and fuzzy capacity[J]. Control Theory and Applications, 2003, 20(6): 894—902. (in Chinese)
- [9] 唐加福, 汪定伟, 徐宝栋. 多品种集约生产计划问题的模糊方法[J]. 管理科学学报, 2003, 6(1): 44—50.
Tang Jiafu, Wang Dingwei, Xu Baodong. Fuzzy modeling approach to aggregate production planning with multi-product[J]. Journal of Management Sciences in China, 2003, 6(1): 44—50. (in Chinese)
- [10] Silver A, Pyke F, Peterson R. Inventory Management and Production Planning and Scheduling[M]. 1998.
- [11] Afentakis P, Gavish B, Karmarkar U. Computationally efficient optimal solutions to the lot-sizing problem in multistage assembly system[J]. Management Science, 1984, 30(2): 229—239.
- [12] Afentakis P, Gavish B. Optimal lot-sizing algorithms for complex product structures[J]. Management Science, 1986, 30(2): 229—239.
- [13] Kuik K, Salomon M. The multi-level lot-sizing program evaluation of a simulated annealing[J]. European Journal of Operation Research, 1990, 45(6): 25—37.
- [14] Dellaert N, Jeunet J. Solving large unconstrained multilevel lot-sizing problems using a hybrid genetic algorithm[J]. International Journal of Production Research, 2000, 38(5): 1083—1099.
- [15] Dellart N, Jeunet J, Jonard N. A genetic algorithm to solve the general multi-level lot-sizing problem with time-varying

- costs[J]. *International Journal of Production Research*, 2000, 68: 241—257.
- [16] Tang Ou. Simulated annealing in lot sizing problems[J]. *International Journal of Production Economics*, 2004, 88: 173—181.
- [17] 韩丽敏, 王 飞, 冯允成. 解批量问题的模拟退火算法[J]. *北京航空航天大学学报*, 1999, 25(5): 588—591.
Han Limin, Wang Fei, Feng Yuncheng. Simulated annealing heuristic for lot sizing problem[J]. *Journal of Beijing University of Aeronautics and Astronautics*, 1999, 25(5): 588—591. (in Chinese)
- [18] Pitakaso R, Almeder C, Doerner K F, *et al.* Combining population-based and exact methods for multi-level capacitated lot-sizing problems[J]. *International Journal of Production Research*, 2006, 44: 4755—4771.
- [19] Dellaert N, Jeunet J. Randomized multi-level lot-sizing heuristics for general product structures[J]. *European Journal of Operation Research*, 2003, 148: 211—228.
- [20] 李 宁, 邹 彤, 孙德宝. 车辆路径问题的粒子群算法研究[J]. *系统工程学报*, 2004, 19(6): 596—600.
Li Ning, Zhou Tong, Sun Debao. Particle swarm optimization for vehicle routing problem[J]. *Journal of Systems Engineering*, 2004, 19(6): 596—600. (in Chinese)
- [21] Dong Y, Tang J, Xu B, *et al.* Application of particle swarm optimization to nonlinear programming problems[J]. *Computers and Mathematics with Applications*, 2005, 49: 1655—1668.
- [22] 马慧民, 柳 毅, 叶春明. 基于改进粒子群算法求解单级多资源约束生产批量问题[J]. *工业工程与管理*, 2005, 6: 66—70.
Ma Huimin, Liu Yi, Ye Chunming. Improved particle swarm optimization algorithm for the single level capacitated dynamic lot-sizing problem[J]. *Industrial Engineering and Management*, 2005, 6: 66—70. (in Chinese)
- [23] Jeunet J, Jonard N. Single-point stochastic search algorithms for the multi-level lot-sizing problem[J]. *Computers and Operations Research*, 2005, 32: 985—1006.
- [24] 谢晓锋, 张文俊, 杨之廉. 微粒群算法综述[J]. *控制与决策*, 2003, 18(2): 129—134.
Xie Xiaofeng, Zhang Wenjun, Yang Zhilian. Overview of particle swarm optimization[J]. *Control and Decision*, 2003, 18(2): 129—134. (in Chinese)
- [25] Shi Y, Eberhart R. A Modified Particle Swarm Optimizer[C]. *Proceedings of IEEE International Conference on Evolutionary Computation*, Anchorage, USA, 1998.
- [26] Clerc M. Discrete particle swarm optimization illustrated by the traveling salesman problem[EB/OD], <http://www.mauriceclerc.net>, 2000.

Particle swarm optimization algorithm for solving uncapacitated multilevel lot-sizing problems

HAN Yi, TANG Jia-fu, MU Li-feng, PAN Zhen-dong

Department of Systems Engineering, Northeastern University, Key Laboratory of Integrated Automation of Process Industry (MOE), Shenyang 110004, China

Abstract: The classical Particle Swarm Optimization (PSO) algorithm is a powerful method to find the minimums of numerical functions on a continuous definition domain. It has been a very important optimization tool in many research fields. So far, papers on the application of PSO algorithm to multilevel lot-sizing (MLLS) problems can not be seen often. In view of this, a PSO algorithm combined with the mutation operator of Genetic algorithm (GA) is come up with to solve MLLS problems. Our aim is to expand the application scope of PSO algorithm. Experiments showed the feasibility and credibility of this algorithm.

Key words: particle swarm optimization; multilevel lot-sizing problem; genetic algorithm; assembly structure; uncapacitated