Algorithm to distribute feed pulp between paralleled thickeners during red-sludge thickening and washing in alumina production

E R Fedorova¹, N V Vasileva¹, E A Pupysheva¹

¹Saint-Petersburg Mining University, 2, 21st Line, Saint-Petersburg, 199106, Russia
E-mail: Fedorova_ER@pers.spmi.ru

Abstract. Distributing the red-sludge flow between paralleled thickeners is a relevant problem for controlling the process of thickening and washing the alumina sludge. Sludge sticks to thickener walls and rakes in operation, which reduces the effective volume of the machine and puts a greater strain on the rakes; raising the rakes is necessary to avoid breaking the stirrer, but this reduces the compacting rate of the thickened product. Redistributing the feed pulp between thickeners over the course of their gradual uneven contamination might solve the problem of the thickened product being under-compacted in the thickening-washing line. There are patents that address this issue; however, they only describe semiautomatic approaches. The problem has been covered to a great extent by Russian scientists and engineers such as M V Levin, T B Potapova, V V Aleksandrov, T G Milberger, P F Minin, I M Fain, R M Khamidov, as well as by the following institutions: All-Union Research and Development Institute of Aluminum, Magnesium, and Electrode Industries, Pikalevo Alumina Plant, Pavlodar Aluminum Plant. The paper describes the algorithm of an artificial immune system for redistributing the red-sludge flow in alumina production between paralleled thickeners.

1. Introduction
Red-sludge thickening and washing is crucial for alumina production [1-6]; it is implemented by a string of paralleled thickeners and washers, some of which run continuously. Currently, thickener power supply is controlled manually by adjusting the gates on the feed box lines. Paralleled thickeners receive nearly identical volumetric flow, which is inappropriate, as continuous run results in sludge being accumulated on the walls and rakes, necessitating redistribution of feed pulp between thickeners over time.

The existing ways to control paralleled thickeners are semi-automatic and use the known control circuits. The goal hereof is to create an algorithm to distribute the pulp flow between paralleled thickeners. Artificial immune system (AIS) based algorithms have already been used for process optimization [7]. Distributing the feed flow between paralleled machines is a relevant problem for alumina production. This research uses a case-tailored AIS algorithm to address this challenge.

2. Facility Controlled
A cylindrical-conical circular thickener is the facility to be controlled. The thickening line comprises 7 thickeners, 5 of which run continuously. The objective parameters of the process control are the turbidity of the aluminate solution exiting the thickening line; and the solid-phase content in the thickened product.
to be washed. Thus, the AIS algorithm aims at maintaining an average density of the thickened product within a specified range, as well as at reducing the total volumetric solid-phase flow in the overflow. When testing and debugging the AIS algorithm, the controlled facility used was a generalized dynamic model of a circular thickener with various inputs on the sludge accumulation on the bottom and on the rakes. For a real thickener, it is proposed to use such parameters as the engine load in rake rotation and the rake height.

3. AIS Training
Molecules recognized by adaptive immune systems are referred to as antigens. When an animal is exposed to an antigen, some subpopulations of its bone marrow-produced cells (B lymphocytes) respond by producing antibodies. Antibodies are molecules that are initially attached to the surface of B cells; they are designed to recognize and specifically bind to antigens. Each B cell secretes a specific antibody type that is relatively specific to this or that antigen. When bound to such antibodies, or receiving a second signal from helper cells such as T helpers, the antigen stimulates the B cell to grow (divide) and mature into antibody-secreting terminal (indivisible) cells referred as plasma cells. In the process of cell division, or mitosis, a clone (one or set of the “offspring” cells) is born.

In addition to proliferation and differentiation into plasma cells, B cells may become long-lived memory B cells. Memory cells circulate in the blood steam, lymph, and tissue; when exposed to a second antigen stimulus, they differentiate into plasma cells capable of produce high-affinity antibodies that are pre-selected for the specific antigen that stimulated the primary response.

4. AIS Algorithm
Feed-flow is redistributed between paralleled thickeners by minimizing the optimization criterion J [6]:

\[ J = \frac{\sum_{i=1}^{5} (Q_{Fi} - U_{Fi})^2}{\sum_{i=1}^{5} Q_{Fi} \phi_{Fi}}, \quad Q_{FX} = \text{Const} \quad (1) \]

where \( Q_{Fi} \) is the volumetric flow of the feed pulp, m³/h; \( U_{Fi} \) - the volumetric flow of the thickened product, m³/h; \( \phi_{Fi} \) is the volumetric fraction of the solid phase; \( C_{OFi} \) is the solid-phase concentration in the overflow, mg/l, i=1..5.

The algorithm distributes the feed flow between the paralleled thickeners by minimizing the optimality criterion \( (J') \) described in the formula (1); it stabilizes the thickened-product solid-phase content at 0.085 to 0.095 vol. fraction while minimizing the total solid-phase content in the overflow of the entire thickening line [6]:

\[ J' = \min_{U'} \left( J(U', V) \right), \quad U' = [Q_{F1}, Q_{F2}, Q_{F3}, Q_{F4}, Q_{F5}] \quad (2) \]

\[ \text{with a constraint: } \phi_{UF\min} \leq \phi_{UF} \leq \phi_{UF\max}. \quad (3) \]

The algorithm is used for the evolution of a population of potential solutions, whereby each individual member seeks to find the best solution. Selection and variability operators (e.g. crossover, mutation, or inversion) optimize the population. Nichening operators expand the evolution algorithms to domains where combined solutions must be found and supported, e.g. combined and multi-objective function optimization.

The algorithm below must demonstrate that such accumulative blind variability based solely on a proportionate affine mutation combined with selective pressure may be able to generate efficient solutions for complex problems.

The algorithm uses a shape space where real numbers are rendered as binary code \( x_i = \text{code} (U_i) \), i.e. each control parameter (modifier) of the thickener is represented by a bit array of length \( L = 7 \) bits [6].

The shape-space model is intended to quantify the antigen-antibody-interaction. The set of molecule properties is referred to as the generalized form of the molecule. The antigen-antibody codification (binary or real) defines its spatial representation; distance is used to compute the interaction between these molecules. Mathematically, molecules (both antibodies and antigens) can be generalized by the set of qualitative features \( L \), which are linked directly to the coordinates axes. Exact physical value of each feature is not related to the development of computational tools. The authors hereof consider binary
or integer strings as molecular representations. It is believed that antigens and antibodies must have the same length L. The length and the representation of a cell are problem-specific.

Therefore, the length of the bit array for storing all modifiers in binary code np = L×i, where i is the number of modifiers. Then Xvect = [x₁, x₂, ..., xₗ] = [code (U₁), code (U₂), ..., code (Uₗ)]. On a numeric axis, real numbers form a continuum, i.e. two numbers can be arbitrarily close to each other, and any segment may contain an indefinite number of values. However, the number of possible values of numbers is finite in machine representations. Therefore, the real number/modifier (Uᵢ) is replaced with a machine code that constitutes a finite discrete set, as each code represents a whole interval of continuum values and equals: Nosnov = 2^L = 2^7 = 128. The number of elements for any control actions in binary code in the discrete settings space is defined in a range [0...2ⁿ], which is the range of Xvect that contains all the variables and modifiers of the controlled facility [6].

Therefore, optimal modifiers are found in Xvect; then the parameters are decoded from binary code to decimals by MatLab fix and mod operations; then the obtained normalized values of the control actions (Uᵢnorm) are scaled to the set system units as [6]:

\[ U_i = U_i \text{min} + \frac{(U_i \text{max} - U_i \text{min})}{\text{Nosnov}} U_i \text{norm} \] (4)

where \( U_i \text{max} \) is the upper boundary/constraint on the parameter \( U_i \); \( U_i \text{min} \) is the lower boundary.

In the Xvect space, the objective function has multiple local minima; however, the proposed algorithm is a parallel-search algorithm that handles searching for the global minimum [6].

Step 1. Outside the loop, generate a matrix of random memory cells (population) M sized [i, j], where i = 1:7 is the number of structures in the array M, j = 1: np. Set an initial value of the criterion \( J_{\text{max}} \) and the affinity \( \text{Maff} \) to the current minimum antigen \( J \) in each structure at \( i = 1:7 \) and \( j = 1 \). For the entire array M at \( j = 1: np \), generate binary-represented cell prototypes by means of the operator ‘rand’ (0|1).

Thus, the array M is the best vector of initial search for settings [6].

Step 2. Similarly to the population of memory cells, generate an out-of-cycle matrix of B cells (initial set of B lymphocytes) sized [i, j], where i = 1:10 is the number of structures in the array B, j = 1: np. Set an initial value of the criterion \( J_{\text{max}} \) and the affinity \( \text{Baff} \) to the current minimum antigen \( J \) in each structure at \( i = 1:7 \) and \( j = 1 \). For the entire array B at \( j = 1: np \), generate binary-represented cell prototypes by means of the operator ‘rand’ (0|1). Array B is a global search vector (an array of structures \( V \)) [6].

Step 3. Generate a population of B cells characterized by a larger affinity/proximity than the initial set of B cells. The algorithm of generating a population of B cells on the basis of M memory cells consists of two parts [6]:

The memory-cell affinity is compared to the B-cell affinity; however, if \( \text{Maff} > \text{Baff} \), then the M array structure / M array memory cell is embedded into the ith slot of the B cell while the B cell is shifted to \( i+1 \). Thus, generate an array of structures \( V_{i,j} \) by the array of structures M: \( V_{i,j} := M_{i,j} \). Test the program stop: if all the elements of the structure array \( V_{i,j} \) are close with a certain tolerance \( \varphi \), stop searching; else, proceed to the next substep;

For search variety, generate three new B cells for \( i = 8:10 \); use the operator ‘rand’ (0|1).

Step 4. To prevent uncontrolled reproduction of B cells and to support the promising B cells, B cells must be stimulated and suppressed by T cells; compare the peptide of the embedded cells to the peptide of the cells \( V_{i,j} \). If there is high affinity/low difference, then i, or the array \( V_{i,j} \) structure/cell is removed from the population.

Step 5. Each B cell is able to produce 10 antibodies per cycle. The number of cycles is set at the algorithm onset. Thus, the array of antibodies \( \text{Ab} \) consists of at least 100 structures: \( \text{Ab} [100, \text{np}] \) [6].

Step 6. Mutate antibody peptides / invert binary-code bits by the probability template of variability \( T \), where \( T = \{0.8; 0.5; 0.3; 0.2; 0.1; 0.05; 0.01\} \) [6].

Step 7. Select 10 random mutated antibodies by the round operation to further combat the antigen [6].

Step 8. Decode from binaries to decimals and scale to the specified range of the physical units of the thickeners modifiers: \([U₁, U₂, ..., Uₗ] = \text{decode (Xvect)} = \text{decode} [x₁, x₂, ..., xₗ]\) [6].
Step 9. Combat the current antigen (Ag) and compute the affinity to it is done in the following steps:
  Calculate the control actions for 10 setting configurations/randomly mutated antibodies decoded at step 8. Test the model parameters for technological constraints. Solve the conditional minimization problem by setting the maximum-value objective function $J_{\text{max}}$ at the maximum known parameters of the facility. The objective-function values are stored in the element $Ab_i$, where $i = 1:10$.

Calculate the currently-strongest Ag. The strongest antigen determines the minimum $J$ for the stated problem. If $Ab_i < J_{\text{min}}$, then $J_{\text{min}} := Ab_i$. Calculate the affinity of the criterion $J$ of the selected mutated structures $Ab_i$ and $V_i$ where $i = 1:10$ for the minimum value $J_{\text{min}}$ as: $Ab_{i \text{aff}} = J_{\text{min}}/(Ab_i \times J)$, $V_{i \text{aff}} = J_{\text{min}}/(V_i \times J)$. The calculation results show the antigen-antibody affinity.

Step 10. Sort the selected antibodies/clones by their affinity to the antigen by the bubble method [6]:
  for $i = 1: (10-1)$ check:
  if clones_y $(i, 1) >$ clones_y $(i + 1, 1)$, swap the values of antibodies, their affinity, and antibody peptides by means of temporary variables, e.g.
  
  time_value = clones_y$(i, 1)$;
  clones_y$(i, 1)$ = clones_y$(i + 1, 1)$;
  clones_y$(i + 1, 1)$ = time_value;

Step 11. Generate memory cells on the basis of best antibodies. Store the first 7 best antibodies sorted at Step 10 in 7 memory cells of the structure array $M$ [6].

Step 12. Go to step 3.

When the search is complete, the controller generates a control action (volumetric flow of feed pulp to each of the thickeners of the thickening line); the actions are submitted to the local thickener control systems that now use the settings until recalculated.

The parallel function-optimum search algorithm has been tested in MatLab. Figure 1 shows the algorithm searching for a global minimum at specific feed-pulp parameters.

![Figure 1](image1.png)

**Figure 1.** Search for the minimum value of the optimality criterion by the AIS algorithm.

![Figure 2](image2.png)

**Figure 2.** Scatterplot of archival values and values as obtained after optimizing by solid-phase content of the thickened product at the thickening-line output.
5. Conclusion
The solutions (circles) cover the graph peaks in all of the tested cases. Although the algorithm steps are stochastic, the results shown in Figure 1 are not significantly different from each of the attempts. Peaks are always clear, while the number of off-peaks is always reduced.

The feed-flow distribution between paralleled thickeners is controlled by an AIS-based algorithm and thickener-specific control models; this stabilizes the solid-phase content in the thickened product of the thickening branch within a volumetric fraction of 0.085 to 0.095 (Figure 2); the algorithm also minimizes the total solid-phase mass that is carried by the aluminate solutions leaving the RS thickening and washing facility (Figure 3).

References
[1] Concha F A 2014 Fluid Mechanics and Its Applications vol 105 (Springer International Publishing Switzerland) p 429
[2] Sizyakov Victor M, Brichkin Vyacheslav N. 2018 Journal of Mining Institute 231 292-298
[3] Fedorova E R, Firsov A Yu 2018 Journal of Physics: Conference Series 1015 032035
[4] Bürger R, Damasceno J R and Karlsen K H 2004 Int. J. Miner. Process. 73 183-208
[5] Bürger R, Garcia A, Karlsen K H and Towers J D 2007 Computers and Chemical Engrg. 6(32) 1173-1194
[6] Fedorova E.R. 2017 Raspredelenie pitayuschego potoka mezhdu parallel'nymi sgustitelyami pri upravlении технологическим переработкой и промывкой красных шламов глиноземного производства [in Russ.] p 22
[7] Safonov D.N. 2012 Upravlenie processom razdeleniya tehnologicheskikh pul'p mednikel'nikel'mogo proizvodstva v sovremennyh fil'tr-pressah [in Russ.] p 108