Immunizing Job Recommender System

Shaha T. AL-Otaibi¹ and Mourad Ykhlef²

¹Department of Information Systems, College of Computer and Information Sciences, Princess Nora Bint Abdulrahman University, Riyadh, Saudi Arabia
²Department of Information Systems, College of Computer and Information Sciences, King Saud University, Riyadh, Saudi Arabia

Corresponding author: Shaha T. AL-OOTAIBI

Abstract
Artificial Immune System is a novel computational intelligence technique inspired by immunology has appeared in the recent few years. It takes inspiration from the immune system in order to develop new computational mechanisms to solve problems in a broad range of domain areas. This article presents a problem oriented approach to design an immunizing solution for job recommendation problem. We will describe the immune system metaphors that are relevant to job recommender system. Then, discuss the design issues that should be taken into account such as, the features of the problem to be modeled, the data representation, the affinity measures, and the immune process that should be tailored for the problem. Finally, the corresponding computational model is presented.

Key words: Recommender systems, Content-based recommender system, Collaborative filtering recommender system, Hybrid based recommender system, Computational Intelligence (CI), Artificial Immune System (AIS), Clonal Selection, Somatic hypermutation, Affinity measures.

1. Introduction
The Job Recommender System has been emerged in e-business online services in the few recent years. While companies published new job positions on the online portals, job-seeker uses them to establish their profiles. For each posted job, thousands of profiles received by companies. Consequently, a huge volume of job descriptions and candidate profiles are becoming available online. The need increases for applying the recommender system technologies that can support recruiters to handle the huge online information efficiently (Färber, Weitzel, & Keim, 2003), (Yi, Allan, & Croft, 2007).

The main challenge faced the applying of recommender system technology in candidates/job matching as recognized by the literature analysis, is the large volume of low qualification of applicants that match the search criteria (Singh, Catherine, & Visweswariah, 2010). Whereas, the best fit between job and candidates depends on underlying aspects that are hard to measure. Additionally, we notice the diversity nature of the job specification that should be considered in candidates/job matching. Certain job requirements should be modeled in many forms to meet a diverse set of candidates that satisfies job’s requirements. Whereas, many applicants missed the recruiting’s opportunity that result from poor job requirements’ determination. Since the determined features may meet set of candidates, but still there are many other applicants suitable for the job but did not match the determined criteria. We need to determine the job’s requirements by a set of vectors of different features that can meet most possible appropriate applicants, and then rank applicants depending on the matching degree between the applicant and job requirements.
Many traditional recommendation techniques have been applied to job recommendation problem. Most of these techniques produced good solutions with relatively some limitations (Al-Otaibi & Ykhlef, 2012b), (Zheng, Hong, Zhang, & Yang, 2012). In order to cope with these limitations, the natural immune system shows many properties that are benefited in this area of research. We will conduct a comprehensive study and develop a recommender system algorithm that aimed to improve the job recommendation process. We treat the recommendation process as an optimization problem that has been successfully applied in many domains such as science, engineering, management, and business. Where as, different families of optimization models are used to formulate and solve decision-making problems. Metaheuristics represent a family of optimization techniques that gained a lot of popularity in the past two decades. They are among the most promising and successful techniques that offer good solutions in a reasonable time for solving hard and complex problems in science and engineering. This explains the significant growth of interest in metaheuristics domain. In recent years, many metaheuristics are inspired by natural processes such as evolutionary algorithms from biology; ants, bees’ colonies, and particle swarm optimization from swarm intelligence; and simulated annealing from physics (Talbi, 2011). More recently, a novel computational intelligence system inspired by immunology has appeared, named Artificial Immune Systems (AIS). This immune system has already been applied in solving a wide range of engineering problems. This work explores the undertaken aspects to develop an immune inspired technique for job recommendation problem. We used the observable immune components and processes as metaphors to produce AIS algorithm. This algorithm encapsulates a number of beneficial features of the natural immune system, and they are utilized towards solving job recommendation problem.

The article is organized as follows: Section 2 presents the related works. Section 3 demonstrates the background of artificial immune system and their related issues. In section 4, we illustrate how to model the job recommendation problem in AIS metaphor and discuss the design issues that should be taken into account. Additionally, in section 5 we present the AIS algorithm that will be applied to the job recommendation problem. We conclude this work in section 6.

2. Related Works

Many researches have been conducted to discuss different issues related to the recruiting problem as well as, the applying of recommender system technologies (Al-Otaibi & Ykhlef, 2012a), (Al-Otaibi & Ykhlef, 2012b). Zheng, et al. summarized job recommendation related issues such as user profiling and similarity calculation. They conducted empirical experiments on a local online recruiting website and presented the details of specific case study (Zheng, Hong, Zhang, & Yang, 2012). In our previous work, we presented a comprehensive survey of job recommender systems and listed the advantages and disadvantages of technical approaches in different job recommender systems (Al-Otaibi & Ykhlef, 2012b).

First, the content-based recommendations which originally recommend items that have similar content information to the matching users. Paparrizos, et al. built an automated system to recommend jobs for applicants based on their past job histories. This system treated the recommendation problem as a supervised machine learning problem (Paparrizos, Cambazoglu, & Gionis, 2011). The PROSPECT system, which is a decision support tool presented in (Singh, Catherine, & Visweswariah, 2010) to shortlist candidate resumes list. It mines resumes to extract features of candidate profiles such as skills, education, and experience. The information retrieval techniques used to rank applicants for a given job position. Additionally, Yu, et al. proposed a preference method based on user’s interaction history and a new similarity measurement method. The recommendation process divided into two parts: job recommendation and job-seeker recommendation. For both parts, the recommendations should be the objects which are the most consistent with their preferences (Yu, Liu, & Zhang, 2011).

Second, hybrid based recommendation which used a combination of different recommendation approaches to handle the job problem. For example, a combination of two or more approaches can be used such as content-based, collaborative filtering and knowledge-based recommendation. A recommendation system initially used
to recommend objects to users such as movies or books have been applied to matching partners (Färber, Weitzel, & Keim, 2003). They used two recommendation approaches content-based filtering and collaborative filtering simultaneously. Later, this recommendation approach utilized and extended in many works. Lee & Brusilovsky integrated the idea of recommender systems and the adaptive hypermediato produce the proactive recommender system (Lee & Brusilovsky, 2007). Additionally, Fazel-Zarandi & Fox improved the matching process by providing an adaptive job offering and discovery environment. They combined different matchmaking strategies in a hybrid approach for matching job seekers and jobs using logic-based and similarity-based matching (Fazel-Zarandi & Fox, 2010). The fuzzy multiple criteria algorithm determined the suitable personality traits and key specialized skills through information statistics and Analytic Hierarchy Process (Chen, 2009). Moreover, Zheng, et al. developed an online system named IHR, which groups users into different clusters and uses different recommendation approaches for different user clusters (Zheng, Hong, Zhang, & Yang, 2013).

3. Background of Artificial Immune Systems

The immune system is one of the most complicated biology systems. The knowledge about how the immune system acts is increasing with the advances in biology and molecular genetics. This knowledge becomes very interesting not only from a biological viewpoint, but also from a computational perspective. The immune system has now started to the emergence of AIS as a novel computational intelligence paradigm. It has emerged in the 1990s as a new branch in Computational Intelligence (CI). A number of AIS models were existent, and they are used in pattern recognition, elimination, optimization, computer security, fault detection, and many other applications researchers are exploring in the field of science and engineering (Dasgupta, 2006). It is also an addition to the increasingly long queue of approaches that are biologically inspired (Garrett, 2005). These approaches comprise the established paradigms such as Genetic and Evolutionary Computation (B"ack, Fogel, & Michalewicz, 1997); as well the Ant Colony (Dorigo, 1992), (Dorigo, 1999) and Particles Swarm ((Kennedy & Eberhart, 1995).

We realize some general features of job recommendation problem that the AIS will really bring some benefits, and thus distinguish it from other techniques. We will attempt to carefully mapping the problem features to mechanisms exhibited by the AIS, taking the problem-oriented perspective outlined and discussed in the following section. AIS can be described as metaphorical computational systems developed using ideas, theories, components, and process derived from the immune system (Castro & Timmis, 2002). Although many details of the immune mechanisms are unidentified to immunologists, however, it is well-known that the immune system is protecting our body against foreign invaders by a multilayered system. Depending on the type of the invader and the way it comes to the body, the immune system uses different response mechanisms to destroy the infected cells. It is comprised of physical barriers such as the skin and respiratory system, physiological barriers such as destructive enzymes and stomach acids and the original immune system, which has two parts, the innate and adaptive immune systems (Castro & Timmis, 2002). The innate immune system is a stable mechanism that perceives and destroys specific invading organisms, whereas the adaptive immune system responds to anonymous foreign invader and provides a response that can persevere in the body over a long period of time. The adaptive immune system is comprised of a collection of different cells accomplishing different functions that spread over the body. The primary cells in this process are two classes of white blood cells, called T-cells and B-cells. Both classes of cells initiate in the bone marrow, although T-cells cross to the thymus to mature before they circulate the body in the blood and lymphatic vessels. First, the T-cells have three classes: T-helper cells which are responsible for the activation of B-cells, Killer T-cells which bind to foreign invaders and inset poisonous into them making their destruction, and suppressor T-cells prevent the action of other immune cells thus inhibit the allergic reactions. Second, the B-cells are responsible for the generation and secretion of antibodies using specific proteins which binding the antigen. Each B-cell can only provide one particular antibody. The antigen is located on the surface of the invading organism, and the binding of an antibody to the antigen is a signal kills the invading
cell. In general, there is more than one mechanism for the immune system (Goldsby, Kindt, & Osborne, 2002). We will focus in this research on the primary processes that utilized in most AIS models: The Clonal Selection and the Somatic Hypermutation theories (Castro & Zuben, 2001), (Castro & Timmis, 2002), (Aickelin, 2004).

3.1. The Clonal Selection Principle

The Clonal Selection Algorithm named CLONALG (Castro & Zuben, 2001) was developed to perform pattern recognition and optimization. It has exposed success on a wide range of engineering problems. Since, the Clonal Selection has a set of important features such as diversity, optimization, exploration and learning. Diversity, while the immune cells can be sparsely spread to cover a wide region of the antigenic space. Optimization fulfilled by the selection and proliferation of high affinity cells that produce a rapidly population with high affinity that supports high response to next infection. Additionally, the exploration achieved by using mutation process to reach higher affinity matches to the invading antigen. Finally, It supports the learning using repeated exposure to an antigenic stimulus can work as a method of reinforced learning. While the memory cells become more specific to the current antigen and capable to respond more rapidly to a reoccurrence (Jennifer & Simon, 2003). The Clonal Selection is an algorithm used to define the basic features of an immune response to an antigenic stimulus. It is defined the idea that only those cells that recognize the antigen are proliferate. It works on both T cells and B cells. When the body is exposed to an antigen, some of the derived B-cells respond by producing antibodies. Each cell secretes only one kind of antibody, which is relatively specific for the antigen. By binding to these receptors, with a second signal from accessory cells (T-helper cell), an antigen stimulates the B cell (lymphocytes) to proliferate and mature into antibody secreting cells, called plasma cells. Although plasma cells are the most active antibody secretors, large B cells divide rapidly and secrete antibodies at a lower rate. Although, B cells secrete antibodies, T cells do not secrete antibodies, but play a central role in the regulation of the B cell response and are preeminent in cell mediated immune. B cells are proliferated or differentiated into plasma cells and also can differentiate into long-lived B memory cells. The memory cells circulate through the blood, lymph and tissues not to build-up antibodies, but when exposed to a second antigenic stimulus start differentiating into large lymphocytes. These lymphocytes capable of producing high affinity antibody, preselected for the specific antigen that had stimulated the primary response (Castro & Zuben, 1999). Figure 1 illustrates the Clonal Selection principle.

![Figure 1: The Clonal Selection principle.](image)

3.2. Somatic Hypermutation

The mutation process is the most commonly used in AIS; it means random genetic changes to the genes of the cloned cells are involved to control the antigen receptor. These changes caused proliferation and variation of high affinities antibodies. This concept was used as the basis for mutation in Clonal Selection algorithm, where the mechanism was the affinity between the antibody and the antigen (Brownlee, 2005). The mutation is applied for most applications such as: for binary strings bits are flipped, for real value strings one value is replaced at random, or for others the order of elements is swapped. Additionally, the mechanism is often
enhanced by the somatic hypermutation that noticed as a process for optimizing the binding affinity of antibodies (Aickelin, 2004).

4. Modeling Job Recommendation in AIS

The immune system is a suitable solution for job recommendation problem because of certain properties that inherent from many immune inspired algorithms. The features that are particularly relevant to recommendation problem are diversity, optimization, exploration, and learning capability. In our case, the main task of the immune system is to search for the qualified applicants (Antigens) for a specific job position (Antibodies).

The important feature that needs justification is the diversification nature of job’s requirements that should be considered in candidates/job matching, it will cause a large number of the matching process. Hence, particular job’s requirements need to be modeled in many forms to meet a diverse set of candidates. For example, there are many possible terms for specialities or skills that can be considered as possible valid solutions. To illustrate the diversification nature of job problem, we can consider a certain job as set of vectors \( n \) composed of different attributes that can meet different candidates’ characteristics. We can calculate the number of possible vectors \( NV \) in the worst case that describes the job’s requirements using the \( r \)-combinations in a particular order such as:

\[
NV = \binom{n}{r} = \frac{n!}{(n-r)!r!}
\]

These vectors should be applied to candidates/job matching to get the best fit between job’s requirements and candidates. For each candidate, we should calculate the affinity with all vectors to get the best fit between job’s requirements and candidates.

Number of matching processes = number of applicants \( \times NV \)

Generally, the diversification problem has been shown to be NP-hard. Thus, to solve large instances of the problem, we need to rely on heuristics (Drosou & Pitoura, 2010). In all metaheuristics, the contradictory criteria must be taken into account: exploration of the search space and diversification as well as exploitation of the best solutions (Talbi, 2011). The complexity of computing \( n! \) certainly takes an exponential time (Cheng, 2004). Whereas, the complexity of NP-hard problems represents the set of all decision problems that can be solved by a nondeterministic algorithm. In our case as shown, it requires an exponential time to be solved in optimality. Finally, the required search time to solve a given problem is an important issue in the selection of an optimization algorithm. Unlike traditional algorithms, metaheuristics allow to tackle large-size problem instances by delivering satisfactory solutions in a reasonable time (Talbi, 2011).

In order to apply an immune model to solve a specific problem, first we should choose the immune algorithm according to the type of problem that is being resolved. Then, determined the elements involved in the problem and how they can be modeled as objects in the immune model. To model such objects, a representation for these elements should be selected, specifically a string representation: binary, integer or real-valued vector representation or a hybrid representation. Next, suitable affinity measure should be selected to define the matching rules (Dasgupta, 2006). The following sub-sections introduce the details of the strategies used in the proposed algorithm.

4.1. Metaphors and Parameters

This section describes the metaphors and parameters that will be used throughout the rest of this article, particularly as they used in the Clonal Selection Algorithm that applied in Job Recommender System named JRS-CSA. JRS-CSA is a simulation of immune system where the antigens attacking our body can stimulate the immune system to produce antibodies. The following lines present the metaphors and the mapping of AIS terminology to JRS-CSA.

- Body is the organization that search for candidates to be employed in an open job position.
- Antibody is the job’s requirements that protect an organization (body) from nonqualified applicants.
• Antigenis any new applicant that considered as a target to be checked.

The JRS-CSA will use the following parameters:

$AB$ represents antibodies pool.

$Ab_i$ is the current antibody

$n$ is the number of antibodies.

$AG$ represents antigens pool.

$Ag_j$ is the current antigen.

$L$ is the number of parameters (length of the vector).

$D_k$ is the distance between $Ag_j$ and $Ab_i$ for parameter $k$.

$Affinity_i$ is the total distance between a given $Ag_j$ and $Ab_i$.

$N$ is the maximum size of antibody pool.

$C$ is the set of selected clones.

$MC$ is the set of maturated clones.

$LM$ is the set of long memory cells.

$Threshold$ is the acceptance level of antigen.

$Aga$ is the set of acceptance antigens.

d% is the percentage of removed antibodies cells.

4.2. **MultiRepresentation**

Along with other metaheuristics, choosing a suitable representation is very important for the algorithm’s success. Antigens and antibodies are represented in the same way. For most problems the most observable representation is a string of numbers with length equal to the number of parameters, the position is the parameter and the value is the actual value of the parameter itself. In JRS-CSA, there are many parameters that should be represented to model the job’s requirements as well as the applicant’s resume. For example, a set of qualifications and their information, skills, languages, experience and etc. Both antigens and antibodies are represented by a vector of mix numbers (binary, integer or real). The information can be divided to six categories: personnel information, qualifications, language, skills, experience and etc. In general, there are more than 70 possible parameters. Most of these parameters will be used in the recommendation process. Some of the parameters have multi levels of representation such as the qualification degrees and languages. To simplify the idea, possible representation of the applicant and the job vectors are:

Applicant: \{(Applicant’ id/ Job code, (PhD degree, PhD GPA, PhD specialty), (MSC degree, MSC GPA, MSC specialty), 
(BSC degree, BSC GPA, BSC specialty), (Diploma degree, Diploma GPA, Diploma specialty), (language, language level), type of skill, experience}\}.

Applicant1: \{(1,0,0,0),(1,4.1,2),(1,4.02,2),(0,0,0),(1,3),1,5\}

The above applicant’s vector means: \{(Applicant id, no PhD degree, MSC degree with 4.1 GPA and specialty number 2 (2 refers to Electric Engineering as determined in the input screen), BSC degree with 4.02 GPA in Electric Engineering, No diploma degree, 1 refers to English language with excellent level (3 refers to excellent level), skill no 1(1 refers to programming language skills), 5 years’ experience}\}.

Job1: \{(1,1,4,5,2),(1,4,5,2),(1,4,0,2),(0,0,0),(1,3),3,5\}

The above Job’s vector means: \{(Job code, PhD degree with 4.5 GPA in Electric Engineering, MSC degree with 4.5 GPA in Electric Engineering, BSC degree with 4.0 GPA in Electric Engineering, no diploma degree required, programming language skills,English language with excellent level, 5 years’ experience}\}.

4.3. **Multilevel Similarity Measures**

The similarity or affinity is very important design choice in building an AIS algorithm, and closely coupled to the representation scheme. In JRS, the affinity is taken as the distance between a given antigen and the antibody. As illustrated in section 4.2, a string of mixed representations was used. Subsequently, the affinity is calculated using a combination of different types of similarity measures depending on the parameters’ types.
Additionally, some of the parameters need multi-level checking. It needs pretest to determine if it is required by the recruiter for the recent job or not. For example, the certain degree that the applicant has may be it is not required for a certain job; in such case this degree should not be scored. Then, the parameters distance is taken as a Boolean match 0, no match 1 as equation (1).

\[
D_K = \begin{cases} 
1 & \text{if } A_g \neq A_b \\
0 & \text{otherwise} 
\end{cases}
\]

For real attributes such as the GPA for a certain degree, the distance is calculated as a normalized distance between a given antigen and antibody as equation (2).

\[
D_K = \frac{-(A_g - A_b)}{Maxvalue} 
\]

Where Maxvalue is the maximum value for current attribute. For example, if the maximum GPA is 5 then the Maxvalue is 5.

Note: if the applicant GPA is more than the job’s required GPA it will be decreased the distance (use minus).

For integer attributes such as a certain specialty or skill, the distance is also taken as equation (1) but without pretest.

Total affinity for an antibody with a given antigen is calculated as equation (3).

\[
Affinity_{j,i} = \frac{1}{L} \sum_{k=0}^{L} D_K
\]

Where \(j\) represents the current antigen \((A_g)\), \(i\) represents the current antibody \((A_b)\), \(k\) is the current parameter and \(L\) is total number of parameters. Finally, the best affinity of an antigen is calculated as equation (4).

\[
Affinity_{j,*} = \text{Min}(Affinity_{j,1}..Affinity_{j,n})
\]

Where * is a wildcard means any antibody, \(j\) is the current antigen, and \(Affinity_{j,1}\) to \(Affinity_{j,n}\) represent affinities of the antibodies from antibody \(1\) to antibody \(n\) for current antigen.

Applying the above affinity measures in the example that introduced in section 4.2,

\text{Applicant1}: \{1,(0,0,0),(1,4.1,2),(1,4.02,2),(0,0,0),(1,3),1,5\}, and

\text{Job1}: \{1,(1,4.5,2),(1,4.5,2),(1,4.0,2),(0,0,0),(1,3),3,5\}.

We have \(Affinity_{j,i} = 0.1235\).

4.4. Cloning process

Cloning can be explained as the process of producing similar populations of genetically identical individuals. In Clonal Selection, the cells with high affinity with current antigen are selected to proliferate. The proliferation rate for any immune cell is proportional to its affinity with the selective antigen: the higher the affinity, the higher the number of offspring generated and vice-versa. The number of clones created from each of the selected antibodies is proportional to their affinity using a rank based measure. This is achieved by first sorting the set of selected antibodies in increasing order by their affinity to the antigen. Then, clones are created according to the rank. Thus, the antigen selects several immune cells to proliferate. In JRS-CSA, cloning cells produce only one clone because the identical clones do not add any benefit to the processing and will give the same resultant vectors after mutation. In such case increasing the number of clones will decrease system efficiency.

4.5. Mutation Strategy

Mutation is a very common process in AIS algorithms, and in most cases it is done in absolutely random fashion by changing each bit in the solution. It is seen as a training operator which provides a small amount of random noise. This kind of unsighted mutation is not suitable for JRS because there are constraints that should be taken into account in job requirements. Additionally, due to the diversification nature of job problem that mentioned above in section 4 we should use a customized scheme. Therefore, we have applied a specific scheme of mutation to the clone cells. Where, the number of attributes changed in the clone cells is equally for all clones using predefined sets of alternatives. For example, we use the specialties and the skills alternatives
that already accepted by the recruiter for performing the mutation process. This mechanism can be seen a somatic hypermutation that noticed as a process for optimizing the binding affinity of antibodies.

5. **AIS Algorithm for Job Recommender System**

The proposed AIS recommender system named JRS-CSA is performed one generation once all available antigens have been exposure to the antibody cells, and all the AIS steps have been performed for each antigen. Each antibody has only one clone and the mutation process performed in the same way for all selected clones. Somatic hypermutation will be implemented using different alternatives (e.g. specialties and skills) that determined as applicable choices by the job’s recruiter. The following steps illustrate the JRS-CSA.

- The initial population is generated using the job’s requirements (antibodies set) and the applicants’ resumes (antigens set). Both antibodies and antigens are encoded into vector space using the representation that illustrated in section 4.2. Additionally, the mutation pools (e.g. specialties and skills) are loaded using information prepared by the recruiter.
- While the stopping condition is not met (no more Antigen), the algorithm proceeds by performing a number of iterations to expose the system all antigens.
  - The system is exposed to the selected antigen, and the affinity is calculated for all antibodies against the antigen.
  - The \( d\% \) of antibodies is selected from the entire antibody’s pool that has the highest affinity with the antigen.
  - Cloning is performed; a number of clones \( nc \) are generated for each antibody.
  - Affinity maturation is performed (mutation) for all selected clones, such that the clones are then subjected to an affinity maturation process to enhance the matching with current antigen.
  - The resulted clones after mutation are then exposed to the same antigen, and the affinity measures are calculated.
  - The antibodies with the highest affinities in the maturated clones are then selected as candidate memory antibodies. The highest affinity with current antigen will be assigned as antigen’s affinity. The antigen’s affinity is compared with a certain threshold that specified by job’s recruiter to determine the acceptance of an antigen.
  - If the affinity of candidate antibodies cells is higher than that of the highest stimulated antigen from the memory pool, then it replaced.
  - The antibodies cells are sorted and percentage \( d\% \) of individuals with the lowest affinity is removed from antibody pool.
  - If the total number of antibodies cells is more than the maximum size of antibody pool \( (N) \), the antibodies cells with lowest affinities will be removed.
  - The memory antibodies pool then taken to expose next antigen.
- Finally, the set of antigens will be ranked depending on their affinities.

The following figure presents a diagrammatic representation of the steps AIS for job recommender system.
Loop
Until no more

Initialize AIS parameters

Select random Ag

Calculate affinities
With all Abs

Select Abs with
highest affinities

Clone & Mutate

Calculate affinities
for all clones

Select clones with
highest affinities

Assign Ag affinity to the
closest clone with highest
affinity

Add clones with
highest affinities to
memory pool

Remove d % Abs
with lowest affinities
from Abs pool

Ranked Antigens
(applicants)

Figure 2: Diagrammatic representation of JRS-CSA for job recommender system.
The following lines of Table 1 introduce the pseudo code for JRS-CSA and use the parameters presented in section 4.1.

Table 1: The pseudo code for JRS-CSA.

<table>
<thead>
<tr>
<th>Algorithm JRS-CSA</th>
</tr>
</thead>
<tbody>
<tr>
<td>1: Initialize parameters and generate AB, AG, and mutation pools</td>
</tr>
<tr>
<td>2: for j = 0 to m-1 do</td>
</tr>
<tr>
<td>3: Select (Ag[j])</td>
</tr>
<tr>
<td>4: for i = 0 to n-1 do</td>
</tr>
<tr>
<td>5: Select (Ab[i])</td>
</tr>
<tr>
<td>6: CalculateAffinity(Ag[j],Ab[i]) using equations 1,2,3</td>
</tr>
<tr>
<td>7: end for</td>
</tr>
<tr>
<td>8: AB ← Sort (AB) according to the affinity</td>
</tr>
<tr>
<td>9: C ← Clone (AB)</td>
</tr>
<tr>
<td>10: Select the d% represents that the best antibodies</td>
</tr>
<tr>
<td>11: for i = 0 to nc - 1 do</td>
</tr>
<tr>
<td>12: MC ← Mutate(C)</td>
</tr>
<tr>
<td>13: end for</td>
</tr>
<tr>
<td>14: for i = 0 to nmc - 1 do</td>
</tr>
<tr>
<td>15: CalculateAffinity (Ag[j],MC[i]) using equations 1,2,3</td>
</tr>
<tr>
<td>16: end for</td>
</tr>
<tr>
<td>17: LM ← Add (MC,LM)</td>
</tr>
<tr>
<td>18: Sort (LM) according to the affinity</td>
</tr>
<tr>
<td>19: Affinity [Agj] = Min (Affinity [Ab0]...Affinity [Abn-1])</td>
</tr>
<tr>
<td>20: using equation 4</td>
</tr>
<tr>
<td>21: if Affinity[Agj] &lt;= Threshold then</td>
</tr>
<tr>
<td>22: Aga ← Add (Ag[j],Aga)</td>
</tr>
<tr>
<td>23: end if</td>
</tr>
<tr>
<td>24: LM ← Remove d% of antibodies with lowest affinities from the LM set</td>
</tr>
<tr>
<td>25: for i = 0 to nlm do</td>
</tr>
<tr>
<td>26: AB ← Add (lm[i],AB)</td>
</tr>
<tr>
<td>27: end for</td>
</tr>
<tr>
<td>28: Sort (AB)</td>
</tr>
<tr>
<td>29: if n &gt; N then</td>
</tr>
<tr>
<td>30: if i &lt;= N then</td>
</tr>
<tr>
<td>31: for i = 0 to n</td>
</tr>
<tr>
<td>32: if i &lt;= N then</td>
</tr>
<tr>
<td>33: AB ← Add(Ab[i],AB)</td>
</tr>
<tr>
<td>34: end if</td>
</tr>
<tr>
<td>35: end if</td>
</tr>
</tbody>
</table>
5.1. System Implementation

From the algorithmic outlines given in previous sections, we construct a recommendersystem written in the c# programming language. It provides data mapping into vector space and Clonal Selection processes as well as, antigens classification algorithm. Visualization tools used to test the program result. The JRS-CSA implementation was prepared as defined in sections 45 and 5. The cloning and somatic hypermutation was implemented as specified above, as well as the affinity calculations were performed using equations 1,2,3,4 that mentioned in section 4.3.

5.2. An Illustrative Example

This example used a simple antibodies pool with four vectors for a specific job position. Minimal parameters were included in this example to simplify the concepts. Additionally, we use only the specialties pool for the mutation process with few numbers of alternatives. A design goal of JRS-CSA is recommended the recognized antigens that represent the qualified applicants. Figure 3 represents four antibodies for a specific job position. The variables A₁ to A₁₆ refer to the parameters that mentioned in section 4.2.

<table>
<thead>
<tr>
<th>Ab₁</th>
<th>A₂</th>
<th>A₃</th>
<th>A₄</th>
<th>A₅</th>
<th>A₆</th>
<th>A₇</th>
<th>A₈</th>
<th>A₉</th>
<th>A₁₀</th>
<th>A₁₁</th>
<th>A₁₂</th>
<th>A₁₃</th>
<th>A₁₄</th>
<th>A₁₅</th>
<th>A₁₆</th>
<th>Affinity</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>4.5</td>
<td>2</td>
<td>1</td>
<td>4.5</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>7</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>10</td>
</tr>
</tbody>
</table>

**Figure 3:** Set of antibodies for a specific job position.

Figure 4 displays the specialties for the current job, such that for PhD the accepted specialty only Computer Science and Telecommunication, MSc are Computer Science and Electric Engineering and etc.

<table>
<thead>
<tr>
<th>Sp-Code</th>
<th>Name</th>
<th>PhD</th>
<th>MSc</th>
<th>BSc</th>
<th>Dip</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Computer Science</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>Electric Engineering</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Telecommunication</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Business Administration</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

**Figure 4:** The Specialties alternatives for current job.

When A₀ enters to the system with d%=50, the following list of Abs will be selected as Clone C cells (table 1):

<table>
<thead>
<tr>
<th>Ab₁</th>
<th>A₁</th>
<th>A₂</th>
<th>A₃</th>
<th>A₄</th>
<th>A₅</th>
<th>A₆</th>
<th>A₇</th>
<th>A₈</th>
<th>A₉</th>
<th>A₁₀</th>
<th>A₁₁</th>
<th>A₁₂</th>
<th>A₁₃</th>
<th>A₁₄</th>
<th>A₁₅</th>
<th>A₁₆</th>
<th>Affinity</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>5</td>
<td>0.1235</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>4.5</td>
<td>2</td>
<td>1</td>
<td>4.5</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>5</td>
<td>0.12975</td>
</tr>
</tbody>
</table>

**Figure 5:** Set of selected Clone C.
Then, the following mutated cells $MC$ will be produced.

The $LM$ cells appear in figure 7:

![Table](image)

The minimum affinity will be assigned to the current antigen; such that $A_{g1} = 0.1235$. The produced $LM$ cells compared with maximum antibodies in the system if it exceeds the maximum then the antibodies cells with low affinities will be removed. The resultant $AB$ cells that will be introduced to the next antigen appear in figure 8.

![Table](image)

Figure 6: The mutated cells $MC$.

Figure 7: The resultant $LM$ cells.

Figure 8: The resultant AB cells.
When 10 antigens exposed to the system, they ranked by affinities as the following set in figure 9. The Test column represents the acceptance result of the antigen with Threshold = 0.15, (Checked: accepted, Unchecked: rejected).

![Table](image)

**Figure 9:** result of 10 antigens that selected by the system.

### 6. Conclusion

The Artificial Immune System established an emergent biologically inspired computing paradigm. Whereas the principles that extracted from the immune system used to design computational algorithms to solve many engineering problems. It has been exposed to be a useful solution for complex problems such as optimization, fault detection, pattern recognition, classification and many other problems. Our proposed algorithm represents an optimization technique based on the abstractions of the Clonal Selection and somatic hypermutation theories. We used the observable immune components and processes as metaphors to produce AIS algorithm. This algorithm encapsulates a number of beneficial features of the natural immune system, and they are utilized towards solving job recommendation problem. These features include optimization, matching, diversity, and learning capability.

### References


