

# Clonal Selection based Genetic Algorithm for Workflow Service Selection

Simone A. Ludwig  
North Dakota State University  
Fargo, ND, USA  
simone.ludwig@ndsu.edu

**Abstract**—Quality of Service (QoS) aware service selection of workflows is a very important aspect for service-oriented systems. The selection based on QoS allows the user to include also non-functional attributes in their query, such as availability and reliability. Several exact methods have been proposed in the past, however, given that the workflow selection problem is NP-hard, approximate algorithms can be used to find suboptimal solutions for requested workflows. Genetic algorithm is one such method that can find approximate solutions in the form of services selected. In this paper, we propose an improved version of the standard genetic algorithm approach by making use of the clonal selection principle from artificial immune systems. Experimental results show that the clonal selection based genetic algorithm achieves much higher fitness values for the workflow selection problem than standard genetic algorithm.

## I. INTRODUCTION

Service-oriented computing is seen as the *de facto* standard for distributed enterprise applications in supporting collaborative business processes and promoting science, research and technology [1]. The idea behind service-oriented computing is that businesses offer their application functionality as services over the Internet, and users or companies can make use of these services by composing and integrating these services into their applications. Service-oriented architecture is the concept that combines this idea, and the basic components making up the so-called “service-oriented triangle” are providers, requesters and registries. The service provider publishes the service description in a service registry, and a service requester can query a service from the registry and dynamically bind it to one of the services that are returned by the search query. This service-oriented triangle can then be implemented using existing technologies from the web services stack [2].

The main idea of service-orientation is to compose these services by discovering them and then dynamically invoking them when building applications, rather than building them from scratch or reusing other applications. The result of this composition process is referred to as a composite service. Choreography describes the message interchanges between the different participants in service-oriented systems. It provides the global distributed model of message exchanges without the need of a central coordinator [3].

One primary requirement of service-oriented systems is the ability of self-adaptivity [4]. Self-adaptivity in the area of service-oriented environments implies that the system should be able to adapt its behavior depending on the changes within

its environment. A possible solution for this adaptability, in particular for service composition, is the concept of QoS. All non-functional attributes of a service such as performance-specific attributes are described by QoS.

QoS attributes generally considered are cost, availability, response time, and reliability. Cost is defined as the fee to be paid to the service provider by the service requester for executing a particular service. The cost is always associated with the value of the service functionality, i.e. the more complex the function it provides, the higher the service cost. Availability of a web service is the probability that the service operation is accessible. This is defined by the proportion of the service’s uptime and downtime. Response time is the expected delay between the time instant when a request is sent and the time when the result is obtained. Reliability is a measure of the service invocation trustworthiness. It is defined as the ratio between the numbers of service invocations that comply with the negotiated QoS over the total number of service invocations.

This paper proposes a clonal selection based Genetic algorithm (CSGA) approach that makes use of an idea used from artificial immune systems applying a clonal selection method that guides the stochastic genetic search. The clonal selection method uses the concept of affinity to evaluate the selected services based on its fitness and clones the ones with higher affinity values. This leads to improved solution quality applied to the workflow service selection problem.

The remainder of the paper is structured as follows. Related work is outlined in the next section. Section III describes the proposed clonal selection based genetic algorithm approach. In Section IV the experiments and results are presented, followed by the conclusions summarized in Section V.

## II. RELATED WORK

Related work in the area of web service selection include approaches using trust and reputation mechanisms, computational intelligence methods, QoS-based selection techniques and fuzzy approaches.

An adaptive service selection approach to service composition that bars problematic external services to be used in a service-oriented application of an organization is introduced in [5]. Service composition can be formed using the highly referenced services by using dynamic WSDL (Web Service Description Language) information in public service registries

to approximate a snapshot of a network of services, and apply link analysis on the snapshot to identify services that are currently more popular by different consumers.

An evidence-based scheme for web service selection is proposed to estimate the degree of consumer trust in a particular service by considering consumers' direct experience and indirect recommendation of the service in [6]. The proposed model enables deception detection and excludes the fraudulent evidence of malicious evaluators from the selection process.

In [7], a reputation measure approach of web services is introduced to enhance the reputation measure accuracy. Three phases, feedback checking, feedback adjustment, and malicious feedback detection, have been employed to improve the service selection process in order to obtain reliable services.

A new approach for designing and developing a QoS ontology and its QoS-based ranking algorithm for web services are introduced in [8]. The QoS ontology is used to support QoS descriptions and facilitates service participants to express their QoS offers and demands. Analytic Hierarchy Process (AHP) is adopted in the QoS-based ranking algorithm to rank the web services with similar functionality in order to select the best services for a request.

A web service selection framework by introducing a QoS-based cost function concept is proposed for composite distributed web services in [9]. The proposed approach composes a few web services chosen from a given service community by their cost function to find the optimal composite service.

A novel ANN-based (Artificial Neural Network) service selection algorithm (ANNSS) is proposed and applied to a ubiquitous web service environment in [10]. The users can acquire effective guidance, and choose the most appropriate service evaluated by an ANN-based evaluation standard. A three-term method that improves the traditional back propagation algorithm is used to satisfy the requirements of time issue in real-time systems.

Linear-programming, an exact optimization method, has been applied to the workflow service selection problem in [11] and [12]. Another implementation has used complex workflow patterns to address the service selection problem [13]. Linear programming is used to solve the optimization problem using an aggregation function for different QoS attributes.

However, one problem the exact optimization methods have is the larger time-complexity. Therefore, approximate techniques can be used to improve the execution time of the selection process, however, only sub-optimal assignments of the services based on the service requests are likely to be found. One such approximate method is Genetic Algorithm (GA).

A stochastic demand multi-product supplier selection model with service level and budget constraints using a GA is described in [14]. The relationship between the expected profit and the number of trials, and between the expected profit and the combination of mutation and crossover rates, are studied to identify better parameter values to run the GA.

A new cooperative evolution algorithm, which consists of Stochastic Particle Swarm Optimization (SPSO) and simulated

annealing, is introduced to solve the service selection problem in [15]. The approach resolves the service selection with multi-objective and QoS global optimization. The proposed cooperative evolution algorithm owns better global convergence ability with faster convergence speed, in addition, the multi-objective SPSO is feasible and efficient.

In [16], a system for supporting the user in the discovery of semantic web services is used to model an ad-hoc service request by selecting conceptual terms rather than using strict syntax formats. The selection exploits the fuzzy formal concept analysis to request the system to return a list of semantic web services that match the user query.

In [17], a service selection method based on the technique for Order Preference by Similarity to an ideal Solution (TOPSIS) with fuzzy opinions is used to evaluate the weights of various criteria and the rating of each alternative web service. The approach uses triangular fuzzy membership functions to represent the weights of criteria and the ratings of web services.

In [18], a fuzzy-based UDDI (Universal Description, Discovery and Integration) with QoS support is proposed to consider the non-functional quality of QoS information for personalized web service selection. This approach considers the objective factors described by service providers, and subjective information with trustability evaluations from users by adapting GAs to learn the user preferences, and to apply fuzzy logic to make decisions. The users can determine the most suitable web service with a fuzzy query interface to provide subjective and objective factors.

A decision model under consumer's vague perception of intuitionistic fuzzy set for QoS-aware web services selection is proposed in [19]. The selection method is modeled as a fuzzy multi-criteria decision-making problem by considering the non-functional QoS properties that heavily rely on the perceptions of service providers and consumers.

In this paper, we are using an approximate technique, namely GA, to optimize the workflow selection process. Furthermore, the selection of services is based on QoS parameters as described in the following section. We combine the clonal selection with GA in order to further enhance and speed up the workflow selection process.

### III. QoS-BASED WORKFLOW SELECTION

#### A. Workflow Example

When a service requester requires a specific functionality, which cannot be provided by one single service, the composition of multiple services needs to be done thereby creating a workflow. In addition, the composition of web services should not only be functionally compatible, but should also be compatible with regards to the defined service levels. In order to show how abstract workflows are provided based on the users request, the following sample workflow as shown in Figure 1 is provided, showing the abstract as well as the concrete services.

The figure displays an example of the processing of an image. The process is as follows: the image is first read in

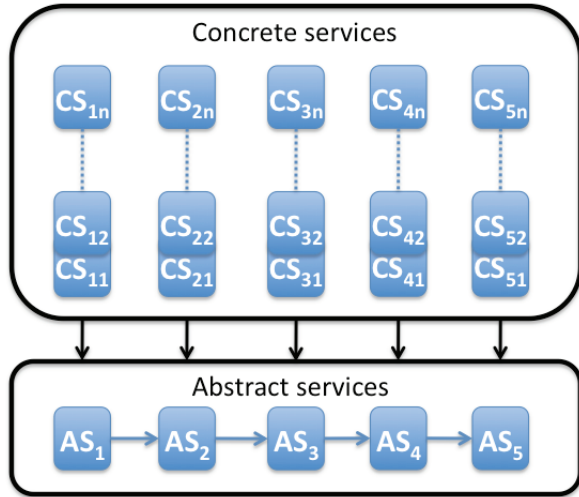


Fig. 1. Example of an image processing workflow

(AS1), and converted to Grey scale (AS2), then it is thresholded (AS3), i.e., image pixels less than a certain amount is set to black and everything higher is set to white. The difference is then taken between this image and an image where the white lines are thinned out, i.e., detail is taken away (AS4). The resulting image is produced, by taking the difference between the detail of the white parts, which were pruned by the ShrinkWhite unit resulting in an outline of an image, and saved as a new compound component (AS5). The concrete services (CS<sub>xy</sub>) for each abstract service (AS<sub>x</sub>) provide the same functional, but different non-functional properties; i.e., QoS attributes. Selecting the services of a workflow based on QoS parameters requires an algorithm that can optimize the assignment of concrete services for a workflow for a given abstract workflow description. Furthermore, we are considering that multiple requests are being served at the same time. Given that performance in terms of execution time in a service-oriented environment is of essence, we argue that we do not need to have optimal assignments, but close to optimal assignments should be found within a reasonable time.

### B. QoS Criteria

There are many measures available for different QoS criteria, however, we consider the following four generic quality criteria for single services, also referred to as QoS parameters: availability, reliability, execution time, and cost.

- The availability of a service is the fraction of time that the service is accessible. It basically measures the fraction of the total amount of time in which the service is available during the last defined period of time (threshold is set by administrator).
- The reliability of a service is the fraction of requests correctly responded to within a maximum expected time frame. Reliability is a measure related to the hardware and software configuration of web services and their network connections. Reliability values are computed

from past data measuring the successful executions in relation to the overall number of executions.

- The execution time denotes the expected delay in seconds from the moment a request is made until the moment when the results are returned. Services advertise their processing time or provide methods to inquire about it.
- The execution cost represents the amount of money a user has to pay for executing a service. Web service providers usually advertise the execution price directly or they provide methods to inquire about it.

However, in this study we are not only concerned with single services but with complete workflows, and therefore, the QoS parameters of the single services have to be aggregated. We assume in our study that we are only using sequential workflows. Therefore, the availability and reliability of a workflow is calculated as the product of each single services availability and reliability respectively. The cost and time of a workflow is the average of each single services cost and time respectively. Our goal is to maximize the overall QoS not only for one workflow, but for several workflows simultaneously.

## IV. APPROACHES

### A. Genetic Algorithms (GA)

GA [20] is a global optimization strategy that models natural evolution. A GA works on a population of chromosomes that is created as a set of possible solutions to the optimization problem. Within a generation of a population, the chromosomes are randomly altered by processes of selection, crossover and mutation in order to create new chromosomes that have better evaluation scores. The next generation population of chromosomes is randomly selected from the current generation with selection probability based on the evaluation score of each chromosome.

In this study, each chromosome has the following structure:

NoS	CS1	CS2	CS3	NoS	CS1	CS2	CS3	CS4	...
2	13	26	31	1	18	23	37	42	...

A gene within the chromosome consists of integers, whereby the first number characterizes the number of services within the workflow (NoS), and the following integers are the concrete services of the workflow. The first digit from the left of a concrete service (CS) characterizes the abstract service number; the second describes the specific concrete service implementation. The number of services the abstract workflow consists of determines the length of the gene. For example, the first workflow consists of 3 services, the second of 4 services, etc. For this study, we consider workflows up to 5 services, having 10 concrete services available for each of the 10 abstract services.

At the beginning, the first population is randomly initialized. After that, the fitness of the individuals is evaluated using the fitness function. After the fitness is evaluated, individuals have to be selected for pairing. The selection method used is tournament selection. Always two individuals are paired, resulting in an offspring of two new individuals. In the pairing phase, a

random crossover mask is used, i.e. the positions (dimensions) for which crossover occurs are selected randomly. If crossover occurs at certain positions (dimensions), individuals that are mated exchange their values at that position and the resulting individuals are used as offspring. The crossover has to make sure that the offspring present a valid match. Therefore, if two values are exchanged, other positions in the two match vectors are usually effected as well. The offspring faces mutation with a certain low probability. After mutation, the fitness of the offspring is calculated. Then, either all individuals from the last generation compete against the whole offspring, or the offspring only compete with its corresponding parents. In this implementation, all individuals from the old generation compete with all individuals in the new generation. After the new generation is selected, the GA will start over, and continue with parent selection and crossover.

### B. Artificial Immune Systems and Clonal Selection

The human immune system is seen as a parallel and distributed adaptive system with information processing abilities that the artificial immune system (AIS) community captures to some extent. AIS is also sometimes referred to as immunocomputing [21][22]. Some of AIS's computational models and algorithms have established themselves as immune clonal algorithm, immune network system, negative selection algorithm, etc. [23][24]. These models can be applied to a wide variety of application areas ranging from robotic control, recognition, data mining, information security, abnormal detection and diagnosis etc.

According to Burnet's clonal selection theory [25], the immune system undergoes a selection mechanism during the lifetime of the individual. The theory states that activation of lymphocytes occurs when the binding with a suitable antigen happens. Once activated, clones of the lymphocyte are produced expressing identical receptors to the original lymphocyte that encountered the antigen. This process ensures that only lymphocytes specific to an activating antigen are produced in large numbers.

The first popular selection algorithm devised is CLONALG (clonal selection algorithm) [26]. This algorithm selects the fittest antibodies and clones them proportionally to their antigenic affinities. There is a hypermutation operator that performs an affinity maturation process that is inversely proportional to the fitness values that generate the matured clone population. An improved clonal selection algorithm that deals with numerical optimization problems was introduced in [27]. The learning mechanism in immune systems is simulated by the Baldwinian learning operator by employing information from within the antibody population to alter the search space.

### C. Clonal-Selection based Genetic Algorithm (CSGA)

The clonal selection operation that we include in our GA implementation to replace the tournament selection can be described as follows with the artificial immune systems' terms used:

- **Antigen and antibody:** An antigen is any substance that causes the immune system to produce antibodies against. The synonym for antigen is a decision variable, and in our case the QoS value.
- **Crowd distance:** The crowd distance between two antibodies are computed in a four-dimensional space  $\Theta = \{availability, reliability, time, cost\}$  composed of the QoS attributes. Since QoS attributes are normalized within the range of  $[0, 1]$ , the distance between the antibodies  $x, y \in [0, 1]^4$  is as follows:  $\rho$  is the distance function and determines the difference between  $X$  and  $Y$ , and  $\omega_\theta$  are the different weight values that can be placed on the different QoS parameters.

$$\rho = \sqrt{\sum_{\theta \in \Theta} \omega_\theta (X_\theta - Y_\theta)^2} \quad (1)$$

- **Affinity:** Affinity is used to evaluate the workflow (antibody) based on its fitness and crowd distance as compared to the global optimum.

$$affinity_m = \frac{fitness_m}{\rho_m + \alpha} \quad (2)$$

where  $\rho_m$  ( $m = 1, \dots, N$ ) stands for the distance between the  $m^{th}$  workflow task (antibody) and the global optimal service, and  $N$  is the population size. An antibody with a higher fitness and shorter distance has a higher affinity. Therefore, affinity is only a better measure than fitness if the balance of population diversity and selection pressure is concerned.  $\alpha$  is a small constant and is set to 0.1.

- **Fitness function:** The fitness of several workflows being optimized simultaneously is computed as follows:

$$fitness = \max \sum_{k=1}^K (\omega_1 \prod_{j=1}^n Q_{1jk} + \omega_2 \prod_{j=1}^n Q_{2jk} + \omega_3 \sum_{j=1}^n Q_{3jk} + \omega_4 \sum_{j=1}^n Q_{4jk}) \quad (3)$$

where  $Q_{ijk}$  ( $i = 1, 2, 3, 4$ ) are the QoS attribute values (*availability, reliability, time, cost*) of the  $j^{th}$  service of the  $k^{th}$  workflow respectively;  $n$  is the number of services in a workflow;  $\omega_i = 1$  and  $0 < \omega_i < 1$  ( $\omega_i$  of the four QoS attributes); and  $K$  is the number of workflows to be optimized.

- **Cloning:** Antibodies with higher affinity are more favorable and therefore should be proliferated in order to sustain the superior properties of the population. The clone of an antibody is decided by the following equation:

$$num_m = \sigma P \times \left[ \frac{fitness_m}{\sum_{m=1}^c fitness_m} \right] \quad (4)$$

where  $\sigma$  is a constant that determines how many times an antibody is cloned within population  $P$ .

- **Hypermutation:** The hypermutation operator takes care of the extensive search for the best antibodies. Random mutation is performed for the assignment of services



for elitism and efficiency. First of all,  $\sigma P$  antibodies are cloned based on Equations (2) and (4) from the current population. Then, the hypermutation operation is performed on every service with a certain mutation probability. One candidate service is randomly selected from all the possible services available, and this service then replaces the current service.

## V. EXPERIMENTS AND RESULTS

### A. Experimental Setup

GA and CSGA as introduced in the previous sections were implemented, as well as the Munkres algorithm [28]. Given that Munkres is an exact optimization algorithm it achieves an optimal assignment that is used as the baseline for the fitness values. Experiments were designed to measure the fitness and the execution time.

The algorithms were analyzed with regard to the number of generations used, as well as the scaling of the workflow size. All measurement points displayed in the graphs are results averaged over 30 independent runs. The data sets for the workflows and services were randomly generated, whereby workflows were created consisting of up to 5 abstract services, out of a pool of 10 concrete services for each of the 10 available abstract services (equals 100 concrete services). Please note that we assume that a particular concrete service can be used in several workflows. There is no maximum number given regarding how often one particular service can be selected simultaneously.

The following parameter settings have been chosen due to their superior performance on the workflow composition problem. The GA/CSGA settings were: population size = 100, mutation probability = 0.05, crossover probability = 0.7, size of the tournament selection = 4, the number of positions that are selected for crossover = 0.1,  $\sigma = 2$  (clone size is double to the population size), hypermutation probability = 0.05. The experiments were conducted on an Intel Xeon 5880 processor running at 3.33 GHz running the Java Version 1.6.2 JDK Runtime Environment.

### B. Experiments and Results

The first measurements were conducted measuring the fitness and the execution time with increasing generations. The number of workflows to be optimized was 300. Figure 2 shows the fitness of GA and CSGA. As can be seen, the optimization of the fitness is much better for CSGA than the GA. After 1,000 generations, CSGA achieves a fitness of almost 99%, whereas GA only reaches a value of 95%.

In terms of execution time as shown in Figure 3, given that CSGA runs the additional clonal selection procedure that is computationally more expensive than the tournament selection used for GA, CSGA takes more time to compute than GA. It takes GA only about 60 seconds to run the optimization with 1,000 generations, and CSGA needs more than 90 seconds to compute. We can see that there is the trade-off between solution quality and execution time.

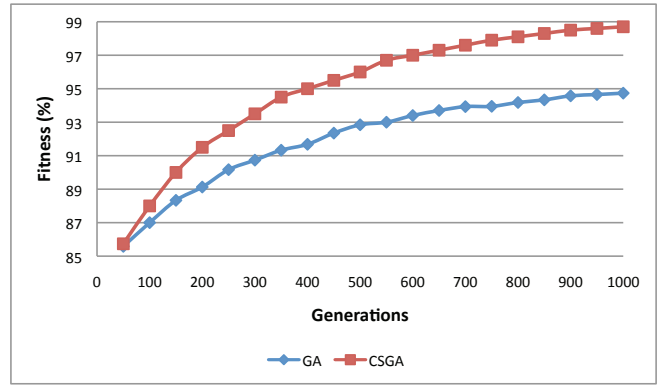


Fig. 2. Fitness vs. generations.

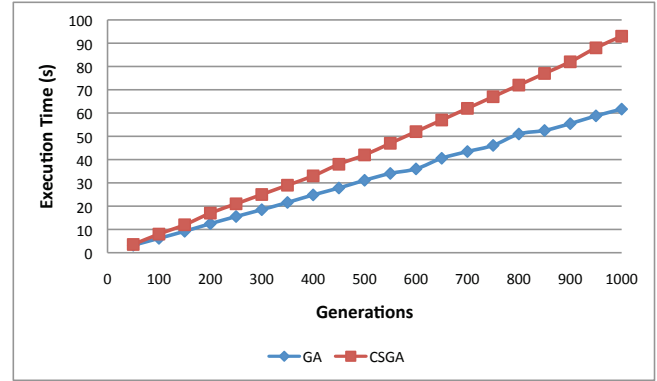


Fig. 3. Execution time vs. generations.

Given that the problem size, in our case, the workflow size is of importance, several different workflow sizes were investigated. The workflow sizes chosen were ranging from 50 to 1,000 with increases of 50 (generation was set to 1,000). Figure 4 shows the fitness of GA and CSGA. The fitness of both GA and CSGA decreases with increasing workflow sizes, however, GA decreases more rapidly than CSGA. For 800 workflows GA's fitness has decreased to approximately 91.5% whereas CSGA's fitness is 93%.

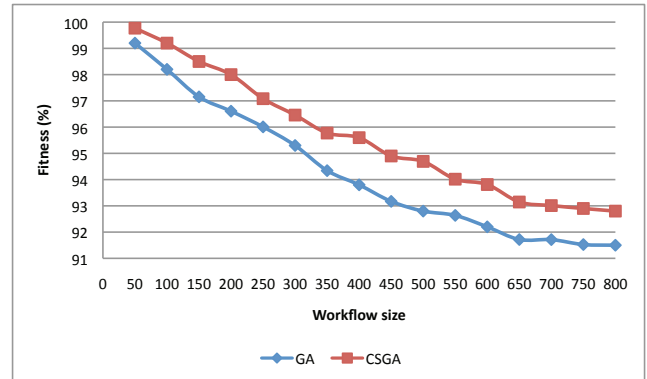


Fig. 4. Fitness vs. workflow size.

The execution time with increasing workflow sizes is il-

illustrated in Figure 5. Again, it can be observed even though GA and CSGA are comparable for smaller workflow sizes, however, for larger workflow sizes CSGA has much larger execution times. The ratio for the workflow size of 800 is 2.5.

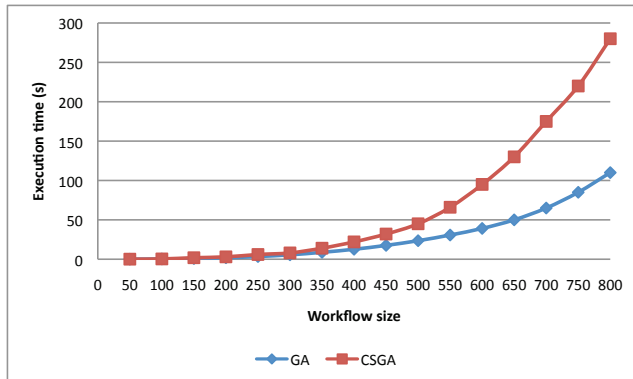


Fig. 5. Execution time vs. workflow size.

Another set of experiments was carried out measuring the generations needed to achieve a fitness of 95% (this value is chosen since an optimization of 95% seems reasonable for the workflow selection problem). Table I shows the generations needed to reach the predefined fitness for different workflow sizes, showing that CSGA needs less generations than GA.

TABLE I  
GENERATIONS NEEDED TO ACHIEVE A FITNESS OF 95%

	Workflow size of				
	100	200	300	400	500
GA	114	389	723	1146	1544
CSGA	57	148	337	789	1103

## VI. CONCLUSIONS

This paper addressed the service composition task using GA to achieve the service selection of workflows. These types of optimization problems were mainly addressed in the past by linear programming methods. All past approaches only optimized one workflow request at a time, whereas our approach serves several requests simultaneously. We argue that in service-oriented architectures, several user requests need to be satisfied at any given time. In addition, given that scalability is a major issue in service-oriented environments, an approximate method should be employed for this selection problem. Therefore, a GA approach was investigated for its suitability. Given that GA suffers from premature convergence, we also implemented a clonal selection based GA combining the idea of clonal selection from artificial immune systems with GA.

As shown by the experimental results, CSGA produces much higher fitness values than GA, however, with the drawback of longer execution times. Looking at the results from a generations point-of-view demonstrates that if a certain predefined fitness is necessary, CSGA needs far less generations than GA for the optimization process. This clearly

demonstrates that CSGA outperforms GA in terms of solution quality.

Future work will follow two directions. First of all, since the predefined workflows and services were fairly similar in range, the effect of larger variations of services in terms of QoS will be investigated. Secondly, given that time is of essence in service-oriented environments the parallelization of CSGA will be conducted in order to improve the execution time of the algorithm.

## ACKNOWLEDGMENT

This paper is partly based on research supported by North Dakota EPSCoR and National Science Foundation Grant EPS-0814442.

## REFERENCES

- [1] S. Dustdar and M.P. Papazoglou, "Services and Service Composition - An Introduction", *Information Technology*, 50(2/2008), Feb. 2008.
- [2] S. Weerawarana, F. Curbera, F. Leymann, T. Storey, and D.F. Ferguson, "Web Services Platform Architecture: SOAP, WSDL, WS-Policy, WS-Addressing, WS-BPEL, WS-Reliable Messaging, and More", Prentice Hall, 1st edition, 2005.
- [3] J.M. Zaha, M. Dumas, A.H.M. ter Hofstede, A.P. Barros, and G. Decker, "Service Interaction Modeling: Bridging Global and Local Views", *Proceedings of the 10th IEEE International Enterprise Distributed Object Computing Conference (EDOC06)*, Hong Kong, China, October 2006.
- [4] G. Denaro, M. Pezze, and D. Tosi, "Designing Self-Adaptive Service-Oriented Applications", *Proceedings of the Fourth International Conference on Autonomic Computing (ICAC07)*, Jacksonville, FL, USA, 2007.
- [5] L. Mei; W.K. Chan; T.H. Tse, "An Adaptive Service Selection Approach to Service Composition", *Proceedings of IEEE International Conference on Web Services*, Sept. 2008.
- [6] P. Wang, K. Chao, C. Lo, R. Farmer, "An Evidence-Based Scheme For Web Service Selection", *Information Technology and Management*, 12.2:161-172, 2011.
- [7] S. Wang, Q. Sun, H. Zou, and F. Yang, "Reputation measure approach of web service for service selection", *IET Software* 5, no. 5: 466-473, 2011.
- [8] V. Tran, H. Tsuji, R. Masuda, "A new QoS ontology and its QoS-based ranking algorithm for Web services", *Simulation Modelling Practice and Theory*, 17(8):1378-1398, 2007.
- [9] A.E. Saddik, and D. Liu, "QoS based Selection Algorithms for Composite Distributed Web Services", *Journal of Interconnection Networks*, 10.4:421-434, 2009.
- [10] H. Cai, X. Hu, Q. Lu, Q. Cao, "A Novel Intelligent Service Selection Algorithm And Application For Ubiquitous Web Services Environment", *Expert Systems with Applications*, 36.2:2200-2212, 2009.
- [11] T. Yu, Y. Zhang, and K.J. Lin, "Efficient Algorithms for Web Services Selection with end-to-end QoS Constraints", *ACM Transactions on the Web*, 1(1), 2007.
- [12] D. Ardagna and B. Pernici, "Adaptive Service Composition in Flexible Processes", *IEEE Transactions on Software Engineering*, 33(6):369384, 2007.
- [13] D. Schuller, J. Eckert, A. Miede, S. Schulte, R. Steinmetz, "QoS-Aware Service Composition for Complex Workflows", *Proceedings of the 2010 Fifth International Conference on Internet and Web Applications and Services*, 2010.
- [14] P. Yang, H. Wee, S. Pai, Y. Tseng, "Solving a Stochastic Demand Multi-Product Supplier Selection Model with Service Level and Budget Constraints using Genetic Algorithm", *Expert Systems with Applications*, 38.12:14773-14777, 2011.
- [15] X. Fan, X. Fang, and C. Jiang, "Research on Web Service Selection Based on Cooperative Evolution", *Expert Systems with Applications*, 38.8:9736-9743, 2011.
- [16] G. Fenza, and S. Senatore, "Friendly Web Services Selection Exploiting Fuzzy Formal Concept Analysis", *Soft Computing - A Fusion Of Foundations, Methodologies and Applications*, 14.8:811-819, 2010.

- [17] C. Lo, D. Chen, C. Tsai, K. Chao, "Service Selection Based on Fuzzy TOPSIS Method", *Advanced Information Networking and Applications Workshops, International Conference on*, pp. 367-372, *Proceedings of IEEE 24th International Conference on Advanced Information Networking and Applications Workshops*, 2010.
- [18] H. Wang, C. Lee, and T. Ho, "Combining Subjective and Objective QoS Factors for Personalized Web Service Selection", *Expert Systems with Applications*, 32.2:571-584, 2007.
- [19] P. Wang, "QoS-Aware Web Services Selection With Intuitionistic Fuzzy Set Under Consumers Vague Perception", *Expert Systems with Applications*, 36.3:4460-4466, 2009.
- [20] J.H. Holland, "Adaptation in Natural and Artificial Systems", University of Michigan Press, Ann Arbor, 1975.
- [21] L.N. De Castro, J.I. Timmis, "Artificial Immune Systems: A New Computational Intelligence Paradigm", Springer-Verlag, 2002.
- [22] T. Burczynski, "Information Sciences Special Issue on Artificial Immune Systems", *Information Sciences*, 179:1377-1378, 2009.
- [23] X.B. Cao, H. Qiao, Y.W. Xu, "Negative selection based immune optimization", *Advances in Engineering Software*, 38:649-656, 2007.
- [24] D. Dasgupta, L.F. Nino, "Immunological computation: theory and applications", CRC Press, 2008.
- [25] F.M. Burnet, "The Clonal selection theory of acquired immunity", Vanderbilt Univ. Press, Nashville TN, 1959.
- [26] L.N. de Castro, F.J. Von Zuben, "Learning and optimization using the clonal selection principle", *Evolutionary Computation, IEEE Transactions on Evolutionary Computation*, vol. 6, no. 3., pp. 239-251, 2002.
- [27] M.G. Gong, L.C. Jiao, L.N. Zhang, "Baldwinian learning in clonal selection algorithm for optimization", *Information Sciences*, 180:1218-1236, 2010.
- [28] H.W. Kuhn, "The Hungarian method for the assignment problem", *Naval Research Logistics*, 52(1), 1955.