

树突细胞算法及其理论研究

方贤进 王丽 康佳 刘佳

(安徽理工大学计算机学院 淮南 232007)

摘要 树突细胞算法(DCA)是受先天性免疫系统中树突细胞(DCs)功能的启发而开发的算法,它已被成功运用于许多计算机安全相关领域。但是对 DCA 理论方面的分析工作很少,对算法大多数理论方面的研究也较少出现。而其它的人工免疫算法如负选择算法、克隆选择算法在理论方面的研究工作却出现在很多文献中。因此对 DCA 算法进行相似的理论分析,确定算法的运行时间复杂度,揭示其它算法的属性就显得非常重要。根据算法执行的 3 个阶段,通过引入 3 个运行时间变量实现对 DCA 算法的理论分析。标准 DCA 算法取得的运行时间复杂度下界为 $\Omega(n)$,而在最坏情况下的时间复杂度为 $O(n^2)$ 。另外,如果利用“分片”方法实现 DCA 的在线分析组件,则算法的运行时间复杂度可以改进为 $O(\max(nN, n\delta))$ 。

关键词 树突细胞算法,形式化描述,运行时间复杂度

中图分类号 TP301 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2015.2.028

On Dendritic Cell Algorithm and its Theoretical Investigation

FANG Xian-jin WANG Li KANG Jia LIU Jia

(School of Computer, Anhui University of Science and Technology, Huainan 232007, China)

Abstract Dendritic cell algorithm (DCA) is inspired by functions of the dendritic cells (DCs) of the innate immune system, and has been successfully applied to numerous security-related problems. However, theoretical analysis of the DCA has barely been performed, and most theoretical aspects of the algorithm have not yet been revealed. Other immune inspired algorithms, such as negative and clonal selection algorithms, are theoretically presented in many literatures. As a result, it is important to conduct a similar theoretical analysis of the DCA, and determine its runtime complexity and other algorithmic properties in line with other artificial immune algorithms. Theoretical analysis was implemented via introduction of three runtime variables in terms of three phases of the algorithm. The standard DCA achieves a lower bound of $\Omega(n)$ runtime complexity and an upper bound of $O(n^2)$ runtime complexity under the worst case. In addition, the algorithm's runtime complexity can be improved to $O(\max(nN, n\delta))$ by using segmentation approach for online analysis component.

Keywords Dendritic cells algorithm, Formal description, Runtime complexity

1 引言

树突细胞算法(Dendritic Cells Algorithm, DCA)^[1]属于第二代人工免疫系统(Artificial Immune Systems, AIS)^[2],是受先天性免疫系统中的一种抗原呈递细胞树突细胞(Dendritic Cell, DC)功能的启发,将 DC 生态学行为的抽象模型作为算法开发的基础。DCA 已成功用于许多计算机安全相关领域,如端口扫描检测^[3]、僵尸网络检测^[4]、机器人安全分类器^[5]等。这些应用的结果显示 DCA 不仅在分类检测率方面具有好的性能,而且引比其它系统(如 SOMs)减少了误警率^[6]。

在算法理论分析方面,Stibor 等人对 DCA 进行了几何学分析^[7],这方面的工作后来被扩展在 Oates 的学位论文的研究工作中^[8];Feng Gu 与 Julie Greensmith 等人利用集合论和数学函数定义了确定性 DCA 算法的数据结构、操作过程,并进一步对算法进行了形式化描述,给出了算法实现的伪代码表示^[9,10]。为提高 DCA 算法的检测效率和性能,将基于分片

(segmentation)思想的在线分析组件集成到了算法中^[11],对在检测阶段 DC 群体中每个 DC 处理过的序列信息根据数据项数或时长进行分片,然后对这些分片信息进行周期性地、连续地分析。由于所有的分片在检测阶段就被处理,因此改进后的 DCA 算法实现的是在线分析而不是离线分析。

尽管 DCA 已成功应用于计算机安全相关领域,但除了文献[7-11]之外,对该算法的理论研究相对较少,而对其它的 AIS 算法,如 Negative selection 算法和 Clonal selection 算法,在理论方面尤其是运行时间复杂度与算法优化方面的研究工作则较多^[12-16]。因此有必要对 DCA 算法的运行时间复杂度以及其它算法的属性方面进行研究。

2 群体树突细胞行为抽象为信息处理系统模型

根据 DC 的生态学行为,可将基于群体的 DC 的功能抽象为图 1 所示的信息系统模型,其中, System 对应于免疫系统中的组织;事件或进程对应于免疫系统中的外部 Antigens(如

到稿日期:2014-03-09 返修日期:2014-05-11 本文受国家自然科学基金(61240023)资助。

方贤进(1970-),男,博士,教授,主要研究方向为计算智能、网络与信息安全, E-mail: xjfang@aust.edu.cn;王丽(1977-),女,硕士,讲师,主要研究方向为数据挖掘;康佳(1990-),女,硕士生,主要研究方向为数据挖掘;刘佳(1989-),女,硕士生,主要研究方向为网络与信息安全。

virus, bacterium 等)、正常的细胞凋亡 (Apoptosis)、非正常的细胞死亡 (Necrotic)、人体组织中的炎症 (inflammation) 等; 观察得到的全局行为对应于免疫系统中由于外部病原体、细胞凋亡、非正常细胞死亡和炎症等产生的 Danger、Safe、PAMP 和 inflammation 等分子信号 (Signals)。由行为数据与事件或进程组合的时间序列数据由 DC 群体进行处理, 过程包括信号转换 (可认为是 data fusion) 和事件采样, 然后由 DC 群体分析算法计算每个 DC 的输出, 最后执行综合度量计算 (可认为是事件与行为数据的 Correlation) 来对事件进行分类。

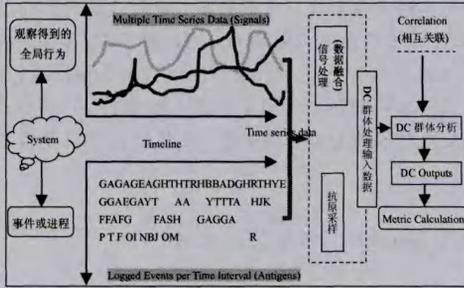


图1 群体DC行为抽象为信息处理系统模型

3 标准 DCA 算法的形式化描述

根据文献[9]对算法进行形式化描述的方法, 首先定义 DCA 算法的数据结构, 包括定义 DC Population 大小为 N , 群体中的每个 DC 细胞包含下列数据结构: $i \in N$ 表示每个 DC 细胞的编号; $lifespan(i) \in R$ 表示第 i 个 DC 细胞的生命期初始值; $K(i) = 0$ 表示第 i 个 DC 细胞的初始信号值; 信号转换权重矩阵 W ; 存储每个 DC 的输出的线性表 lst 等; 然后定义每个 DC 的抗原采样函数、DC 的信号转换函数、DC 的生命期更新函数、DC 的信号值更新函数、DC 的输出值记录函数、抗原类型计数函数、DC 的信号值抽取函数、某一抗原类型 α 的异常值度量函数等。由此得到标准 DCA 算法实现的伪代码描述如图 2 所示。根据伪代码可以看出, DCA 算法由 3 个阶段组成: 初始化阶段 (从第 1 行到第 3 行)、检测阶段 (从第 4 行到第 19 行)、离线分析阶段 (从 20 行到 26 行)。

input: time series data (由抗原和信号组成)

output: antigen types + $K\alpha$ (异常度量系数)

```

1. For each DC object do /* initialization phase */
2.   Call DC initialization;
3. End
4. while input data stream do /* detection phase */
5.   if antigen then
6.     Select a DCi;
7.     Call antigen profile update 函数 H(t, i);
8.   End
9.   if signal then
10.    Call signal transformation 函数 O(t);
11.    For each DC object do
12.      Call lifespan update 函数 F(t, i);
13.      Call signal profile update 函数 G(t, i);
14.      if F(t-1, i) < 0 then
15.        Call DC output record 函数 L(j); /* 输出到 lst 表 */
16.      End
17.    End
18.  End

```

```

19. End
20. while output list do /* offline analysis phase */
21.   For each antigen type  $\alpha$  do
22.     Call antigen counter 函数 C(j,  $\alpha$ );
23.     Call signal profile abstraction 函数 R(j,  $\alpha$ );
24.     Call anomaly metric calculation 函数

$$\beta = \sum_{j=0}^{N-1} C_{\alpha}(j), \gamma = \sum_{j=0}^{N-1} R_{\alpha}(j) \Rightarrow K(\alpha) = \frac{\lambda}{\beta}$$

25.   End
26. End

```

图2 标准 DCA 算法的伪代码实现

4 DCA 算法的运行时间复杂度分析

算法运行时间复杂度分析涉及到被算法执行的主要操作或步骤的次数, 这种分析通常基于逼近理论, 而它的目的是从理论上推导出随着输入规模 n 增长的算法的运行复杂度。

DCA 算法的输入是由 Antigen 和 Signal 组成的时间序列数据。设 p 为输入数据中含有 Antigen 样本的个数, q 是 Antigen 类型的个数。 $N = |\text{Population}|$ 是 DC 细胞群体的大小, 很显然 N 是独立于输入数据规模大小 n 的, 并且 $1 \leq N \leq n$, 因为输入数据要么是 Antigen, 要么是 Signal, 所以在输入数据序列中含有的信号样本的个数是 $n-p$ 。

4.1 标准 DCA 算法的时间复杂度分析

根据图 2 中的算法, 设 $T_1(n), T_2(n), T_3(n)$ 分别表示算法 3 个阶段的运行时间变量, 令 $T(n) = T_1(n) + T_2(n) + T_3(n)$ 为算法总的运行时间。相应于图 2 中的算法, 其所有的原始操作如表 1 所列。

表1 算法中的所有原始操作

# Line	Operation or steps description	Executed Times	phase
1	For each DC object do	N	Initialization
2	DC initialization	N	phase
4	while input data stream do	n	
5	If statement	p	
6	Select DC _i	p	
7	Call antigen profile update	p	
9	If statement	$n-p$	Detection
10	Call signal transformation	$n-p$	phase
11	For each DC object do	$(n-p) * N$	
12	Call lifespan update	$(n-p) * N$	
13	Call signal profile update	$(n-p) * N$	
14	If statement	$(n-p) * N$	
15	Call DC output record	$(n-p) * N$	
20	While output list do	p	
21	For loop	$p * q$	Analysis
22	Call antigen counter	$p * q$	phase
23	Call signal profile abstraction	$p * q$	
24	Call anomaly metric calculation	$p * q$	

初始化阶段, 由于 DC 群体中每个 DC 细胞的初始生命期的值与初始信号值是指定的, 因此 $T_1(n) = N + N = O(N)$ 。

检测阶段的运行时间与输入数据的规模 n 有关, 若抗原数量为 p , 则信号的数量为 $n-p$, 群体的大小为 N 。检测阶段的运行时间变量的值可按下式计算:

$$T_2(n) = n + 3p + 2(n-p) + 5(n-p) * N = 3n + 5N * (n-p) + p$$

$$T_2(n) = O(nN) \text{ (if } n \rightarrow +\infty \text{)}$$

分析阶段的运行时间变量涉及到时间序列数据中抗原样本的数量 p 以及抗原类型的数量 q 。在最坏情况下, 如果 $p =$

q ,则在分析阶段的运行时间变量可以按如下计算:

$$\begin{aligned} T_3(n) &= p + 4pq \\ \because 1 \leq q \leq p \leq n \\ \therefore T_3(n) &\leq n + 4n^2 \\ \therefore T_3(n) &= O(n^2) \end{aligned}$$

定理 1 对于输入数据规模大小 n , 标准 DCA 算法的运行时间复杂度上下界分别为 $\Omega(n)$ 和 $O(n^2)$ 。

证明: $T(n) = T_1(n) + T_2(n) + T_3(n)$

$$T(n) = O(N) + O(nN) + O(n^2) \Rightarrow \Omega(n) \leq T(n) \leq O(n^2)$$

这样, 标准 DCA 在最好情况下能获得的运行时间复杂度下界为 $\Omega(n)$, 而在最坏情况下的运行时间复杂度的上界为 $O(n^2)$ 。因此相比较于其它异常检测算法, DCA 算法在处理大数据集的情况是具有竞争力的。

4.2 具有在线分析组件的 DCA 算法的运行时间复杂度分析

很显然, 在图 2 的算法中执行的是离线分析(图 2 中从 20 行到 26 行)。但是在线分析组件是开发一个高效率的入侵检测系统的根本, 它对于获得高的检测速度和检测精度是非常重要的, 而这两个方面也是衡量一个入侵检测系统性能的两个主要指标^[9]。Feng Gu 等人使用分片思想(segmentation)来实现 DCA 算法在线分析组件^[11], 该思想是在算法的检测阶段(图 2 中从 4 行到 19 行)同时执行分析, 由于算法在检测阶段已处理的信息都输出到线性表 lst 中(图 2 中行 15), 根据时长(Time Based Segmentation, TBS)或采集的抗原数(Antigen Based Segmentation, ABS)将存储在表 lst 中的输出分成很多片(slice), 然后对这些片执行周期性的、连续的分析, 因而在检测阶段就能不断地在线检测入侵。

假如使用 ABS 方法来实现 DCA 的在线分析组件, 设 δ 是分片大小且 $1 \leq \delta \leq n$ 。那么一旦输出线性表 lst 的大小达到 δ 时, 一个分片就产生了, 接着对在输出线性表中当前批次的已处理信息执行分析。很显然, 分片产生的数量为 $\lceil n/\delta \rceil$, 这些分片被标以序号 $\{1, 2, \dots, k, \dots, \lceil n/\delta \rceil\}$, 设 p_k 与 q_k 分别为在第 k 个分片中抗原的数量和抗原类型的数量, 根据 4.1 节内容, 在分析阶段对每个分片进行分析的运行时间复杂度为:

$$\begin{aligned} T_3^k(n) &= p_k + 4p_kq_k \\ \because 1 \leq q_k \leq p_k \leq \delta \\ \therefore T_3^k(n) &\leq \delta + 4\delta^2 \\ \therefore T_3^k(n) &= O(\delta^2) \end{aligned}$$

定理 2 具有在线分析组件的 DCA 算法的运行时间复杂度为 $O(\text{Max}(nN, n\delta))$, 此处输入数据规模的大小为 n , DC 群体的大小为 N , 分片的大小为 δ 。

证明: $T(n) = T_1(n) + T_2(n) + \sum_{k=1}^{\lceil n/\delta \rceil} T_3^k(n)$

$$T(n) = O(N) + O(nN) + \sum_{k=1}^{\lceil n/\delta \rceil} T_3^k(n)$$

$$T(n) \leq O(N) + O(nN) + \lceil n/\delta \rceil * O(\delta^2)$$

$$\because 1 \leq N \leq n, 1 \leq \delta \leq n$$

$$\therefore T(n) = O(N) + O(nN) + O(n\delta)$$

$$T(n) = O(\text{Max}(nN, n\delta))$$

因此在线分析组件的引入改变了 DCA 算法的运行时间复杂度, 使其从 $O(n^2)$ 降低为 $O(\text{Max}(nN, n\delta))$ 。若 $N=n$ 或 $\delta=n$, 则算法的时间复杂度为平方级即为 $O(n^2)$; 若 $N \ll n$ 且 $\delta \ll n$, 则算法的时间复杂度为线性即为 $O(n)$ 。该结果对实现在线入侵检测任务非常重要, 因为使用分片方法实现在线分析组件的 DCA 算法比标准 DCA 算法能产生更好的结果。

结束语 本文给出了基于群体树突细胞行为的信息处理系统的抽象模型, 根据算法的形式化描述, 推算出标准 DCA 在最坏情况下的时间复杂度为平方级 $O(n^2)$ 。研究引入“分片”方法来实现 DCA 的在线分析组件, 能改变算法的运行时间复杂度, 在某些情况下, 其时间复杂度能近似达到线性级 $O(n)$, 此结果对实现在线检测任务非常重要。

本文工作独立于算法具体应用, 因此对算法未来的开发具有指导意义。DCA 算法未来的开发目标之一就是将其转换成一个自动的、自适应的、在线的检测系统, 而实现这个目标必须满足高计算效率、检测过程中适应实时场景的要求。

下一步的研究工作包括: (1) DC 群体中分化为成熟 DC 的数量与 DC 更新频率的关系, 进而研究 DC 群体中每个 DC 生命期值的分布(高斯分析、均匀分布)与 DC 群体中分化为成熟 DC 的数量之间的关系; (2) 被成熟的 DC 处理的抗原数量与输入序列数据大小之间的关系; (3) 每个 DC 产生的活动窗口的影响、DC 群体大小、信号转换权值矩阵与算法检测性能之间的关系。

参考文献

- [1] Greensmith J. The Dendritic Cell Algorithm[D]. Nottingham: University of Nottingham, 2007
- [2] Leandro N, De Castro J T. Artificial Immune Systems; A New Computational Intelligence Approach [M]. London: Springer-Verlag, Inc, 2002
- [3] Greensmith U A. Dendritic Cells for SYN Scan Detection[C]// Proceedings of the Genetic and Evolutionary Computation Conference (GECCO2007). London, UK: ACM Press, 2007: 49-56
- [4] Al-Hammadi Y, Aickelin U, Greensmith J. DCA for Bot Detection[C]// Proceedings of the IEEE World Congress on Computational Intelligence (WCCI 2008). Berlin Heidelberg: Springer-Verlag, 2008: 1807-1816
- [5] Oates R, J G, Aickelin U, et al. The Application of a Dendritic Cell Algorithm to a Robotic Classifier[C]// Proceedings of the 6th International Conference on Artificial Immune (ICARIS 2007). 2007: 204-215
- [6] Greensmith J, Feyereisl J, Aickelin U. The DCA: SOME Comparison A comparative study between two biologically-inspired algorithms[J]. Evolutionary Intelligence, 2008, 1(2): 85-112
- [7] Stibor T R O, Kendall G, et al. Geometrical insights into the dendritic cell algorithm[C]// Proc. of the Genetic and Evolutionary Computation Conference (GECCO). 2009: 1275-1282
- [8] Oates R. The Suitability of the Dendritic Cell Algorithm for Robotic Security Applications[D]. Nottingham: School of Computer Science, University of Nottingham, 2010
- [9] Feng Gu, Julie G, Uwe A. The Dendritic Cell Algorithm for Intrusion Detection[J]. Bio-Inspired Communications and Networking, IGI Global, 2011: 84-102
- [10] Greensmith J, U A. The Deterministic Dendritic Cell Algorithm [C]// Proceedings of the 7th International Conference on Artificial Immune Systems (ICARIS 2008). Phuket, Thailand. Lecture Notes in Computer Science Volume 5132, 2008: 291-302
- [11] Gu Feng, J G, Aickelin U. Integrating Real-Time Analysis With The Dendritic Cell Algorithm Through Segmentation[C]// Proceedings of the Genetic and Evolutionary Computation Conference (GECCO2009). Montreal, Canada, 2009

结束语 目前将通用的需求工程方法应用到 C4ISR 系统需求获取中普遍存在以下问题:1)这些方法不是以军事应用为背景的,由于领域差异,军事人员难以理解和掌握;2) C4ISR 需求涉及宏观、微观两个方面,而一般需求工程方法的开发机制往往直接面向微观层次的设计问题,通常难以从宏观把握 C4ISR 需求,更难以指导 C4ISR 系统的顶层设计;3)更重要的是,C4ISR 系统的策划和研制必须与部队的使命任务紧密联系,从充分发挥武器装备的功能和效能、提高部队作战能力出发,才能获得满足用户需要的需求。

在 C4ISR 体系结构建模方面,虽然已经逐渐认识到能力概念在系统设计、开发中的重要性,但还很少有方法将能力作为核心概念去引导用户建立系统需求模型。

本文旨在以能力概念为核心,提出了一种面向 C4ISR 能力分析的领域建模语言。主要工作为:从 C4ISR 能力概念化入手,定义了 C4ISR 能力的元本体,利用 UML 元层扩展机制定义了领域特定描述语言,并利用 OWL DL 定义了语言的形式语义,使得基础概念推理的模型一致性验证得以进行;采用 3 层概念抽象定义 C4ISR 能力需求模型,为 C4ISR 能力需求复用奠定基础。本文工作具有以下特点:(1)C4ISR 能力概念化的元本体是在体系结构元模型 M3 基础上提出的,对国际流行的 C4ISR 体系结构标准有良好的支持,适用性较广,同时可以很好地支持基于 UML 技术的软件建模与开发。(2)采用 3 层概念抽象来描述系统能力需求,不仅可以通过定义领域特定建模语言来复用领域知识,而且可以运用领域概念化约束规范应用建模行为,检验应用模型的领域适用性和正确性。(3)采用 OWL DL 对概念模型进行形式化描述,给出形式语义,可以实现领域概念和应用概念模型的一致性验证。

参考文献

[1] US Department of Defense. DoD Architecture Framework Version 1.5 (Volume I-II-III) [OL]. 2007. <http://www.defenselink.mil/cio-nii/>

[2] UK Ministry of Defence. The MOD Architectural Framework v1.2 [OL]. 2008. <http://www.modaf.org.uk>

[3] CapDEM CEP Team. Capability Engineering Process Version 1 [R]. Defence Research and Development Canada, Valcartier, 2004

[4] OMG. Unified Profile for the Department of Defense Architecture Framework (DoDAF) and the Ministry of Defence Architecture Framework (MODAF) Specification [OL]. 2008. <http://www.updm.com/index.htm>

[5] OMG. ALert Management Service (ALMAS) Specification (Be-

ta 2) [OL]. 2008. http://www.omg.org/technology/documents/domain_spec_catalog.htm

[6] 金芝. 基于本体的需求自动获取[J]. 计算机学报, 2000, 23: 486-492

[7] Jin Zhi, Bell D, Wilkie F D. Automated requirements elicitation: Combining a model-driven approach with concept reuse [J]. International Journal of Software Engineering and Knowledge Engineering, 2003, 13(1): 53-82

[8] 陆汝钤, 金芝. 从基于知识的软件工程到基于知识的软件工程 [J]. 中国科学 F 辑, 2008, 38(6): 843-863

[9] 舒凤笛, 赵玉柱, 王继喆, 等. 个性化领域知识支持的用户主导需求获取方法 [J]. 计算机研究与发展, 2007, 44(6): 1044-1052

[10] 翟健, 杨秋松, 肖俊超, 等. 一种形式化的组件化软件过程建模方法 [J]. 软件学报, 2011, 22(1): 1-16

[11] 张伟, 梅宏. 一种面向特征的领域模型及其建模过程 [J]. 软件学报, 2003, 14(8): 1345-1356

[12] 王波, 赵海燕, 张伟, 等. 问题驱动的需求捕获中问题分析与解决技术研究 [J]. 计算机研究与发展, 2013, 50(7): 1513-1523

[13] 杨美清, 吕建, 梅宏. 网构软件技术体系: 一种以体系结构为中心的途径 [J]. 中国科学 F 辑, 2008, 38(6): 818-828

[14] 吕建, 马晓星, 陶先平, 等. 面向网构软件的环境驱动模型与支撑技术研究 [J]. 中国科学 F 辑, 2008, 38(6): 864-900

[15] 李德毅. 需求工程——对复杂系统的软件工程的基础研究 [J]. 中国基础科学, 2009, 11(2): 4-8

[16] Wang Jian, He Ke-qing. RGPS: a unified requirements meta-modeling frame for networked software [C] // Proceedings of the 3rd International Workshop on Applications and Advances of Problem Frames at 30th International Conference on Software Engineering. 2008: 29-35

[17] 马国普, 余滨, 沙基昌, 等. 系统能力集成与开发的 SCFP 模型 [J]. 火力与指挥控制, 2008, 33(7): 36-39

[18] 付广胜, 陈洪辉. 基于能力的指挥信息系统需求获取方法 [J]. 火力与指挥控制, 2009, 34(9): 72-75

[19] 谢仲文, 李彤, 代飞, 等. 基于特征组合的软件需求建模 [J]. 计算机学报, 2012, 39(1): 130-141

[20] 王聪, 王智学. UML 活动图的操作语义 [J]. 计算机研究与发展, 2007, 44(10): 1801-1807

[21] 王聪, 王智学. 军事电子信息系统需求模型 [J]. 解放军理工大学学报: 自然科学版, 2008, 9(4): 328-334

[22] 王智学, 董庆超, 陈彬, 等. 基于 UML 模型的 C4ISR 系统能力需求分析与验证 [J]. 系统工程与电子技术, 2009, 31(9): 2167-2171

[23] 王智学, 董庆超, 朱卫星, 等. 一种支持能力需求概念一致性和合理性分析的方法 [J]. 计算机学报, 2013, 36(1): 10-21

(上接第 133 页)

[12] Elberfeld M, J T. Negative selection algorithms on strings with efficient training and linear-time classification [J]. Theoretical Computer Science, 2011, 412(6): 534-542

[13] Zarges C. Rigorous runtime analysis of inversely fitness proportional mutation rates [C] // Proceedings of Parallel Problem Solving from Nature (PPSN). LNCS 5199, 2008: 112-122

[14] Zarges C. On the utility of the population size for inversely fitness proportional mutation rates [C] // Proceedings of the 10th

ACM SIGEVO Workshop on Foundations of Genetic Algorithms (FOGA). 2009: 39-46

[15] Timmis J, Home A, Stibor T, et al. Theoretical advance in artificial immune systems [J]. Theoretical Computer Science, 2008 (403): 11-32

[16] Janse T, Zarges C. Analyzing different variants of immune inspired somatic contiguous hypermutations [J]. Theoretical Computer Science, 2011, 412(6): 517-533