# Resource Allocation for Cloud-based Social TV Applications Using Particle Swarm Optimization

Gosala Kulupana, Dumidu S. Talagala, Hemantha Kodikara Arachchi and Anil Fernando
I-Lab, Centre for Vision, Speech and Signal Processing, University of Surrey, United Kingdom
Email: {g.kulupana, d.talagala, h.kodikaraarachchi, w.fernando}@surrey.ac.uk

***Abstract*—Social interaction of groups of users, amongst themselves and with the media content itself, is increasingly becoming popular due to the advancements in the Internet access technologies. However, multimedia resource provisioning for dispersed user groups poses a challenge and demands innovative technologies. This paper proposes a novel approach based on Particle Swarm Optimization (PSO) to optimally allocate computational and networking resources to a group of interactive users, such that the group Quality-of-Service (QoS) is maximized. We evaluate the performance of the proposed improved PSO method with respect to the state-of-the-art greedy resource allocation mechanisms and related PSO approaches. The ability to find a feasible solution (i.e., the serving probability) and the accuracy of such solutions are compared for different network topologies. The proposed method demonstrates reduced computational complexity, an up to 40% increase in the serving probability compared to the greedy methods, and up to 60 times faster convergence compared to the basic PSO approach. Overall, the comparable QoS level to the optimal solution suggests that the proposed solution efficiently allocates the resources available in the network.**

***Index Terms*—Cloud computing, data center location, interactive television, particle swarm optimization, Quality-of-Service, resource optimization, social multimedia applications, video distribution.**

## I. Introduction

DURING the past decade social networking has spread to all aspects of human activities, with services by Facebook, Twitter, Google+ and Qzone becoming dominant players in the field [1]. Building on these trends, the next generation of social networking applications such as Interactive TV (ITV) and multi-player online games are expected to penetrate human life more thoroughly in the near future [2]. For example, ITV is envisaged to transform the passive uni-directional TV broadcasting concept into an actively engaged bi-directional TV concept, eventually becoming the evolution of the existing social networks. In ITV applications such as [3], the multimedia content is distributed as a personalized multicast stream among a group of users known as a 'social group', where the members of the group can interact with each other via the multicast multimedia content stream itself. However, in order to realize a superior Quality-of-Experience, balancing the needs of the social components of the media networks (i.e., TV viewers) with the technological capabilities of networks (i.e., cloud and networking infrastructure) becomes critical.

Multimedia service providers typically opt to leverage the cloud-based processing resources due to their inherent benefits in terms of high scalability, easy deployment and easy maintenance [4]. Tobias *et al.* [4] described the major challenges encountered when a multimedia service is moved to the cloud, of which, the artifacts introduced due to the increasing distance between the service and the customer is a major concern for delay critical applications. In addition to these, the latency becomes the most stringent requirement for ITV applications, since a minimal latency [5] between the application server and the consumer is essential to maintain the perception of near real-time interaction. In order to fulfill these stringent requirements, multiple datacenters at different locations (facilitated by the multi-cloud and inter-cloud communication concepts [6]) resembling the fog computing paradigm can be selected [7]. However, in order to maintain the consistency and maximize each individual's Quality-of-Experience, media processing should occur at a single location for each group of users; thereby maximizing the QoS of groups of users. Therefore, determining the optimal datacenter allocation (to individual 'social groups') that maximizes the Quality-of-Experience of the end-users becomes an essential but difficult problem to solve.

Although a number of resource allocation schemes for media applications have been proposed in the literature, several drawbacks limit their applicability to the problem described above. For example, the significance of underlying network conditions such as delay, jitter and packet losses on the QoS has not been considered in the user QoS maximization problem in [8]. In [9] and [10], a sequential data center allocation mechanism (for media tasks) that considers network conditions have been proposed based on a heuristic method and a machine learning method, respectively. In both cases, although the queuing approach may well suit tasks of a bursty nature, it does not provide a globally optimum solution for a long duration continuous media task such as personalized interactive group video distribution. More sophisticated simultaneous resource allocation methods for multiple media tasks, while maintaining QoS requirements, have also been

This work was supported by the EU FP7 project ACTION-TV (http://www.action-tv.net), funded under the European Commission's 7th Framework Program (Grant Number: 611761).

proposed in [11],[12], [13] and [14]. In [11], the objective function was modelled as a linear combination of latency, carbon emissions and energy supply costs, whereas in [12] and [13], the objective was to minimize the monetary cost (subjected to QoS and delay limitations). However, these simultaneous resource allocation mechanisms for multiple media tasks [11],[12],[13] have not considered the group processing nature of the potential applications. For example, the location yielding the best QoS for a particular user may violate the latency requirement of another who resides at a distant access node. Therefore, when deciding the optimal datacenter/cloud location for a particular group of users, the QoS requirement of all the users in that group should be considered [14].

This paper proposes an improved Particle Swarm Optimization (PSO) solution to the abovementioned resource optimization problem. The PSO approach is motivated by the reduction in complexity and convergence time that can be achieved in comparison with linear programming approaches, especially in the case of large numbers of users and user groups. Here, near real-time interactivity is guaranteed by imposing a maximum allowed latency between the server and the end-users, where QoS is modelled over a resource constrained network. The specific improvements in the proposed PSO approach over the regular binary PSO methods are; (i) the reduction of the particle position vector length, and (ii) the novel bit allocation algorithm for the position vector.

The remainder of this paper is organized as follows. In Section II we formulate the problem with respect to the ITV application using matrix notations. Section III consists of different solution approaches considered, namely the proposed improved PSO method together with two other PSO methods, the mixed integer linear programming (MILP) method, as well as the greedy best fit (BF) and first fit (FF) approaches. Section IV describes the simulation configuration, and is followed by the results and discussion in Section V and concluding remarks in Section VI.

## II. Problem Formulation

### A. System Description

Fig. 1 illustrates an example of a simplified topological view of the node connectivity and the user composition of the interactive media application considered in this work. Here, five users belonging to two user groups are connected to three

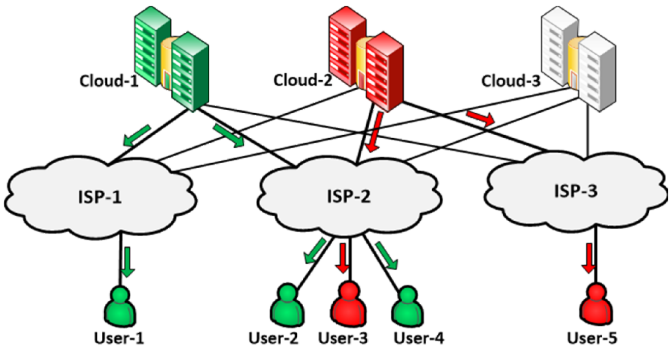


Fig. 1. Logical network architecture diagram of an example interactive TV distribution system.

Internet Service Provider (ISP)s (i.e., access nodes). The three cloud computing resources (i.e., processing nodes) and virtual links between clouds and ISPs form the remainder of the network. "User-1", "User-2" and "User-4" form "Social Group-1", the media content of which is processed by "Cloud-1". "Social Group-2" is served by "Cloud-2" and constitutes of "User-3" and "User-5". Each group consists of both interactive users who actively engage in the content and non-interactive users who passively watch the TV show. The interactive users engage in the multimedia stream by means of uploading their virtual presence to the network. For example, when a prospective ITV user wishes to participate in a quiz show, the virtual presence of the user is inserted into the media stream and his/her presence can be seen by the rest of the group members as part of their personalized video stream. It should be noted that more than one active user can be part of the same 'social group'. In this scenario, the personalized television show will include all the active users, together, in the same scene. We assume that the following applies to the network:

(A.1). Capacities of the processing nodes and network remain fixed during the resource allocation process.
(A.2). A single processing node serves each group of users.
(A.3). Link bandwidths between each user and his/her ISP is sufficient to carry the media multicast transmission.

Let $G(V, E)$ represent the connected network where $V=\{S, A\}$ is the set of nodes, that include $S=\{s_1, s_2, \ldots, s_S\}$ the set of processing nodes and $A=\{a_1, a_2, \ldots, a_A\}$ the set of access nodes available in the network. $E=\{E_v, E_a\}$ is the set of edges connecting different nodes, where $E_v$ represents the set of virtual links between clouds and ISPs and $E_a$ represents the set of access links to the end users. Let $U=\{u_1, u_2, \ldots, u_U\}$ be the set of ITV viewers uniquely belonging to the set of user groups $N=\{n_1, n_2, \ldots, n_N\}$. We define $\boldsymbol{d} = [\boldsymbol{d}_1^T, \ldots \boldsymbol{d}_n^T, \ldots \boldsymbol{d}_N^T]^T$ as the vector representation of the processing location for each social group, where $\boldsymbol{d}_n$ is a binary $S\times 1$ vector with a single non-zero element corresponding to its processing cloud index (e.g., $\boldsymbol{d}_n = [0\ 1\ 0]^T$ when the $n^{\text{th}}$ social group is processed at the "Cloud-2" in Fig. 1). The set of constraints imposed on the ITV system can therefore be expressed as follows:

$$\text{(C.1)} \quad B \cdot \boldsymbol{B}_{(A\times N)}[\boldsymbol{d}_1, \boldsymbol{d}_2, \ldots, \boldsymbol{d}_N]^T_{(S\times N)} \preceq \boldsymbol{B}_{0(A\times S)}$$

$$\text{(C.2)} \quad P \cdot \boldsymbol{1}_{(1\times N)}[\boldsymbol{d}_1, \boldsymbol{d}_2, \ldots, \boldsymbol{d}_N]^T_{(S\times N)} \preceq \boldsymbol{p}_{0(1\times S)}$$

$$\text{(C.3)} \quad \max\left\{ \boldsymbol{D}_{(U\times NS)} \boldsymbol{d}_{(NS\times 1)} \right\} \le \Delta_D$$

$$\text{(C.4)} \quad \boldsymbol{I}_{(N\times NS)} \boldsymbol{d}_{(NS\times 1)} = \boldsymbol{1}_{(N\times 1)}$$

The constraint (C.1), above, ensures that the total bandwidth demand of each group does not exceed the available bandwidth of each virtual link, where $\boldsymbol{B}[a,n] \in \{0,1\}$ indicates the existence of a "Social Group-$n$" user in the $a^{\text{th}}$ access node, $B$ indicates the multicast bandwidth per group and $\boldsymbol{B}_0$ indicates the available bandwidths in the virtual link set $E_v$. Similarly, $P$ indicates the processing resource requirement per 'social group', and $\boldsymbol{p}_0$ represents the available processing resources in each cloud. The constraint (C.2) specifies that

each processing node has sufficient processing capacity to process all the user groups assigned to it. $\boldsymbol{D}$ in the Constraint (C.3) indicates the delay between each user and the potential processing clouds and it sets the maximum latency tolerable for near real-time interactive behavior. In other words the maximum latency between the user and the respective social group's processing cloud is kept below a threshold so that action-to-reaction delay is unnoticeable. The constraint in (C.4) ensures that only one cloud processes a given social group, satisfying (A.2).

### *B. Group Quality-of-Service Maximization*

We model each user's QoS as the sum of end-to-end link QoS parameters from the processing node to the user by adopting a similar approach to Hyun *et al.* [15]. Extending this to maintain an acceptable Quality-of-Experience throughout the interactions (by the imposition of a delay bound), the link QoS cost metric for the ITV application can be modelled for a fixed rate transmission as

$$c_{n,s}^{u} = \alpha_1 \times (L^u) + \alpha_2 \times (J^u) + \alpha_3 \times (D^u), \tag{1}$$

where $c_{n,s}^{u}$ refers to the QoS cost of the user $u$ from the processing cloud $s$ to the user, $L^u$ refers to the average packet loss along the path from the cloud to the user, $J^u$ refers to the average jitter along the path and $D^u$ refers to the average delay along the path.$\{\alpha_1, \alpha_2, \alpha_3\}$ are appropriate constants that adequately parameterize the QoS cost metric for IPTV [15].

The Social Group-$n$'s cumulative QoS cost for some processing node $s$, can therefore be expressed as

$$c_{n,s}^{G} = \frac{1}{\sum_{u \in U^n} u} \left( \sum_{u \in U^n} c_{n,s}^{u} \right), \tag{2}$$

where $U^n$ refers to the set of users who belong to "Social Group-$n$". Consequently, the cumulative QoS cost of the overall network can be expressed in matrix form as

$$\boldsymbol{c}_{(N\times 1)}^{G} = \boldsymbol{C}_{(N\times NS)} \boldsymbol{d}_{(NS\times 1)} . \tag{3}$$

$\mathbf{C}$ therefore becomes a block diagonal matrix of dimension $(N \times NS)$ whose $n^{th}$ row is $[\,\boldsymbol{0}_{(1\times(n-1)S)}, [\,c_{n,1}^{G}, c_{n,2}^{G}, \ldots, c_{n,S}^{G}\,]_{(1\times S)}, \boldsymbol{0}_{(1\times(N-n)S)}\,]$. Thus, maximizing the 'Group Quality-of-Service' $\boldsymbol{c}^{\mathbf{G}}$, implies a minimization of the $l_1$-norm of the cumulative QoS cost of the overall network; i.e.,

$$\text{minimize} \quad \left\| \mathbf{C} \cdot \boldsymbol{d} \right\|_1, \tag{4}$$

## III. SOLUTION METHODOLOGY

In this section, we describe our proposed improved PSO method to solve the optimization problem outlined in the previous section. Two existing PSO methods and two greedy resource allocation methods (BF, FF) are briefly described afterwards for the sake of completeness.

### *A. Proposed Improved PSO Method*

PSO is a population based optimization technique, which consists of a set of collaborating particles (i.e., solution vectors) that oscillate within a solution space. Originally proposed by Kennedy and Eberhart, the algorithm was applied to a continuous objective function [16], but was later extended to a discrete binary version [17] in which the solution vector, and hence the particle position, is expressed in binary. Initially, each particle is assigned with a random location, which represents a feasible solution to the problem. During each iteration, every particle moves further towards the optimal solution. In order to do so, each particle compares the position of the previous iteration with its historical best solution (*pbest*) and the best solution advertised by the other member particles (*gbest*). Each particle gradually starts moving partially towards *pbest* and *gbest*.

Let $\widehat{\boldsymbol{d}}_{\boldsymbol{i}}$ and $\widehat{\boldsymbol{v}}_{\boldsymbol{i}}$ denote the position vector and the velocity vector of the $i^{th}$ particle. It should be noted that $\widehat{\boldsymbol{d}}_{\boldsymbol{i}}$ essentially represents a solution for the processing cloud vector $\boldsymbol{d}$ described in the previous section. The vector $\widehat{\boldsymbol{d}}_{\boldsymbol{i}}$ can therefore be expressed as $\widehat{\boldsymbol{d}}_{\boldsymbol{i}}=[x_{i,1}, x_{i,2}, \ldots, x_{i,m}, \ldots, x_{i,(N\times S)}]^T$, where $x_{i,m} \in \{0,1\}$ represents the particle $i$'s position in the $m^{th}$ dimension (i.e., $m \in \{1, 2, \ldots, N\times S)\}$ ). A similar notation applies to $\widehat{\boldsymbol{v}}_{\boldsymbol{i}}$ (i.e. $\widehat{\boldsymbol{v}}_{\boldsymbol{i}}=[v_{i,1}, v_{i,2}, \ldots, v_{i,m}, \ldots, v_{i,(N\times S)}]^T$ ). Consequently, the velocity of the $i^{th}$ particle in $m^{th}$ dimension becomes [17]

$$v_{i,m} = v_{i,m} + \varphi(p_{i,m} - x_{i,m}) + \varphi(p_{g,m} - x_{i,m}), \tag{5}$$

where $\varphi$ is a constant and $p_{i,m}$, $p_{g,m}$ are *pbest* and *gbest* in the $m^{th}$ dimension, respectively. The velocity, $v_{i,m}$ is limited to the range $-v_{max}$ and $+v_{max}$, such that

$$\begin{aligned} &if(v_{i,m}) < -v_{max} \; then \; v_{i,m} = -v_{max} \\ &if(v_{i,m}) > +v_{max} \; then \; v_{i,m} = +v_{max}. \end{aligned} \tag{6}$$

A smaller $v_{max}$ causes a larger span of the search space and a higher mutation rate. The new position of $i^{th}$ particle $x_{i,m}$ is derived by comparing a randomly generated number (*rand*( )) in the range [0 1] with the sigmoidal function, i.e.,

$$\begin{aligned} if\ (rand() < Sig(v_{i,m}))\ then\ x_{i,m} = 1; \\ else\ x_{i,m} = 0; \end{aligned} \tag{7}$$

where $Sig(v_{i,m})$ is the sigmoidal function. In order to expand the span of the search space in our solution we selected $\varphi$=1 and $v_{max}$=4.

The original PSO formulation is applicable to problems with no associated constraints. For problems that include constraints, Coath *et al*. [18] proposed two extensions for PSO, namely the Feasible Solutions Method (FSM) and Penalty Function Method (PFM). In FSM, a set of feasible solutions is filtered from the search space prior to the objective function evaluation. In contrast, PFM deals with constraint violations by applying the appropriately defined penalty values embedded within the objective function itself. Thus, PFM often exhibits faster convergence times, whereas FSM results in better accuracy. However, the drawbacks of the FSM method become apparent when the feasible solution space is very small within the overall solution search space, which forces the particle initialization process to become impractically time consuming [18].

Several major differences can be observed in our optimization problem in comparison to the typical binary PSO problem. First, there exist only a few feasible solutions within the solution space. Moreover due to the scarcity of feasible solutions, it makes impossible to filter solution space

intuitively and apply the FSM method. We therefore incorporate a hybrid of PFM and FSM approach in order to obtain more accurate results with reduced convergence times. Thus, of the four constraints described in (C.1) to (C.4), only (C.2) to (C.4) are evaluated prior to the objective function assessment. (C.1) is embedded in the objective function itself, in terms of a penalty. This enables the following key developments in the proposed improved PSO algorithm;

*1) Dimension reduction of the particle position vector*

The length reduction of the position vector $\widehat{\boldsymbol{d}}_{\boldsymbol{i}}$ is obtained through the evaluation of (C.3). The processing location vector for "Social Group-*n*" is obtained from the original vector $\boldsymbol{d}_n$ by initially filtering out processing clouds that violate the delay constraint in (C.3). Therefore, the size of the modified processing location vector for "Social Group-*n*", $\boldsymbol{d}_n^f$, becomes $S_n^f \times 1$, where $S_n^f \leq S$. Consequently, all the position vectors and velocity vectors are also appropriately filtered and the discarded bits shall no longer participate in iterative optimization process. (i.e. $\widehat{\boldsymbol{d}}_{\boldsymbol{i}}$ becomes $\widehat{\boldsymbol{d}}_{\boldsymbol{i}}^{\boldsymbol{f}}$ and $\widehat{\boldsymbol{v}}_{\boldsymbol{i}}$ becomes $\widehat{\boldsymbol{v}}_{\boldsymbol{i}}^{\boldsymbol{f}}$). This dimension reduction of the position vector immediately leads to a smaller search space, and ultimately faster convergence.

*2) Constrained bit allocation*

After evaluating the constraint (C.3) as proposed above, the remaining constraints (C.2) and (C.4) are evaluated in conjunction with a novel bit allocation algorithm, Algorithm 1. Hence, the inputs to the algorithm are the filtered particle position vectors from the dimension reduction process. It should be noted that Algorithm 1 is executed in place of (7) in the binary PSO method.

**Algorithm 1.** Proposed bit allocation procedure.

**procedure** PARTICLE POSITIONING ($\widehat{\boldsymbol{v}}_{\boldsymbol{i}}^{\boldsymbol{f}}$, $\mathrm{p}_0$, $P$ )

1. **for** each $v_{i,m} \in \widehat{\boldsymbol{v}}_{\boldsymbol{i}}^{\boldsymbol{f}}$
2. calculate $k_{i,m} = Sig(v_{i,m}) - rand()$ ***Step 1***
3. **end for**
4. $k'_{i,m}$←sort $k_{i,m}$ in descending order ***Step 2***
5. Initialize all $x_{i,m}$ to zeros.
6. **for** each $k'_{i,m}$
7. Extract social group *n* and processing cloud *s* from dimension *m* ***Step 3***
8. **if** (no cloud is assigned to group *n* **and** $\boldsymbol{p_0}[\mathrm{s}] > P$)
9. Assign $x_{i,m} \leftarrow 1$; % cloud *s* is assigned to group *n*
10. Assign $\boldsymbol{p_0}[\mathrm{s}] \leftarrow \boldsymbol{p_0}[\mathrm{s}] - P$; ***Step 4***
11. **end if**
12. **end for**
13. **return** vector $\boldsymbol{d'}_{\boldsymbol{i}}$ % $\boldsymbol{d'}_{\boldsymbol{i}}$ is constructed using $x_{i,m}$

**end procedure**

In Step 1, the difference between the sigmoidal function and the randomly generated number is calculated and stored. Then in Step 2 the stored values are sorted in descending order which essentially agrees with the original PSO concept since the set bits suggested by PSO always comes to the top of the sorted list. In Step 3 first the social group and processing cloud represented by dimension *m* is deduced. Finally, Step 4 eliminates the most probable invalid processing clouds by ensuring that constraints (C.2) and (C.4) are evaluated.

After evaluating (C.2) to (C.4), for the resultant vectors containing a single non-zero processing cloud for each 'social group', the derived vector $\boldsymbol{d'}_{\boldsymbol{i}}$ is applied to the group QoS cost function in (8). Here, the evaluation of the constraint (C.1) is embedded into the objective function, resulting in a composite objective function,

$$\text{minimize} \left\| \mathbf{C} \cdot \boldsymbol{d}'_{\mathbf{i}} \right\|_1 + \eta(\mathrm{g}_1) * \mathrm{PF}(\boldsymbol{d}'_{\mathbf{i}}) \quad (8)$$

where $\mathrm{g}_1 = \left\| \mathrm{B} \cdot \boldsymbol{B}_{(\mathrm{A}\times\mathrm{N})} [\boldsymbol{d}'_{\mathrm{i},1}, \boldsymbol{d}'_{\mathrm{i},2}, \ldots, \boldsymbol{d}'_{\mathrm{i},\mathrm{N}}]^{\mathrm{T}}_{(\mathrm{S}\times\mathrm{N})} - \boldsymbol{B}_{0(\mathrm{A}\times\mathrm{S})} \right\|_1$ is the (C.1) violation function, $\eta$ is the non-stationary cost function and 'PF($\boldsymbol{d'}_{\boldsymbol{i}}$)' is the penalty function derived from [19]. The sub-vector $\boldsymbol{d}'_{\mathrm{i,n}}$ which is deduced from $\boldsymbol{d'}_{\boldsymbol{i}}$ represents the processing location vector for "Social Group-*n*" . Based on this output, *pbest* and *gbest* values are updated. This process is performed iteratively until the particles converge to a feasible solution.

*B. Combinations of PFM and FSM Methods*

In order to visualize the time gain from each of the developments proposed in the previous subsection, we introduce two additional PSO implementations; namely 'Pure PFM' and 'Hybrid PFM & FSM' methods. In Pure PFM all the constraints except (C.4) are evaluated in the objective function. The position vector update procedure follows (7), and accordingly the derived first fit cloud is evaluated in the objective function. The modified objective function contains imposed penalties for every constraint violation. In Hybrid PFM & FSM, the constraints (C.2) to (C.4) are evaluated prior to the objective function assessment whereas (C.1) is handled in the objective function similar to the proposed improved PSO method. Therefore, the objective function for hybrid method is essentially same as (8). However, (C.2) and (C.4) are evaluated directly without incorporating Algorithm 1. In other words, first, the particle position vector is truncated using the dimension reduction process. Later, the set bits after performing (7) is used in the (C.2) and (C.4) evaluation. Only the first fit instances that satisfy all the constraints (C.2) to (C.4) are eligible for the objective function evaluation stage. A similar positioning approach has been utilized in [20] the during backfilling operation.

*C. Optimum Mixed Integer Linear Programming Method*

The solution to our optimization problem entails calculating an optimal resource allocation that satisfies (4), subject to the constraints (C.1) to (C.4). An LP (Linear Programming) solver, which supports binary decision variables, can therefore be used to solve this problem. In this paper, we use MATLAB toolboxes (YALMIP [21] for modelling and MOSEK [22] for solving) to model and compute an optimal solution and compare the performance of the proposed PSO solution. Although theoretically capable of achieving an optimal solution, this style of MILP solvers suffer from drawback such as the large memory requirement and exponentially rising computational complexity on the order of approximately $O(\mathrm{N}^2)$ - $O(\mathrm{N}^3)$.

### D. Best Fit (BF) and First Fit (FF) Methods

In this approach, groups of users are sequentially assigned to the best processing cloud, in terms of the QoS cost, in a partially greedy fashion. Thus, the later social groups will have fewer available resources since the earlier groups have already acquired better processing locations and network resources. Ultimately this leads to infeasible solutions for larger numbers of social groups. The First Fit method is largely similar to the Best Fit method, where the only difference is in the allocation of the first processing cloud for individual social groups. Here too the resources are allocated in a greedy fashion, sequentially.

## IV. Simulation and Results

The simulation environment, which is generated in MATLAB, consists of 70 Monte Carlo simulations of different network conditions. In each Monte Carlo instance, the performance for different social group configurations is compared while keeping the network conditions fixed. For simplicity, we assume that each social group's transmission stream is a HD H.264 video, which occupies 8Mbps bandwidth in the network and consumes 12 processing units in the cloud. The maximum allowable delay for each individual is restricted to 100 ms as per [5]. The network is assumed to be made up of 10 ISPs (access nodes) and 10 cloud computing resources (processing nodes). In order to obtain generalized conclusions, the network conditions are assumed vary as follows. The available bandwidth, link latency, jitter and packet loss is restricted to be within $\boldsymbol{B}_0[i,j] \in (20,60)$ Mbps, $\boldsymbol{D}[i,j] \in (20,100)$ ms, $\boldsymbol{J}[i,j] \in (5,80)$ ms, and $\boldsymbol{p}_0[1,j] \in (15,85)$, respectively. The link latency between each user and his/her ISP is a random variable in the interval (10,20) ms. On average there are 12 users per social group who are randomly distributed between multiple access nodes. As per [15] $\{\alpha_1, \alpha_2, \alpha_3\}$ are derived as $\{\alpha_1=0.02, \alpha_2=0.0011, \alpha_3=0.00024\}$.

Fig. 2 illustrates the "Average Group QoS Cost" (i.e., the cumulative QoS cost in (3) averaged across the number of trials and user groups) obtained using the different solution approaches. As the number of social groups increase, this 'per user group cost' also increases. This can be attributed to the fact that the competition for resources also increases among the social groups and hence results in higher costs. Moreover BF and FF methods seldom generate feasible solutions (as seen in Fig. 3) for larger numbers of social groups. Hence, when comparing average Group QoS cost, we plotted only the instances for which all the four methods yield feasible solutions. As seen in the figure, the MILP method demonstrates the least cost solutions through its complex calculations. However, our proposed PSO based approach has outperformed BF and FF methods significantly and achieves a near-optimal solution.

The serving probability for each method is illustrated in Fig. 3. Here, we evaluate the likelihood of a particular optimization method finding a feasible solution, if one exists. The probability is calculated accordingly (i.e., if social group $n$ generated feasible solutions for 35 out of 70 instances, the serving probability for group $n$ would be 0.5). As illustrated in the Fig. 3, when the number of social groups increases the demand for resources also increases and consequently results

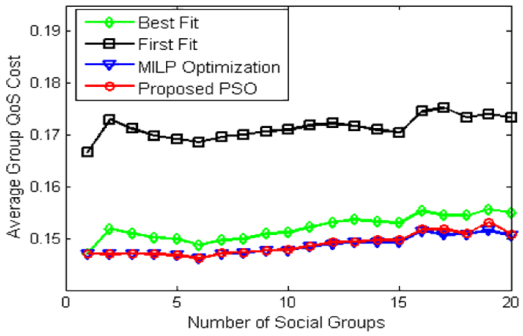


Fig. 2 Average Group QoS Cost for different social groups (colour online).

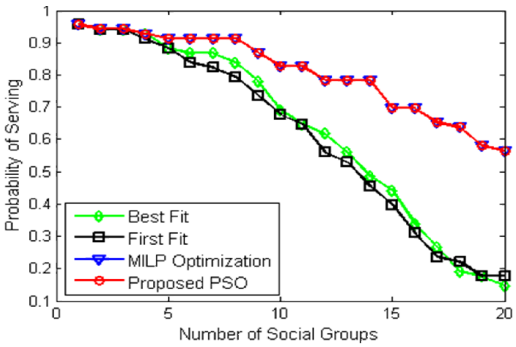


Fig. 3. Serving probability for different social groups (colour online).

improved PSO method to find a feasible solution is proportionately greater compared to the two greedy approaches and is very much similar to the MILP approach.

In addition, it can also be seen that the BF method has outperformed FF method in terms of QoS cost. This can be explained from the fact that once the FF method identified a feasible location for current social group allocation, it stops the search, whereas the BF method searches every cloud location for a better assignment, even though it already possesses a feasible solution. However this greedy behavior of the BF method sometimes lead to very few alternate locations for subsequent social groups and ultimately results in infeasible solutions for which the FF method can still find a feasible solution. This particular scenario can be observed in Fig. 3 for social groups 18 and 20. The main source of success of the proposed algorithm is its global knowledge of the network topology, the social group composition and the holistic approach used in the resource allocation phase. In contrast, the BF and FF only have local knowledge about the social group composition. In other words, they choose the best processing location for a particular social group solely based on the knowledge they possess about that social group, where ultimately, the initially assigned social groups possess an unfair advantage. Finally, it should be noted that although we randomly initialize the particles to enable a fairer comparison of the convergence behavior, the use of the output of the BF or FF methods as initialization vectors could enable a further improvement of the convergence time of the proposed method.

The effectiveness of the proposed improved PSO method compared to the original binary PSO approach is illustrated in the Fig. 4. Here, we find the number of iterations required to reach 5% of the optimal QoS cost. The proposed improved PSO, which incorporates both position vector dimension reduction and the bit allocation scheme in Algorithm 1, converges significantly faster compared to the two other PSO

methods. The iteration gap between the pure PFM and hybrid method illustrates the gain that can be achieved through the dimension reduction operation alone, whereas the gap between the hybrid and the proposed methods exhibits the gain due to Algorithm 1. The pure PFM has to perform large number of iterations within the entire solution space prior to reaching a feasible region; thus, results in high convergence times. In contrast, the hybrid method fluctuates within a much smaller solution space and therefore converges much faster. However, this method also spends a significant time without evaluating the objective function due to incapability of the original bit allocation mechanism to predict a solution which satisfies (C.3) and (C.4). The proposed improved PSO method addresses both (C.3) and (C.4) during the bit allocation process itself, and hence results in a lower convergence time in the order of $O(PN)$, where $P$ corresponds to the number of particles.

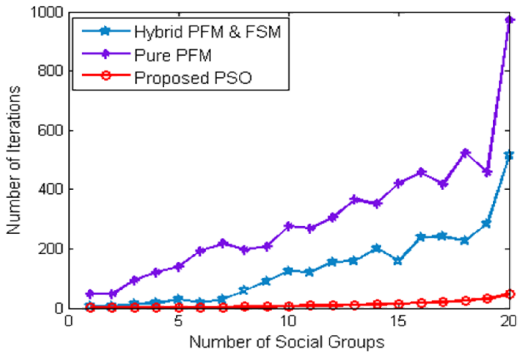


Fig. 4. Iteration count for different social groups (colour online).

## V. Conclusion

In this study we propose an improved PSO algorithm to optimally allocate dispersed cloud resources for an interactive multimedia application consisting of multiple user groups, such that the average Group QoS is maximized. We formulate the problem using matrix notations and describe an improved PSO based solution to the problem that reduces the complexity of the solution by limiting the probable solution space. Monte Carlo simulations consisting of different group compositions and different network conditions are used to evaluate the performance of our proposed method. For comparison purposes we incorporate an optimal solution generation method, which relies on a MILP solver, two greedy methods, and related PSO approaches. The simulation results demonstrate the superior performance of our method, achieving up to 40% improvement in the ability to allocate resources compared to the greedy approaches. The demonstrated reduction in the required convergence time illustrates the reduction in the computational complexity, and suggests that the proposed method is well suited for this type of problem in comparison with other related resource allocation mechanisms.

## References


[1] "List of social networking websites." [Online]. Available: en.wikipedia.org/wiki/List_of_social_networking_websites#Q.

[2] E. Mantzari, G. Lekakos, and A. Vrechopoulos, "Social tv: introducing virtual socialization in the tv experience," in *Proc. of the 1st Int. conf. on Designing Interactive User Experiences for TV and Video - uxtv '08,* Silicon Valley, California, 2008, pp. 81–84.

[3] "User Interaction Aware Content Generation and Distribution For Next Generation Social TV." [Online]. Available: http://www.action-tv.net/.

[4] T. Hoßfeld, R. Schatz, M. Varela, and C. Timmerer, "Challenges of QoE management for cloud applications," *IEEE Commun. Mag.*, vol. 50, no. 4, pp. 28–36, 2012.

[5] "Response Times: The 3 Important Limits." [Online]. Available: http://www.nngroup.com/articles/response-times-3-important-limits/.

[6] K. Hwang, G. Fox, and J. J. Dongarra, "Inter-cloud resource management," in *Distributed and Cloud Computing*, San Francisco: Morgan Kauffmann Publishers, 2011, p. 246.

[7] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *Proc. of the first edition of the MCC workshop on Mobile cloud computing,* Helsinki, 2012, pp. 13–16.

[8] A. Filali, A. S. Hafid, and M. Gendreau, "Adaptive resources provisioning for grid applications and services," in *IEEE Int. Conf. on Commun.,* Beijing, China, 2008, pp. 186–191.

[9] X. Nan, Y. He, and L. Guan, "Optimal resource allocation for multimedia application providers in multi-site cloud," in *IEEE Int. Symp. on Circuits and Syst.* , Beijing, China, 2013, pp. 449–452.

[10] K. Sembiring and A. Beyer, "Dynamic resource allocation for cloud-based media processing," in *Proc. of the 23rd ACM Workshop on Network and Operating Syst. Support for Audio and Video,* New York, 2013, pp. 49–54.

[11] P. Gao, A. Curtis, B. Wong, and S. Keshav, "It's not easy being green," in *Proc. of the ACM SIGCOMM 2012 Conf. on Applications, Technologies, Architectures, and Protocols for Computer Commun.,* Helsinki, Finland, 2012, pp. 211–222.

[12] R. Hans, U. Lampe, M. Pauly, and R. Steinmetz, "Cost-Efficient capacitation of cloud data centers for QoS-Aware multimedia service provision," in *4th Int. Conf. on Cloud Computing and Services Sci.,* Barcelona, Spain,, 2014, pp. 158–163.

[13] Í. Goiri, K. Le, J. Guitart, J. Torres, and R. Bianchini, "Intelligent placement of datacenters for internet services," in *Proc. Int. Conf. on Distributed Computing Syst.,* Minneapolis, 2011, pp. 131–142.

[14] G. Kulupana, D. S. Talagala, H. Kodikara Arachchi, and A. Fernando, "Optimized resource distribution for interactive TV applications," in *IEEE Int. Conf. on Consumer Electronics*, 2015 (accepted).

[15] H. Kim and S. Choi, "A study on a QoS/QoE correlation model for QoE evaluation on IPTV service," in *12th Int. Conf. on Advanced Commun. Technology,* Phoenix Park, Ireland, 2010, pp. 1377–1382.

[16] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *MHS'95. Proc. of the Sixth International Symposium on Micro Machine and Human Science,* Nagoya, Japan, 1995, pp. 39–43.

[17] J. Kennedy and R. C. Eberhart, "A discrete binary version of the particle swarm algorithm," in *1997 IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and* Simulation,Orlando*, FL*, 1997, vol. 5, pp. 4104–4108.

[18] G. Coath and S. Halgamuge, "A comparison of constraint-handling methods for the application of particle swarm optimization to constrained nonlinear optimization problems," in *The Congress Evolutionary Computation*, 2003, vol. 4, pp. 2419–2425 Vol.4.

[19] and C.-Y. K. Jinn-Moon Yang, Ying-Ping Chen, Jomg-Tzong Horng, "Applying family competition to evolution strategies for constrained optimization," in *Evolutionary Programming VI*, 1997, pp. 201 – 211.

[20] S. Wang, Z. Liu, and Z. Zheng, "Particle swarm optimization for energy-aware virtual machine placement optimization in virtualized data centers," in *Int. Conf. on Parallel and Distributed systems, S*eoul, South Korea, 2013, pp. 102–109.

[21] "YALMIP: A Toolbox for modeling and optimization in MATLAB." [Online]. Available: Available: http://users.isy.liu.se/johanl/yalmip.

[22] "The MOSEK optimization toolbox for MATLAB." [Online]. Available: http://www.mosek.com.