A Pareto Optimal Solution for Secure-Low Cost Web Service Composition in multi-cloud Market

Mirsaeid Hosseini Shirvani
Department of Computer Engineering
Islamic Azad University, Sari Branch
Sari, IRAN
Email: mirsaeid_hosseini@iausari.ac.ir

Abstract—**Although the cloud computing is a promising paradigm, there are still several challenges. For instance, a single cloud cannot meet all category user requests. The reason why multi-cloud market attracted a lot of attention in literature. Therefore, by utilizing multi-cloud and web service composition technology adoption, a user/company can fulfil its complicated business process. In multi-cloud market, every cloud provider publishes its web services with especial pricing schemes and security level. In this article, we propose a new method to quantify each cloud security level based on probability and vulnerability of cybersecurity attacks on cloud components which lingers business continuity. Then, we model user request as a bi-objective optimization problem in multi-cloud market with security and cost viewpoints. To solve the combinatorial problem, we have extended a bi-objective optimization algorithm to find Pareto points indicating trade-off between objectives. The result of execution in different scenarios prove the optimality of our proposed approach in terms of accuracy and overhead.**

Keywords—Web service composition; multi-cloud computing; combinatorial optimization;

I. INTRODUCTION

Cloud computing is the new information technology paradigm which provisions wide spectrum of configurable hardware, software and middleware services to their world-wide subscribers via distributed networks [1-2]. Several competing providers provision their services in cloud market; for instance, infrastructure as a service (IaaS) by Amazon [3], platform as a service (PaaS) by Google [4] and software as a service (SaaS) by Salesforce [5]. Web service composition technology which applies software oriented architecture (SOA) techniques is platform- and technology- independent; so, it has attracted a lot of attention in literature for the sake of software development cost reduction [6]. MCE makes fault tolerant system and avoiding vendor lock-in which is an economic concept where the user does not rely on one supplier [7]. Web services are self-documented and provide transparent special functional operation to user independent from the underlying technology [8-9]. Brokers in clouds are applied with both local and distributed manner; they act as an interface when new services are generated by providers; they registers the new services in service directory to be accessed in future for users [10]. Although MCE provisions myriad atomic web services, maybe it does not cover user's business process. So, web service composition technology integrates atomic web services with standard protocols to make coarse value-added services [10]. When a user submits his/her request to broker; the broker searches to find appropriated web services based on SLA to meet suitable QoS for user. Finding an appropriate web service composition in ever-increasing MCE is computationally NP-hard [11-12]. Several papers have been published in literature to cope with the web service composition problem with different perspectives. For instance, Karimi et al. have been published a QoS-aware web service composition technique for only quick and online manner [13]. Another approach has been done by Kurdi et al. to minimize the number of used clouds in MCE for the sake of reduction of overall cost [14]. The same work has been propounded to select composition with low network delay by paying attention to network parameters in [15]. Surveys in [16] along with works in [13-15] shows that the majority works are QoS-, SLA- and Network-aware which do not pay attention to security tenets. Research upon literature demonstrates that security tenets are availability, integrity and confidentiality in cloud environment [17-20]. If MCE does not cover business-critical security requirement, it makes financial loss or even failure. Such business may tolerate network delay, but cannot tolerate data leakage, distributed denial of service attacks etc. So, the current approaches are not applicable for mission critical businesses. To quantify business financial losses owing to cloud disability to cover business-critical security requirement, we have extended mean failure cost (MFC) concept [21] to advanced mean failure cost (AMFC). On the other hand, in MCE, each cloud provisions the same functional web service with different cost and security level. So, we solve the problem of web service composition with cost and cybersecurity risk perspectives in where it should minimize both cost and risk simultaneously. The main contribution of this paper is to present a bi-objective optimization algorithm with cost and cybersecurity risk perspective whereas the cost and risk do not have any meaningful relation. A genetic-based algorithm is extended to figure out the problem; it find non-dominated points so called, Pareto set [24]. The rest of the paper is followed by system framework, service cost, AMFC concept, problem statement, bi-objective GA, simulation setup, evaluation and conclusion respectively.

978-1-5386-5824-6/18/$31.00 ©2018 IEEE

II. System Framework

Our system framework which is similar to frameworks in [14, 22-23], but differ in log file is depicted in Fig. 1.

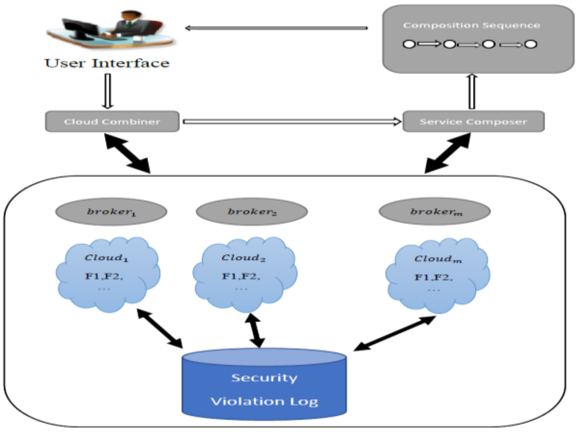

Fig. 1. SYSTEM FRAMEWORK

It has several modules which are described below:

. A multi-cloud environment (*MCE*) which contains set of *m* cloud providers whereas *MCE* = $\{C_1, C_2, \dots, C_m\}$ such as in [14, 22-23]. Every cloud has a set of service files $F = \{F_1, F_2, \dots, F_f\}$ where every service file includes a set of services $S = \{S_1, S_2, \dots, S_s\}$. Moreover, each cloud has its own pricing schemes and security level. The same service with the same functionality from different providers have different pricing schemes and variation in security ability.

. The user interface receives a request and returns service composition sequence

. The cloud combiner selects suitable cloud set, which has the most appropriate services to accomplish the user's security requirement along with cost, and produces a cloud combination list based on the set.

. The service composer receives cloud combination list from the cloud combiner module and determines which services from which cloud can best accomplish the user's security requirement.

. Security violation log: this module logs all of abnormal behavior by specifying normal manner; for instance the number of repudiation information/service, number and duration of server unavailability, data leakage due to malicious insider/outsider attacks etc. can be profiled to calculate probability of threat materializing.

A. Service Cost

Total web service cost is calculated via equation (1).

$$\text{Cost}= \sum_{i=1}^{n} sp^{i,j} \qquad (1)$$

$$\text{Whereas } \ C_j \in MCE$$

The variable $sp^{i,j}$ is the service price of S_i which is provisioned from cloud C_j to fulfill task T_i.

B. Advanced Mean Failure Cost (AMFC)

The random variable $AMFC_i$ is used to indicate the amount of cloud C_i financial losses owing to disability to cover security requirement; so, total security risk in multi-cloud is calculated by equation (2).

$$\text{AMFC} = \sum_{i}^{m} AMFC_i \qquad (2)$$

Whereas $C_i \in MCE$

How $AMFC_i$ is calculated is brought in equation (3) below; it is measured by chain multiplying of 2D matrixes.

$$AMFC_i\text{=}STM \times DM \times ITFM_i \times \text{TFM} \times TPM_i \qquad (3)$$

The matrices value can be fulfilled by system security specialists. Whereas stakeholder matrix (STM) is *1*×*R* matrix and cell $(1, R_j)$ means that user's loss per year if security requirement R_j fails. It is independently calculated from underlying cloud. Decision matrix (DM) is *R*×*C* matrix and cell (R_j, C_k) means the amount of component C_k's rule in requirement R_j. This values is the probability that requirement R_j is violated in course of operation the system for some period of time. It is also independently calculated from underlying cloud; the column shows cloud components. Impact Threat Family Matrix ($ITFM_i$) is a *C*×*F* matrix and cell $(C_{k,i}, F_{m,i})$ means the probability that component $C_{k,i}$ fails when the threat family $F_{m,i}$ is materialized by sourcing over $Cloud_i$. The matrix value depends on underlying used cloud. Threat Family Matrix (TFM) is *F* ×*T* matrix and cell (F_m, T_h) means the probability of threat family F_m occurance when threat T_h has been materialized. Also, one column is added for no threat. It is also independently calculated from underlying cloud. Threat Probability Matrix (TPM_i) is a *T*×*1* matrix and cell $(T_{h,i}, 1)$ means the probability of a threat happening per unit of operation epoch on $Cloud_i$. The value of TPM matrix value depends on underlying used cloud.

III. Problem Statement

The web composition problem is converted to a bi-optimization problem below which must minimize both Cost and AMFC functions simultaneously; it is observed in equation (4).

Bi-optimization problem:

$$\begin{cases} Min \ \ Cost \ = \ \sum_{i}^{n} sp^{i,j} \\ Min \ \ AMFC = \sum_{i}^{m} AMFC_i \\ \quad s.t \ \ m \leq k \leq M \\ \quad k \ is \ number \ of \ used \ clouds \\ \qquad Whereas \ C_j \in MCE \end{cases} \qquad (4)$$

Where *k* is the number of used clouds from *MCE* and parameters *m* and *M* indicate to minimum and maximum number of allowable cloud usage.

IV. BI-OBJECTIVE OPTIMIZATION GENETIC ALGORITHM

Genetic algorithm is inspired by natural evolutionary process. It is one the most popular search algorithm in large-scale problem to find optimal solutions. A GA-based algorithm typically consists of some operators. Since GA is typically designed to optimize one criterion such as to minimize cost function. We change GA algorithm toward our bi-optimization problem. We particularly design our roulette wheel so that solutions are fairly distributed along both criteria and from accurate Pareto curve. Algorithm1 details the advanced GA procedure.

```
Input: (1) All service prices from MCE to calculate Cost
       (2) All matrices' information along with Probabilities
           from system log to calculate AMFC
Output: The optimal location of service deployment in
        multi-cloud as Pareto Frontier

Step 1: Generate Initial Population
Step 2: Generate Smart Roulette Wheel
Step 3: Generate the Intermediate Population
Step 4: Smart Selection of Intermediate Population to
        Generate the Next Population
Step 5: Copy Next Population to Current Population
Step 6: While the Pareto Frontier is still improving repeat
        Steps 2-5
            If the Pareto Frontier has not changed for 50
            consecutive iteration, then terminate;
Step 7: End
```

Algorithm 1. Enhanced GA to make Pareto frontier

In our proposed GA, design of chromosomes and genes represent all solution possibility of a problem. Assume that the business needs 9 web services to cover their workload's functional requirement along with security objectives availability, integrity and confidentiality as non-functional requirement; preprocessing and investigation over multi-cloud market show that 50 providers provision 9 requested web services with different pricing schemes and security level the reason why each chromosome is designed such as a vector which numbers range from 1 to 50. For instance, a chromosome [20, 7, 7, 20, 40, 40, 35, 49, 50] means that web service numbered 1 and 4 are deployed on $Cloud_{20}$, web service numbered 2 and 3 are deployed on $Cloud_{7}$, web service numbered 5 and 6 are deployed on $Cloud_{40}$, web service numbered 7 is deployed on $Cloud_{35}$, web service numbered 8 is deployed on $Cloud_{39}$ and web serviced numbered 9 is deployed on $Cloud_{50}$ respectively. This arrangement incurs specific cost and risk level. Moreover, we empirically assign two parameters *m* and *M* to determine minimum and maximum number of allowable used clouds. Also, Step 1 generates initial population which consists of several limited chromosomes. Moreover, we take 100 individuals as population size in this problem. Roulette wheel in our improved GA is designed in favor of both equally important criteria of bi-objective total Cost and Security Risk discrete functions. So, since the minimum value of both criteria are unknown, we consider relative quality of each chromosome in comparison with other chromosomes in current population. Algorithm2 calculates fitness value for all chromosomes in current population.

```
Input: (1) The Current Population and (2) Frontier
       Resolutions: FntResCost and FntResRisk
Output: Fitness value of all chromosomes
Step1: Find the minimum and maximum of Total Service
       Cost generated by all chromosomes in current population
       namely MinCost, MaxCost respectively.
Step2: Find the minimum and maximum of Security Risk
       generated by all chromosomes in current population
       namely MinRisk, MaxRisk respectively.
Step3: Let $BandCost = \frac{MaxCost-MinCost}{FntResCost}$ and
       $BandRisk = \frac{MaxRisk-MinRsik}{FntResRisk}$
Step4: For x=0 to FntResCost-1 do
           For y=0 to FntResRisk-1 do
               X1=MinCost+BandCost*x
               X2=MinCost+BandCost*(x+1)
               T1= MinRisk+BandRisk*y
               T2= MinRisk+BandRisk*(y+1)
               Find all chromosomes with total cost between
                   [X1,X2]
               and security risk between [T1,T2] and store
                   them into ChrmSet.
               Let Fitness=Max(FntResCost-x,FntReRisk-
               y)*(FntResCost-x)*(FntResRisk-y)
               Let Fitness value of all chromosomes in
                   ChrSet be Fitness.
           Next y
       Next x
Step5 :End
```

Algorithm2. Calculation Algorithm of Fitness values for all chromosomes

In spite of original GA, we subjectively constitute intermediate population with specific feature to cover several objectives. In this regard, the concerns are to divergence, making outlier individuals and speed of algorithm. To cope with the aforementioned problems, we consider intermediate population size 5-10 times more than current population. The number five and ten are set experimentally. We copy current population directly into intermediate population to preclude divergence. Then, chromosomes are selected by roulette wheel that are delivered to crossover operation to be split into predetermined three point sections and genes of parents are interleaved to generate new offspring. The mutation operator receives new generated chromosomes by crossover and randomly changes their genes with predefined probability. Constituting intermediate population ameliorates speed of algorithm, convergence and quality of solution along with trade-off between them. During process of crossover and mutation operations, few infeasible solutions as outliers may be generated when the chromosomes interpreting that web services are deployed on providers' datacenters, the number of used clouds are more than *M* or less than *m*. In this circumstance, it violates the predefined constraints as minimum and maximum suppliers involved. In the next phase, chromosomes of intermediate population are sorted based on some criteria. Afterwards, the best of them are copied into next population. Since the aim is to constitute Pareto set to concentrate on both criteria. In fact, 10% of population which have the lowest composition total cost, 10% of population which have the lowest security risk and the rest of 80% are filled by summation of total cost and security risk in decreasing order. Meanwhile, the duplicated chromosomes are omitted. The portion 10, 10 and 80 percent of population are set empirically. As GA is endless algorithm, the typical criterion or even mixing of criteria are applied to terminate the algorithm common criteria are total execution time, number of iterations, fitness values of the best chromosomes etc. In this algorithm we take 100 iterations as termination criterion.

V. Simulation Setup

All experiments were implemented using Sony VAIO laptop with a 2.26 GHz Intel Core 2 Duo processor and 4 GB RAM and using Matlab 2015. Take a business model which needs 10 different web services. For simulation, we have conducted seven scenarios by considering $n \in [30..90]$ number of cloud providers increasing by 10 in MCE which can deliver requested web services. Each web service price has been taken between 10 and 100 $/month into account for each provider; after composited web services are constituted, we can multiply this summation amount by 12 to calculate overall cost in $/year for composited web services. Also, to measure amount of cybersecurity risk into *AMFC* variable related to each provider, we considered fixed value for *STM*, *DM* and *TFM* matrices and range of [0.005..0.1] and [0.0005..0.009] for $ITFM_i$ and TPM_i matrices respectively. Moreover, all the values are derived from uniform distribution. Also, for the sake of simplicity we take parameters *m*=1 and *M*=10 into account.

VI. Evaluation Methodology

Because our approach is considered for mission-critical businesses and there is no approach with cybersecurity risk perspectives for web service composition technology, to evaluate our methodology, we considered single objective GA which only takes cost function into account along with our bi-objective GA which takes cost and risk functions. In this section, we compare fitness value of our proposed model based on algorithm 2 with a single objective GA based on pure cost neglecting risk function. Our bi-objective GA algorithm minimizes both cost and risk functions simultaneously to deliver a solution. Then set of non-dominated points, Pareto set, are delivered. For instance, Fig. 2 depicts Pareto set points in scenario which the number of cloud providers are 50 (in the third scenario).

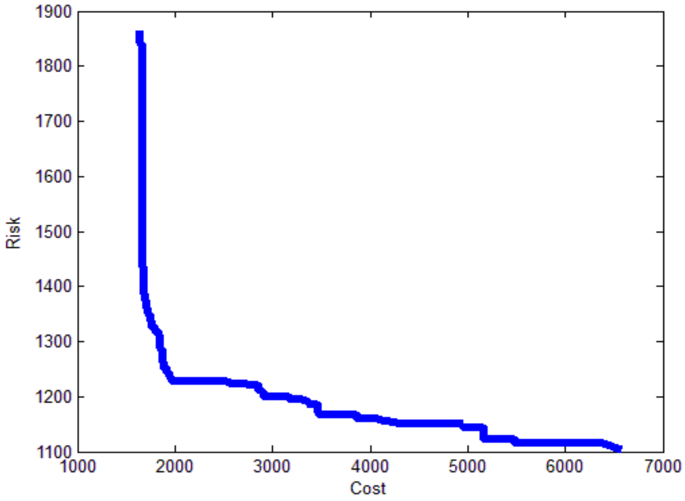

Fig. 2. Pareto set solving bi-objective optimization in scenario which n=50 cloud providers

Table *I* compares our bi-objective GA and a single objective GA with cost, amount of cybersecurity risk and total cost which is the sum of two terms cost and risk in term of $/year for each scenario. Moreover, in a single objective GA the goal is to minimize the only cost function whereas our bi-objective GA tries to minimize both cost and risk functions simultaneously. For instance, our approach leads a composition with cost=3000, risk=1000 and total cost=4000 whereas a single GA leads a composition with cost=2000 when the risk is 2500 based on our approach which were not calculated by the single objective GA; it shows total cost is 4500. It indicates that an approach which considers only a cost function and neglects risk is misleading procedure. Therefore, we separately calculated the amount of risk for single objective GA which was neglected by it.

TABLE I. Approaches comparison in terms of cost, risk and total cost in all scenarios

Number of Providers	Approaches	Cost/ Risk /Total Cost
***n*=30**	bi-objective GA	1.6904*10^3 1.3732*10^3 **3.0636*10^3**
	single-objective GA	1.5012*10^3 1.6104*10^3 3.1116*10^3
***n*=40**	bi-objective GA	1.7899*10^3 1.3805*10^3 **3.1704*10^3**
	single-objective GA	1.5045*10^3 1.7015*10^3 3.2060*10^3
***n*=50**	bi-objective GA	1.6851*10^3 1.4238*10^3 **3.1089*10^3**
	single-objective GA	1.6001*10^3 1.5160*10^3 3.1161*10^3
***n*=60**	bi-objective GA	1.7407*10^3 1.4160*10^3 **3.1567*10^3**

	single-objective GA	$1.7020*10^3$ $1.4810*10^3$ $3.1830*10^3$
***n*=70**	bi-objective GA	$1.6175*10^3$ $1.4111*10^3$ **$3.0286*10^3$**
	single-objective GA	$1.5056*10^3$ $1.7098*10^3$ $3.2154*10^3$
***n*=80**	bi-objective GA	$1.6707*10^3$ $1.3381*10^3$ **$3.0088*10^3$**
	single-objective GA	$1.6120*10^3$ $1.4100*10^3$ $3.0220*10^3$
***n*=90**	bi-objective GA	$1.7241*10^3$ $1.3007*10^3$ **$3.0248*10^3$**
	single-objective GA	$1.7022*10^3$ $1.3439*10^3$ $3.0461*10^3$

The results of Table *I* show that although cost column for single objective GA approach is minimum for all scenarios, the total cost is the worst for it. It means that considering only cost function with neglecting risk amount never yield sustainable decision. Our algorithm, on the other hand, outperforms compared with single objective GA in all scenarios in term of total cost.

VII. PERORMANCE AND SCALABILITY

To evaluate optimality of our bi-objective GA algorithm, we compare fitness value of our algorithm with a single objective GA. Results of implementation show that in all scenarios our algorithm outperforms on single objective GA even in ever-increasing large search space. Inasmuch as the problem is NP-hard in nature, to assess the performance and scalability of our algorithm, we conducted 10 different scenarios by considering from 30 to 500 providers which can deliver requested web services. The eight of them are from 30 to 100 increasing by 10 providers whereas in the ninth and tenth scenarios the number of providers are 200 and 500 respectively. As the Fig. 3 illustrates, our algorithm is scalable and has good convergence speed even for large search space in which one can make online decision. In contrast, maybe the execution time of single objective GA is lowest, but in all cases the decision is not sustainable it is due to considering only cost function and neglecting cybersecurity impact.

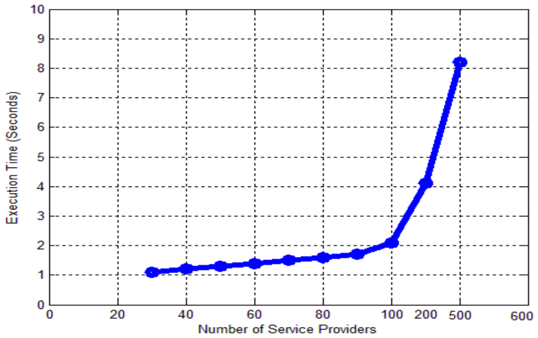

Fig. 3. Execution Time for Web service Composition

VIII. CONCLUSION AND FUTURE WORK

In this paper, a web service composition problem has been investigated for business-critical model by considering cost and cybersecurity perspectives in MCE; where each provider provisions the same atomic service with different pricing scheme and security level. Security principles are considered as availability, integrity and confidentiality whereas each cyber security attack on them can occur business loss or even business failure. To measure the amount of cybersecurity loss in terms of monetary parameter we extended AMFC variable. Then, we have developed a bi-objective GA optimization algorithm that minimizes cost and security risks for composited services simultaneously. Since price and security are not in the same direction, we have found set of non-dominated points called Pareto set by executing our GA algorithm. Therefore, one of them is applied as the optimal web service composition. Our algorithm are more optimal and stable in comparison with single objective GA which considers only cost function. Also, execution results indicates that our algorithm are scalable in ever-increasing MCE even we can decide online for web service composition because it is as quick as possible. This paper does not take into account communication costs between clouds for interaction amongst web services. For future work, we envisage to develop the bi-objective cost-risk optimization model for web service composition by taking workload manner and communication cost into account in the model.

REFERENCES

[1] P. Mell, T. Grance, The NIST definition of cloud computing, Natl. Inst. tand.Technol. 53 (6) (2009) 50.

[2] M. Armbrust, A. Fox, R. Griffith, A.D. Joseph, R.H. Katz, A. Konwinski, G. Lee, D.A. Patterson, A. Rabkin, I. Stoica, M. Zaharia, Above the clouds: A Berkeley view of cloud computing, University of California, Berkeley, 2009.

[3] Amazon EC2 aws.amazon.com/EC2 [Visited 2016].
[4] Google.com [Visited 2016].
[5] Salesforce.com [Visited 2016].

[6] Yu Q., Chen L., Li B., Ant colony optimization applied to web service compositions in cloud computing, Computers and Electrical Engineering 41 (2015) 18–27.

[7] Kiran S., Anusha A., Gowtham K., Praveen K. R., SELECTION OF MULTI-CLOUD STORAGE USING COST BASED APPROACH, International Journal of Computer and Electronics Research, 2013: 2(2).

[8] Bichier M, Lin K-J. Service-oriented computing. Computer 2006;39(3):99–101.

[9] Hatzi Ourania, Vrakas Dimitris, Nikolaidou Mara, Bassiliades Nick, Anagnostopoulos Dimosthenis, Vlahavas L. An integrated approach to automated semantic web service

composition through planning. IEEE Trans Serv Comput 2012;5(3):319–32.

[10] Wang D., Yang Y., Mi Z., A genetic-based approach to web service composition in geo-distributed cloud environment, Computers and Electrical Engineering 43 (2015): 129–141.

[11] Canfora G, Penta MD, Esposito R, Villani ML. An approach for QoS-aware service composition based on genetic algorithms. In: Proceedings of the 7th annual conference on genetic and evolutionary computation; 2010. p. 123–8.

[12] Rao J, Su X. A survey of automated web service composition methods. In: Proceedings of the first international workshop on semantic web services and web process composition SWSWPC; 2004. p. 43–54.

[13] Karimi M.B, Isazadeh A, Rahmani A.M. ,QoS-aware service composition in cloud computing using data mining techniques and genetic algorithm, Journal of Supercomputing (2016), doi:10.1007/s11227-016-1814-8.

[14] Kurdi H., Al-Anazi A., Campbell C., Al Faries A., A combinatorial optimization algorithm for multiple cloud service composition, Computers and Electrical Engineering 42 (2015) 107–113.

[15] Klein Adrian, Ishikawa Fuyuki, Honiden Shinichi. Towards network-aware service composition in the cloud. In: Proceedings of the 21st international conference on World Wide Web. ACM; 2012. p. 959–68.

[16] Jula A., Sundararajan E., Othman Z., Cloud computing service composition: A systematic literature review, Expert Systems with Applications 41 (2014) 3809–3824.

[17] ENISA, Cloud Computing: Benefits, Risks and Recommendations for Information Security (ENISA, 2009).

[18] Cloud Security Alliance. Whitepaper: Security Guidance for Critical Areas of Focus in Cloud Computing. 2009.

[19] Catteddu D, Hogben G. Cloud Computing: Benefits, risks and recommendations for information security. European Network and Information Security Agency, Crete, Greece, 2009.

[20] Federal Information Processing Standards Pub 199: Standards for Security Categorization of Federal Information and Information Systems http://csrc.nist.gov/publications/fips/fips199/FIPS-PUB-199-final.pdf accessed on Jan 7, 2010.

[21] Aissa A.B., Robert K. Abercrombie, Frederick T. Sheldon, Ali M., Quantifying security threats and their potential impacts: a case study, Innovations Syst Softw Eng (2010) 6:269–281, DOI 10.1007/s11334-010-0123-2.

[22] Zou G, Chen Y, Xiang Y, Huang R, Xu Y. AI planning and combinatorial optimization for web service composition in cloud computing. In: Proceedings of the international conference on cloud computing and virtualization; 2010. p. 1–8.

[23] OWLS-Xplan Service Composition Planner. <http://www-ags.dfki.uni-sb.de/~klusch/owls-xplan/> [Visited 2016].

[24] Wikipedia. (Visited 2016). Pareto efficiency.

[25] Ben, A. R., Mili, A., et al. (2013). A cybersecurity model in cloud computing environments. Journal of King Saud University–Computer and Information Sciences, 25(1), 63–75.