

# 多模式多项目选择计划的动态克隆选择算法

夏林丽<sup>1</sup>, 雷 宏<sup>1</sup>, 苏显方<sup>2</sup>

(1. 铜仁学院 数学与计算机科学系, 贵州 铜仁 554300;  
2. 湖南大学 工商管理学院, 湖南 长沙 410012)

**摘 要:**针对多模式多项目多任务选择计划问题的特点, 随机选取各任务的执行模式, 综合考虑资源限制、项目承继约束、任务时序约束、项目合同时间约束等, 建立多模式多项目多任务选择计划的数学模型。采用二层编码表示调度的编码, 利用半随机的方式生成抗体群, 依据克隆选择原理, 实施动态克隆、自适应变异等策略, 设计动态克隆选择算法求解模型的最佳决策方案。通过数值实验, 与启发式算法、免疫克隆选择算法相比, 该算法具有更好的性能。

**关键词:**承继约束; 资源均衡; 克隆选择; 自适应变异; 动态克隆

中图分类号: TP399; TP301.6

文献标识码: A

文章编号: 1673-629X(2015)06-0060-04

doi: 10.3969/j.issn.1673-629X.2015.06.013

## Dynamic Colonel Algorithm for Multi-mode Multi-project and Multi-job Selection Planning

XIA Lin-li<sup>1</sup>, LEI Hong<sup>1</sup>, SU Xian-fang<sup>2</sup>

(1. Department of Mathematics and Computer Science, Tongren University,  
Tongren 554300, China;  
2. Business School of Hunan University, Changsha 410012, China)

**Abstract:** For the analysis of the characteristics of multi-project and multi-job selection planning, considering the resource limitation, projects inheritance constrains, timing constrains of tasks, project contract time constrains and so on comprehensively, the multi-mode, multi-project and multi-job selection planning model was proposed. It is used to obtain the optimal scheduling sequences so that resource equilibrium are utilized. According to the colonel selection principle, some strategies are adopted such as dynamic clone, adaptive mutation and so on, an Dynamic Colonel Selection Algorithm (DCSA) is designed to find the best solution of the model. By numerical experiments, compared with the immune colonel selection algorithm and the heuristic algorithm, show that the algorithm has better performance.

**Key words:** inheritance constrains; resource equilibrium; colonel selection; adaptive mutation; dynamic clone

### 1 概 述

实际的生活中,尤其在飞机制造、建筑工程等小批量生产方式的企业里,根据各种资源投入量的不同,任务的完成时间也不同,各任务有多种不同的执行模式。这类问题不仅要调整项目内部各项任务的实际开工时间,还要选取各项任务合适的执行模式,此类问题是多模式项目调度问题。针对多模式单项目调度问题,解决此类问题的算法有模拟退火算法<sup>[1-2]</sup>、遗传算法<sup>[3-5]</sup>、克隆选择优化算法<sup>[6]</sup>、蚁群算法<sup>[7]</sup>、微粒群算法<sup>[8-9]</sup>等。

近年来,企业为了发展,承接的项目越来越多且越

来越复杂,项目内部的各任务有多种不同的执行模式,继续用单项目管理方法对多项目进行管理是不可行的,也不利于企业发展。另外,企业资源是有限的,加上一些不可控意外情况的发生,企业只能选择部分项目执行,其余项目要进行外包。针对此类问题,不仅要合理地选择项目执行,还要调整项目内部各项任务的实际开工时间以及选取每个项目内部各项任务合适的执行模式,这就是多模式多项目选择计划问题。此类问题是多模式单项目调度问题的扩展,约束条件多,问题更加复杂,求解难度增加,但更接近于实际。此类问题也是企业需要解决的难题。目前还很少见到解决多

收稿日期: 2014-07-09

修回日期: 2014-10-15

网络出版时间: 2015-05-06

基金项目: 贵州省科技计划项目(黔科合J字 LKT[2012]20号,铜市科研(2013)9-3号(科技局政产学研))

作者简介: 夏林丽(1982-),女(侗族),讲师,硕士研究生,研究方向为项目管理、系统建模与智能优化。

网络出版地址: <http://www.cnki.net/kcms/detail/61.1450.TP.20150506.1627.013.html>

模式多项目选择计划问题的文献。

基于此,文中在已有研究的基础上,根据多模式多项目多任务选择计划问题的特点,建立多执行模式多项目多任务选择计划模型,并设计动态克隆选择算法(Dynamic Colonel Selection Algorithm, DCSA)对其求解。数值试验说明了该算法是有效的、可行的。

## 2 问题描述

设包含  $n$  个项目的项目集为  $C = \{c_i \mid i = 1, 2, \dots, n\}$ , 各项目所需的资源种类、资源数可能不同。某企业一共有  $R$  种资源,第  $k$  种资源在第  $t$  天的总限量为  $R_{tk}$ ,第  $k$  种资源在  $T$  天内总限量为  $R_k$ ,资源的权重为  $\mu_k$ ,  $0 \leq \mu_k \leq 1$ 。项目之间存在承继关系,  $S_i$  表示  $c_i$  的紧前项目集。在同一项目内,任务之间满足时序约束,  $c_i$  的任务数记为  $J_i$ ,  $c_{is}$  表示项目  $c_i$  的第  $s$  个任务,  $P_{ik}$  表示  $c_{ik}$  的紧前任务集。  $s_i$ 、 $t_{is}$  分别表示  $c_i$  开始时间及其第  $s$  个任务的开始时间。每个任务有多种执行模式,  $c_{is}$  有  $M_{is}$  种执行模式,  $c_{is}$  必须选择其中之一执行,  $M_{is} = \{1, 2, \dots, |M_{is}|\}$ , 执行模式选定后在整个执行过程中不得中断或更改。  $m_{is} \in M_{is}$  表示  $c_{is}$  在  $M_{is}$  中选择的执行模式,  $d_{ism_k}$  表示  $c_{is}$  选择模式  $m_{is}$  执行的持续时间,  $r_{ism_k}$  表示  $c_{is}$  选择模式  $m_{is}$  执行时对应每天使用第  $k$  种资源的资源量;  $e_i$ 、 $T_i$  分别为  $c_i$  的合同日期、持续时间。决策变量定义如下:

$$b_i = \begin{cases} 1, & c_i \text{ 被选中执行} \\ 0, & \text{otherwise} \end{cases}$$

$$b_i^j = \begin{cases} 1, & c_i \text{ 在第 } j \text{ 天被选中执行} \\ 0, & \text{otherwise} \end{cases}$$

$$a_i^j = \begin{cases} 1, & c_i \text{ 在第 } s \text{ 个任务在第 } j \text{ 天被选中执行} \\ 0, & \text{otherwise} \end{cases}$$

则多模式多项目多任务选择计划资源均衡模型建立如下:

$$\text{Min} \sum_{k=1}^R \mu_k \sigma_k \quad (1)$$

s. t.

$$T_0 = \max_{i=1,2,\dots,n} \{s_i + T_i\} - \min_{i=1,2,\dots,n} s_i \quad (2)$$

$$T_i = \max_{j=1,2,\dots,J_i} \{t_{ij} + d_{ijm_j}\} - \min_{j=1,2,\dots,J_i} t_{ij} \quad (3)$$

$$\sum_{i=1}^n b_i^j \sum_{s=1}^{J_i} a_{is}^j r_{ijm_jk} \leq R_{jk} \quad (4)$$

$$\sum_{i=1}^n b_i \sum_{j=1}^{J_i} d_{ijm_j} r_{ism_jk} \leq R_k \quad (5)$$

$$t_{ik} - t_{is} \geq d_{ijm_j}, c_{is} \in P_{ik}, s \neq k \quad (6)$$

$$s_i \leq e_i - T_i, s_i - s_j \geq T_j, \forall c_j \in S_i \quad (7)$$

$$\sigma_k = \frac{1}{T_0} \sqrt{\sum_{j=1}^{T_0} (R_{tk} - \sum_{i=1}^n b_i^j \sum_{s=1}^{J_i} a_{is}^j r_{ijm_jk})^2} \quad (8)$$

$$\sum_{k=1}^R \mu_k = 1 \quad (9)$$

模型中,式(1)为目标函数,表示资源利用偏差最小;式(2)表示被选中项目的总工期;式(3)表示被选中的每个项目的实际执行时间;式(4)表示在第  $j$  天第  $k$  种资源的资源约束;式(5)表示第  $k$  种资源的总资源约束;式(6)表示任务的时序约束;式(7)分别表示合同约束及项目的承继约束;式(8)表示第  $k$  种资源的资源利用偏差;式(9)表示资源的权重之和。

## 3 动态克隆选择算法(DCSA)

借鉴生物体免疫系统对自适应抗原刺激的动态过程,设计动态克隆选择算法,其依据抗体与抗体间的亲和力和抗体与抗原的亲和力动态分配记忆单元和一般抗体种群的规模,进而实现记忆单元和一般抗体进化的自适应调节<sup>[10]</sup>。

### 3.1 算法描述:

Step1:置  $k \leftarrow 0$ , 设定算法参数,依据抗体生成规则,生成规模为  $N$  的初始抗体群  $A(0)$ 。

Step2:根据亲和力函数计算每个抗体的亲和力。

Step3:对  $A(k)$  所有的抗体根据亲和力的大小进行降幂排序,将抗体群分为记忆池抗体群  $M(k)$  和一般抗体群  $Ab(k)$ ,记忆池和一般抗体的规模进行动态分配,其规模分别为  $M, N - M$ , 其中

$$M = \text{fit}[N \times (S + D)]$$

$$D =$$

$$\frac{1}{N \times (N - 1)} \sum_{j=1}^N \sum_{i=1}^N | \text{aff}(A_i(k)) - \text{aff}(A_j(k)) |$$

$$\max | \text{aff}(A_i(k)) - \text{aff}(A_j(k)) |$$

式中,  $\text{fit}(\bullet)$  为取整函数;  $S$  是保证记忆池最小规模而设计的常数;  $D$  是为了增加抗体与抗体间的多样性。

Step4:将  $A(k)$  中抗体执行动态克隆操作,获得克隆群  $B(k)$ 。

Step5:对  $A(k)$ 、 $B(k)$  中的抗体实施自适应变异,分别获新群体  $A'(k)$  和  $C(k)$ 。

Step6:对  $A'(k)$ 、 $C(k)$  实施一点交叉,获得  $A''(k)$ 、 $C'(k)$ 。

Step7:对  $A''(k) \cup C'(k)$  实施群体更新,选择新的进化抗体群  $A(k+1)$ 。

Step8:置  $k \leftarrow k + 1$ ,若满足终止条件,则输出结果,并结束;否则,返回 Step2。

### 3.2 抗体编码

抗体的编码分为两层:项目层和任务层。

第一层表示最终项目的调度顺序及被选中的情况。表示为:

$$\mathbf{Ab}_1 = \begin{bmatrix} \mathbf{C} \\ \mathbf{B} \end{bmatrix} = \begin{bmatrix} c_1 & c_2 & \cdots & c_n \\ b_1 & b_2 & \cdots & b_n \end{bmatrix}$$

其中,第一行表示项目的调度顺序;第二行表示对应项目被选中的情况。

第二层表示各项目内部任务的调度序列及对应的执行模式,表示为:

$$\mathbf{Ab}_2 = \begin{bmatrix} \mathbf{J} \\ \mathbf{M} \end{bmatrix} = \begin{bmatrix} J_{11} & J_{1L_1} & \cdots & J_{n1} & & J_{n2} & \cdots & J_{nL_n} \\ M_{11} & \cdots & M_{1L_1} & \cdots & M_{n1} & M_{n2} & \cdots & M_{nL_n} \end{bmatrix}$$

其中,第一行是项目调度顺序中各项目内部任务的调度顺序,第二行表示各任务的执行模式。

### 3.3 抗体生成规则

采用半随机的方式生成抗体,项目的调度顺序和各项目内部任务的执行顺序利用拓扑排序<sup>[11]</sup>生成,然后在各任务对应的模式集中随机选择一种执行模式。根据资源约束按项目的调度顺序和对应内部任务的执行顺序安排任务的开始时间及完成时间,进而确定项目的开始时间和完成时间。满足合同约定则选择此项目执行,否则不选。按此方式生成足够多的抗体。

### 3.4 抗体亲和力

对于最小化问题,亲和力与目标函数值成反比,则抗体 Ab 的亲和力  $\text{aff}(\text{Ab})$  设计为

$$\text{aff}(\text{Ab}) = \frac{N_1}{N(1 + f(\text{Ab}))} \quad (10)$$

其中,  $N$  表示所承接的项目总数;  $N_1$  表示被选择执行的项目数;  $f(\text{Ab})$  表示 Ab 对应的目标函数值。

### 3.5 动态克隆

将抗体群  $A(k)$  中的抗体依据亲和力进行降幂排序,第  $i$  个抗体记为  $A_i(k)$ ,根据  $A_i(k)$  的抑制半径  $r(A_i(k))$  找出被抗体  $A_i(k)$  抑制掉的所有抗体构成的集合  $E(A_i(k))$ ,  $E(A_i(k))$  中元素个数为  $A_i(k)$  的克隆规模,其中

$$\begin{aligned} r(A_i(k)) &= \frac{\min_{j=1,2,\dots,N} |\text{aff}(A_i(k)) - \text{aff}(A_j(k))| + 2}{\max_{j=1,2,\dots,N} |\text{aff}(A_i(k)) - \text{aff}(A_j(k))| + 1} \\ E(A_i(k)) &= \{ A_j(k) \mid 0 \leq \text{aff}(A_i(k)) - \text{aff}(A_j(k)) \leq r(A_i(k)), A_j(k) \in A(k) \} \end{aligned}$$

如果  $A_j(k) \in E(A_i(k))$ ,即  $A_j(k)$  被  $A_i(k)$  抑制掉,则  $A_j(k)$  不进行克隆繁殖。 $A_i(k)$  的抑制半径越大,则它抑制掉其他抗体的数目越多,也即繁殖规模越大。

### 3.6 自适应变异

对于  $A(k)$  中的抗体  $A_i(k)$  ( $i = 1, 2, \dots, N$ ),每个抗体的变异概率为

$$p_m(A_i(k)) = 1 - \alpha * \delta$$

$$\delta = \exp\left(-\frac{\max_{j=1,2,\dots,N} (\text{aff}(A_j(k))) - \text{aff}(A_i(k))}{\max_{j=1,2,\dots,N} (\text{aff}(A_j(k))) - \min_{j=1,2,\dots,N} (\text{aff}(A_j(k)))}\right)$$

其中,  $0 < \alpha < 1$ ,随机产生一个 0 至 1 之间的随机数  $r$ ;如果  $r$  小于变异概率,则对  $A(k)$  中抗体进行插入式变异<sup>[12]</sup>操作。

对于  $B(k)$  中的抗体  $B_i(k)$  ( $i = 1, 2, \dots, N$ ),每个抗体的变异概率为

$$p_m(B_i(k)) = \beta + (1 - \beta)p(B_i(k))$$

$$p(B_i(k)) = \frac{1}{N} \sum_{B_j(k) \in B(k)} W(B_j(k), B_i(k))$$

$$W(B_j(k), B_i(k)) =$$

$$\begin{cases} 1, & \text{if}(\text{aff}(B_j(k)) < \text{aff}(B_i(k))) \\ 0, & \text{otherwise} \end{cases}$$

其中,  $\beta$  ( $0 < \beta < 1$ ) 可以调节。然后随机产生一个 0 至 1 之间的随机数;如果随机数小于变异概率,则随机生成不同的新抗体取代  $B_i(k)$ 。

### 3.7 一点交叉

对于  $A'(k)$  中的抗体,如果有  $A'_i(k) = A'_j(k)$ ,  $i, j = 1, 2, \dots, N$ ,  $i \neq j$ ,则从  $A'(k)$  中随机选取两个不相同的抗体进行一点交叉<sup>[13]</sup>,交叉产生的新个体按一定概率取代  $A'_i(k)$  或  $A'_j(k)$ 。对于  $C'(k)$  中的每个抗体,则从  $C'(k)$  中随机选取一个不相同的抗体进行一点交叉<sup>[13]</sup>,交叉操作产生的新个体取代当前进行交叉的抗体。

## 4 数值实验

选取基于不同优先准则的两种启发式<sup>[14]</sup>(MaxR、LongT)算法、免疫克隆选择算法<sup>[15]</sup>(ICSA)与动态克隆选择算法(DCSA)进行比较。MaxR、LongT 各自执行 1 次;ICSA 与 DCSA 运行 20 次;算法的参数设置为:最大迭代次数 Max=2 00,初始群体规模  $N = 40$ 。测试平台 Windows 7, Visual C++ 6.0,电脑主频 2.13 GHz,内存 2 G。进行如下实验:

贵州某企业承接了 50 个项目,每个项目的任务数不同,每个任务使用的资源种类、资源量不同,且有多种执行模式,承接的项目必须在  $T = 150$ (天)中完成,企业一共有 5 种资源,资源的权值分别为 0.25、0.1、0.3、0.1、0.25,单位时间内的资源量分别为 10、10、10、10、10,在  $T$  天内提供的各种资源总量分别为 850、900、800、900、850。由于资源的限制,企业只能选择部分项目执行,其余项目实行外包(其他相关数据省略),数值实验结果如表 1、表 2 所示。

由表 1、表 2 可知,几种算法均获得了问题的可行调度方案。MaxR 和 LongT 虽然都选中 41 个项目,但是选中的项目不同且执行的顺序不同,进而所选中的项目总工期及资源利用不同。原因在于两种算法的优先规则不同。ICSA 和 DCSA 两种算法的结果明显好于前面两种启发式算法,DCSA 的平均项目选中率、平

均资源利用率分别为 92.6%、95.613 3%, 好于 ICSA 的平均项目选中率 91%、平均资源利用率 94.248 3%。DCSA 最好方案中选中的项目有 48 个, 最好方案的项目总工期为 103, 明显好于其他算法。

表 1 50 个项目的实验结果统计表

算法	方案 总工期	项目选 中率/%	平均资源 利用率/%	最好目标 函数值	平均运行 时间/s
MaxR	130	82	91.352 2	0.450	0.45
LongT	113	82	88.363	0.422 7	0.499
ICSA	106	91	94.248 3	0.326	31
DCSA	103	92.6	95.613 3	0.307	16.9

表 2 被选中执行的项目统计表

算法	决策方案 (按项目的调度顺序) (被选中项目编号, 完工时间, 合同时间)
MaxR	(24, 15, 111), (38, 15, 121), (37, 12, 121), (46, 24, 110), (32, 19, 108), (4, 23, 110), (8, 29, 109), (16, 30, 108), (18, 41, 121), (5, 34, 120), (6, 44, 130), (22, 41, 128), (0, 44, 108), (1, 53, 118), (7, 53, 127), (25, 51, 109), (26, 62, 122), (28, 52, 118), (45, 53, 109), (15, 59, 108), (17, 72, 120), (48, 66, 123), (41, 68, 111), (42, 79, 123), (44, 92, 136), (3, 72, 127), (9, 75, 108), (30, 77, 108), (10, 82, 118), (19, 78, 117), (11, 85, 105), (12, 99, 116), (13, 109, 126), (27, 93, 129), (31, 93, 105), (33, 108, 116), (34, 120, 129), (35, 117, 128), (36, 130, 138), (2, 98, 116), (20, 103, 116)
LongT	(24, 13, 111), (38, 14, 121), (37, 13, 121), (46, 23, 110), (4, 21, 110), (5, 30, 120), (32, 24, 108), (6, 41, 130), (28, 29, 118), (22, 34, 128), (25, 36, 109), (26, 47, 122), (8, 43, 109), (16, 47, 108), (18, 60, 121), (7, 52, 127), (0, 38, 108), (1, 56, 118), (45, 60, 109), (48, 64, 123), (41, 66, 111), (42, 76, 123), (44, 90, 136), (15, 67, 108), (17, 79, 120), (30, 72, 108), (9, 77, 108), (3, 75, 127), (10, 84, 118), (19, 80, 117), (27, 87, 129), (11, 87, 105), (12, 100, 116), (14, 113, 129), (13, 108, 126), (2, 89, 116), (47, 95, 121), (20, 96, 116), (21, 109, 126), (31, 97, 105), (33, 110, 116)
ICSA	(41, 11, 111), (30, 10, 108), (19, 8, 117), (42, 20, 123), (31, 11, 105), (32, 17, 108), (33, 24, 116), (8, 24, 109), (20, 24, 116), (46, 29, 110), (0, 29, 108), (34, 38, 129), (39, 27, 132), (15, 32, 108), (38, 46, 121), (21, 39, 26), (40, 39, 116), (16, 42, 108), (11, 45, 105), (25, 49, 109), (9, 50, 108), (45, 49, 109), (2, 52, 116), (1, 56, 118), (35, 56, 128), (37, 62, 121), (3, 63, 127), (29, 61, 115), (43, 62, 122), (4, 66, 110), (22, 67, 128), (10, 72, 118), (17, 75, 120), (48, 75, 123), (28, 76, 118), (12, 80, 116), (23, 80, 137), (24, 86, 111), (13, 89, 126), (47, 90, 121), (26, 97, 122), (27, 103, 129), (18, 103, 121), (44, 104, 136), (5, 106, 120), (36, 103, 138)
DCSA	(16, 10, 108), (45, 11, 109), (29, 7, 115), (28, 13, 118), (31, 14, 105), (46, 21, 110), (19, 17, 117), (38, 24, 121), (11, 22, 105), (32, 29, 108), (30, 28, 108), (15, 29, 108), (24, 35, 111), (12, 43, 116), (22, 38, 128), (14, 56, 129), (33, 43, 116), (37, 48, 121), (39, 42, 132), (9, 46, 108), (41, 51, 111), (20, 51, 116), (0, 54, 108), (4, 58, 110), (25, 59, 109), (23, 58, 137), (1, 64, 118), (3, 73, 127), (5, 67, 120), (2, 67, 116), (8, 76, 109), (10, 84, 118), (48, 71, 123), (40, 75, 116), (7, 79, 127), (26, 84, 122), (17, 83, 120), (27, 93, 129), (18, 91, 121), (47, 88, 121), (49, 95, 130), (21, 93, 126), (13, 96, 126), (34, 102, 129), (6, 103, 130), (42, 102, 123), (35, 102, 128), (43, 102, 122)

产生以上结果的原因在于 MaxR、LongT 的优先规则确定, 则项目的调度顺序和任务的执行顺序就确定, 对于复杂的问题, 很难获得较好解。而 ICSA 和 DCSA

是随机搜索算法, 项目的调度顺序和任务的执行顺序是拓扑排序生成的, 其顺序有多种, 因而有多个可行解。从两种智能算法的平均搜索曲线(见图 1)可知, DCSA 的搜索效果好于 ICSA, 主要原因在于 DCSA 中动态分配记忆池和一般抗体群的规模, 动态克隆、自适应变异、交叉等模块, 将全局搜索、局部搜索、随机搜索及基于概率的非确定性搜索有机结合, 增加了群体的多样性。从表 1 还可知, ICSA 和 DCSA 两种算法 20 次独立的平均运行时间不同, DCSA 的执行效率明显高于 ICSA。

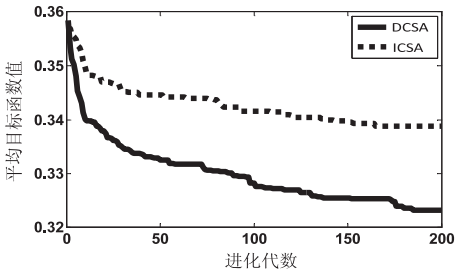


图 1 算法的平均搜索曲线比较

5 结束语

针对多执行模式多项目多任务选择计划问题, 文中建立了多执行模式多项目多任务选择计划模型, 同时设计了动态克隆选择算法对其求解。数值实验结果说明, 设计的算法能较好地得出问题的决策方案。由于文中的问题研究较少且复杂, 还需进一步的完善。

参考文献:

[1] Bouleimen K, Lecocq H. A new efficient simulated annealing algorithm for the resource - constrained project scheduling problem and its multiple mode version[J]. European Journal of Operational Research, 2003, 149(2): 268-281.

[2] 何正文, 刘入境, 徐渝. 基于现金流均衡目标的多模式项目调度问题研究[J]. 管理工程学报, 2011, 25(2): 124-130.

[3] Kolisch R, Hartmann S. Experimental investigation of heuristics for resource - constrained project scheduling; an update [J]. European Journal of Operational Research, 2006, 174(1): 23-37.

[4] 王为新, 李原, 张开富. 基于遗传算法的多模式资源约束项目调度问题研究[J]. 计算机应用研究, 2007, 24(1): 72-74.

[5] 喻瑛. 多模式资源受限项目调度问题的混合遗传算法[J]. 东南大学学报: 自然科学版, 2008, 38(4): 736-740.

[6] 潘晓英, 刘芳, 焦李成. 多执行模式项目调度问题的克隆选择优化[J]. 模式识别与人工智能, 2008, 21(3): 303-309.

[7] Chiang Chuan-Wen, Huang Yuqing, Wang Wen-Yen. Ant



据量足够大时,各参数的训练趋于收敛状态,所以最终表现为检测概率趋于稳定值。此外,采用 CHMM 方法明显比采用 DHMM 方法具有更高的检测概率,这是由于 CHMM 方法克服了 DHMM 方法存在的量化误差问题,使训练和匹配都更加准确。

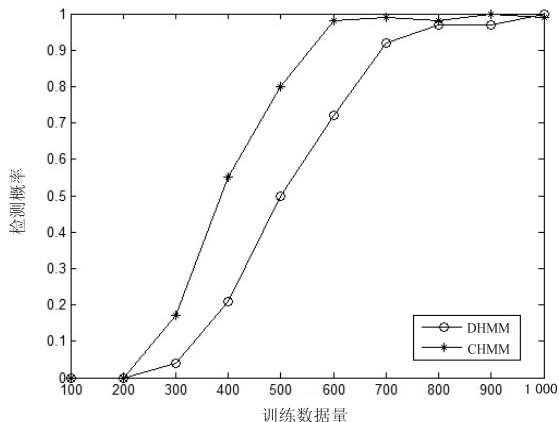


图 5 不同训练数据情况下的检测概率值

## 4 结束语

文中首次将 CHMM 引入频谱检测,把 CHMM 这种机器学习算法和认知无线电中的频谱检测进行有机结合,充分利用了 CHMM 能够处理连续数据的特点,通过  $S_U$  采集  $P_U$  的能量值组成能量向量对 CHMM 进行训练,将当前采集到的待测矩阵输入到训练好的多个模型中进行匹配,似然概率最大的即为当前信道的状态。

### 参考文献:

- [1] 张平,冯志勇. 认知无线网络[M]. 北京:科学出版社,2010.
- [2] Minho J O, Han Longzhe, Kim D, et al. Selfish attacks and detection in cognitive radio Ad-Hoc networks[J]. IEEE Network, 2013, 27(3): 46-50.
- [3] 赵知劲,郑仕铨,尚俊娜. 认知无线电技术[M]. 北京:科学出版社,2013.

(上接第 63 页)

- colony optimization with parameter adaptation for multi-mode resource-constrained project scheduling[J]. Journal of Intelligent & Fuzzy Systems, 2008, 19: 345-358.
- [8] 周蓉,叶春明,谢阳,等. 基于混沌粒子群算法的多目标多执行模式项目调度问题研究[J]. 计算机科学, 2013, 40(4): 259-262.
- [9] 郭研,李南,李兴森. 多模式多资源均衡及基于动态种群的多目标微粒群算法[J]. 控制与决策, 2013, 28(1): 131-136.
- [10] 焦李成,杜海峰,刘芳,等. 免疫优化计算、学习与识别[M]. 北京:科学出版社,2006.
- [11] 邓林义,林焰,金朝光,等. 资源约束下多项目调度的拓

出版社,2013.

- [4] 王伟刚,胡海峰. 基于压缩感知的协作频谱检测[J]. 计算机技术与发展, 2012, 22(12): 241-244.
- [5] Li Deng, Li Xiao. Machine learning paradigms for speech recognition: an overview[J]. IEEE Transactions on Audio, Speech, and Language Processing, 2013, 21(5): 1060-1089.
- [6] 吴敏,王汝传,蔡涛涛. 机器学习在 P2P 流量检测中的研究[J]. 计算机技术与发展, 2010, 20(11): 133-136.
- [7] 李艺颖,邓皓文,王思齐,等. 基于机器学习和 NetFPGA 的智能高速入侵防御系统[J]. 信息安全, 2014(2): 12-19.
- [8] 陆悠,李伟,罗军舟,等. 一种基于选择性协同学习的网络用户异常行为检测方法[J]. 计算机学报, 2014, 37(1): 28-40.
- [9] Thilina K M, Choi K W, Saquib N, et al. Machine learning techniques for cooperative spectrum sensing in cognitive radio networks[J]. IEEE Journal on Selected Areas in Communications, 2013, 31(11): 2209-2221.
- [10] He Xiaofan, Dai Huaiyu, Ning Peng. HMM-based malicious user detection for robust collaborative spectrum sensing[J]. IEEE Journal on Selected Areas in Communications, 2013, 31(11): 2196-2208.
- [11] Ranjan R, Mitra D. Order estimation of HMM discrete channel model for OFDM systems[C]//Proc of international conference on communications, devices and intelligent systems. [s. l.]: [s. n.], 2012: 41-44.
- [12] 黎洁,刘羽西,李奇越. 基于隐马尔可夫模型的认知无线电频谱切换方法[J]. 电子测量与仪器学报, 2014, 28(1): 69-74.
- [13] Pan Shing-tai, Chen Ching-fa, Tsai Yi-heng. Genetic algorithm on speech recognition by using DHMM[C]//Proc of 7th IEEE conference on industrial electronics and applications. Singapore: IEEE, 2012: 1333-1338.
- [14] Liu Guohai, Jiang Xingke, Mei Congli. Soft-sensing modeling method based on continuous hidden Markov model for microbial fermentation process[C]//Proc of Chinese control and decision conference. [s. l.]: [s. n.], 2010: 1106-1110.
- 扑优化方法[J]. 系统仿真学报, 2007, 19(16): 3846-3849.
- [12] Bector F F. Resource-constrained project scheduling by simulated annealing[J]. International Journal of Production Research, 1996, 34: 2335-2351.
- [13] Hartmann S. A competitive genetic algorithm for resource-constrained project scheduling[J]. Naval Research Logistics, 1998, 45: 733-750.
- [14] 白思俊. 资源有限的网络计划与启发式优化方法及其评价与选择: 启发式优化方法综述[J]. 中国管理科学, 1993, 1(2): 30-38.
- [15] 夏林丽,雷宏. 多项目多任务选择计划模型及其智能决策[J]. 计算机应用, 2012, 32(5): 1316-1319.

# 多模式多项目选择计划的动态克隆选择算法

作者：[夏林丽](#)，[雷宏](#)，[苏显方](#)，[XIA Lin-li](#)，[LEI Hong](#)，[SU Xian-fang](#)  
作者单位：[夏林丽, 雷宏, XIA Lin-li, LEI Hong \(铜仁学院 数学与计算机科学系, 贵州 铜仁, 554300\)](#)  
，[苏显方, SU Xian-fang \(湖南大学 工商管理学院, 湖南 长沙, 410012\)](#)  
刊名：[计算机技术与发展](#)[ISTIC](#)  
英文刊名：[Computer Technology and Development](#)  
年，卷(期)：2015(6)

引用本文格式：[夏林丽](#). [雷宏](#). [苏显方](#). [XIA Lin-li](#). [LEI Hong](#). [SU Xian-fang](#) [多模式多项目选择计划的动态克隆选择算法](#)[期刊论文]-[计算机技术与发展](#) 2015(6)