An Immune Artificial Algorithm for Dynamic Optimization Problems: A Case of Study

Victoria S. Aragón, Susana C. Esquivel[1] and Carlos A. Coello Coello[2]

[1] Laboratorio de Investigación y Desarrollo en Inteligencia Computacional[***]
Universidad Nacional de San Luis
Ejército de los Andes 950
(5700) San Luis, ARGENTINA
{vsaragon, esquivel}@unsl.edu.ar
[2] CINVESTAV-IPN (Evolutionary Computation Group)[†]
Departamento de Computación
Av. IPN No. 2508, Col. San Pedro Zacatenco
México D.F. 07300, MÉXICO
ccoello@cs.cinvestav.mx

Abstract. In this paper, we present an algorithm inspired on the T-Cell model of the immune system, i.e., an artificial immune system (AIS). The proposed approach (called DTC) is intented to solve dynamic optimization problems. We realize a first study and we validate our algorithm using dynamic functions taken from the specialized literature. Results are promising when we compare them with respect to the results obtained by five AIS representative of the state of the art in this field.

Keywords: Artificial immune systems, dynamic optimization, heuristics optimization.

1 Introduction

In general, the conditions of an optimization problem changes by one of the following reasons or a combination of both [1]: 1) The objective function changes itself, 2) The constraints change. In this paper the first problematic is addressed.

In recent years, a bio-inspired metaheuristic known as the "artificial immune system" (AIS) has gained popularity in a wide variety of tasks [12]. The AIS is inspired on our natural immune system, which has a number of very interesting features, from a computational point of view, that make it a very good candidate to be modelled in a computer. For example, it is a distributed system, it is fault-tolerant, it has memory, it is able to distinguish between its own components and those which are foreign, and it learns by experience. AISs have been used for solving dynamic optimization problems (see for example [16, 7, 13]), but in most

[***] LIDIC is financed by Universidad Nacional de San Luis and ANPCyT (Agencia Nacional para promover la Ciencia y Tecnología).

[†] The third author acknowledges support from CONACyT project no. 45683-Y.

cases, it has been applied only to global optimization problems. In this paper, we precisely focus on solving dynamic optimization problems with an AIS that we have proposed, and which we believe that can be a viable alternative for solving these kind of problems.

The remainder of the paper is organized as follows. In Section 2, we describe the benchmark used to generate the dynamic functions. Section 3 describes the existing AIS for dynamic optimization. In Section 4, we describe our AIS, which is based on the T-Cell Model. In Section 5, we present our experimental setup, our results and we discuss them. Finally, in Section 6, we present our conclusions and some possible paths for continuing with this research line.

2 Moving Peaks Benchmark (MPB)

Moving Peaks Benchmark was proposed in [2]. MPB defines a dynamically changing fitness landscape $f : X \times T \longrightarrow \Re$, where T stands for the (discrete) time, and $X = x_1, \ldots, x_5$ is the set of admissible solutions. The landscape is build of a set of peaks (scenario 1) or cones (scenario 2). Every ith peak or cone has its height h_i , width w_i, and the coordinates of its maximum $cmax_i$. All the parameters characterizing each peak are generated randomly from the corresponding interval. The fitness function for ith peak is evaluated as follows: $f_i(x_1, \ldots, x_5) = \frac{h_i}{1+w_i \prod_{j=1}^{5}(x_j - cmax_i[j])^2}$, while the equation for an ith cone is $f_i(x_1, \ldots, x_5) = h_i - w_i\sqrt{\prod_{j=1}^{5}(x_j - cmax_i[j])^2}$. Then the value of the overall fitness function $f(x_1, \ldots, x_5)$ is computed as: $f(x_1, \ldots, x_5) = \max_{i=1,\ldots,N} f_i(x_1, \ldots, x_5)$, where N is a number of peaks or cones defined in the scenario. For such a fitness landscape, all the three features of each peak or cone can be modified to perform landscape changes [16].

3 Previous Related Work

Gaspar et. al [7] proposed a Simple Artificial Immune System (Sais). Sais starts with an initial random population of B-Cells, each able to detect a given antigen specified by a binary bits long string. Then, it applies at each generation three operators : Evaluation, Clonal Selection and Recruitment (elimination of undesirable B-cells). Sais was validated with pattern tracking problem.

Trojanowski K. in [14] analyze the efficiency of two mutation operators applied in a clonal selection based optimization algorithm (AIIA) for non-stationary tasks. Both operators use a $\alpha-$stable random number generator. The author argues that appropriate tuning of the α parameter allows to outperform the results of algorithms with the traditional operators. The algorithms were tested with six environments generated with two test-benchmarks: a Test Case Generator and a Moving Peaks Benchmark.

Nanas et. al [10] compared Evolutionary Algorithms and Artificial Immune Systems under Multimodal Dynamic Optimization. They review the basic evolutionary and immune-inspired approaches to multimodal dynamic optimisation and they identify correspondences and differences and point out essential computational elements.

Trojanowski K. in [13] analyze the efficiency of the B-Cell algorithm applied to Moving Peaks Benchmark. The algorithm starts with a population of solutions randomly generated and performs the process of iterated improvement of the solutions by the repetition of: 1) affinity evaluation and 2) clonal selection and expansion.

Trojanowski K. et. al compared in [16] five instances of AISs: 1) Artificial Immune Iterated Algorithm (AIIA) [15], 2) B-Cell Algorithm (BCA) [8], 3) Clonal Selection Algorithm (CLONALG) [4], 4) opt-Ainet algorithm [11], and a Simple Artificial Immune System (Sais) [7]. All of them implement non-deterministic iterated process of search and all of them work with a population of antibodies or B-cells. These represent candidate solutions to the problem. The coordinates of these points are represented by real numbers or can be coded as bit strings. Each algorithm starts with a population of randomly generated tentative solutions which are iteratively improvement. The authors tested those approaches with seven types of mutations. M_1 to M_7 (see [16] for details). The algorithms were tested with six environments generated with two test-benchmarks: a Simple Test Case Generator and a Moving Peaks Benchmark.

4 T Cell Theory

In this paper, we present what we believe to be a new adaptive immune system model based upon the immune responses mediated by the T-cells. Our model is called TCELL, and it considers many of the processes that the T cells suffer from their origin in the hematopoietic stem cells in the bone marrow until they become memory cells.

T cells belong to a group of white blood cells known as lymphocytes. They play a central role in cell-mediated immunity. They present a special receptor on their cell surface called T cell receptors (TCR[3]).

T cells can be classified in different populations according to the antigen receptor they express,TCR-1 or TCR-2. Additionally, TCR-2 cells express CD4 or CD8[4].

Also, T cells can be divided into three groups according to their maturation or development level (phylogenies of the T cells [5]). Virgin cells are those which had never been activated (i.e., they did not suffer proliferation or differentiation). At the beginning, these cells do not express CD4 nor CD8. However, later on, they develop and express both marks, CD4 and CD8. Finally, virgin cells mature and express only one mark, either CD4 or CD8. Before these cells release the thymus, they are subject to both positive selection [6] and negative selection [6]. Positive

[3] TCRs are responsible for recognizing antigens bound to major histocompatibility complex (MHC) molecules.

[4] Lymphocytes express a large number of surface molecules that can be used to mark different cellular populations. CD means Cluster Denomination and indicates the group to which lymphocytes belong.

selection guarantees that the only survivors are the cells with TCRs that present a moderate affinity with respect to the self MHC. Negative selection eliminates the cells with TCRs that recognize self components unrelated to the MHC.

Effector cells are a type of cells that express only one mark, CD4 or CD8. They can be activated by co-stimulating signals plus their ability to recognize an antigen [3, 9]. The immune cells interact through the secretion of cytokines.[5] Cytokines allow cellular comunication. Thus, an immune cell c_i influences the activities (proliferation and differentiation) of another cell c_j by the secretion of cytokines, modulating the production and secretion of cytokines by c_j [5]. In order to activate an effector cell, a co-stimulated signal is necessary. Such signal corresponds to the cytokines secreted from another effector cell. The activation of an effector cell implies that it will be proliferated and differentiated. The proliferation process has as its goal to replicate the cells and the differentiation process changes the clones in order to they acquire specialized functional properties.

Finally, the memory cells are cells that persist into the host even when the infection or danger have been overtaken, so that in the future, they are able to get stimulated by the same or by a similar antigen. It is worth noting that, although the effector and memory cells are proliferated, they are not subject to somatic hypermutation. For the effector cells, the differentiation process is subject to the cytokines released by another effector cell. In our model, the differentiation process of the memory cells relies on their own cytokines.

4.1 Our Proposed Algorithm Based on TCELL

DTC (Dynamic T-Cell) is an algorithm inspired on our TCELL model, which we propose to solve dynamic optimization problems. DTC operates on four populations, corresponding to the groups in which the T-cells are divided: (1) Virgin Cells (VC), (2) Effector Cells with cluster denomination CD4 (CD4), (3) Effector Cells with cluster denomination CD8 (CD8) and (4) Memory Cells (MC). Each population is composed by a set of T-cells whose characteristics are subject to the population to which they belong.

Virgin Cells (VC) do not suffer the activation process. They have to provide diversity.Virgin cells are represented by 1) a *TCR* (TCR_b): represented by a bitstring using Gray coding and 2) a *TCR* (TCR_r): represented by a vector of real numbers.

Effector Cells are composed by 1) a TCR_b or TCR_r, if belongs to CD4 or CD8, respectively , 2) a proliferation level and 3) a differentiation level. The goal of this type of cell is to explore in a global way the search space.

The goal of memory cells is to explore the neighborhood of the best found solutions. These cells are represented by the same components that CD8.

For our propose the TCR identifies the decision variables of the problem, independently of the TCR representation. The proliferation level indicates the

[5] Proteins act as signal transmitters between cells, and also induce growth, differentiation, activation, etc.

number of clones that will be assigned to a cell and the differentiation level indicates the number of bits or decision variables (according with the TCR representation) that will be changed, when the differentiation process is applied.

The activation of an effector cell, ce_i, implies the random selection of a set of potential activator (or stimulating) cells. The closer one to ce_i, using Hamming or Euclidean distance according to the TCR, in the set is chosen to become the stimulating cell, say ce_j. Then, ce_i proliferates and differentiates.

At the beginning, the proliferation level of each stimulated cell, ce_i, is given by a random value between [1, 5], but then it is determined taking into account the proliferation level of its stimulating cell (ce_j). If the ce_i is better than ce_j, then ce_i keeps its own proliferation level; otherwise, ce_i receive a level which is 10% lower than level of ce_j.

Memory cells proliferate and differentiate according to their proliferation level (random between 1 and the size of MC) and differentiation level (random between 1 and the 90% of the number of decision variables), respectively. Both levels are independent from the others memory cells.

Each type of cell has its own differentiation process, which is blind to their representation and population.

Differentiation for CD4: the differentiation level is determined by the Hamming distance between the stimulated (ce_i) and stimulating (ce_j) cells. Each decision variable and the bit to be changed are chosen in a random way. The bit changes according to a mutation (or reaction) probability prob$_{mut-CD4}$.

Differentiation for CD8: the differentiation level from each cell is related to its stimulating cell (ce_j). If ce_j is better than the stimulated cell ce_i, then the level (for ce_i) is a random between $[1, \mid dv \mid]$; otherwise is a random between $[1, \mid dv \mid /2]$, where $\mid dv \mid$ is the number of decision variables of the problem. Each variable to be change is chosen in a random way and it is modified according to the following expression: $x^{'} = x \pm U(0, lu - ll)^{U(0,1)}$, where x and $x^{'}$ are the original and the mutated decision variables, respectively. lu and ll are the upper and lower bounds of x, respectively. At the moment of the differentiation of a cell (ce_i), the value of the objective function of its stimulating cell (ce_j) is taken into account. In order to determinate if $r = U(0, lu - ll)^{U(0,1)}$, will be added or subtracted to x, the following criteria are considered: 1) if ce_j is better than ce_i and the decision variable value of ce_j is less than the value of ce_i, or if ce_i is better than ce_j and the decision variable value of ce_i is less than the value of ce_j, then r is subtracted to x; otherwise r is added to x. Both criteria are motivated by our aim to guide the search towards the best solutions found so far.

Differentiation for MC : Each variable to be change is chosen in a random way and it is modified according to the following expression: $x^{'} = x \pm \left(\frac{U(0,lu-ll)}{100iter}\right)^{U(0,1)}$, where x and $x^{'}$ are the original and the mutated decision variables, respectively. lu and ll are the upper and lower bounds of x. $iter$ indicates the number of iterations until reach the amount maximum of

Algorithm 1 DTC Algorithm

```
Initialize_Function();
Initialize-Evaluate_VC();
Assign_Proliferation();
Divide_CDs();
Positive_Selection_CD4_CD8();// eliminate the cells in CD4 and CD8 with worst objective function value
Negative_Selection_CD4_CD8();// eliminate the most similar cells in CD4 and CD8
while Repeat a predetermined number of change do
  while Repeat a predetermined number of evaluations do
    Active_CD4();
    Sort_CD4();
    Active_CD8();
    Sort_CD8();
    Insert_CDs_en_MC();
    while Repeat a predetermined number of times (rep$_{MC}$) do
      Active_MC();
    end while
    Sort_CM();
  end while
  Statistics();
  Change_Function();
  Re-evaluate_Populations();
end while
```

evaluations for a change. In a random way, we decide if $r = \left(\frac{U(0,lu-ll)}{100iter}\right)^{U(0,1)}$, will be added or subtracted to x.

In both differentiation processes $U(0,w)$ refers to a random number with a uniform distribution in the range (0,w).

The general structure of our proposed algorithm for dynamic optimization problems is given in Algorithm 1.

The algorithm works in the following way. At the beginning TCR_b and TCR_r from virgin cells are initialized in a random way. Then, they are evaluated. The negative and positive selections are applied, the first eliminates the similar cells (keeping the best cells) and second eliminate a 10% of the worst cells. Next, the best virgins cells are divided in order to compose the populations CD4 (TCR_b) and CD8 (TCR_r).

Once the CD4 and CD8 populations have been activated (proliferation and differentiation) the best solutions from these populations are inserted or replace the worst solutions in MC (if MC is empty or not, respectively). When a cell from CD4 has to be inserted into MC, the cell first has to be converted in its real-value vector through the application of the following equation: $dv_j = l_{lj} + \frac{\sum_{i=0}^{L_j} 2^{L_j - i} dv'_{ij}(l_{uj} - l_{lj})}{2^{L}_{j} - 1}$, where dv_j is the j^{th} decision variable with $j = 1, \ldots,$ number of decision variables of the problem, L_j is the number of bits for the

j^{th} decision variable, l_{uj} and l_{lj} are the upper and lower limits for the decision variable dv_j and dv'_{ij} is the i^{th}-bit of the binary string that represents dv_j.

Next, the cells from MC are activated a predefined number of times (rep_MC).

The algorithm finishes when a predefined number of evaluations for each change is reached. We assume that the algorithm `knows` when the environment has changed, in order to re-evaluate the populations.

5 Numerical Experiments

In order to evaluate the performance of the algorithm we use the measure [16] Offline error (oe), it represents the average deviation of the best individual evaluated since the last change from the optimum. It is defined by: $oe = \frac{1}{N_c}\sum_{j=1}^{N_c}(\frac{1}{N_e(j)}\sum_{i=1}^{N_e(j)}(f_j^* - f_{ji}^*))$, where N_c is the number of fitness landscape changes within a single experiment, $N_e(j)$ is the number of solution evaluations performed for the jth state of the landscape, f_j^* is the value of the optimal solution for the jth landscape and f_{ji}^* is the current best fitness value found for the jth landscape [16].

To validate DTC, we use two fitness landscapes created with MPB. Each landscape consist of a number of peaks, changing in a random way their height, width and location. The two environments were used according to standard settings given in the web page[6]: scenarios 1 (5 dimensions - 5 peaks) and 2 (5 dimensions - 50 peaks, see the web page for details). For each scenario the function changes every 5000 evaluations of the objective function.

The required parameters for DTC are: size of VC, CD4, CD8 and MC; number of repetitions MC (rep_MC); percentage of replacement for MC; Probability of mutation $prob_{mutCD4}$. For the experiments, we adopted the following parameters, which were empirically derived alter numerous experiments, we used a population size, for VC, of 100 cells. For CD4 and CD8 we adopted a population size of 50 and 70 cells , for scenario 1 and 2 respectively. For MC 5 cells. The mutation probability $prob_{mut}$ was 0.07. rep_MC was set to 10 and 100, for scenario 1 and 2 respectively. Finally, 50% replacement was adopted for replacing from CDs to MC. 50 independents runs and 110 change of the objective function were performed for each scenario. For both scenarios we use a binary Gray code (for VC and CD4) with 40 bits for each decision variable.

Our results were compared respect to the results obtained for the best combination AIS-mutation operator presented in [16] for each scenario, they are: AIIA-M_6- $\alpha = 2.0$, CLONALG-M_2, Sais-M_3 and opt-Ainet -M_2 for scenario 1 and AIIA-M_5- $\alpha = 1.5$, CLONALG-M_3, Sais-M_2 and opt-Ainet -M_1 for scenario 2. BCA- M_5- $\alpha = 1.75$, for scenario 1 and - $\alpha = 2.0$ for scenario 2.

[6] http://www.aifb.uni-karlsruhe.de/ jbr/MovPeaks/movpeaks/.

5.1 Analysis of Results

Comparing our proposed DTC with respect to the AIS presented in [16] (see Table 1), our approach obtained better results in scenario 1 and competitive results for scenario 2.

Fig. 1a shows the average of the best found solutions, of each population (CD4, CD8 and MC)and the optimum before the objective function change, for scenario 1. Here, we can see that the solutions found by CD4 are better than the solutions found by CD8. MC performs a good local search for the solutions provide by CD4. The performance of DTC does not deteriorate in the presence of a change. While for scenario 2 (see Fig. 1b), we can observe that both populations, CD4 and CD8, provide good solutions in order to MC find solutions near the optimum and its performance is affected. This fact makes us think that scenario 2 requires more diversity due to the number of peaks, fifty peaks while scenario 1 has only five peaks.

Fig. 1c shows the average of the offline error, over the 50 runs, for both scenarios. The presence of a change affects the performance of DTC worst in scenario 2 than scenario 1.

Table 1. Offline Error obtained for each Approach.

Approach	Scenario 1	Scenario 2
AIIA	0.71	3.44
CLONALG	11.71	10.53
Sais	12.40	11.57
BCA	0.39	**2.69**
opt-Ainet	2.39	4.76
DTC	**0.37**	3.02

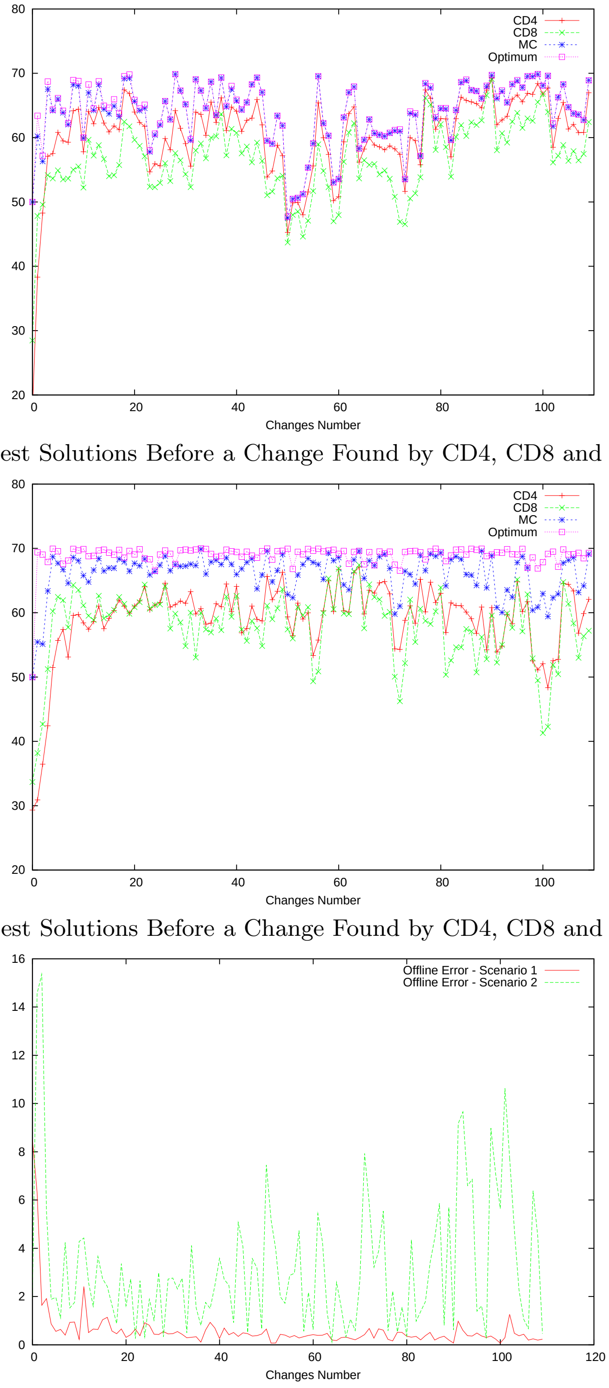
6 Conclusions and Future Work

We have presented an artificial immune system based on the T-Cell model, which has been proposed to solve dynamic optimization problems. The proposed approach is inspired on the processes suffered by the T-Cells within our immune system. In this first study, the proposed approach has been tested with two scenarios generated by a generator of dynamic functions (MPB). The results obtained by our proposed approach are promising, resulting better in one of the cases to those generated by the other algorithms with respect to which it was compared. This fact encourages us, as future work, to deep this study over different kind of dynamic environments.

References

1. *How to Solve It: Modern Heuristics.* Springer, December 2004.
2. Jurgen Branke. Memory enhanced evolutionary algorithms for changing optimization problems. In *IN CONGRESS ON EVOLUTIONARY COMPUTATION CEC99*, pages 1875–1882. IEEE, 1999.
3. P. Bretscher and M. Cohn. A theory of self-nonself discrimination, 1970. Science, 169, 1042-1049.
4. Leandro Nunes De Castro and Fernando J. Von Zuben. The clonal selection algorithm with engineering applications. In *In GECCO 2002 - Workshop Proceedings*, pages 36–37. Morgan Kaufmann.
5. Dipankar Dasgupta and Fernando Nino. *Immunological Computation: Theory and Applications.* Auerbach Publications, Boston, MA, USA, 2008.
6. David B. Roth David Male, Jonathan Brostoff and Ivan Roitt. *Inmunologia.* Elsevier Science Health Science div, Madrid, 7a edition, 1997. Ed. Harcourt BraceISBN 84-8174-180-9.
7. Alessio Gaspar and Philippe Collard. From gas to artificial immune systems: Improving adaptation in time dependent optimization. In *In Proceedings of the Congress on Evolutionary Computation*, pages 1859–1866. IEEE Press, 1999.
8. Johnny Kelsey and Jon Timmis. Immune inspired somatic contiguous hypermutation for function optimisation. In E. Cantú-Paz, J. A. Foster, K. Deb, L. Davis, R. Roy, U. O'Reilly, H. Beyer, R. K. Standish, G. Kendall, S. W. Wilson, M. Harman, J. Wegener, D. Dasgupta, M. A. Potter, A. C. Schultz, K. A. Dowsland, N. Jonoska, and J. F. Miller, editors, *Genetic and Evolutionary Computation - GECCO 2003, Genetic and Evolutionary Computation Conference*, pages 207–218, Chicago, Illinois, USA, July 2003. Springer. Lecture Notes in Computer Science Vol. 2723.
9. P. Matzinger. Tolerance, danger and the extend family. *Annual Review of Immunology*, 12:991–1045, April 1994.
10. Nikolaos Nanas and Anne N. De Roeck. Multimodal dynamic optimization: From evolutionary algorithms to artificial immune systems. In *ICARIS*, pages 13–24, 2007.
11. L. Nunes de Castro and J. Timmis. An artificial immune network for multimodal function optimization. In *Proccedings of the 2002 Congress on Evolutionary Computation (CEC'2002)*, volume 1, pages 669–674, Honolulu, Hawaii, May 2002.
12. Leandro Nunes de Castro and Jonathan Timmis. *Artificial Immune Systems: A New Computational Intelligence Approach.* Springer-Verlag, New York, 2002.
13. Krzysztof Trojanowski. B-cell algorithm as a parallel approach to optimization of moving peaks benchmark tasks. *Computer Information Systems and Industrial Management Applications, International Conference on*, 0:143–148, 2007.
14. Krzysztof Trojanowski. Clonal selection approach with mutations based on symmetric alpha stable distributions for non-stationary optimization tasks. In *ICANNGA '07: Proceedings of the 8th international conference on Adaptive and Natural Computing Algorithms, Part I*, pages 184–193, Berlin, Heidelberg, 2007. Springer-Verlag.
15. Krzysztof Trojanowski and Slawomir T. Wierzchon. Studying properties of multipopulation heuristic approach to non-stationary optimisation tasks. In *IIS*, pages 23–32, 2003.
16. Krzysztof Trojanowski and Slawomir T. Wierzchon. Immune-based algorithms for dynamic optimization. *Inf. Sci.*, 179(10):1495–1515, 2009.

Fig. 1. Optimum, Best Solutions and Offline Errors for DTC.



a. Optimum and Best Solutions Before a Change Found by CD4, CD8 and MC for Scenario 1

b. Optimum and Best Solutions Before a Change Found by CD4, CD8 and MC for Scenario 2

c. Offline Error Obtained by DTC