Risk-aware Social Cloud Computing based on Serverless Computing Model

Pavlos Athanasios Apostolopoulos*, Eirini Eleni Tsiropoulou*, and Symeon Papavassiliou‖
{pavlosapost@unm.edu, eirini@unm.edu, papavass@mail.ntua.gr}
* *Dept. of Electrical and Computer Engineering, The University of New Mexico*, Albuquerque, NM, USA
‖ *School of Electrical and Computer Engineering, National Technical University of Athens*, Athens, Greece

***Abstract*—In this paper, a flexible resource sharing paradigm is introduced, to enable the allocation of users' computing tasks in a social cloud computing system offering both *Virtual Machines (VMs)* and *Serverless Computing (SC)* functions. VMs are treated as a safe computing resource, while SC due to the uncertainty introduced by its shared nature, is treated as a common pool resource, being susceptible to potential over-exploitation. These computing options are differentiated based on the potential satisfaction perceived by the user, as well as their corresponding pricing, while taking into account the social interactions among the users. Considering the inherent uncertainty of the considered computing environment, Prospect Theory and the theory of the Tragedy of the Commons are adopted to properly reflect the users' behavioral characteristics, i.e., gain-seeking or loss-averse behavior, as well as to formulate appropriate prospect-theoretic utility functions, embodying the social-aware and risk-aware user's perceived satisfaction. A distributed maximization problem of each user's expected prospect-theoretic utility is formulated as a non-cooperative game among the users and the corresponding *Pure Nash Equilibrium (PNE)*, i.e., optimal computing jobs offloading to the VMs and the SC, is determined, while a distributed low-complexity algorithm that converges to the PNE is introduced. The performance and key principles of the proposed framework are demonstrated through modeling and simulation.**

***Index Terms*—Social Computing, Virtual Machines, Serverless Computing, Prospect Theory, Risk-aware Behavior.**

I. Introduction

The remarkable growth of social networks over the last decade - evidenced by the more than 1.62 billion Facebook users and the 270 million Twitter users in 2019 - has concluded to new solutions for communication networks and mobile computing. It is predicted that by 2020, 67% of the overall enterprise information technology infrastructure and software development will be served by cloud-based offerings [1].

A. Related Work and Motivation

The *Social Cloud Computing* is arising as a resource sharing framework, which exploits the users' social ties to improve the services offered by the cloud providers. In [2], the trust levels among users in social networks are exploited to create a dynamic social cloud computing environment, where the users are sharing their cloud computing resources, creating a volunteer social cloud computing environment. In [3], the authors tackle the problem of placing the users' computing tasks over multiple clouds considering the social-aware services and the users' social ties.

While social cloud computing is still in its infancy, a new cloud computing solution is coined by the industry, named *Serverless Computing (SC)*, where the users' computing tasks are defined as a workflow of event-triggered functions [4]. In contrast to the model of *Virtual Machines (VMs)*, where the users are renting the VMs from the cloud provider and the resources could remain idle in the case of sporadic requested computing tasks, concluding to unwanted monetary cost, the SC model allows the users to offload computing tasks to the cloud provider, who remains responsible to manage the infrastructure and respective resources [5]. The users run stateless functions at the cloud providers' servers and are charged with respect to the allocated memory and the actual required CPU time of executing them [6], thus promising a more cost-efficient and flexible model compared to the VMs. Example platforms supporting the SC model include: AWS Lambda [7], Google Cloud Functions [8], etc.

However, all these efforts in social cloud computing and serverless computing have been progressing in isolation of each other. Thus, despite the advances that have been achieved in both these areas independently, the lack of joint consideration and exploitation of the users' social relation and the available VMs and SC by the cloud provider, limits their potential exploitation and adoption in a realistic scenario.

B. Contributions & Outline

In this paper, we aim to fill the aforementioned research gaps by introducing a risk-aware social computing framework, which exploits the computing capabilities of the available VMs and SC offered by the cloud provider, while accounting for the users' social ties and their risk-aware behavior. The latter stems from the risk imposed by the shared nature of the SC, which may become non-responsive due to its over-exploitation. The specific contributions of this paper are as follows.

1. Each user dynamically offloads part of its computing jobs to the VMs and/or the SC (Section II). Fixed price is assumed for the use of VMs, while a social-aware SC pricing is considered based on the ”social importance” of a user within the system (Section II-A). A holistic user's actual utility function is introduced to capture the user's satisfaction by executing its jobs in a specific time frame, while considering the corresponding price (Section II-B).
2. Each VM is characterized as a ”safe resource”, as it is exclusively rent by a user, and accordingly the user enjoys

guaranteed computing service. In contrast, the SC is typically more cost-efficient, having the potential of providing high satisfaction to the user. Given that its computing resources are shared among many users, it is characterized as Common Pool Resource (CRP), which introduces risk in users' decisions to offload their computing jobs to it, as it can potentially become over-exploited. We capture this phenomenon via adopting the theory of the Tragedy of the Commons [9] (Section III-A). The problem of users' risk-aware offloading decision to the VMs and the SC, is formulated by using the Prospect Theory [10]. Each user's prospect-theoretic utility function is introduced by considering its actual utility, its behavioral patterns, and the probability of the SC's failure (i.e., non-responsiveness) (Section III-B).

3. The problem of each user determining the number of computing jobs that will be offloaded at the SC and the VMs, is formulated as a maximization problem of its expected prospect-theoretic utility, and treated as a non-cooperative game among the users (Section IV). The existence and uniqueness of a Pure Nash Equilibrium (PNE) is shown (Section V), while a distributed and low-complexity algorithm is introduced, and its convergence to the unique PNE is proven (Section VI).

4. A series of simulation experiments is performed to evaluate the performance of the proposed risk-aware social computing framework. A comparative study demonstrates its superiority, in terms of user's satisfaction and proper system operation (Section VII). Finally, Section VIII concludes the paper.

II. System Model

A Cloud Provider (CP) consisting of Virtual Machines (VMs) and Serverless Computing (SC) functions is considered. A set of $\mathbb{N} = \{1, \cdots, N\}$ users is assumed, while a set of $\mathbb{T} = \{1, \cdots, T\}$ time slots is defined, where D_t denotes the duration of each time slot t. Each user i has a number $J_i^{(t)}$ of computing jobs that wants to offload to the CP for remote execution per time slot. In the VMs case, a user can reserve a VM with its own operating system and predefined on demand computational and storage capabilities, while in the SC the user executes its own serverless instance as an application in a common operating system without any control over the resources on which the job is executed. Given a specific type of VMs, we define as $\lambda_{vm}^{(t)}(D_t)$ the maximum number of jobs that can be executed by the VM in the duration D_t, where $\lambda_{vm}^{(t)}(D_t)$ is an increasing function of the duration D_t. Each user i aims at determining the number $\lambda_i^{(t)}$ ($\lambda_i^{(t)} \leq J_i^{(t)}$) of computing jobs to be executed at the SC, while the rest ($J_i^{(t)} - \lambda_i^{(t)}$) jobs are offloaded to the VMs.

A. Social-aware Cloud Aspects and Pricing

With respect to the social aspects of the cloud computing system, we define an overlay virtual representation of the system as follows: $\mathbb{S} = \{\mathbb{N}, \mathbb{E}, \mathbb{W}\}$, where the users $\mathbb{N} = \{1, \cdots, N\}$ may interact with each other. Specifically, the edge set, i.e., interactions, is denoted as $\mathbb{E} = \{(i,j) : e_{i,j} = 1, \forall i, j \in \mathbb{N}\}$, where $e_{i,j} = 1$ indicates the existence of information flow from user i to user j. The weight set $\mathbb{W} = \{w_{i,j}, \forall i, j \in \mathbb{N}\}$ is defined, where $w_{i,j} \in \mathbb{R}$ depicts the strength of the interaction (e.g., criticality of information) that is exchanged between the source user i and the destination user j, while $w_{i,j} = 0, \forall i, j \in \mathbb{N}$ such that $e_{i,j} = 0$. Therefore, each user is characterized by its social factor $f_i = \frac{\omega_1 \sum_{j \in \mathbb{N}, j \neq i} w_{i,j} + \omega_2 \sum_{j \in \mathbb{N}, j \neq i} w_{j,i}}{\sum_{j \in \mathbb{N}} f_j}$, where $\omega_1, \omega_2 \in [0,1]$, $\omega_1 + \omega_2 = 1$ depict the weights of a user's interactions by acting as a sender or receiver of information, respectively.

In the VMs case, each user is charged based on the reserved VMs: $p_{i,vmp}^{(t)} = \left\lceil \frac{J_i^{(t)} - \lambda_i^{(t)}}{\lambda_{vm}^{(t)}(D_t)} \right\rceil \cdot p_{vm}$, where p_{vm} is a fixed VM's price [6]. In the SC case, the user is charged based on its execution time: $p_{i,sc}^{(t)} = \lambda_i^{(t)} D_t f_i^{-1} p_{sc}^{(t)}(\lambda_T^{(t)})$, where the average SC's response time is D_t and f_i^{-1} shows that the more important is a user for the social cloud computing system the greater is the incentive for the SC to assign a lower price. The $p_{sc}^{(t)}(\lambda_T^{(t)})$ is the SC's rate of return function, which is a function of the overall number of offloaded jobs at the SC, i.e., $\lambda_T^{(t)} = \sum_{i \in \mathbb{N}} \lambda_i^{(t)}$, and is formulated as:

$$p_{sc}^{(t)}(\lambda_T^{(t)}) = \begin{cases} \frac{\Lambda_{sc}^{(t)} - \lambda_T^{(t)}}{\Lambda_{sc}^{(t)}} \cdot p_{sc} & \text{, if } \lambda_T^{(t)} < \Lambda_{sc}^{(t)} \\ p_{sc}^{f} & \text{, otherwise} \end{cases} \tag{1}$$

where $\Lambda_{sc}^{(t)}$ is the number of jobs threshold that the SC can operationally process during D_t. If $\lambda_T^{(t)} \geq \Lambda_{sc}^{(t)}$, the SC's response time is greater than D_t and the SC "fails", thus, the SC's price is the minimum one ($p_{sc}^{f} < p_{sc}$). This phenomenon is known as the Tragedy of the Commons [9]. In the case of SC's failure, the user's successfully executed jobs during D_t are only the ones executed at the VMs.

Proposition 1: The SC's rate of return function $p_{sc}^{(t)}(\lambda_T^{(t)})$ is strictly decreasing with respect to $\lambda_T^{(t)}$, since as the $\lambda_T^{(t)}$ increases the less the SC can guarantee that the average response time is D_t, and the lower is the SC's price.

B. Actual Utility Function

The user's i actual utility $z_i^{(t)}$ expressing its satisfaction from executing $\lambda_i^{(t)}$ jobs at the SC and the rest ($J_i^{(t)} - \lambda_i^{(t)}$) at the VMs is formulated. This satisfaction is captured by the portion of jobs that are executed successfully during the timeslot t and the user's overall cost, as follows.

$$z_i^{(t)}(\lambda_i^{(t)}, \lambda_{-\mathbf{i}}^{(\mathbf{t})}) = \frac{E_i^{(t)}}{J_i^{(t)}} - \frac{p_{i,vmp}^{(t)} + p_{i,sc}^{(t)}}{B_i^{(t)}} \tag{2}$$

where $\lambda_{-\mathbf{i}}^{(\mathbf{t})}$ is the vector of the offloading decisions of all users except i, $E_i^{(t)}$ are the jobs that are executed successfully during D_t (see Section III-A), and $B_i^{(t)}$ is the user's i total budget.

III. The Prospect of Cloud

A. Risk-aware Behavior: The Tragedy of the Commons

The SC is a CPR since all the users can arbitrarily offload part of their jobs to it and share its resources. Towards maximizing the actual utility, each user aims at determining in an autonomous and distributed manner the number of jobs $\lambda_i^{(t)}$

offloaded to the SC, by accounting for the uncertainty of the SC's failure due to over-exploitation. Based on this uncertainty, we introduce the SC's probability of non-responsiveness.

Assumption 1: SC's probability of non-responsiveness $PnR^{(t)}(\lambda_T^{(t)})$ is strictly increasing, convex and twice continuously differentiable with respect to $\lambda_T^{(t)} \in [0, \Lambda_{sc}^{(t)})$, with $PnR^{(t)}(\lambda_T^{(t)}) = 1$, $\forall \lambda_T^{(t)} \geq \Lambda_{sc}^{(t)}$.

We consider a linear probability of non-responsiveness, i.e., $PnR^{(t)}(\lambda_T^{(t)}) = \frac{\lambda_T^{(t)}}{\Lambda_{sc}^{(t)}}$, $\forall \lambda_T^{(t)} < \Lambda_{sc}^{(t)}$. Other forms of $PnR^{(t)}$ that follow Assumption 1 can be considered, e.g., logarithmic, exponential, without damaging the applicability and validity of the following analysis.

B. Risk-aware Resource Allocation under Prospect Theory

Prospect Theory is adopted to address the users' subjectivity in decision-making [10]. Following this behavioral model, the users make actions under risk and uncertainty regarding the corresponding payoff of their actions. Each user's satisfaction by offloading a number of jobs to the SC and the VMs is evaluated with respect to a reference point (reference dependence property). Each user's reference point is the guaranteed utility $z_{i,0}^{(t)}$ that the user obtains by offloading all the jobs at the VMs (referred to as the safe resource), thus, $\lambda_i^{(t)} = 0$. Therefore, each user's i reference point is $z_{i,0}^{(t)} = 1 - \left\lceil \frac{J_i^{(t)}}{\lambda_{vm}^{(t)}(D_t)} \right\rceil \cdot \frac{p_{vm}}{B_i^{(t)}}$, where $\frac{E_i^{(t)}}{J_i^{(t)}} = 1$ (Eq. 2) since all the jobs are executed successfully during the time slot t.

Based on Prospect Theory, each user's $i, i \in \mathbb{N}$ prospect-theoretic utility is defined as follows [11]:

$$u_i^{(t)}(\lambda_i^{(t)}, \lambda_{-\mathbf{i}}^{(\mathbf{t})}) = \begin{cases} (z_i^{(t)} - z_{i,0}^{(t)})^{\alpha_i} & \text{, if } z_i^{(t)} \geq z_{i,0}^{(t)} \\ -k_i \cdot (z_{i,0}^{(t)} - z_i^{(t)})^{\beta_i} & \text{, if } z_{i,0}^{(t)} > z_i^{(t)} \end{cases} \quad (3)$$

The parameters $a_i, \beta_i \in (0, 1]$ express the user's i sensitivity to gains and losses of its actual utility $z_i^{(t)}$, respectively. Small values of a_i parameter reflect a gain-seeking and loss-aversion behavior. Small values of β_i capture a higher decrease of the prospect-theoretic utility $u_i^{(t)}$, when the user's actual utility $z_i^{(t)}$ is lower than its reference point $z_{i,0}^{(t)}$. In our study, without loss of generality, we consider a similar behavior in gains and losses, thus $a_i = \beta_i, \forall i \in \mathbb{N}$. The parameter $k_i \in [0, \infty)$ expresses how users weigh losses compared to gains. If $k_i > 1$ the user's prospect-theoretic utility $u_i^{(t)}$ has a greater slope of decrease in losses compared to the slope of decrease in gains. The exact opposite holds true if $k_i \leq 1$.

If the SC does not fail due to the overall offloaded number of jobs $\lambda_T^{(t)}$, then $z_i^{(t)} \geq z_{i,0}^{(t)}$, and by appropriate mathematical derivations based on the first branch of Eq. 3, we conclude that its prospect-theoretic utility is $u_i^{(t)}(\lambda_i^{(t)}, \lambda_{-\mathbf{i}}^{(\mathbf{t})}) = (\lambda_i^{(t)})^{a_i}(\frac{\gamma_i p_{vm}}{\lambda_{vm}^{(t)}(D_t) B_i^{(t)}} - \frac{D_t f_i^{-1} p_{sc}^{(t)}(\lambda_T^{(t)})}{B_i^{(t)}})^{a_i}$, where γ_i is the user's i regulator factor, such that $\frac{-\lambda_i^{(t)}}{\lambda_{vm}^{(t)}(D_t)} \cdot \gamma_i = \left\lceil \frac{-\lambda_i^{(t)}}{\lambda_{vm}^{(t)}(D_t)} \right\rceil$.

On the other hand, if the SC "fails", user's i experienced actual utility $z_i^{(t)}$ is lower than its reference point $z_{i,0}^{(t)}$, and the user's i prospect-theoretic utility is obtained as: $u_i^{(t)}(\lambda_i^{(t)}, \lambda_{-\mathbf{i}}^{(\mathbf{t})}) = -k_i(\lambda_i^{(t)})^{a_i}(\frac{1}{J_i^{(t)}} - \frac{\gamma_i p_{vm}}{\lambda_{vm}^{(t)}(D_t) B_i^{(t)}} + \frac{f_i^{-1} D_t}{B_i^{(t)}} p_{sc}^f)^{a_i}$, based on the second branch of Eq. 3, where the price of the SC is the minimum one, thus $p_{sc}^{(t)} = p_{sc}^f$. For notational convenience we define $\epsilon_i^{(t)} = (\frac{1}{J_i^{(t)}} - \frac{\gamma_i p_{vm}}{\lambda_{vm}^{(t)}(D_t) B_i^{(t)}} + \frac{f_i^{-1} D_t}{B_i^{(t)}} p_{sc}^f)^{a_i}$, and $h_i^{(t)}(\lambda_T^{(t)}) = (\frac{\gamma_i p_{vm}}{\lambda_{vm}^{(t)}(D_t) B_i^{(t)}} - \frac{D_t f_i^{-1} p_{sc}^{(t)}(\lambda_T^{(t)})}{B_i^{(t)}})^{a_i}$, where $h_i^{(t)}(\lambda_T^{(t)}) > 0$ if the SC does not fail. Thus, considering the probability of non-responsiveness $PnR^{(t)}$ of the SC, the user's prospect-theoretic utility can be written as:

$$u_i^{(t)}(\lambda_i^{(t)}, \lambda_{-\mathbf{i}}^{(\mathbf{t})}) = \begin{cases} (\lambda_i^{(t)})^{a_i} h_i^{(t)}(\lambda_T^{(t)}) & \text{with prob. } 1 - PnR^{(t)} \\ -k_i \epsilon_i^{(t)} (\lambda_i^{(t)})^{a_i} & \text{with prob. } PnR^{(t)} \end{cases} \quad (4)$$

Thus, user's i expected prospect-theoretic utility is given as:

$$\begin{aligned} \mathbb{E}(u_i^{(t)}) &= (\lambda_i^{(t)})^{a_i} h_i^{(t)}(1 - PnR^{(t)}) - (\lambda_i^{(t)})^{a_i} k_i \epsilon_i^{(t)} PnR^{(t)} \\ &= (\lambda_i^{(t)})^{a_i}[h_i^{(t)}(1 - PnR^{(t)}) - k_i \epsilon_i^{(t)} PnR^{(t)}] \\ &= (\lambda_i^{(t)})^{a_i} g_i(\lambda_T^{(t)}) \end{aligned} \quad (5)$$

where $g_i(\lambda_T^{(t)}) = [h_i^{(t)}(1 - PnR^{(t)}) - k_i \epsilon_i^{(t)} PnR^{(t)}]$ is the user's effective rate of return from the SC.

IV. Optimizing Resource Allocation: Problem Formulation

Each user's i goal is to maximize its perceived expected prospect-theoretic utility (Eq. 5) via determining its best recourse allocation strategy, i.e., the number of jobs $\lambda_i^{(t)}$ that are offloaded at the SC at timeslot t. This problem is formulated as a maximization problem of each user's i expected prospect-theoretic utility function (Eq. 5), as follows.

$$\max_{\lambda_i^{(t)} \in S_i^{(t)}} \mathbb{E}(u_i^{(t)}) = (\lambda_i^{(t)})^{a_i} g_i(\lambda_T^{(t)}) \quad (6)$$

where $S_i^{(t)}$ is the user's i strategy space as it is defined later.

The above maximization problem can be treated as a non-cooperative game $\mathbb{G} = \{\mathbb{N}, \{S_i^{(t)}\}, \{\mathbb{E}(u_i^{(t)})\}\}$ among the N users, where $S_i^{(t)} = [0, min(J_i^{(t)}, \Lambda_{sc}^{(t)})]$ is the strategy space of each user i, and $\mathbb{E}(u_i^{(t)})$ is its expected prospect-theoretic utility. Towards solving the non-cooperative game, the concept of Pure Nash Equilibrium (PNE) is adopted. Let $\lambda^{*,(\mathbf{t})} = [\lambda_1^{*,(t)}, \cdots, \lambda_N^{*,(t)}]$ denote the users' resource allocation strategies and $\lambda_{-\mathbf{i}}^{*,(\mathbf{t})}$ the vector of all the users' resource allocation strategies except user i at the PNE point.

Definition 1: The resource allocation vector $\lambda^{*,(\mathbf{t})} \in S^{(t)} = S_1^{(t)} \times \cdots \times S_N^{(t)}$, is a PNE of $\mathbb{G}$, if $\mathbb{E}(u_i^{(t)}(\lambda_i^{*,(t)}, \lambda_{-\mathbf{i}}^{*,(\mathbf{t})})) \geq \mathbb{E}(u_i^{(t)}(\lambda_i^{(t)}, \lambda_{-\mathbf{i}}^{*,(\mathbf{t})}))$, $\forall \lambda_i^{(t)} \in S_i^{(t)}, \forall i \in \mathbb{N}$.

V. Existence and Uniqueness of PNE

The best response strategy of user i is $B_i(\lambda_{-\mathbf{i}}^{(\mathbf{t})}) = \underset{\lambda_i^{(t)} \in S_i^{(t)}}{\arg\max} \mathbb{E}(u_i^{(t)}(\lambda_i^{(t)}, \lambda_{-\mathbf{i}}^{(\mathbf{t})})) : S_{-i}^{(t)} \rightrightarrows S_i^{(t)}$, $S_{-i}^{(t)} = \times_{j \in \mathbb{N} - \{i\}} S_j^{(t)}$.

***Theorem** 1:* For each user i, its best response strategy exists and it is single-valued, such that $\lambda_i^{*,(t)} = B_i(\lambda_{-\mathbf{i}}^{(\mathbf{t})})$.

We adopt the notation $\lambda_{-i,T}^{(t)} = \sum_{j \in \mathbb{N}, j \neq i} \lambda_j^{(t)}$ to depict the total number of offloaded jobs at the SC of all users except user i. The proof of Theorem 1 can be readily concluded based on Berge's Theorem [12] and the following Lemmas 1-3.

***Lemma** 1:* For each user i the following holds true: i) there exists a value $\overline{\lambda}_i^{(t)}$, such that $g_i(\overline{\lambda}_i^{(t)}) = 0$, ii) if $\lambda_{-i,T}^{(t)} \geq$

$\overline{\lambda}_i^{(t)}$ then $\lambda_i^{*,(t)} = 0$, and iii) if $\lambda_{-i,T}^{(t)} < \overline{\lambda}_i^{(t)}$ there exists an user-specific interval $A_i^{(t)} \subset [0, \overline{\lambda}_i^{(t)})$ such that all user's best responses are positive, and $\lambda_i^{*,(t)} + \lambda_{-i,T}^{(t)} \in A_i^{(t)}$.

The proof of Lemma 1 is omitted due to space limitations. Below the notation (t) is dropped for notational convenience.

***Lemma** 2:* The best response $\lambda_i^*, \forall i \in \mathbb{N}$ is single-valued $\forall \lambda_{-i,T} \in [0, \Lambda_{sc}]$.

Proof: Based on Lemma 1 we know that $\forall \lambda_i > 0$ such that $\lambda_i + \lambda_{-i,T} \in A_i$, we have $g_i(\lambda_T) > 0$ and $\frac{\partial g_i(\lambda_T)}{\partial \lambda_T} < 0$, where $\lambda_T = \lambda_i + \lambda_{-i,T}$. Also, since $g_i(\lambda_T)$ is concave in interval A_i (Lemma 1), the user's i expected prospect-theoretic utility is concave, i.e., $\frac{\partial^2 \mathbb{E}(u_i)}{\partial \lambda_i^2} = \lambda_i^{a_i} \frac{\partial^2 g_i(\lambda_T)}{\partial \lambda_T^2} + 2a_i \lambda_i^{a_i - 1} + a_i(a_i - 1)\lambda_i^{a_i - 2} g_i(\lambda_T) < 0$. As a result, since any best response λ_i^* satisfies $\lambda_i^* + \lambda_{-i,T} \in A_i$, λ_i^* is an argument of maximum of $\mathbb{E}(u_i)$, and therefore is unique. ■

***Lemma** 3:* The user's best response $\lambda_i^* : S_{-i} \rightrightarrows S_i$ is continuous for $\lambda_{-\mathbf{i}} \in S_{-i}$.

The proof of Lemma 3 is derived based on Berge's Theorem [12] and Lemma 2.

***Theorem** 2:* A Pure Nash Equilibrium $\lambda^* = [\lambda_1^*, \cdots, \lambda_N^*]$ of the non-cooperative game $\mathbb{G} = [\mathbb{N}, \{S_i\}, \{\mathbb{E}(u_i)\}]$ exists.

Proof: The strategy set S_i is a convex compact subset of the Euclidean space and so is the joint strategy space, $S = S_1 \times \cdots \times S_N \subset \mathbb{R}^N$. By defining a mapping $T : S \to S$ such that $T(\lambda_1, \cdots, \lambda_N) = (\lambda_1^*, \cdots, \lambda_N^*)$, from Lemma 2, T is single-valued and from Lemma 3 is continuous. Brouwer's fixed point theorem guarantees the existence of a strategy profile $s = \{\lambda_i^*\}_{i \in \mathbb{N}} \in S$ that is invariant under the best response mapping and therefore is a PNE of $\mathbb{G}$ [12]. ■

***Lemma** 4:* The function $d_i(\lambda_T) = \frac{-a_i g_i(\lambda_T)}{\frac{\partial g_i(\lambda_T)}{\partial \lambda_T}}$ is strictly decreasing with respect to λ_T, $\forall \lambda_T \in A_i$.

Proof: The first-order derivative of $d_i(\lambda_T)$ is $\frac{\partial d_i(\lambda_T)}{\partial \lambda_T} = -a_i \frac{(\frac{\partial g_i(\lambda_T)}{\partial \lambda_T})^2 - g_i(\lambda_T) \frac{\partial^2 g_i(\lambda_T)}{\partial \lambda_T^2}}{(\frac{\partial g_i(\lambda_T)}{\partial \lambda_T})^2}$. When $\lambda_T \in A_i$, based on Lemma 1 it holds true that $g_i(\lambda_T) > 0$ and $\frac{\partial^2 g_i(\lambda_T)}{\partial \lambda_T^2} \leq 0$, therefore it hollows directly that $\frac{\partial d_i(\lambda_T)}{\partial \lambda_T} < 0$, $\forall \lambda_T \in A_i$ ■

***Theorem** 3:* The Pure Nash Equilibrium of the non-cooperative game $\mathbb{G}$ is unique.

Proof: We use the notation λ_T^* to denote the total offloaded number of jobs at the SC at the PNE of game $\mathbb{G}$. The proof of Theorem 3 is based on the reduction to absurdity. Let $\lambda_{T(1)}^*, \lambda_{T(2)}^*$ be two distinct PNE points. Without loss of generality we assume that $\lambda_{T(2)}^* > \lambda_{T(1)}^*$. We define the set $Sup \triangleq \{i \in \mathbb{N} : \lambda_T^* < \overline{\lambda}_i\}$, thus it includes every user that offloads a non-zero number of jobs at the SC. Thus, $Sup_2 \subset Sup_1$. Also, we have $\sum_{j \in Sup_1} d_j(\lambda_{T(1)}^*) = \lambda_{T(1)}^*$, $\sum_{j \in Sup_2} d_j(\lambda_{T(2)}^*) = \lambda_{T(2)}^*$. So, $\sum_{j \in Sup_2} d_j(\lambda_{T(1)}^*) + \sum_{j \in Sup_1 \setminus Sup_2} d_j(\lambda_{T(1)}^*) = \lambda_{T(1)}^* \Rightarrow \sum_{j \in Sup_2} d_j(\lambda_{T(1)}^*) \leq \lambda_{T(1)}^* < \lambda_{T(2)}^* = \sum_{j \in Sup_2} d_j(\lambda_{T(2)}^*)$. However, $d_j(\lambda_T)$ is decreasing, so $d_j(\lambda_{T(1)}^*) > d_j(\lambda_{T(2)}^*), \forall j \in Sup_2$, which is contradiction. Thus, $\lambda_{T(1)}^* = \lambda_{T(2)}^*$. ■

VI. Algorithm-Convergence to PNE

Based on Lemma 4, each user's best response strategy λ_i^* is decreasing with respect to the total number of offloaded jobs $\lambda_{-i,T}$ of the rest users. Thus, $\mathbb{G}$ belongs to the *best-response potential games*, and therefore the sequential best response dynamics converge to the PNE [13]. Each user i first receives the total number of offloaded jobs of the rest users, i.e., $\lambda_{-i,T}$, in order to compute its best response λ_i^* and it determines if $\lambda_i^* = 0$, thus, whether $g_i(\lambda_{-i,T}) \leq 0$ and $\frac{\partial g_i(\lambda_T)}{\partial \lambda_T}|_{\lambda_T = \lambda_{-i,T}} < 0$ holds true (conditions stemming from Lemma 1). If the user i finds that $\lambda_{-i,T} < \overline{\lambda}_i$, then its λ_i^* exists and is single-valued (Theorem 1). Specifically, due to the existence of the unique root of $\frac{\partial \mathbb{E}(u_i)}{\partial \lambda_i} = 0$, and considering that $\frac{\partial \mathbb{E}(u_i)}{\partial \lambda_i}$ is a continuously differentiable and decreasing (i.e., $\frac{\partial^2 \mathbb{E}(u_i)}{\partial \lambda_i^2} < 0$, Lemma 2) with respect to λ_i, then the unique root r_i^* can be found via binary search into $[0, \Lambda_{sc}]$ with an approximation $\epsilon \to 0$, and finally user's i best response to be $\lambda_i^* = min(J_i, r_i^*)$. The complexity of the binary search is $\mathcal{O}(\log_2 \Lambda_{sc})$. In each iteration of the sequential best response dynamics, only one user i determines its best response strategy via executing arithmetical calculations (Algorithm 1). By denoting as Ite the number of iterations that are needed for convergence to the PNE, the complexity of the Algorithm 1 is $\mathcal{O}(N * Ite * \log_2 \Lambda_{sc})$. It is noted that the execution time of Algorithm 1 scales very well with respect to the number of users (see Section VII-B).

Algorithm 1 Distributed Algorithm for Convergence to PNE

```
Input/Initialization: $\mathbb{S}, D_t, f_i, p_{sc}, p_{sc}^f, \Lambda_{sc}, p_{vm}, \lambda_{vm}$
    $Ite = 0, \lambda_i \in [0, min(J_i, \Lambda_{sc})], \forall i \in \mathbb{N}$
Output: PNE profile $\lambda^* = [\lambda_1^*, \cdots, \lambda_N^*]$
while PNE not reached do
    $Ite = Ite + 1$
    for $i = 1$ to $N$ do
        User $i$ receives the $\lambda_{-i,T}$
        if ($g_i(\lambda_{-i,T}) \leq 0$ && $\frac{\partial g_i(\lambda_T)}{\partial \lambda_T}|_{\lambda_T = \lambda_{-i,T}} < 0$) then
            $\lambda_i^* = 0$
        else
            $r_i^* = BinarySearch([0, \Lambda_{sc}], \epsilon)$, $\epsilon \to 0$
            $\lambda_i^* = min(J_i, r_i^*)$
        end if
    end for
    Check convergence to PNE
end while
```

VII. Numerical Results

In this section, we provide detailed numerical results to illustrate the performance of the proposed approach in terms of the following aspects: basic operation of our framework (Section VII-A), scalability (Section VII-B), and framework's behavior under heterogeneous users in terms of loss aversion parameter k_i (Section VII-C). Finally, a comparative evaluation of our approach against alternative resource allocation techniques is provided (Section VII-D).

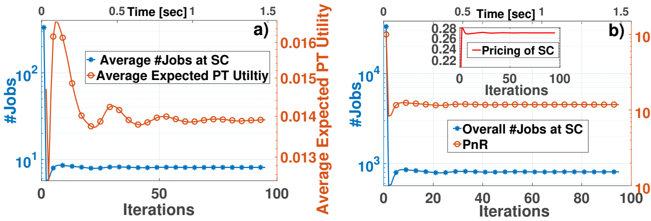

Fig. 1: Pure operation of the proposed framework

In our study, the duration of each timeslot is $D_t = 1sec$ and the price of reserving a VM for D_t is $p_{vm} = 10$, while the SC's price per unit of time is $p_{sc} = 0.3$ and $p_{sc}^f = 0.2$ [6]. The maximum number of jobs that can be executed by an VM instance during D_t is $\lambda_{vm}^{(t)} = 10$. A directed social network is created with random topology and 100 users, where each user has $J_i^{(t)} \in [400, 1000]$ number of jobs. Each user is associated with its social factor f_i. For the SC, we have $\Lambda_{sc}^{(t)} = 10\% \times \sum_{i \in \mathbb{N}} J_i^{(t)}$. Unless otherwise stated, we assume homogeneous users with parameters $a_i = 0.2$, $k_i = 5$.

A. Pure Operation of the Proposed Framework

Fig. 1a illustrates the average number of offloaded jobs to the SC (left vertical axis) and the average expected prospect theoretic utility (right vertical axis expressed in logarithmic scale), as a function of the iterations (low horizontal axis) and the execution time (upper horizontal axis). Fig. 1b presents the overall number of jobs at the SC (left vertical axis) and the SC's probability of non-responsiveness (right vertical axis), while in the contained sub-figure the corresponding SC's pricing is depicted. From the results in Fig. 1a and Fig 1b, we confirm that starting from a random initial strategy, as the time evolves the algorithm converges to a stable point (i.e. unique PNE point), where each user has determined its best response strategy. Throughout this evolving process and till we reach the PNE point, the users either offload a larger number of jobs at the SC in order to increase their expected prospect theoretic utility, or they follow an opposite resource allocation strategy, i.e., a lower number of jobs at the SC, when the SC's probability of non-responsiveness increases.

B. Scalability Evaluation

Fig. 2a illustrates each user's average number of offloaded jobs at the SC (and the sub-figure presents the total number of jobs at the SC) and the average expected prospect theoretic utility with respect to the number of users. As the number of users increases, the SC becomes more congested (increased total number of offloaded jobs at the SC - contained sub-figure in Fig. 2a), while each user offloads a smaller number of jobs, since its incentive is reduced due to the higher SC's probability of non-responsiveness (Fig. 2b), while experiencing a lower expected prospect theoretic utility (Fig. 2a). Fig. 2b shows the actual required time for our algorithm to converge to the PNE. As observed from the results our algorithm's execution time presents sublinear behavior with respect to the number of users and is well aligned with our scalability analysis (Section VI).

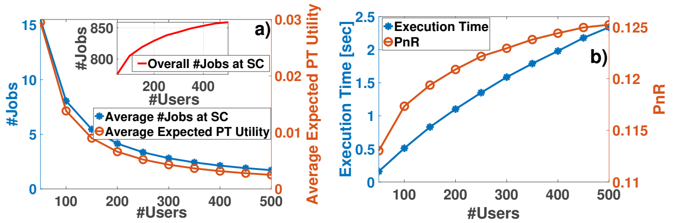

Fig. 2: Scalability Evaluation

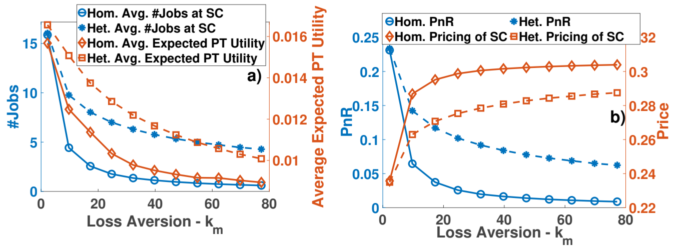

Fig. 3: Heterogeneous users - loss aversion impact study

C. Heterogeneous Users - Loss Aversion

In this section, the impact of users' heterogeneous loss aversion prospect theoretic behavior on the achievable performance is studied. In particular, in Fig. 3a and Fig. 3b we compare a scenario of heterogeneous users, where each user is associated with a different personalized loss aversion index k_i, against a homogeneous scenario where all users assume the same exactly loss aversion parameter k_m. For fairness in the comparison we consider that k_m is equal to the average loss aversion parameter value of all the members of the heterogeneous group. It is noted that the more loss averse is the users' behavior (higher loss aversion parameter), the less number of jobs they offload at the SC (Eq. 5). The opposite holds true for the risk seeking users, which may lead the SC to "failure", thus the users' expected prospect theoretic utility will decrease. Indeed, based on Fig. 3a and Fig. 3b, the heterogeneous users led the system to higher congestion levels, as there is an increase in the average number of offloaded jobs at the SC and a decrease in the SC's pricing $p_{sc}(\lambda_T)$ (Eq.1). However, in our case, Fig. 3a illustrates that the increase of the average number of offloaded jobs at the SC led the heterogeneous users to achieve a higher average expected prospect theoretic utility compared to the homogeneous case.

D. Comparative Analysis

In this section, we present a comparative study of our proposed theoretic framework (that assumes prospect theoretic (pt) users) with five other alternatives, assuming user behaviors as follows: (a) non prospect theoretic (npt) users, but expected actual utility $\mathbb{E}(z_i(\lambda_i, \lambda_{-\mathbf{i}}))$ maximizers instead, taking into account the SC's probability of non-responsiveness, (b) actual utility maximizers (ut) users, where each user maximizes its actual utility (Eq. 2) without considering the SC's probability of non-responsiveness, (c) social (soc) users, where each user based on its social factor f_i offloads $f_i * J_i$ number of jobs at the SC, (d) (sp) users where each user i offloads all of its jobs J_i at the SC, and (e) (vm) users where each user offloads all of its jobs J_i at the VMs.

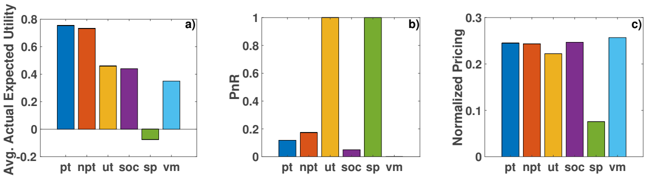

Fig. 4: Comparative Evaluation

The comparative evaluation is performed in terms of: (i) the expected actual utility, (Fig. 4a), (ii) SC's probability of non-responsiveness (Fig. 4b), and (iii) users' average normalized pricing, i.e., $\frac{p^{(t)}_{i,vmp}+p^{(t)}_{i,sc}}{B^{(t)}_{i}}$ (Fig. 4c). In particular, Fig. 4a shows that both (pt) and the (npt) frameworks achieve a higher average actual expected utility (with (pt) slightly outperforming) compared to the rest of the approaches, due to the realistic consideration of the system's uncertainty (through the SC's probability of non-responsiveness). Both the (ut) and the (sp) frameworks, by ignoring the SC's probability of non-responsiveness, lead the SC to "failure", i.e., $PnR(\lambda_T) = 1$ (Fig. 4b), and therefore these two approaches conclude to a lower user average actual expected utility compared to the (pt) and the (npt) (Fig. 4a). Please note that although the (ut) approach leads the SC to "failure", the users still offload part of their jobs at the VMs, and therefore achieve a positive average actual expected utility, while on the other hand, under the (sp) alternative, users achieve a negative average expected actual utility, since none of their jobs is executed successfully.

On the other hand, the users under the (soc) approach, by offloading a number of jobs simply based on their social factor f_i, they do not lead the SC to "failure", however conclude to a lower average actual expected utility compared to the (ut) approach, since they do not perform any optimization. Under (vm) alternative the SC option is not exploited and each user offloads all its jobs at the VMs. Thus its actual expected utility is its reference point, which is lower compared to the ones achieved by the (ut) and the (soc). Finally, it is stressed that the (pt) framework operates better than the (npt), achieving lower SC's probability of non-responsiveness (Fig. 4b) and higher average expected actual utility (Fig. 4a), due to the fact that in the (npt) case, each user does not follow a risk-aware behavior and determines its best response strategy λ_i^* by only considering its guaranteed actual utility, and as a result the SC's utilization is better by the (pt) users.

Fig. 4c presents the average users' normalized pricing for all the scenarios. In the (sp) case, the users by offloading all of their jobs at the SC perceive the lowest pricing p^f_{sc} per unit of time, while for the opposite reason highest price is experienced in the (vm) case. The (soc) users perceive the second highest average normalized price, since they offload a small portion of their jobs. Comparing the (npt) alternative with the (pt), we notice that they present very similar performance, with (npt) concluding to slightly lower average normalized price compared to the (pt), since by offloading a higher portion of their jobs at the SC, they perceive a lower price from the SC. The same holds true for the (ut) users, who offload a larger number of jobs at the SC compared to the (pt) and (npt) users (Fig. 4b), and as a result they perceive the second lowest average normalized price (Fig. 4c).

VIII. Concluding Remarks

In this paper, a novel risk-based distributed approach, towards determining each user's computing tasks optimal allocation strategy, in a social cloud computing environment offering both options of VM and SC computing, is designed. Based on the properties of Prospect Theory and the theory of The Tragedy of the Commons, we take into account the loss averse and gain seeking behavior of the users, as well as the uncertainty introduced due to the shared nature of the SC model. In order to address the decision-making problem at hand, a non-cooperative game is formulated among the users, where the goal of each user is to maximize its perceived expected prospect theoretic utility. The existence and uniqueness of the non-cooperative game's PNE is proven, and a distributed low-complexity algorithm that converges to the PNE is devised. Detailed numerical results were presented highlighting the performance benefits of our proposed approach.

Our current and future research work focuses on the extension of the above approach in an environment where multiple Cloud Providers co-exist acting as common pool of resources.

Acknowledgment

The research of Dr. Eirini Eleni Tsiropoulou was conducted as part of the NSF CRII-1849739.

References

[1] T. Qiu, B. Chen, A. K. Sangaiah, J. Ma, and R. Huang, "A survey of mobile social networks: Applications, social characteristics, and challenges," *IEEE Systems J.*, vol. 12, no. 4, pp. 3932–3947, Dec 2018.

[2] K. Chard, K. Bubendorfer, S. Caton, and O. F. Rana, "Social cloud computing: A vision for socially motivated resource sharing," *IEEE Transactions on Services Computing*, vol. 5, no. 4, pp. 551–563, 2012.

[3] L. Jiao, J. Lit, W. Du, and X. Fu, "Multi-objective data placement for multi-cloud socially aware services," in *IEEE INFOCOM*, 2014, pp. 1–9.

[4] A. Pérez, G. Moltó, M. Caballer, and A. Calatrava, "Serverless computing for container-based architectures," *Future Generation Computer Systems*, vol. 83, pp. 50–59, 2018.

[5] W. Lloyd, S. Ramesh, S. Chinthalapati, L. Ly, and S. Pallickara, "Serverless computing: An investigation of factors influencing microservice performance," in *Int. Conf. on Cloud Eng.* IEEE, 2018, pp. 159–169.

[6] T. Elgamal, "Costless: Optimizing cost of serverless computing through function fusion and placement," in *IEEE/ACM Symposium on Edge Computing (SEC)*. IEEE, 2018, pp. 300–312.

[7] AWS Lambda. [Online]. Available: https://aws.amazon.com/lambda/

[8] Google Cloud Functions. [Online]. Available: https://cloud.google.com/

[9] P. Vamvakas, E. E. Tsiropoulou, and S. Papavassiliou, "On the prospect of uav-assisted communications paradigm in public safety networks," in *IEEE INFOCOM WKSHPS: WCNEE*, 2019. (*preprint available at: shorturl.at/oBLP1*).

[10] ——, "Dynamic spectrum management in 5g wireless networks: A real-life modeling approach," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE, 2019, pp. 2134–2142.

[11] ——, "On controlling spectrum fragility via resource pricing in 5g wireless networks," *IEEE Networking Letters*, 2019.

[12] E. A. Ok, *Real analysis with economic applications.* Princeton University Press, 2007, vol. 10.

[13] P. Dubey, O. Haimanko, and A. Zapechelnyuk, "Strategic complements and substitutes, and potential games," *Games and Economic Behavior*, vol. 54, no. 1, pp. 77–94, 2006.