

A New Learning Automata Based Approach for Increasing Utility of Service Providers

Hossein Morshedlou¹, Mohammad Reza Meybodi²

Abstract

Utility is an important factor for service providers and they try to increase their utilities through adopting different policies and strategies. Due to unpredictable failures in systems there are many scenarios in which the failures may cause random losses for service providers. Loss sharing can decrease negative effects of unexpected random losses. Due to capabilities of learning automata in random and stochastic environments, in this paper a new learning automaton based method is presented for loss sharing purpose. It is illustrated that the loss sharing can be useful for service providers and helps them to decrease negative effect of the random losses. The presented method can be employed especially in collaborative environments such as federated clouds. Results of the conducted experiments show the usefulness of the presented approach to improve utility of service providers.

Keywords: Learning Automata, Service Provider, Utility, Loss Sharing.

¹ Hossein Morshedlou

Soft Computing Lab - Department of Computer Engineering - AmirKabir University of Technology, Tehran, Iran

Email: morshedlou@aut.ac.ir

² Mohamad Reza Meybodi

Soft Computing Lab - Department of Computer Engineering - AmirKabir University of Technology, Tehran, Iran

Email: mmeybodi@aut.ac.ir

1 Introduction

Reaching more utility and profitability are important goals of service providers in market oriented environments such as clouds [7]. Due to unpredictable failures in the service provisioning or system errors there are many scenarios in which SLA violations can conduce to losses for service providers e.g. in form of penalty. These losses have negative effects on utility and business of service providers. Therefore it's very natural that service providers try to decrease effects of the losses. It is very plausible and in accordance with socio-biological arguments to assume that service providers behave in a risk-averse manner [25]. A risk averse service provider prefers more certain condition when it is confronted with two choices with same expected utility. Thus if collaboration with other service providers can decrease uncertainty, a risk averse service provider will be interested to collaborate. Such collaboration can be useful in environments such as federated clouds. In federated clouds two or more independent geo-graphically distinct Clouds share either authentication, files, computing resources, command and control or access to storage resources [8]. This paper aims to show that extending these sharing agreements to the losses can be useful and valuable. This is for the reason that when there are two or more risk averse service providers, an appropriate loss sharing agreement can increase utility of all them. Due to randomness of the losses and their unknown distributions currently establishing a theoretic method for finding an appropriate loss sharing agreement is very difficult and complex. In this paper we aim to present a method for finding appropriate loss sharing agreements. Our method is based on Irregular Cellular Learning Automata (ICLA). ICLA which is introduced formally in [13] is a powerful mathematical model for decentralized applications. ICLA and its structured form, CLA [4], has been found to perform well in many application areas such as graph applications [17], channel assignment in cellular networks [3], solving NP-hard problems [24] and etc. Thus we think capabilities of ICLA can be employed to present a method for loss sharing intention. To the best knowledge of the authors this is the first work which tries to investigate a method for increasing utility of service providers through loss sharing. The rest of this paper is organized as follow: In section 2 we introduce some preliminary concepts which are needed for understanding the next sections. Section 3 review related works. Section 4 contains problem definition. Section 5 presents an ICLA based method for loss sharing problem. Results of conducted numerical experiments are presented in section 6 and we conclude paper in section 7.

2 Preliminaries

This section offers a brief introduction to preliminaries and concepts which are needed in following sections.

2.1 Utility

Utility (or usefulness) is the perceived ability of things to satisfy needs. Utility is an important concept in game theory, economics and business, because it represents satisfaction experienced by an individual[27]. Frequently in economics literatures [25] utility is defined as a function of income or wealth. In this paper, when a service provider provisions its users according to their agreed SLA, income of service provider increases to I and its utility will be $u(I)$. When service provider violates SLAs, it pays penalties to users and its income decreases to $(I-L)$. Therefore its utility will be $u(I-L)$.

2.2 Risk Aversion

A service provider is said to be "risk averse" if, confronted with two choices with same expected utility, it prefers the smaller and more certain of the options. Utility function of a risk averse service provider has two main features: $u' > 0$ and $u'' < 0$ [25]. These features indicate that for a risk averse service provider a possible increase in income is valued less than an equally likely decrease in income of the same amount. Having utility function of a service provider, its risk aversion can be calculated by $-u''/u'$.

Assumption: It is very plausible and in accordance with socio-biological arguments to assume that human beings behave in a risk-averse manner [25]. Therefore it is acceptable to assume that the service providers behave in a risk-averse manner as well.

2.3 Learning Automata

A learning automaton is represented by a triple $\langle \underline{\beta}, \underline{a}, T \rangle$, where $\underline{\beta}$ is the set of inputs, \underline{a} is the set of actions, and

T is learning algorithm. Actions of learning automata are inputs to environments. Let $a_i(k) \in \underline{a}$ and $\underline{p}(k)$ respectively denote the selected action by learning automaton and probability vector defined over the action set at iteration k . Let r and b indicate the reward and penalty parameters and determine the amount of increases and decreases of the action probabilities. m is number of actions can be taken by learning automaton. At each iteration, the action probability vector $\underline{p}(k)$ is updated by the linear learning algorithm (T) given in Eq. (1), if the selected action $a_i(k)$ is rewarded by the environment, and it is updated as given in Eq. (2) if the selected action is penalized.

$$p_j(k+1) = \begin{cases} p_j(k) + r[1 - p_j(k)] & j = i \\ (1 - r)p_j(k) & \forall j \neq i \end{cases} \quad (1)$$

$$p_j(k+1) = \begin{cases} (1 - b)p_j(k) & j = i \\ (\frac{b}{r-1}) + (1 - b)p_j(k) & \forall j \neq i \end{cases} \quad (2)$$

If $r = b$, the recurrence Eq. (1) and (2) are called linear reward-penalty (L_{R-P}) algorithm. If $r \gg b$ the given equations are called linear reward- \mathcal{E} penalty ($L_{R-\mathcal{E}P}$), and finally if $b = 0$ they are called linear reward-Inaction (L_{R-I}).

2.3.1 Variable Set Action Learning Automata

The content of this section is from [17]. A variable action set learning automaton is an automaton with m actions in which the number of available actions at each instant changes with time [21]. Let $\underline{a} = \{a_1, a_2, \dots, a_m\}$ to be a finite set of all actions. Define $\underline{A} = \{A_1, A_2, \dots, A_s\}$ as a set of action subsets and $A(k) \subseteq \underline{a}$ denotes the subset of all the actions can be selected by the learning automaton, at each instant k . According to the probability distribution $q(k) = \{q_1(k), q_2(k), \dots, q_n(k)\}$, an external agency chooses randomly the particular action subsets, where: $q_i(k) = \text{prob}[A(k) = A_i \mid A_i \in \underline{A}, 1 \leq i \leq 2^m - 1]$. $\hat{p}_i(k) = \text{prob}[a(k) = a_i \mid A(k), a_i \in A(k)]$ is the probability of choosing action a_i , if the action subset $A(k)$ has already been selected and also $a_i \in A(k)$. The scaled probability $\hat{p}_i(k)$ is defined as:

$$\hat{p}_i(k) = \frac{p_i(k)}{K(k)} \quad (3)$$

where $K(k) = \sum_{a_i \in A(k)} p_i(k)$ is the sum of the probabilities of the actions in subset $A(k)$, and $p_i(k) = \text{prob}[a(k) = a_i]$.

In a variable action set learning automaton, the procedure of selecting an action and updating the action probabilities can be described as follows. Let $A(k)$ is the action subset selected at instant k . Before selecting an action, the probabilities of all the actions in the selected subset are scaled using equation (3). Then, the automaton randomly selects one of its possible actions according to the scaled action probability vector $\hat{p}_i(k)$. The learning automaton updates its scaled action probability vector based on the response received from the environment. Note that in this step, the probability of the available actions is only updated. Finally, the probability vector of the actions of the selected subset is rescaled as follows for all $a_i \in A(k)$:

$$p_i(k+1) = \hat{p}_i(k+1).K(k) \quad (4)$$

2.4 ICLA

An ICLA is defined as an undirected graph in which, each vertex represents a cell which is equipped with a learning automaton. The learning automaton residing in a particular cell determines its state (action) on the basis of its action probability vector. There is a rule that the ICLA operates under. The rule of the ICLA and the actions selected by the neighboring LAs of any particular LA determines the reinforcement signal to the LA residing in a cell. The neighboring LAs of any particular LA constitute the local environment of that cell. The local environment of a cell is non-stationary because the action probability vectors of the neighboring LAs vary during the evolution of the ICLA.

2.5 Notations

In this section some notations, which are used in following sections, are introduced:

$X_i(t)$: A random variable that shows the losses which service provider i has sustained during period $(t-I, t)$. These losses can be due to for example SLA violations (penalty payment or ...).

$\alpha_i^j(t)$: The percentage of $X_i(t)$ that service provider j has accepted to compensate in loss sharing agreement.

$pr_i^j(t)$: A premium value that service provider i pays to service provider j when service provider j accepts to compensate $\alpha_i^j(t) \times X_i(t)$. $pr_i^j(t)$ is similar to premiums in an insurance contract.

$RP_i(t)$: The remained percentage of $X_i(t)$ that other service providers haven't accepted yet to compensate. When service provider j accepts to compensate $\alpha_i^j(t)$ from $X_i(t)$, service provider i updates $RP_i(t)$ and puts $RP_i(t) = RP_i(t) - \alpha_i^j(t)$.

LA_{ij} : A learning automaton in cell i that decides about $\alpha_j^i(t)$.

N_i : Neighbors of cell i (The service providers that collaborate with SP_i in loss sharing strategy). we assume that for each i , $i \in N_i$.

$I_i(t)$: Income of service provider i from its business in iteration t .

$PI_i(t)$: The pure income of service provider i in round t . $PI_i(t)$ is calculated as:

$$PI_i(t) = I_i(t) - \sum_{j \in N_i} (\alpha_i^j(t) \times X_j(t)) + \sum_{j \in N_i} pr_i^j(t) \quad (5)$$

Loss Sharing Agreement (LSA): A tuple as:

$$\left(\left((\alpha_1^1(t), pr_1^1(t)), (\alpha_2^1(t), pr_2^1(t)), \dots, (\alpha_n^1(t), pr_n^1(t)) \right), \dots, \right. \\ \left. \left((\alpha_1^i(t), pr_1^i(t)), (\alpha_2^i(t), pr_2^i(t)), \dots, (\alpha_n^i(t), pr_n^i(t)) \right), \dots, \right. \\ \left. \left((\alpha_1^n(t), pr_1^n(t)), (\alpha_2^n(t), pr_2^n(t)), \dots, (\alpha_n^n(t), pr_n^n(t)) \right) \right) \quad (6)$$

Where $\sum_{j=1}^n \alpha_i^j(t) = 1$ and $\forall i, t \quad pr_i^i(t) = 0$.

2.6 Benefit of Loss Sharing

In this section we provide a short discussion on benefit of loss sharing for risk averse service providers. We explain why risk averse service providers are interested to share their losses. As said before, utility function of a risk averse service provider has two main feature: $u' > 0$ and $u'' < 0$. Figure 1 shows a utility function for a risk averse service

provider. Let X to be a random variable and assume its value indicates the magnitude of the loss. When all requests are provided according to the agreed SLAs, X is zero and service provider's utility is $u(I_0)$. In case of SLA violation, X is L and utility will be decreased to $u(I_0 - X)$. Therefore when there is likelihood of SLA violation with probability π , the expected utility of the service provider is $\pi \times u(I_0 - X) + (1 - \pi) \times u(I_0)$. Now let to have another service provider exactly with same utility function and loss distribution. If both service providers accept to share their losses according to a LSA such as $((0.5, 0), (0.5, pr)), ((0.5, pr), (0.5, 0))$, then the expected utility of both the service providers are:

$$\pi^2 \times u(I_0 - X) + 2 \times \pi \times (1 - \pi) \times u(I_0 - X/2) + (1 - \pi)^2 \times u(I_0) \quad (7)$$

Since $u' > 0$ and $u'' < 0$ as a result we have:

$$u(I_0) - u(I_0 - X/2) < u(I_0 - X/2) - u(I_0 - X) \quad (8)$$

Eq. (8) means $u(I_0) + u(I_0 - X)$ is smaller than $2 \times u(I_0 - X/2)$, therefore we have:

$$\pi^2 u(I_0 - X) + 2\pi(1 - \pi)u(I_0 - X/2) + (1 - \pi)^2 u(I_0) > \pi u(I_0 - X) + (1 - \pi)u(I_0) \quad (9)$$

Eq. (9) means both the service providers can reach higher utility in case of loss sharing. So rational service providers are interested to collaborate and share their losses.

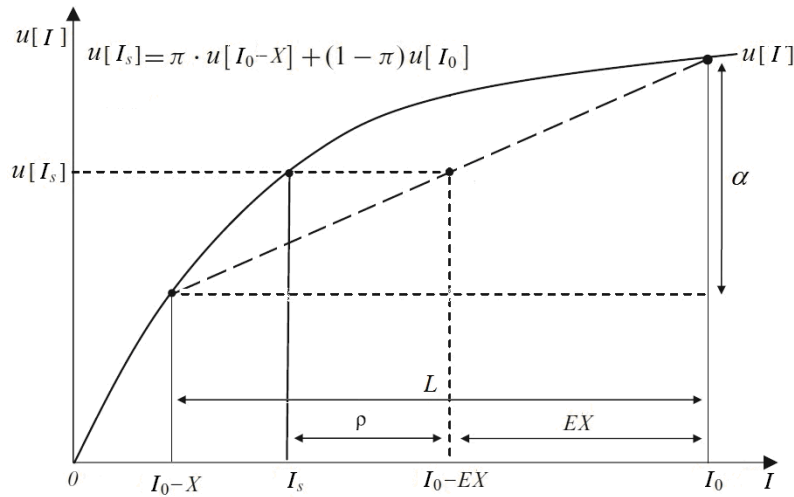


Fig. 1- Utility function of a risk averse service provider

3 Related Works

Nowadays sharing ideas to increase utility are common in many distributed computing infrastructures like as p2p networks, grids and clouds. For example in federated clouds [14, 15, 18, 19, 22, 23] providers share their unused capacities to increase their profits. There are plenty of other works that focus on idea of resource sharing in federated clouds. For example [6] presents a market-oriented model among clouds for VM exchange and capacity sharing. Reference [20] offers a formulation for resource sharing among cloud providers that leads to optimal revenue. To the best knowledge of the authors there is no a notable work for loss sharing in distributed computing literature and this work is the first one in this scope. However idea of loss sharing is not a new idea and there are plenty of works in economic and insurance literatures that focused on topic of risk and loss sharing [1, 2, 5, 9-12]. For example reinsurance [26] is one of the loss sharing approaches in insurance industry. The ultimate goal of reinsurance is to reduce insurance companies' exposure to loss by passing part of the risk of loss to a reinsurer or a group of reinsurers. By choosing a particular reinsurance method, the insurance company may be able to create a more balanced and homogeneous portfolio of insured risks. This would lend greater predictability to the portfolio results on net basis and would be reflected in income smoothing. In view of the fact that loss sharing process is

similar to reinsurance approaches, some of the existing works on finding equilibriums of reinsurance markets [1, 2, 5] can inspire designing an appropriate approaches for loss sharing in distributed computing environments. Loss sharing approaches can reduce exposure to random losses that threat profitability and utility. This paper is the first work that tries to employ loss sharing approach to increase utility of service providers in a distributed computing environment.

4 Problem Definition

Assume there are n risk averse service providers. Let to show these service providers using a graph in which each vertex represents a service provider. Each edge in this graph represents tendency between two service providers for collaboration (e.g. sharing losses). We call this graph tendency graph. Since the service providers are risk averse so their utility functions satisfy two conditions: $u'_i(I_i) > 0$ and $u''_i(I_i) < 0$. Utility of each service provider is not known by the other service providers and X_i is announced just to the service providers that are interested to collaborate with service provider i . We also assume that service providers announce their satisfaction level from a LSA to each other using a satisfaction signal. Values of satisfaction signals are in range $[0, 1]$ and bigger value means higher satisfaction level is obtained. Distribution of X_i is not known. Now we want to find an appropriate LSA such that collaboration of the service providers under this LSA increases their utility as possible.

5. ICLA-based Method for Loss Sharing

To check does a LSA increase utilities of service providers we need detailed information about distribution of losses and utility functions of service providers. Generally finding an appropriate LSA that increases utility of all service providers is difficult especially when there is no complete information about utility function forms and loss distributions. This will be much more complicated when there are too many service providers with different utility functions and different loss distributions. In addition to the mentioned difficulties, $X_i(t)$ and $X_j(t) (\forall j \in N_i)$ are random variables thus $PI_i(t)$ and $u_i(PI_i(t))$ (see Eq. (5)) will be as well. This means even with same LSA in two different iterations, service providers may experience different utilities. Therefore theoretic analysis similar to the one in section 2.6 is not possible. It seems that non-theoretic methods must be considered as alternatives for finding an appropriate LSA. Due to capabilities of learning automata and ICLA in environments with high uncertainty, in this section we present an ICLA-based method for this problem.

In the proposed method, each cell of ICLA represents a service provider (a vertex of tendency graph) and neighborhood relation is defined based on tendency for collaboration between service providers (edges of tendency graph). Per each neighbor j there is a learning automaton LA_{ij} in cell i . In an iterative procedure this learning automaton selects one of its actions that this action determines $\alpha_j^i(t)$. Neighbor j (service provider j) pays $pr_j^i(t)$ to service provider i for $\alpha_j^i(t) \times X_j(t)$. Amount of $pr_j^i(t)$ can be zero or be determined according to a premium calculation method in insurance literature [25]. Then all service providers calculate $u(PI(t))$ and generate satisfaction signals. Figure 2 illustrates the proposed method in detail. There is a local rule in ICLA that determines the reinforcement signal to any particular learning automaton. In our method this rule uses *GenRSignal* to generate reinforcement signal. Figure 3 shows the pseudo code of *GenRSignal*. The mentioned procedure are repeated every iteration by every service provider until a predefined threshold to be met. Figure 4 illustrates pseudo code to check the threshold satisfaction.

```

Each round t Do {
    Simultaneously for each service provider i {
        Simultaneously for each  $SP_j \in N_i$  {
             $SP_j$  waits for a random time and then asks  $SP_i$  about  $RP_i(t)$ .
             $SP_i$  announces  $RP_i(t)$  to  $SP_j$ .
             $SP_j$  rescales actions of  $LA_{ji}$  according to the response of  $SP_i$ .
            Using the rescaled actions,  $LA_{ji}$  selects one action to specify  $\alpha_i^j(t)$ .
             $SP_j$  announces  $\alpha_i^j(t)$  to  $SP_i$ .
             $SP_i$  updates  $RP_i(t)$ .
        }
         $SP_i$  calculates  $X_i(t)$  in round t and announces it to all the  $SP_j$  that  $\alpha_i^j(t) \neq 0$ .
         $SP_i$  calculates  $PI_i(t) = I_i(t) - \sum_{j \in N_i} (\alpha_i^j(t) \times X_j(t)) + \sum_{j \in N_i} pr_i^j(t)$  using all the announced  $X_j(t)$  s.
         $SP_i$  calculates satisfaction signal ( $\beta_i(t)$ ) and announces it to all the  $SP_j$  that  $\alpha_i^j(t) \neq 0$ .
         $SP_i$  receives satisfaction signals ( $\beta_j(t)$ ) of the other  $SP_j$  that  $\alpha_i^j(t) \neq 0$ .
         $SP_i$  generates reinforcement signals for  $LA_{ij}$  using GenRSigal( $PI_i(t), \beta_j(t)$ ).
        Probability vector of  $LA_{ij}$  is updated using the generate reinforcement signal.
    }
} While (is_Threshold_Satisfied())

```

Fig. 2- Pseudo code of the proposed method for loss sharing

```

GenRSigal( $PI_i(t), \beta_j(t)$ ). {
    Rand = Generate_Random_Number(0,1); // 0 < Rand < 1
    If (t==1) {
         $u_i^{\max} = u_i(PI_i(1))$ ;
        If (Rand > 0.5) {
            Reinforcement_Signal =1;
        } Else {
            Reinforcement_Signal =0;
        }
    } else {
        If ( $u_i(PI_i(t)) > u_i^{\max}$ ) {  $u_i^{\max} = u_i(PI_i(t))$ ; }
         $\beta_i(t) = \frac{u_i(PI_i(t))}{u_i^{\max}}$ ; //  $\beta_i(t)$  is satisfaction signal of  $SP_i$ 
         $\beta(t) = w_{ij}\beta_i(t) + (1 - w_{ij})\beta_j(t)$ ; //  $0.5 \leq w_{ij} \leq 1$ 
        If (Rand >  $\beta(t)$ ) {
            Reinforcement_Signal =1;
        } Else {
            Reinforcement_Signal =0;
        }
        // end of If (Rand >  $\beta(t)$ )... Else ...
    }
    // end of If (t==1) ... Else ...
    Return Reinforcement_Signal;
}

```

Fig. 3- Pseudo code of GenRSigal

```

n: Let n denotes number of cells (or service providers).
ni: Let ni denotes number of neighbors of cell i.
mij: Let mij denotes number of actions of LAij.
pijk: Let pijk denotes probability of k-th action in LAij.
is_Threshold_Satisfied() {
    for i=1 to i=n concurrently do:
        for j=1 to j=ni concurrently do {
            boolVar = false;
            for (k=1, k ≤ mij, k++) {
                if (1 - pijk < ε)
                    boolVar = true;
            }
            If (boolVar) STOP( LAij );
        }
    If (All LAij are stopped) return true;
    Else return false;
}

```

Fig. 4- Pseudo code for checking threshold satisfaction

6 Experiments

In this section first the experimental environment is introduced. Then the evaluation approach is described in 6.2. Results of the numerical experiments are reported in section 6.3.

6.1 Experimental Environment

The experiments are conducted using 5 and 8 service providers. Figure 5 illustrates the tendency relation among the service providers in three different cases. In the conducted experiments the exponential form functions similar to Eq.(10) are used to describe utility functions of service providers. This form of utility function satisfies the conditions required for a risk averse service provider.

$$u_i(PI_i) = c_i e^{-a_i PI_i} - c_i \quad (c_i < 0, a_i > 0) \quad (10)$$

For all the service providers in the conducted experiments, c_i and I_i in Eq.(10) are set to -5 and 2000 respectively. I_i is used to calculate PI_i via Eq. (5). In each setting a_i in Eq.(10), which is risk aversion of service provider, has been set according to table 1.

Because it is common in risk management literatures [16] to model random losses with exponential distributions, we assume that random variable X_i has exponential distribution with $\theta_i = 0.005$. The learning algorithms of learning automata are L_{R-I} and the learning rates are set to 0.001.

In the experiments, we assume existence of eleven different levels of shares in loss sharing, including {0, 0.1, ..., 0.9, 1}. For example when service provider i decides to compensate 40% of X_j , its shares will be equal to 0.4. Such categorizations are not unusual in approaches which dealing with random losses and risks. For example in risk management techniques, instead of computing exact value for the likelihood (probability) of occurrence of a risk, likelihood of occurrence is categorized to {very low, low, medium, high, very high}. One reason for such categorizations in dealing with random risks is that calculating an exact value for random variables is too hard and impossible in most cases. Even in case of calculating an exact value, this value is not reliable and dependable. For example if a service provider agrees to compensate 40% of loss X_j , the loss it must compensate in reality may be even greater than the loss it must compensate in another time, when it agreed to compensate 42% of X_j .

According to the eleven levels of shares, each learning automaton has 11 actions which are labeled with action 0, action 1, ..., and action 10. Convergence of learning automata to i th action (action i) means that learning automaton has decided to put its share ($\alpha_i^j(t)$) equal to $(0.1 \times i)$.

6.2 Evaluation Approach

Due to lack of similar work for comparison, the evaluation is based on comparing utility of service providers by adopting a loss sharing strategy versus their utility without any loss sharing. Also it's shown that the obtained LSAs in the conducted experiments reach results near to the results of a pareto optimal LSA. A LSA is pareto optimal if it is impossible to increase the utility of any service provider by changing its share ($\alpha_i^j(t)$) without any decrease in the utility of its neighbor service providers. We focused on pareto optimality concept, because it is assumed that loss sharing among service providers is a collaborative process. However the proposed method can operate under any conditions with unknown utility function forms and loss distributions, but to prove pareto optimality of a LSA it is needed to have exact form of utility functions and loss distributions.

Using utility function of Eq.(10) and exponential distribution of losses, choosing an $\alpha_i^j(t)$ appropriate to inverse of risk aversion of service providers causes to reach a pareto optimal LSA (For proof of this claim see Theorem 2 in appendix A). The conducted experiments compare results of the obtained LSA versus a pareto optimal LSA. Note that using the proposed method there is no need to know form of utility functions and loss distributions.

6.3 Results

As said in section 6.1, the experiments are conducted using 5 and 8 service providers in three different cases: case A, B and C. Figure 5 shows the tendency graphs for each case. The results illustrates the capability of ICLA in finding an appropriate LSA with results near to the results of a pareto optimal LSA. The applicability of the proposed method for a large number of service providers is also discussed in the following section.

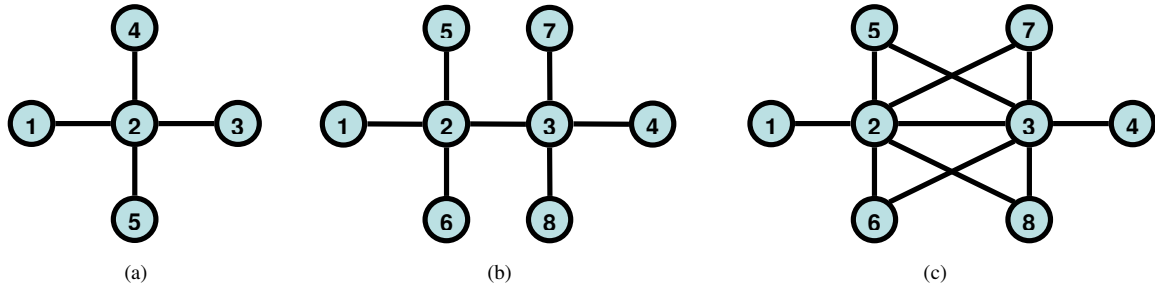


Fig. 5- Tendency graphs for (a) case A, (b) case B, (c) case C

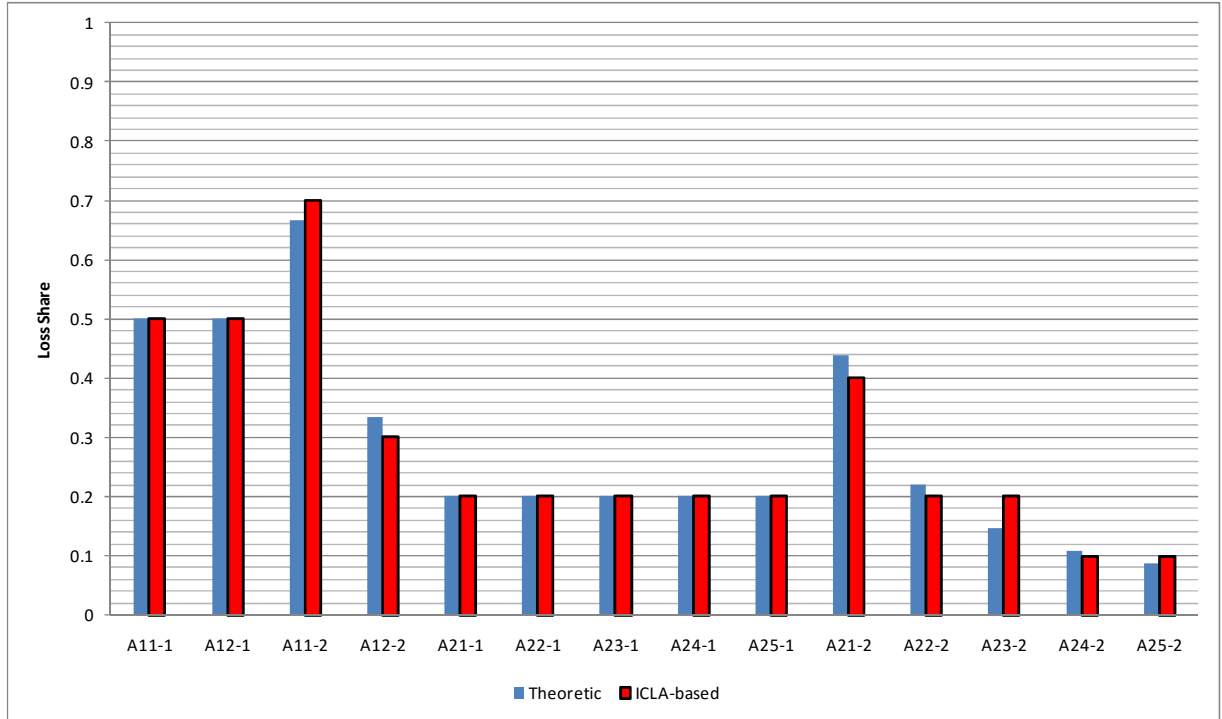
6.3.1 Capability of ICLA in finding pareto optimal LSA

To evaluate the capability of ICLA in finding an appropriate LSA, the results of the obtained LSA is compared with the results of a pareto optimal LSA under two different settings as shown in table 1. For each setting, risk aversion of a service provider is set according to table 1. In setting 1 all service providers have similar risk aversion but in setting 2 they have different risk aversion.

Table 1- Risk Aversion of Service Providers

<i>Risk Aversion</i>	<i>SP1</i>	<i>SP2</i>	<i>SP3</i>	<i>SP4</i>	<i>SP5</i>	<i>SP6</i>	<i>SP7</i>	<i>SP8</i>
<i>Setting 1</i>	0.0005	0.0005	0.0005	0.0005	0.0005	0.0005	0.0005	0.0005
<i>Setting 2</i>	0.0001	0.0002	0.0003	0.0004	0.0005	0.0006	0.0007	0.0008

Figures 6-13 illustrates the shares proposed by the obtained LSA using the proposed method (ICLA-based) and pareto optimal LSA (Theoretic). A_{ij-k} in horizontal axis represents α_i^j under setting k (k=1,2). Figure 6 illustrates the proposed shares for X_1 and X_2 under setting 1 for case A. As illustrated in this figures the difference between the shares proposed by ICLA and theoretic method is negligible. Figure 7 shows the proposed shares for X_3 , X_4 and X_5 have minor differences as well. Diagrams of figures 8-13 illustrate that even for more complex tendency graphs, such as case B and C, the shares proposed by the ICLA-based method are very close to shares proposed by the theoretic approach. By comparing the diagrams for case A (Figures 6 and 7) with diagrams of case B (Figures 8, 9 and 10) and C (Figures 11, 12 and 13), it can be seen that number of service providers (nodes in tendency graphs) have no considerable impact on accuracy of the proposed method. This is a valuable feature for the proposed method and shows capability of ICLA for reaching an appropriate LSA near to a pareto optimal LSA without even knowledge about loss distributions or utility function forms. However, for more complex tendency graphs, ICLA needs more iteration for finding an appropriate LSA.

Figure 6 – The proposed shares for X_1 and X_2 using the ICLA-based and theoretic methods under case A

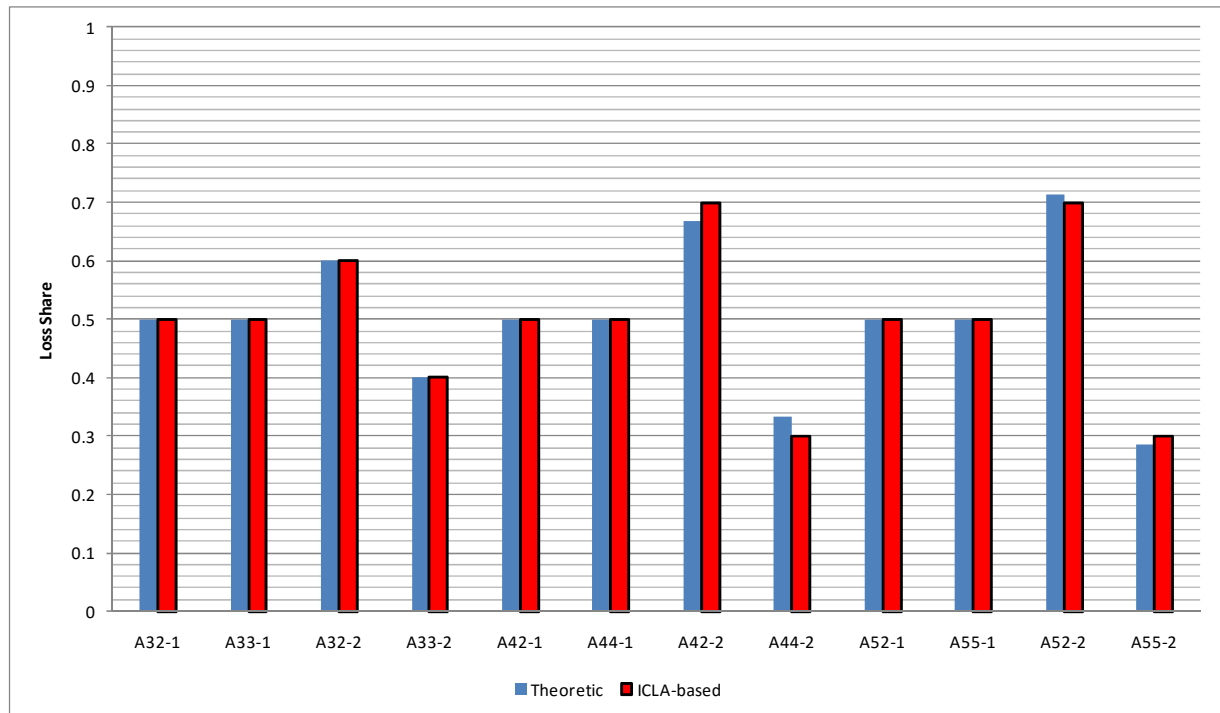


Figure 7 - The proposed shares for X_3 , X_4 and X_5 using the ICLA-based and theoretic methods under case A

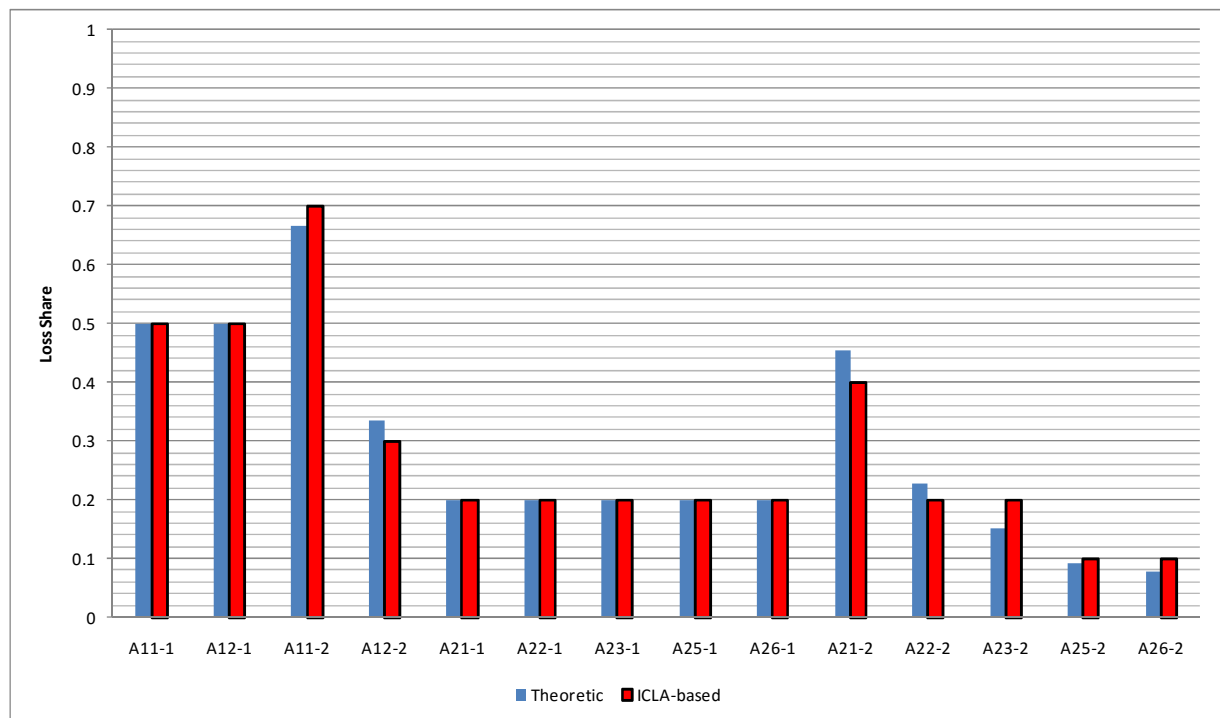


Figure 8 - The proposed shares for X_1 and X_2 using the ICLA-based and theoretic methods under case B

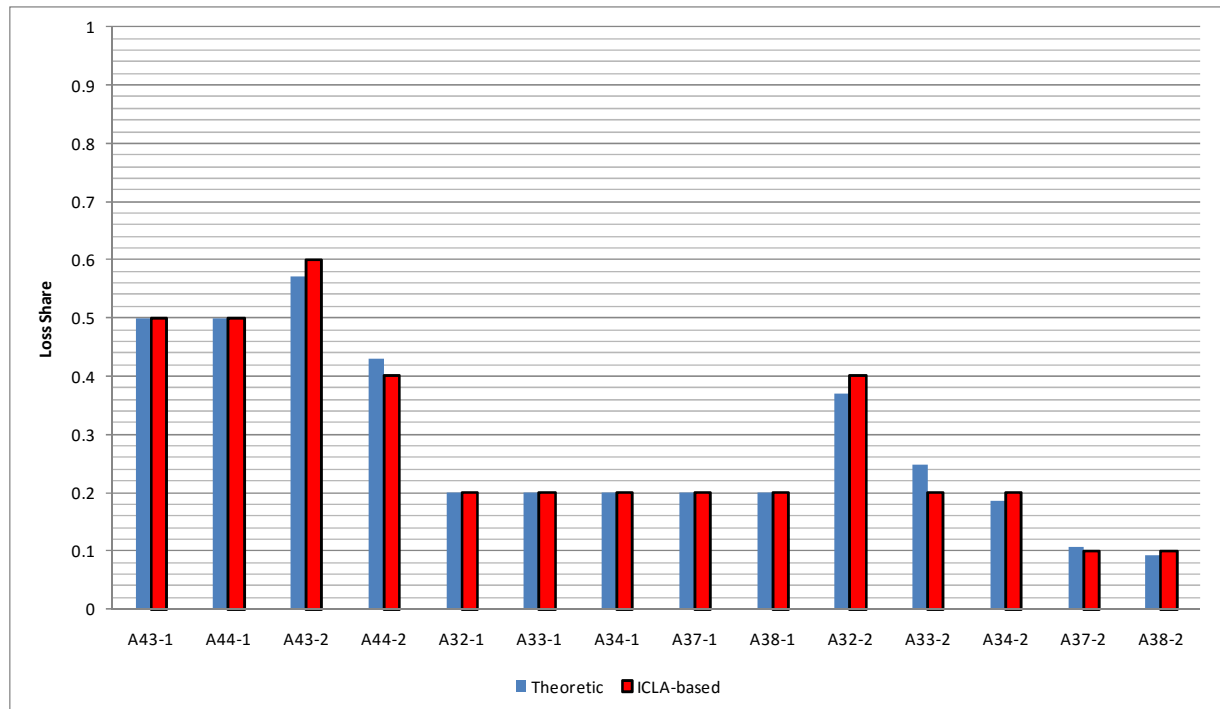


Figure 9 - The proposed shares for X_3 and X_4 using the ICLA-based and theoretic methods under case B

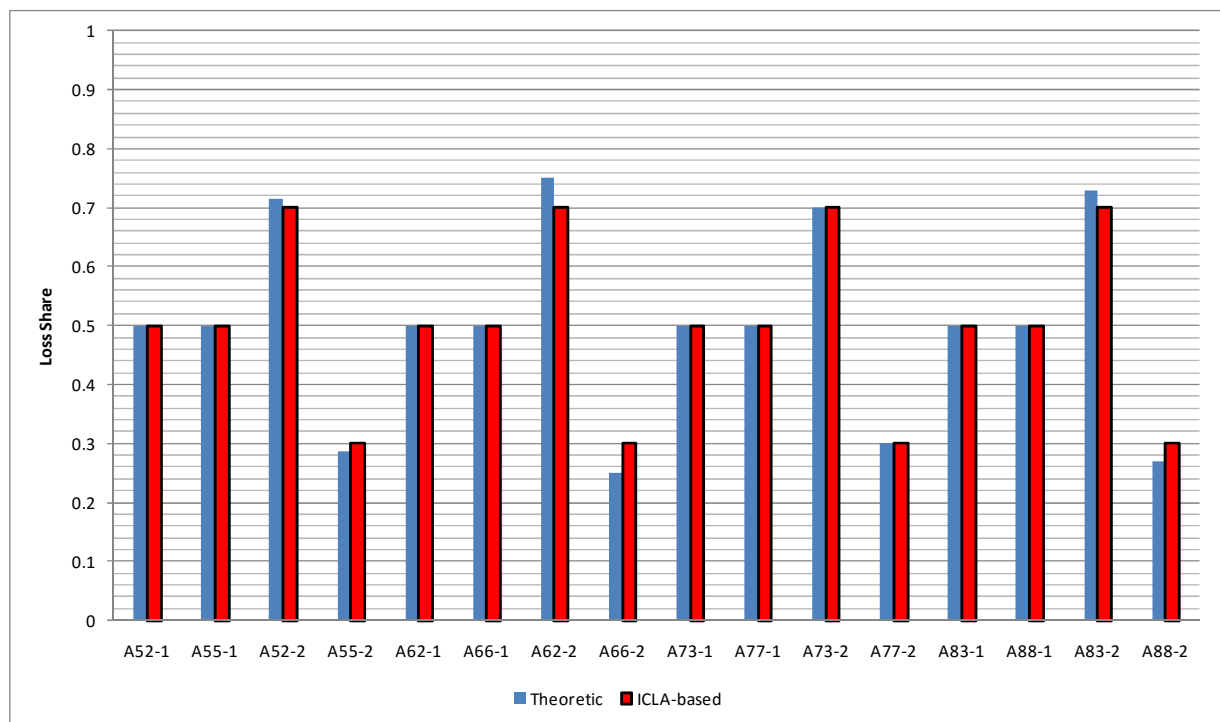


Figure 10 - The proposed shares for X_5 , X_6 , X_7 and X_8 using the ICLA-based and theoretic methods under case B

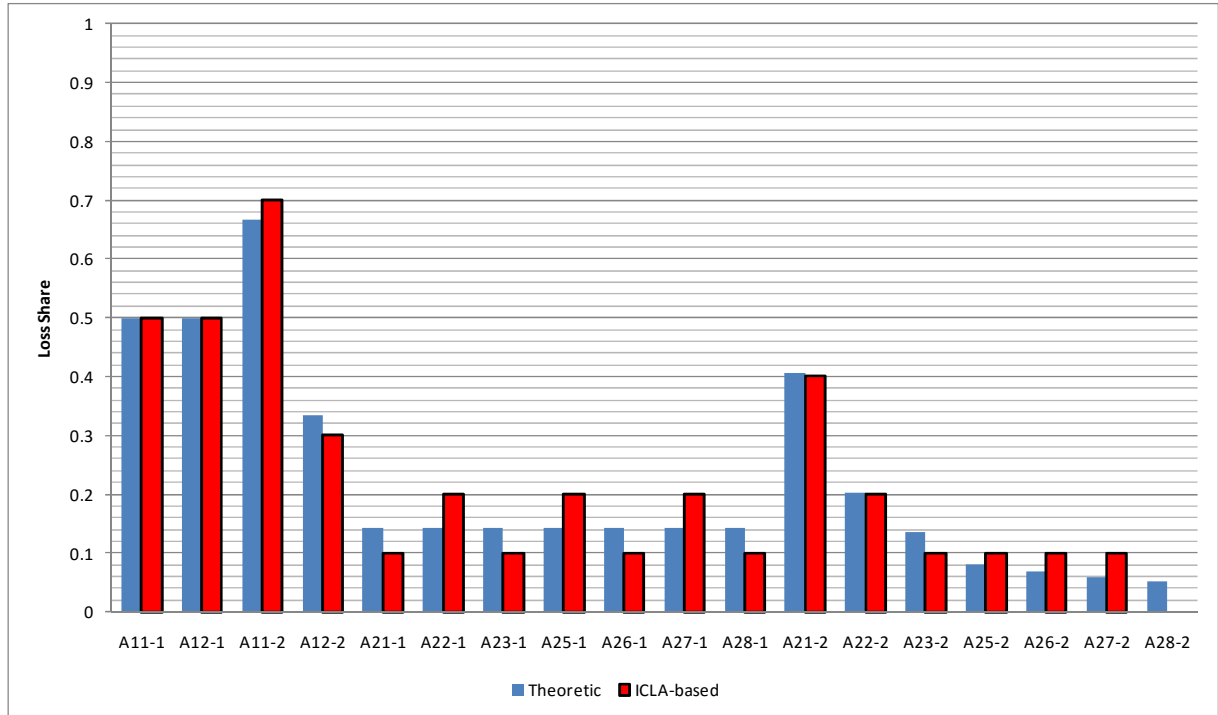


Figure 11 - The proposed shares for X_1 and X_2 using the ICLA-based and theoretic methods under case C

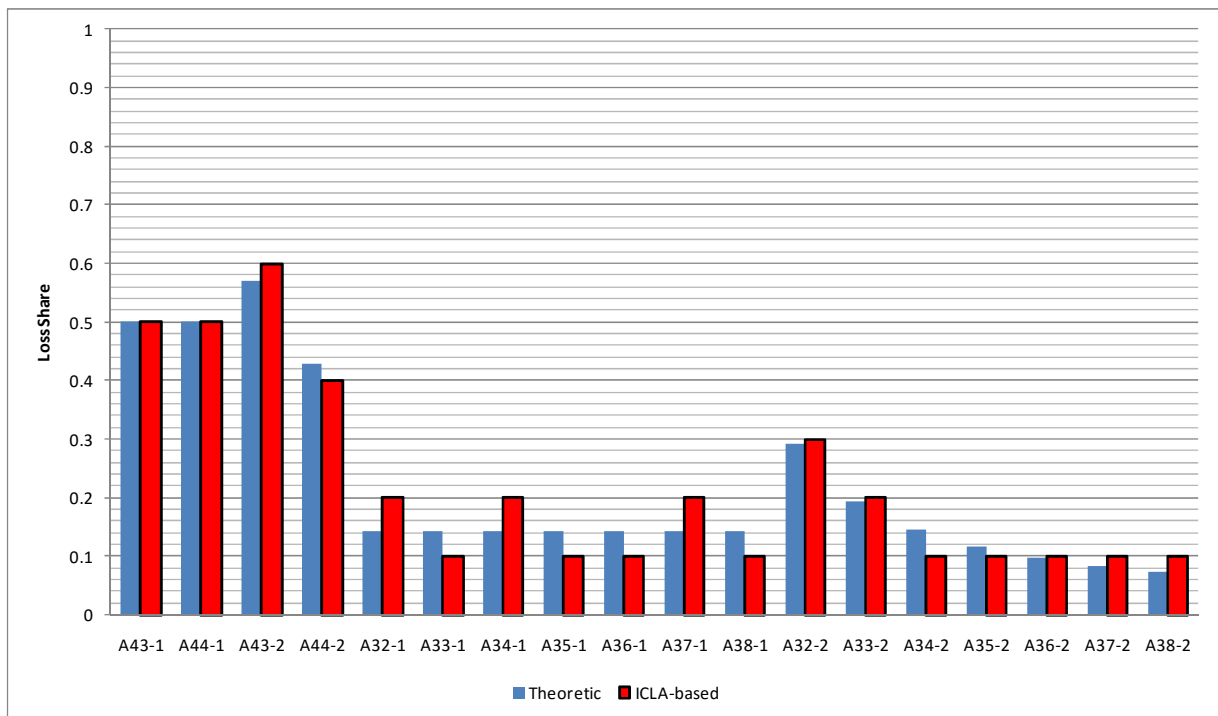


Figure 12 - The proposed shares for X_3 and X_4 using the ICLA-based and theoretic methods under case C

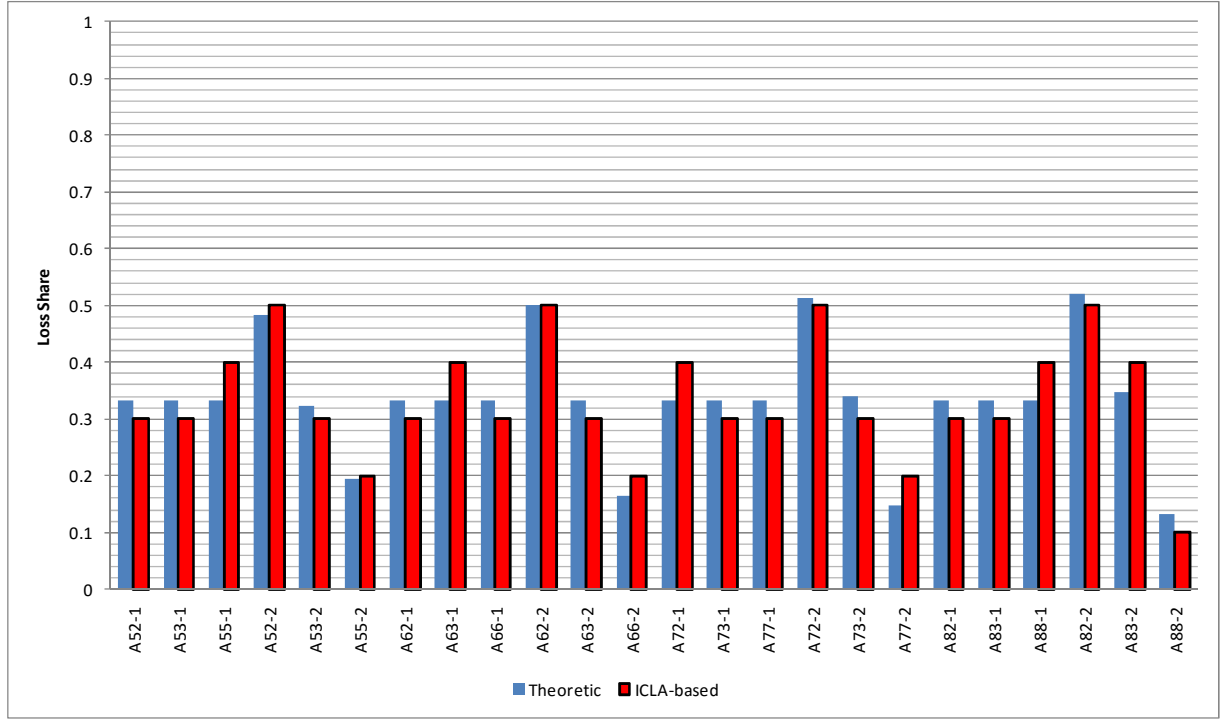


Figure 13 - The proposed shares for X_5 , X_6 , X_7 and X_8 using the ICLA-based and theoretic methods under case C

Figure 14(a) plots the evolution of the probability vector of LA_{21} which decides about shares of X_1 in case A. Figures 14 (b) and (c) show same diagrams for LA_{22} and LA_{25} . As illustrated in Figure 5(a), two, five and two service providers collaborate each other by sharing of X_1 , X_2 and X_5 respectively. The required iterations for convergence of LA_{21} , LA_{22} and LA_{25} to an action last about 700, 1600 and 650 iterations. This means that the required iterations for convergence depend on the number of the neighbors. Comparing the required iterations for convergence of LA_{21} , LA_{22} and LA_{25} in case B with case A and C (see figures 14(d)) illustrates that the required iterations for convergence are approximately invariant of the number of service providers, while it depend on the number of neighbors. LA_{21} , which decides for X_1 , needs 706, 762 and 837 iterations for convergence in cases A, B and C respectively, but number of the required iterations by LA_{22} have significant growth for case C. This is for the reason that number of the neighbors of SP2 increase from 4 to 6 neighbors in case C. Therefore in the proposed method the required iterations for finding a LSA is more dependent on maximum degree of the tendency graph than on the size of graph. This result illustrate that the proposed method is usable even for large graphs with reasonable maximum degree.

6.3.2 Usefulness of LSA for Risk Averse Service Providers

In this section, the utility of service providers are compared under three different situations: ICLA-sharing situation in which the service providers share their losses according to the obtained LSA by the proposed method, Opt-Sharing situation in which the loss sharing is according to the pareto optimal LSA and Non-Sharing situation in which the service providers operate isolated and there is no any loss sharing. Table 2 compares average utility of the service providers over 200 different iterations in case A, B and C. As illustrated in this table by participating in loss sharing, all the service providers can improve their utility. It is expected that the Opt-Sharing to reach better utility than ICLA-Sharing but there are some exceptions in table 2. These are due to randomness of the environment and in long time opt-sharing always reach better average utility. As illustrated in table 2, the difference of average utility

between ICLA-Sharing and Opt-Sharing is trivial. This shows that the proposed method has a good performance without even knowledge about form of utility functions and loss distributions. Now assume $\Delta u = u_{ICLA-Sharing} - u_{Non-Sharing}$ shows the utility difference between ICLA-sharing and Non-sharing situations.

Figure 15 shows the average of the utility difference ($\overline{\Delta u(t)}$) for SP1 and SP5 over the 200 different iterations. In this experiment risk aversion of SP1 and SP5 are 0.02 and 0.05 respectively. By comparing average utility difference of SP1 and SP5, it can be concluded that under similar conditions for two service providers, $\overline{\Delta u(t)}$ will be greater for the more risk averse service provider.

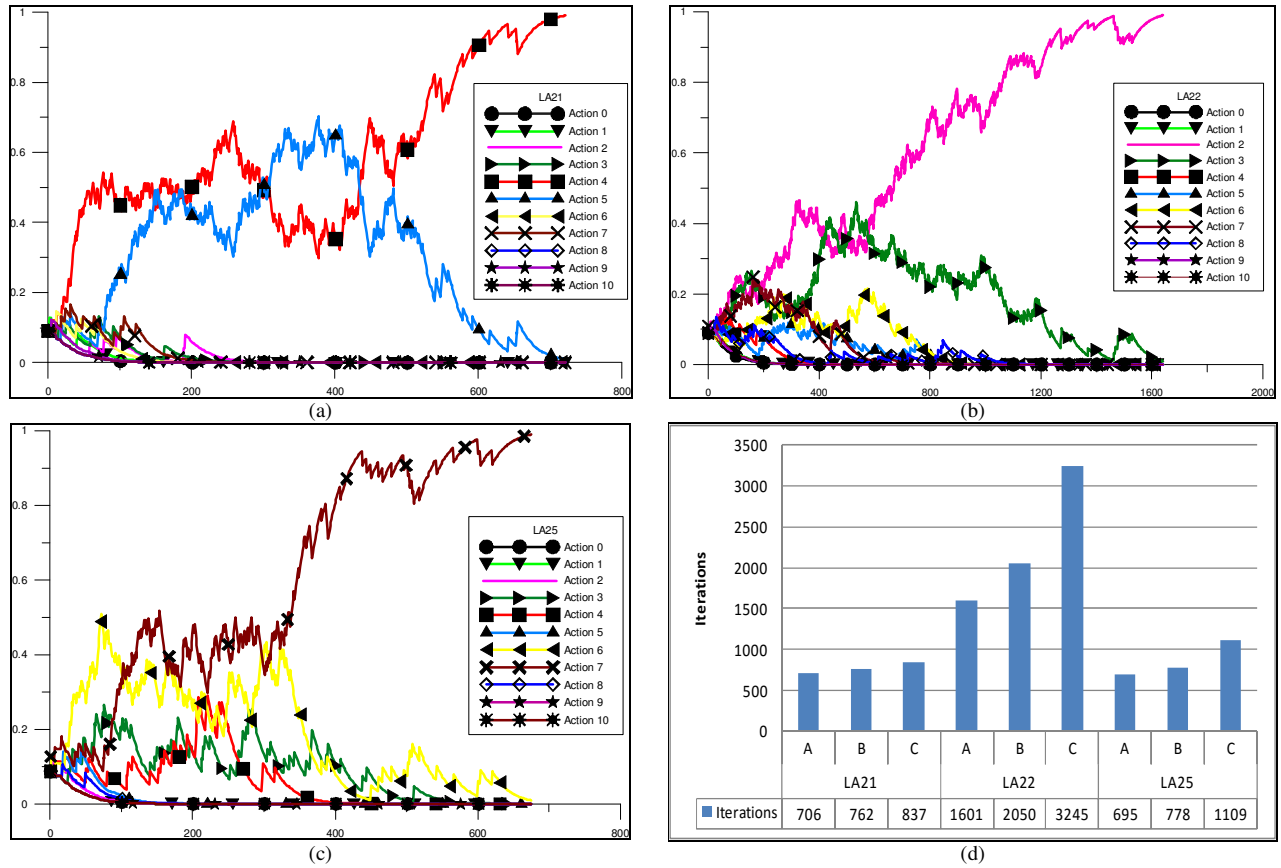


Fig. 14- Evolution of Probability Vector of (a)LA21 (b)LA22 (c)LA25 in case A (d) comparing the required iterations in different cases

Table 2 – The average utility of the service providers over 200 different iterations

		SP1	SP2	SP3	SP4	SP5	SP6	SP7	SP8
Case A	ICLA-Sharing	1.574	1.597	2.693	2.752	3.104	-	-	-
	Opt-Sharing	1.574	1.598	2.686	2.752	3.106	-	-	-
	Non-Sharing	1.5	1.501	2.541	2.571	2.964	-	-	-
Case B	ICLA-Sharing	1.578	1.589	2.676	2.652	3.102	3.095	3.932	3.94
	Opt-Sharing	1.578	1.589	2.678	2.652	3.104	3.097	3.926	3.935
	Non-Sharing	1.507	1.505	2.57	2.539	2.951	2.925	3.764	3.799
Case C	ICLA-Sharing	1.586	1.611	2.72	2.668	3.127	3.124	3.974	3.978
	Opt-Sharing	1.585	1.616	2.729	2.67	3.126	3.123	3.966	3.971
	Non-Sharing	1.518	1.514	2.585	2.571	2.963	2.948	3.776	3.825

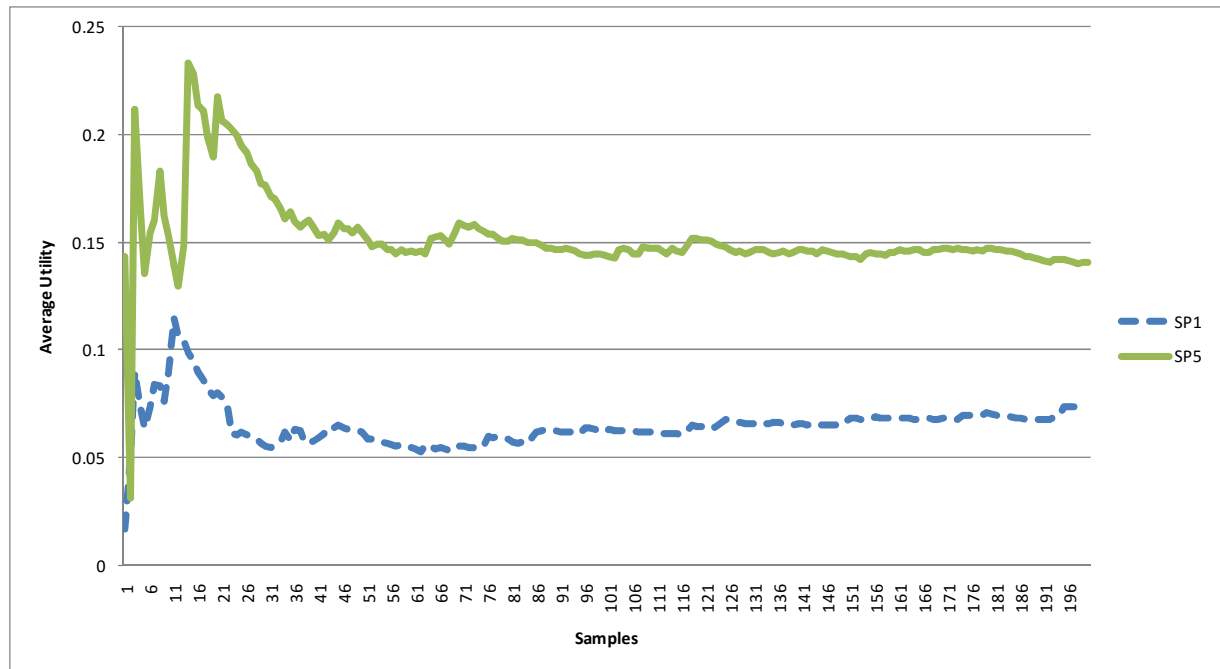


Fig. 15- The average of utility difference for SP1 and SP5

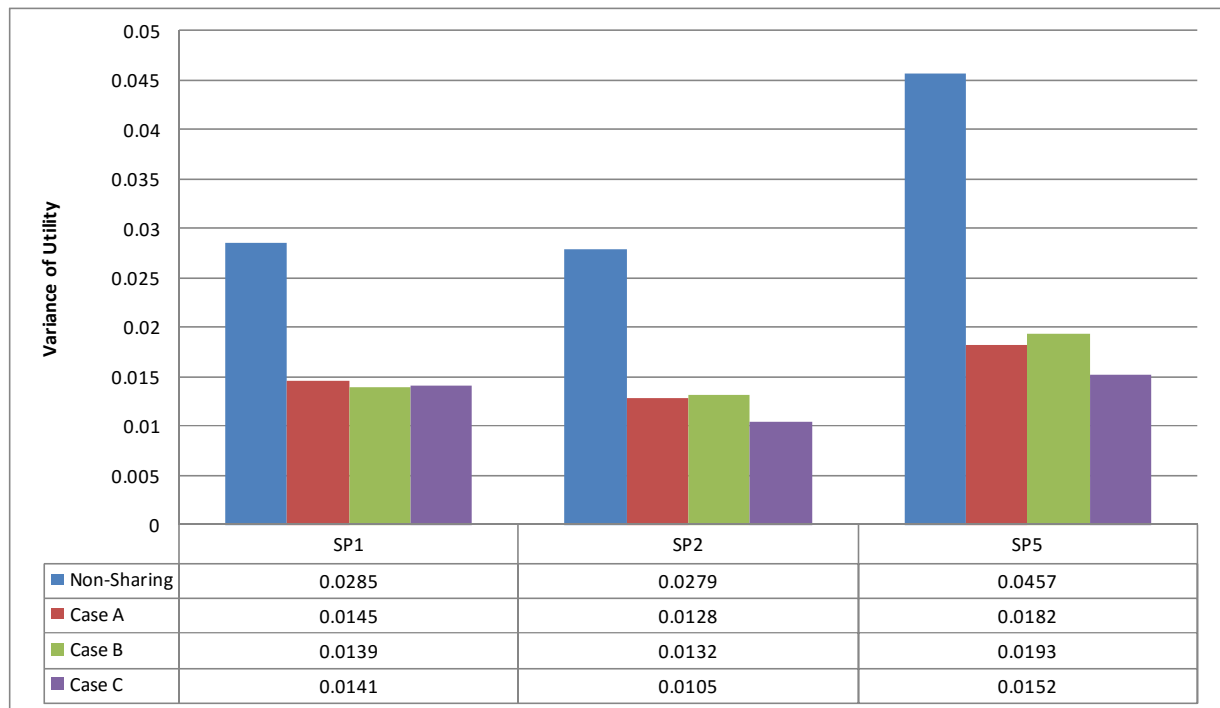


Fig. 16- variances of utilities of SP1, SP2 and SP5 under Non-Sharing and ICLA-Sharing situations in Cases A, B and C

Figure 16 illustrates the utility variances of SP1, SP2 and SP5 under Non-Sharing and ICLA-Sharing situations in Cases A, B and C. Because variance of utility in Non-Sharing situation is approximately identical for the three cases so it is plotted once. As shown in figure 16, under ICLA-Sharing situation, the utility variance drops considerably for all the service providers. Because the service providers are risk averse so they prefer a more stable utility and this drop is desirable for them. The differences of variances are considerable in cases B and C for SP2 and SP5 while this is not the case for SP1. This is for the reason that SP1 has same collaboration tendency in both cases B and C, while SP2 and SP5 extend their collaboration tendency in case C. Existence of more number of service providers in loss sharing causes more drops in variance of utility and service providers experience more stable utility.

7 Conclusion

In this paper a new method is presented for loss sharing to increase utility of risk averse service providers. Due to risk aversion property, service providers prefer more certain conditions when they are confronted with two choices with same expected utility. Thus if collaboration with other service providers can decrease uncertainty, a risk averse service provider is interested to collaborate. Based on this fact, a new method is proposed for loss sharing and decreasing variance of random losses. Due to capabilities of learning automata and ICLA in environments with high uncertainty, ICLA is employed to present the loss sharing method. In the proposed method learning automata decide about level of loss sharing. Results of the conducted experiments illustrated that the proposed method is capable of finding an appropriate LSA such that it increases utilities of all service providers. In addition the utilities obtained by the proposed method have trivial difference with the utilities of a pareto optimal LSA. Finding a pareto optimal LSA needs detailed information about utility functions and loss distributions, while there is no need to such information by the proposed method and this is a notable feature of this method. Also the results illustrated that the proposed method can be employed for a large number of service providers and the time needed for finding a LSA is almost invariant of the number of service providers. As a future work we aim to evaluate this approach in real world applications in service oriented environments. Federated clouds can be a suitable choice for employing the proposed method.

References

- [1] Aase, K. K. (1992). Dynamic equilibrium and the structure of premiums in a reinsurance market. *The Geneva Papers on Risk and Insurance Theory*, 17(2), 93-136.
- [2] Aase, K. K. (1993). Equilibrium in a reinsurance syndicate; Existence, uniqueness and characterization. *Astin Bulletin*, 23(02), 185-211.
- [3] Beigy, H., & Meybodi, M. R. (2003). A self-organizing channel assignment algorithm: A cellular learning automata approach. In *Intelligent Data Engineering and Automated Learning* (pp. 119-126). Springer Berlin Heidelberg.
- [4] Beigy, H., & Meybodi, M. R. (2004). A mathematical framework for cellular learning automata. *Advances in Complex Systems*, 7(03n04), 295-319.
- [5] Borch, K. (1992). Equilibrium in a reinsurance market. In *Foundations of Insurance Economics* (pp. 230-250). Springer Netherlands.
- [6] Buyya, R., Yeo, C. S., Venugopal, S., Broberg, J., & Brandic, I. (2009). Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Generation computer systems*, 25(6), 599-616.
- [7] Buyya, R., Yeo, C. S., & Venugopal, S. (2008, September). Market-oriented cloud computing: Vision, hype, and reality for delivering it services as computing utilities. In *High Performance Computing and Communications*, 2008. HPCC'08. 10th IEEE International Conference on (pp. 5-13).
- [8] Buyya, R., Vecchiola, C., & Selvi, S. T. (2013). *Mastering cloud computing: foundations and applications programming*. Newnes.
- [9] Chateauneuf, A., Mostoufi, M., & Vyncke, D. (2015). Multivariate risk sharing and the derivation of individually rational Pareto optima. *Mathematical Social Sciences*, 74, 73-78.
- [10] Cheung, K. C., Sung, K. C. J., Yam, S. C. P., & Yung, S. P. (2014). Optimal reinsurance under general law-invariant risk measures. *Scandinavian Actuarial Journal*, 2014(1), 72-91.
- [11] Chi, Y., & Tan, K. S. (2013). Optimal reinsurance with general premium principles. *Insurance: Mathematics and Economics*, 52(2), 180-189.

- [12] Deelstra, G., & Plantin, G. (2014). Optimal Reinsurance. In *Risk Theory and Reinsurance* (pp. 59-76). Springer London.
- [13] Esnaashari, M. and Meybodi, M. R., "Irregular Cellular Learning Automata", *IEEE Transactions on Cybernetics, Issue 99, 2014, DOI: 10.1109/TCYB.2014.2356591*
- [14] Goiri, I., Guitart, J., & Torres, J. (2010, July). Characterizing cloud federation for enhancing providers' profit. In *Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on* (pp. 123-130). IEEE.
- [15] Goiri, Í., Guitart, J., & Torres, J. (2012). Economic model of a Cloud provider operating in a federated Cloud. *Information Systems Frontiers, 14*(4), 827-843.
- [16] Haimes, Y. Y. (2005). *Risk modeling, assessment, and management* (Vol. 40). John Wiley & Sons.
- [17] Rezvanian, A., & Meybodi, M. R. (2015). Finding Minimum Vertex Covering in Stochastic Graphs: A Learning Automata Approach. *Cybernetics and Systems, 46*(8), 698-727.
- [18] Rochwerger, B., Breitgand, D., Levy, E., Galis, A., Nagin, K., Llorente, I. M., ... & Galan, F. (2009). The reservoir model and architecture for open federated cloud computing. *IBM Journal of Research and Development, 53*(4), 4-1.
- [19] Rochwerger, B., Breitgand, D., Epstein, A., Hadas, D., Loy, I., Nagin, K., ... & Toffetti, G. (2011). Reservoir—when one cloud is not enough. *IEEE computer, 44*(3), 44-51.
- [20] Samaan, N. (2014). A novel economic sharing model in a federation of selfish cloud providers. *IEEE Transactions on Parallel and Distributed Systems, 25*(1), 12-21.
- [21] Thathachar, M. (1987). Learning Automata with Changing Number of Actions. *IEEE Transactions on Systems, Man, and Cybernetics 17* (6): 1095–1100.
- [22] Toosi, A. N., Calheiros, R. N., Thulasiram, R. K., & Buyya, R. (2011, September). Resource provisioning policies to increase IaaS provider's profit in a federated cloud environment. In *High Performance Computing and Communications (HPCC), 2011 IEEE 13th International Conference on* (pp. 279-287). IEEE.
- [23] Toosi, A. N., Thulasiram, R. K., & Buyya, R. (2012, November). Financial option market model for federated cloud environments. In *Proceedings of the 2012 IEEE/ACM Fifth International Conference on Utility and Cloud Computing* (pp. 3-12). IEEE Computer Society.
- [24] Torkestani, J. A., & Meybodi, M. R. (2011). A cellular learning automata-based algorithm for solving the vertex coloring problem. *Expert Systems with Applications, 38*(8), 9237-9247.
- [25] Zweifel, P., & Eisen, R. (2012). *Insurance economics*. Springer.
- [26] <https://en.wikipedia.org/wiki/Reinsurance>
- [27] http://en.wikipedia.org/wiki/Utility#Quantifying_utility

APPENDIX A

We claimed that provided the exponential distribution of losses and utility functions in form of Eq.(10), choosing a $\alpha_i^j(t)$ appropriate to inverse of risk aversion of service provider i conduces to a pareto optimal LSA. To prove this claim first we need a definition and a theorem.

Definition- Indifferent Level of Income (PI_{ILL}): PI_{ILL} is certain amount of income for which, a risk averse service provider is indifferent between having a risky income PI_i (See Eq. (5)) and having this certain amount of income (PI_{ILL}^i). In other word we have $u_i(PI_{ILL}^i) = E(u_i(PI_i))$

$$PI_{ILL}^i = u_i^{-1}(E(u_i(PI_i))) \quad (A-1)$$

Note: If we substitute utility function of Eq. (10) in Eq. (A-1) we obtain Eq. (A-2):

$$PI_{ILL}^i = u_i^{-1}(E(u_i(PI_i))) = \frac{\ln(E(e^{-a_i PI_i}))^{-1}}{a_i} \quad (A-2)$$

Theorem 1- A LSA is pareto optimal if and only if for each j , $\sum_{i \in N_j} PI_{ILL}^i$ reaches its maximum value.

Proof: Assume that a non-pareto optimal LSA such as C , whose income for provider i is PI_i , gets maximum value for $\sum_{i \in N_j} PI_{ILL}^i$ for each j . According to (A-2) we have (A-3).

$$\sum_{i \in N_j} PI_{ILL}^i = \sum_{i \in N_j} \frac{\ln(E(e^{-a_i PI_i}))^{-1}}{a_i} = \max_{PI'} \left(\sum_{i \in N_j} \frac{\ln(E(e^{-a_i PI'_i}))^{-1}}{a_i} \right) \quad (A-3)$$

PI' is a vector and its elements are pure income of service provider j and its neighbors. Since C is not pareto optimal so there is a LSA such as C'' whose income is PI''_i for service provider i ($i \in N_j$) and for each $i \in N_j$ we have:

$$E(u_i(PI''_i)) > E(u_i(PI_i)) \Rightarrow E(c_i e^{-a_i PI''_i} - c_i) > E(c_i e^{-a_i PI_i} - c_i) \Rightarrow (E(e^{-a_i PI''_i}))^{-1} > (E(e^{-a_i PI_i}))^{-1} \quad (A-4)$$

For the reason that $a_i > 0$ and \ln is an increasing function, so for each $i \in N_j$

$$\frac{\ln(E(e^{-a_i PI''_i}))^{-1}}{a_i} > \frac{\ln(E(e^{-a_i PI_i}))^{-1}}{a_i} \quad (A-5)$$

Therefore we have (A-4):

$$\sum_{i \in N_j} \frac{\ln(E(e^{-a_i PI''_i}))^{-1}}{a_i} > \sum_{i \in N_j} \frac{\ln(E(e^{-a_i PI_i}))^{-1}}{a_i} \quad (A-6)$$

According to (A-3), RHS of (A-6) is equal to maximum value and this means LHS is bigger than maximum. This is contradiction and means **C is pareto optimal LSA**.

Now assume a pareto optimal LSA such as C whose income is PI_i for service provider i , but it doesn't satisfy (A-3). As a result there is a LSA such as C'' whose income is PI_i'' for service provider i ($i \in N_j$) and satisfy (A-7).

$$\sum_{i \in N_j} \frac{\ln(E(e^{-a_i PI_i''}))^{-1}}{a_i} > \sum_{i \in N_j} \frac{\ln(E(e^{-a_i PI_i}))^{-1}}{a_i} \quad (\text{A-7})$$

Now consider a LSA like C' which has shares (α_i^j) similar to shares of C'' ($(\alpha_i^j(C') = \alpha_i^j(C''))$) but different pr_i^j ($pr_i^j(C') \neq pr_i^j(C'')$). Let $pr_i^j(C')$ to be defined as (A-8).

$$pr_i^j(C') = pr_i^j(C'') - \frac{\ln(E(e^{-a_i PI_i''}))^{-1}}{a_i} + \frac{\ln(E(e^{-a_i PI_i}))^{-1}}{a_i} + \phi_i \quad (\text{A-8})$$

Where ϕ_i is a positive value as (A-9):

$$\phi_i = \frac{1}{|N_j|} \times \left(\sum_{i \in N_j} \frac{\ln(E(e^{-a_i PI_i''}))^{-1}}{a_i} - \sum_{i \in N_j} \frac{\ln(E(e^{-a_i PI_i}))^{-1}}{a_i} \right) \quad (\text{A-9})$$

Because $\alpha_i^j(C') = \alpha_i^j(C'')$ thus

$$PI_i' = I_i + \sum_{j \in N_i} pr_j^i(C') - \sum_{i \in N_j} \alpha_j^i(C') X_j = I_i + \sum_{j \in N_i} (pr_j^i(C'') - \frac{\ln(E(e^{-a_i PI_i''}))^{-1}}{a_i} + \frac{\ln(E(e^{-a_i PI_i}))^{-1}}{a_i} + \phi_i) - \sum_{i \in N_j} \alpha_j^i(C'') X_j \quad (\text{A-10})$$

$$\text{Let } \gamma_i = |N_i| \times \left(\phi_i - \frac{\ln(E(e^{-a_i PI_i''}))^{-1}}{a_i} + \frac{\ln(E(e^{-a_i PI_i}))^{-1}}{a_i} \right) \text{ so consequently we have } PI_i' = PI_i'' + \gamma_i.$$

Since $E(\gamma_i) = \gamma_i$ as a result:

$$\frac{\ln(E(e^{-a_i PI_i'}))^{-1}}{a_i} = \frac{\ln(E(e^{-a_i PI_i''}))^{-1}}{a_i} + \gamma_i \quad (\text{A-11})$$

By substitution of value of γ_i , we have (A-12).

$$\frac{\ln(E(e^{-a_i PI_i'})^{-1}}{a_i} = \frac{\ln(E(e^{-a_i PI_i}))^{-1}}{a_i} + |N_i| \times \phi_i \quad (\text{A-12})$$

Since ϕ_i is positive so we have (A-11).

$$\frac{\ln(E(e^{-a_i PI_i'})^{-1}}{a_i} > \frac{\ln(E(e^{-a_i PI_i}))^{-1}}{a_i} \quad (\text{A-13})$$

(A-13) is valid for each $i \in N_j$ and this means LSA C is not pareto optimal. This is contradiction and means that a

pareto optimal LSA maximizes $\sum_{i \in N_i} PI_{ILL}^i$ \square .

Now using Theorem 1 we can prove Theorem 2.

Theorem 2- Using utility function of Eq.(10) and exponential distribution of losses, choosing an α_i^j appropriate to inverse of risk aversion of service providers causes to reach a pareto optimal LSA.

Proof: From theorem 1 we know that A LSA is pareto optimal if and only if $\sum_{i \in N_j} PI_{ILL}^i$ reach its maximum value for each j . So if we show that the shares appropriate to inverse of risk aversion maximizes $\sum_{i \in N_j} PI_{ILL}^i$, the proof will be complete. Using Eq. (5) we can write:

$$PI_{ILL}^i(t) = u_i^{-1}(E(u_i(PI_i(t)))) \quad (A-14)$$

For simplicity purpose, in following we skip iteration index t and also we use pr_i instead of $\sum_{j \in N_i} pr_i^j(t)$. According to (A-2) we obtain (A-15).

$$PI_{ILL}^i = -\frac{1}{a_i} \ln(E(e^{-a_i PI_i})) \quad (A-15)$$

At each iteration pr_i has a fixed value while $\sum_{j \in N_i} \alpha_j^i X_j$ depends on random losses, thus we haven (A-16).

$$PI_{ILL}^i = -\frac{1}{a_i} \ln(E(e^{-a_i PI_i})) = -\frac{1}{a_i} \ln(e^{-a_i pr_i} E(e^{a_i \sum_{j \in N_i} \alpha_j^i X_j})) \quad (A-16)$$

Because losses are independent random variables thus

$$PI_{ILL}^i = -\frac{1}{a_i} (\ln(e^{-a_i pr_i}) + \sum_{j \in N_i} \ln(E(e^{a_i \alpha_j^i X_j}))) \quad (A-17)$$

(A-17) can be written in form of (A-18):

$$PI_{ILL}^i = pr_i - \frac{1}{a_i} \sum_{j \in N_i} \ln(E(e^{a_i \alpha_j^i X_j})) \quad (A-18)$$

Since for each random loss X_j have exponential distribution with parameter of θ_j , so we have (A-19).

$$E(e^{a_i \alpha_j^i X_j}) = \frac{1}{1 - \theta_j a_i X_j} \quad (A-19)$$

By substituting (A-19) in (A-18) we have:

$$PI_{ILL}^i = pr_i - \frac{1}{a_i} \sum_{j \in N_i} \ln\left(\frac{1}{1 - \theta_j a_i \alpha_j^i}\right) = pr_i + \frac{1}{a_i} \sum_{j \in N_i} \ln(1 - \theta_j a_i \alpha_j^i) \quad (\text{A-20})$$

So for each z we have:

$$\sum_{i \in N_z} PI_{ILL}^i = \sum_{i \in N_z} pr_i + \sum_{i \in N_z} \sum_{j \in N_i} \frac{1}{a_i} \ln(1 - \theta_j a_i \alpha_j^i) \quad (\text{A-21})$$

Since $\sum_{i \in N_z} pr_i$ are fixed so if we show that the shares equal to inverse of risk aversions maximizes second term of

RHS of (A-21) for each z such that $\sum_{j \in N_i} \alpha_j^i = 1$, the proof will be completed. This is the case using Karush-Kuhn-

Tucker conditions and it is straight forward that $\alpha_j^i = \frac{1}{\sum_{j \in N_i} \frac{a_i}{a_j}}$ maximizes $\sum_{i \in N_z} PI_{ILL}^i$ \square .