第35卷 第23期 计 算 机 工 程 2009年12月
Vol.35 No.23 Computer Engineering December 2009
·人工智能及识别技术· 文章编号：1000—3428(2009)23—0194—04 文献标识码：A 中图分类号：TP312

一种改进的树突状细胞算法

杨晨旭[1]，吴耿锋[1]，胡 珉[2]
(1. 上海大学计算机工程与科学学院，上海 200072；2. 上海大学悉尼工商学院，上海 201800)

摘 要：针对危险状态识别问题，提出一种改进的树突状细胞算法(IDCA)。在原算法的基础上引入“时间窗”、未成熟DC尽快成熟、衰减因子等概念与策略，提高算法的响应速度和识别精度，解决传统树突状细胞算法一遍运行可能无法评价输入序列末端抗原的问题。在UCI标准数据集上的对比实验证明了IDCA算法的有效性。

关键词：免疫算法；危险理论；树突状细胞算法；时间窗；衰减因子

Improved Dendritic Cells Algorithm

YANG Chen-xu[1], WU Geng-feng[1], HU Min[2]
(1. School of Computer Engineering & Science, Shanghai University, Shanghai 200072;
2. Sydney Institute of Language & Commerce, Shanghai University, Shanghai 201800)

【Abstract】Aiming at the detect the danger status, this paper proposes an Improved Dendritic Cell Algorithm(IDCA). It adopts the idea of “time window”, takes the strategy of maturing as soon as possible to the immature DCs, and introduces the attenuation factor based on the original algorithm to achieve the advantages of short response time and higher recognition precision. The problem that DCA may not be able to evaluate the antigens located at the end of input time sequence if only running algorithm once is also solved. The experiments on UCI dataset prove that IDCA is really available.

【Key words】immune algorithm; danger theory; Dendritic Cell Algorithm(DCA); time window; attenuation factor

1 概述

在目前人工免疫系统的算法研究中，以负选择算法、克隆选择算法和免疫网络算法最具代表性，而后产生了许多与之相关的改进、创新与应用。但是，正如Aickelin等人指出的，现有的大多数人工免疫研究成果都是基于20世纪70年代至80年代生物免疫研究成果的产物[1]。近10年来，生物学界对生物免疫机制的研究又有了突飞猛进的发展。其中最具代表性的就是Polly Matzinger在1994年提出的生物免疫危险理论模式[2]，开创了一种有别于著名的“自我-非我”的全新免疫应答模式——“危险理论”，并由英国诺丁汉大学的Aickelin教授等人首次将其引入人工免疫系统[3]。同时，该小组的Julie Greensmith等人在2005年第4届国际人工免疫学会议上提出以一种基于“危险理论”的全新算法——树突状细胞算法(Dendritic Cell Algorithm, DCA)[4]，并应用于入侵检测系统中。

从外界环境摄取复杂抗原并将其表达在自身表面以被淋巴细胞识别的过程就是抗原提呈，树突状细胞(Dendritic Cell, DC)就是目前所知功能最强大的专职抗原提呈细胞。而DCA算法就是从生物免疫系统中的抗原提呈的角度出发，对输入抗原抽象出“输入信号”(摄取抗原)进行计算得到“输出信号”(抗原表达)，由此得到抗原的“危险程度”，再根据预先设定的阈值做出评价。相对于传统的免疫算法，DCA算法具有简单、快速、无需大量训练样本等优点，开创了免疫算法的一个新思路。但标准DCA算法也存在着一些局限性。本文结合数据挖掘与免疫学原理，对DCA进行了改进，提出了一种改进的DCA算法IDCA(Improved DCA)，并用实验验证了IDCA的有效性。

2 标准DCA算法

2.1 DCA算法的基本原理与定义

DC在生物免疫过程中共有3种状态，即未成熟DC(iDC)、亚成熟DC(smDC)和成熟DC(mDC)。机体组织产生iDC，收集和感知环境抗原信息，当iDC感知到的信号是细胞正常死亡产生的信号(Safe Signal)，则转换到亚成熟状态；如果iDC感知到病原体相关模式(PAMPs)或细胞非正常死亡产生的信号(Danger Signal)，则转换到成熟状态，该过程如图1所示。

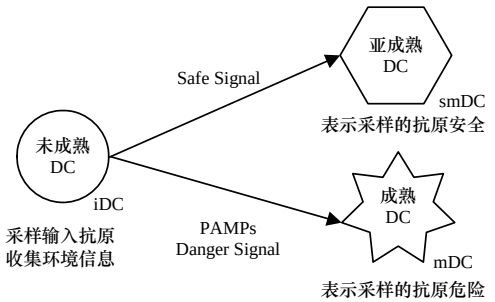


图1 DC状态转换过程

DCA算法抽象了一个与生物免疫学上的DC相对应的数据结构，这里将其称之为“DC”(下文提到的DC如无特别指

基金项目：国家自然科学基金资助项目(50778109)；上海市科技攻关计划基金资助项目(08511501702)；上海市重点学科建设基金资助项目(J50103)

作者简介：杨晨旭(1983－)，男，硕士研究生，主研方向：智能信息处理；吴耿锋，教授、博士生导师；胡 珉，副教授、博士

收稿日期：2009-07-08 **E-mail：**yangchenxu@hotmail.com

出，均指算法上的概念)，其中记录了曾经采样过的抗原、输入输出信号、成熟度阈值和当前 DC 的状态等信息。DCA 算法的基本原理就是仿真了在生物免疫中 DC 状态转换的过程，使用 3 种输出信号来描述 DC 的状态：共刺激分子浓度 csm(concentration of costimulatory molecules)是判断 DC 是否需要进行状态转换的参数；亚成熟 DC 因子 semi(smDC cytokines)是判断 DC“安全程度”的依据；成熟 DC 因子 mat(mDC cytokines)是判断 DC“危险程度”的依据。DCA 中的“采样”是指对某一个抗原提取输入信号 PAMPs, Danger Signal 和 Safe Signal，再通过权值公式和权值矩阵的计算得到输出信号并累加的过程。当 DC 的输出信号 csm 达到某一阈值时，则认为该 DC 达到状态转换条件。然后判断另外两个输出信号 semi 和 mat，如果 semi>mat 则认为 DC 进入“亚成熟”状态，否则进入“成熟”状态。

定义 1(权值公式和权值矩阵) 由免疫学家通过对生物免疫中的 DC 研究，给出的各个输入信号对不同输出信号的影响，其中的权值可以根据实际应用进行调整。文献[5]中给出的权值公式见式(1)：

$$O_{ip}=\sum_{i}\frac{\sum_{j=0}^{2}W_{jp}S_{ij}}{\sum_{j=0}^{2}\left|W_{jp}\right|} \tag{1}$$

其中，i 表示采样抗原的序列位置；j 分别代表 3 种输入信号；p 分别代表 3 种输出信号；O_{ip} 表示输入序列中第 i 个抗原的输出信号强度；W_{jp} 表示对应的权值(见表 1)；S_{ij} 表示第 i 个抗原的输入信号，具体含义见表 1 列出的权值矩阵。

表 1　权值矩阵

W_{jp}	PAMPs (j=0)	Danger Signal (j=1)	Safe Signal (j=2)
csm (p=0)	2	1	2
semi(p=1)	0	0	2
mat(p=2)	2	1	-2

从权值矩阵可以看出，输入输出信号之间的相互影响关系：PAMPs 影响 csm, mat；Danger Signal 影响 csm, mat；Safe Signal 影响 csm, semi 和 mat。所以当根据实际应用调整权值矩阵时同样要满足以上条件，输入输出信号作用关系如图 2 所示。

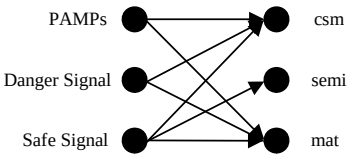


图 2　输入输出信号作用关系

定义 2(上下文环境成熟抗原值 *mcav*) *mcav*(mature context antigen value) 用来评价一个抗原所处环境的成熟程度。由上文对成熟与亚成熟 DC 的描述，*mcav* 也表示抗原的危险程度，其值可通过式(2)计算获得。

$$mcav=\frac{o_1}{o_1+o_2} \tag{2}$$

其中，o_1 是参与该抗原采样的 DC 集合中标识为“成熟”的数量；o_2 是标识为“亚成熟”的数量。

2.2　标准DCA算法流程

由于标准 DCA 算法让每个 DC 在 DC 池中以一定的概率参与对当前抗原的采样，使得每个 DC 随着输入抗原的增加而逐渐成熟，因此算法将后续抗原的影响计入对当前抗原的评价，处于类别转换边界上的抗原容易被误识别。而且，对排在输入序列末端的抗原，算法运行一遍的时候可能无法对这些抗原做出评价，必须进行多次迭代。然而从算法思想来看，DCA 算法并不是一个通过多次迭代逐渐逼近最优解的算法，即算法的多次迭代并没有有效提高算法的识别精度，而仅仅是为了解决能够对所有抗原的评价问题。另外，标准 DCA 算法采取保留所有成熟 DC，迭代完成时统计 *mcav* 的方法，在输入抗原数量巨大，特别是数据流的情况下，除了需要维护大量的成熟 DC，也无法直接对数据流状态下的输入抗原进行评价。所以，标准 DCA 算法不适合大数据量特别是数据流输入的环境应用。针对以上这些局限性，本文提出了改进的 DCA 算法 IDCA。

Julie Greensmith 等人提出的标准 DCA 算法流程如图 3 所示。

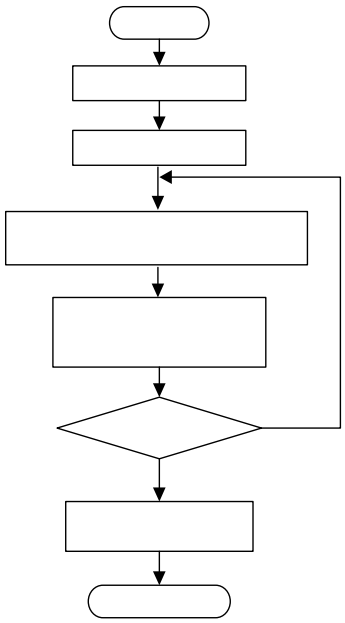


图 3　标准 DCA 算法流程

3　改进的DCA算法IDCA

3.1　IDCA算法的流程

IDCA 算法在 DCA 算法的基础上，引入了一些新的概念：

定义 3(时间窗) 输入抗原序列中，当前正在处理的抗原位置之前的部分连续抗原所形成的一个较短序列，称为时间窗。

定义 4(衰减因子) 0~1 的一个小数，衡量其他抗原对当前正在处理的抗原的影响程度，距离当前抗原越远，对其影响越小。

定义 5(未成熟 DC 集合) 所有参与过采样，但未能成熟的 DC 集合，集合中所有 DC 参与下一次采样。

IDCA 算法放弃了从 DC 池中随机选取 DC 对当前抗原采样的方法，而是每次产生一定量的新 DC 进行采样。不同于标准 DCA 算法保留所有成熟 DC，迭代完成后评价所有抗原，IDCA 算法在采样当前抗原以后，所有参与采样的 DC 立即评价该抗原，统计得出 *mcav*，删除成熟 DC，仅保留未成熟的 DC 进入未成熟 DC 集合，而且集合中的 DC 必定参与下次抗原采样，使其能够尽快成熟。为了获得更快的响应速度，IDCA 算法对 DC 采样的范围进行了限制，仅在“时间窗”的范围中采样，并且引入“衰减因子”，进一步减小在时间序列上处于较远的抗原对当前正在处理的抗原的影响，以提高 DC 输出信号对当前抗原的响应速度。

IDCA 算法流程如图 4 所示。

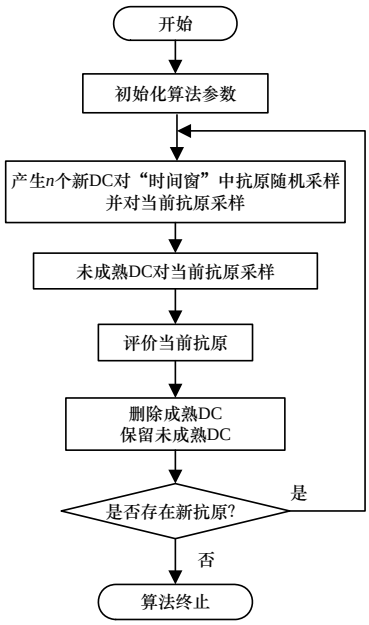


图 4　IDCA 算法流程

IDCA 算法的具体描述如下:

Step1 初始化算法参数，抽象出输入信号；

Step2 创建空的未成熟 DC 集合；

Step3 获取一个输入抗原，生成 *n* 个新 DC，对当前抗原之前“时间窗”中的抗原以逐渐衰减的原则进行随机采样，最后对当前抗原采样；

Step4 未成熟 DC 集合中的所有 DC 采样当前抗原，重新计算输出信号；

Step5 所有参与采样当前抗原的 DC 进行状态设置，全部标记为“亚成熟”或“成熟”，并计算 *mcav*，评价抗原；

Step6 删除所有已成熟 DC，把未成熟的 DC 加入未成熟 DC 集合，以备参与下一次抗原采样；

Step7 判断是否存在新抗原，是则转 Step3，否则转 Step8；

Step8 算法终止。

3.2　IDCA算法中几个关键技术的实现

3.2.1　DC采样抗原的选择策略

IDCA 算法引入了数据流挖掘中“时间窗”(或“滑动窗口”)的概念来限制新 DC 对抗原的采样范围。对每个当前正在处理的抗原产生 *n* 个新 DC。为每个新产生的 DC 在“时间窗”中随机选择部分抗原进行采样计算。然后采样当前正在处理的抗原，使得 *n* 个新 DC 通过不同的抗原采样序列来评价当前正在处理的抗原。采样抗原的选择策略如图 5 所示。

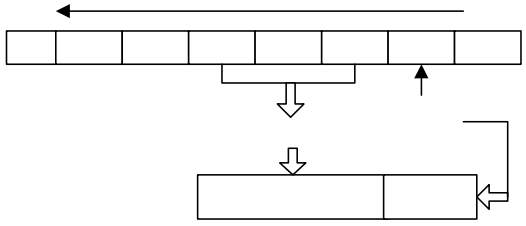


图 5　新 DC 第 1 次采样抗原选择

设“时间窗”长度为 *K*，那么 IDCA 算法从当前处理的抗原位置向前看 *K* 个，以历史输入抗原作为当前抗原评价依据。当 *K* 取值越大，采样抗原可选范围就越大，则增强了抗原的多样性与相互验证的机会，但是可能受到早期环境的干扰也越多；而 *K* 取值越小，采样抗原可选范围越小，早期环境产生的干扰越小，但是对当前处理抗原附近的噪声点越敏感。所以，*K* 取一个合适值是十分重要的。

3.2.2　促使未成熟DC尽快成熟策略

由于选择抗原的随机性，某些新 DC 可能采样的抗原个数偏少，未能达到成熟条件，从而进入未成熟 DC 集合。未成熟 DC 集合中的 DC 全部参与下一个输入抗原的采样，使得集合中的DC能够尽快成熟,从而尽早从集合中删除。IDCA 算法采用的这种尽快成熟策略，减少系统资源开销，适应大数据量环境的应用。此外，保留未成熟 DC 集合的机制使得 DC 不仅仅局限于采样“时间窗”中的抗原，而有机会采样更“远”的抗原，获得更多“相互验证”的机会，提高了算法对噪声数据的鲁棒性。

3.2.3　引入衰减因子加速输出信号的响应速度

为了进一步提高 DCA 算法对环境状态在“安全”与“危险”之间变换时输出信号的响应速度，IDCA 算法引入了衰减因子 α，进一步减小在时间序列上距离当前正在处理的抗原较远的抗原对当前状态的影响。如果被选择的采样抗原简单地按照抗原输入顺序排序，或者按照其对当前抗原影响程度排序，可以采用一个正态分布函数作为衰减因子，如式(3)所示，产生一个 0~1 随距离增大而减小的影响因子。

$$\alpha = \mathrm{e}^{-\frac{(x-\mu)^2}{\sigma^2}} \tag{3}$$

其中，x 是正在计算的抗原位置；μ 是该 DC 选择的采样抗原总数；σ 为控制因子的衰减程度。在对每个被采样抗原计算输出信号以后乘以相应的衰减因子，然后累加计算，得到最后的输出信号。引入衰减因子使算法能更快地体现新环境下的状态。

3.2.4　抗原的评价

在抗原评价方面，IDCA 算法对当前抗原采样完成以后立即计算获得的 *mcav* 做出评价。必须指出的是，无论 DC 成熟与否，全都参与 *mcav* 的计算。这样做不仅能够及时评价当前抗原，而且能尽快删除成熟 DC，释放资源。更重要的是未成熟 DC 参与评价当前抗原的机制解决了 DCA 算法中评价处于输入序列末端的抗原时，参与评价的成熟 DC 不足的情况，使得 IDCA 算法不需要多次迭代，只需运行“一遍”，更体现了 DCA 算法的优势。

4　实验与结果分析

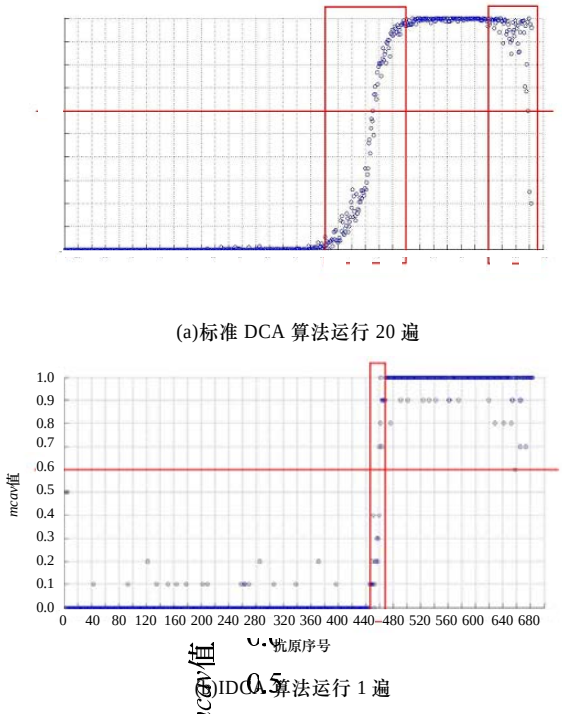
本文选用了文献[4]中提到的 UCI 著名的 Breast Cancer 数据集[6]作为实验数据，把 IDCA 与标准 DCA 算法进行了对比实验。该数据集有 700 个数据，去除某些特征不完整的数据以后，留下 683 个可供实验数据。其中，每一个数据是一个 11 维向量，第一维表示数据编号，最后一维标识所属类别，中间 9 维表示数据特征，描述了一个潜在乳腺癌患者的各方面检测结果。在 683 个数据中，正向数据(标识为良性)有 444 个，余下的 239 个数据为反向数据(标识为恶性)。每个数据选取中间的 9 维特征形成的向量，作为一个“抗原”，所有数据经过归一化处理。

实验 1 IDCA 和 DCA 在一次环境状态转换下的性能比较

首先输入 444 个正向数据，接着输入 239 个反向数据，即让算法经历一次环境状态转换。

参数设置：*csm* 成熟阈值取 5，权值矩阵如表 1 所示，时间窗长度取 20，每个抗原用 10 个 DC 参与采样，IDCA 算法中对每个抗原产生 10 个新 DC 采样，*mcav* 取 0.6 作为危险阈值，不加入衰减因子。

mcav 值随着抗原序列的分布图如图 6 所示(横坐标表示抗原输入序列，纵坐标是 *mcav* 的值)，其中，图 6(a)为标准 DCA 运行 20 遍的结果，图 6(b)为 IDCA 运行 1 遍的结果。



(a)标准 DCA 算法运行 20 遍

(b)IDCA 算法运行 1 遍

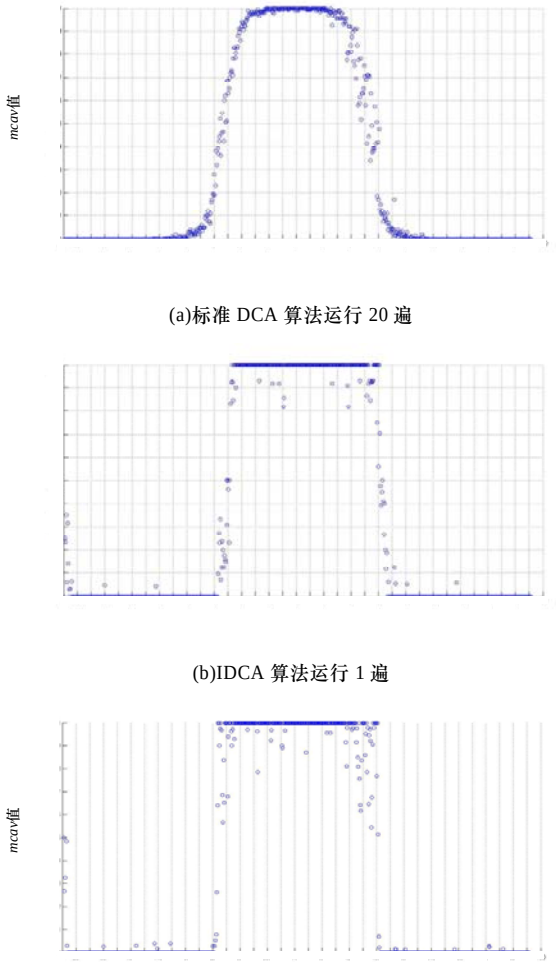
图 6　算法对环境一次切换响应情况比较

从图 6(a)可以看到，实验结果与文献[4]中得到的结果基本一致，平均识别精度在 97.6%左右。根据输入数据的所属类别，应在横坐标为 444 的时候从正向数据转换到反向数据。标准 DCA 从第 380 个数据开始转换，到 500 左右转换完成，有 120 个左右数据的过渡。在第 620 个左右数据以后，*mcav* 开始不稳定，证明了前文提到的由于序列末尾处的后续抗原不足，导致参与采样的 DC 没有成熟而没有参与抗原评价，*mcav* 出现不稳定情况。而图 6(b)表明，IDCA 仅需运行 1 遍，且只经历了大约 20 个数据即完成了转换，由于 2 种算法仅仅经历了一次环境状态的转换，标准 DCA 算法的识别精度比 IDCA 算法略高，达到了 99.1%左右，IDCA 算法识别精度为 97.5%，但是相对于标准 DCA 算法，IDCA 算法 *mcav* 值的切换速度有很大提高。

实验 2 IDCA 和 DCA 在 2 次、4 次环境状态转换下的性能比较

将正向 444 个数据一分为二，输入顺序为正向数据 222 个，反向数据 239 个，正向数据 222 个，让算法经历 2 次环境状态转换，算法参数设置同实验 1。结果如图 7 所示。

标准 DCA 经历 2 次类别切换以后，平均识别精度相对于实验 1 的一次切换骤降到 78.2%左右。而 IDCA 的平均识别精度在 96.9%左右，已大大优于标准 DCA 算法。另外，将本实验正向数据前 222 个再一分为二，反向数据也一分为二，形成输入序列 111(正)-119(反)-111(正)-120(反)-222(正)，即经历 4 次环境状态切换，保持算法参数不变，标准 DCA 算法识别精度下降到 65.0%，所有 DC 被标识为“安全”，结果不可接受。IDCA 算法识别精度虽有下降但仍然达到 94.7%左右。实验表明，环境状态切换越频繁，对算法的识别精度影响越大，而在同样条件下，IDCA 算法受到的影响较小，因此 IDCA 更适合应用于频繁转换的环境，更具有实用价值。



(a)标准 DCA 算法运行 20 遍

(b)IDCA 算法运行 1 遍

(c)IDCA 加入衰减因子

图 7　经历 2 次环境切换算法响应情况

实验 3 衰减因子 α 对 IDCA 的影响

在上述实验 2 的基础上，使用相同的参数，但加入衰减因子 α (见式(3))，其中，σ 取 4，实验结果如图 7(c)所示。可见加入衰减因子后，相对于图 7(b)中未加入衰减因子时的 IDCA 算法具有更快的响应速度，2 类数据边界处的 *mcav* 值区分得更加清晰，识别精度有一定的提高，实验表明衰减因子的加入确实有效提高了 IDCA 算法的识别精度。

上述实验的详细结果综合如表 2 所示，其中的 TPR(True Positive Rate)是指正向数据识别精度，即正向数据确实识别为正向的比例，TNR(True Negative Rate)是指反向数据识别精度，即反向数据确实识别为反向的比例。

表 2　算法识别精度实验结果汇总表　(%)

算法	切换次数								
	1			2			4		
	精度	TPR	TNR	精度	TPR	TNR	精度	TPR	TNR
标准 DCA	99.12	99.5	98.3	78.2	100.0	37.7	65.0	100.0	0.0
IDCA(无衰减因子)	97.5	100.0	92.9	96.9	99.1	92.9	94.7	97.7	89.1
IDCA(加入衰减因子)	97.8	100.0	93.7	97.4	99.3	93.7	95.6	98.4	90.4

(下转第 200 页)