Optimal Resource Allocation in Next Generation Network Services using Engineering Optimization with Linear Constraint Particle Swarm

Hassan Yeganeh
Iran Telecommunication Research Center

Maryam Shakiba
Iran Telecommunication Research Center

Mehdi Samie
Iran Telecommunication Research Center

Abstract

In this paper, we consider the problem of pricing for optimal resource allocation service using Engineering Optimization with Particle Swarm algorithm that ensures efficient resource allocation that provides guaranteed quality of service while maximizing profit in multiservice networks. We formulate our generalized optimization algorithm based on the notion of a "profit center" with an arbitrary number of service classes, linear revenue and nonlinear cost functions and general performance constraints. To ensure the resource constraint is satisfied, we incorporate adaptive resource bounds to guide the search. Specifically, we develop a fast, low complexity algorithm for online dynamic resource allocation, and examine its properties. Finally, its performance is evaluated through an extensive numerical study.

Keywords: Nonlinear resource allocation problem; Adaptive resource bounds; particle swarm optimization; mathematical programming.

1. INTRODUCTION

The next generation Internet will provide advanced services, such as Quality of Service (QoS) guarantees, to users and their applications. As a result of, these enhancements, it is expected that service providers will face an increasing number of users as well as a wide variety of applications. Under these demanding conditions, network service providers must carefully provision and allocate network resources (e.g. bandwidth, buffer size, CPU capacity) for their customers. Provisioning is the acquisition of large end-to-end network services (connections) over a long time scale. In contrast, allocation is the distribution of these provisioned services (via pricing) to individual users over a smaller time scale [1]. Determining the optimal amounts to provision and allocate remains a difficult problem under realistic conditions. Service providers must balance user needs in the short-term while provisioning connections for the long term. Furthermore, this must be done in a scalable fashion to meet the growing demand for network services, while also being adaptable to future network technologies. This paper presents a modified particle swarm optimization (PSO) algorithm for engineering optimization problems with constraints. PSO is started with a group of feasible solutions and a feasibility function is used to check if the newly explored solutions satisfy all the constraints. All the particles keep only those feasible solutions in their memory. Several engineering design optimization problems were tested and the results show that PSO is an efficient and general approach to solve most nonlinear optimization problems with inequity constraints [2].

In this paper, we propose a service pricing model that ensures efficient allocation of resources in a dynamic manner in the aforementioned multiservice networks. The scheme requires close on-line monitoring of the incoming traffic.

We assume a Fractional Brownian Motion traffic model, because of its ability to adequately capture characteristics of real network traces, such as self-similarity and the presence of heavy tailed marginal distributions [3]. Optimal resource allocation is also studied in [4-6]. Specifically, Peng *et al.* propose a measurement-based resource allocation scheme based on a linear pricing model and average queue delay guarantees. This scheme has the disadvantage of not being scalable to large number of service classes. Moreover, average queue delay is not always an appropriate QoS constraint. In [5], they perform maximization over a utility function provided from the network users and resources are shared based on the solution of that optimization problem. In [6], the authors study the problem of resource allocation with dynamic pricing in which the network administrator controls the price of the resources that users demand; based on the demand the prices are dynamically changed over different time periods so as to maximize the revenue of the administrator. Finally, measurement-based resource allocation has also been studied in different contexts in [7-9].

Manuscript received November 5, 2008
Manuscript revised November 20, 2008

2. MODELING FRAMEWORK

The employed modeling framework was introduced in [4], [10] and is depicted in Figure 1. In its present form it represents a single network element, which can be either a traditional network component, such as switch or a router, or a modern network "service center", like IBM's Data Power Service Oriented appliances [11] or CISCO's Application Oriented Network (AON) message routing system [12]. It is assumed that the network element serves two categories of traffic classes; deterministic delay-bound classes and flexible delay-bound ones. Due to the fact that deterministic delay-bound classes have strict requirements, their service level agreement (SLA) can be satisfied only by traffic shaping and admission control schemes [13, 14]. Thus, an amount of resources is dedicated to them and these classes are excluded from subsequent analysis. Examples of these inelastic classes of service include teleconferencing, remote seminars, real-time distributed computation/simulation and high-precision medical imaging.

Therefore, the proposed system is responsible for optimally allocating the excess resources to the remaining flexible delay bound classes. These classes enter the Measurement Based Optimal Resource Allocation (MBORA) system proposed in [10] and shown in Figure 2. The MBORA system consists of a measurement module, an optimization module and a resource orchestrator module. The statistics of the arrival traffic are measured by the measurement module. It is assumed that the traffic can be accurately approximated by a Fractional Brownian motion model, which can account for the burst ness and long-range dependence observed in real traffic traces. Such a model can be fully described by the following parameters: the Hurst parameter H, the *mean* arrival rate $\overline{\alpha}$ and the variance σ. An algorithm for on-line measurement of these parameters is discussed in [15].

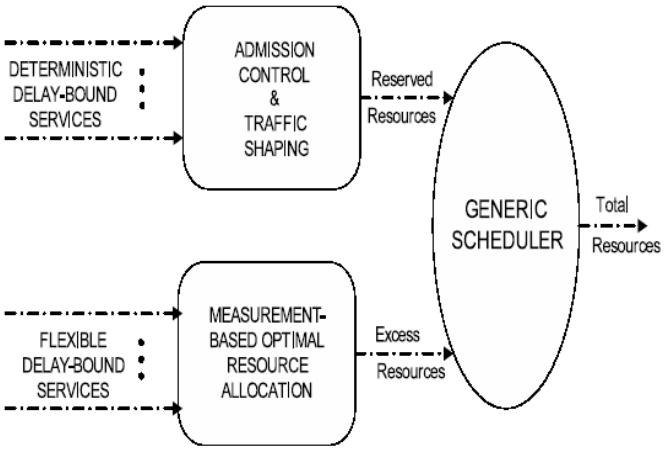

Fig. 1: Depiction of the proposed framework

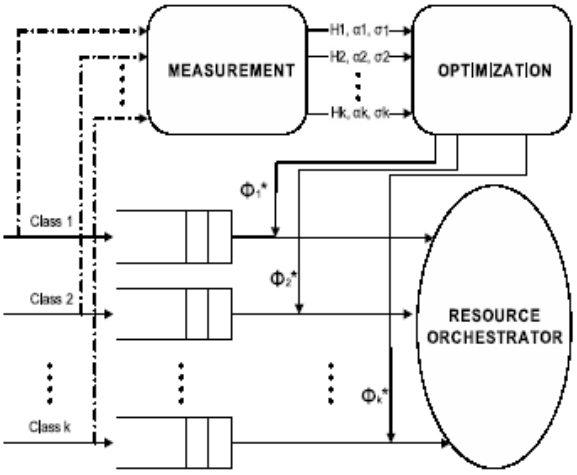

Fig. 2: The MBORA system

The optimization module receives the traffic characteristics of each class and calculates the optimal allocation of resources by solving the optimization problem discussed in Section III. It should be noted that the optimization problem is solved only when there is a significant change in traffic characteristics. The optimal solution is fed to the resource orchestrator which dynamically updates the allocation of resources for each traffic class and forwards the packets (or, more generally, the messages, for example XML) toward their destination.

3. PRICING MODEL AND OPTIMIZATION PROBLEM FORMULATION

We start by introducing the pricing model, whose solution yields the optimal allocation of resources to the network service node we described in the previous section.

A. Non-Linear Pricing Model

Suppose that the node can provide *K* different types of services. The proportions of these services to be allocated are denoted by $\phi = (\phi_1, \ldots, \phi_K)$.

According to [16], the profit of a provider is the difference between the revenue $r(\phi)$ that is obtained for providing these services and the cost $c(\phi)$ that incurs from producing them. The aim of this provider is to maximize the profit function subject to the feasibility constraints:

$$\pi = \max\{r(\phi) - c(\phi)\} = \max\sum_{k=1}^{K}(r_k(\phi_k) - c_k(\phi_k)) \quad (1)$$

Subject to the feasibility constraints: $\phi_k \geq 0, \quad k = 1, \ldots, K, \sum_k \phi_k \leq 1$. The revenue is given by a linear function, while the cost by a nonlinear one. Specifically, $r_i(\phi_i) = p_i.\phi_i$ while the cost function has the form $c_i(\phi_i) = b_i.D_i(\phi_i).exp[\beta(D(\phi_i) - d_i)]$.

The coefficient p_i corresponds to the price that the provider charges for service *i* and the parameter b_i is the amount that the provider has to reimburse the users whenever the SLAs are not met. A higher priority class *u* requires better service than a lower one *v* and thus it is charged more (i.e., $p_u > p_v$ and $b_u > b_v$). The parameter β controls the steepness of the cost function, while $D(\phi_i)$ denotes the value of the performance metric experienced by users of service *i* and d_i the target level under the SLA. Hence, if $D(\phi_i) > d_i$ the users are not receiving adequate resources from the provider, which would incur a cost, until the situation is rectified. This function is monotone in $D_i(\phi_i)$ and is shown in Figure 3. The steep increase in the cost observed beyond the desired by the users SLA value of d_i would force the provider to adjust the allocation of resources (if possible), in order to satisfy the QoS requirements and maximize profit.

Probabilistic Delay Constraints: We employ stochastic delay bounds as the metric for QoS considerations. Specifically, we adopt the approach used in [17], [18], where traffic is treated as Long Range Dependent (LRD) and is characterized by the Hurst parameter H , the mean $\overline{\alpha}$ and the variance σ .

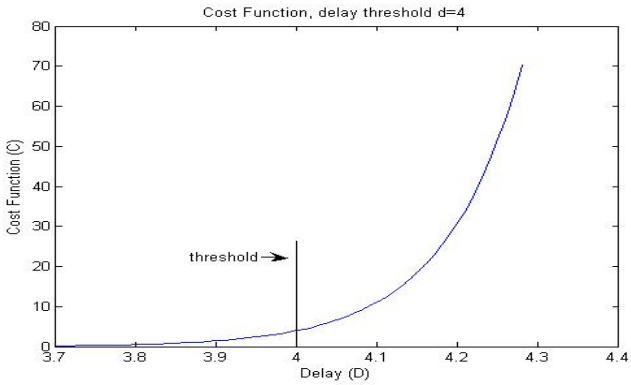

Fig. 3: Cost function

Notice that even a small increase of 2.5% above the delay threshold yields an increase above 100% in the cost function. In this case parameter β = 10.

It is shown that the queue length at any given time *t* is bounded by a value $q_{\max}$ with probability $\varepsilon > 0$ related to the desired QoS. It is shown that for a specific class the following holds:

$$\Pr(Q(t) > q_{\max}) \approx \varepsilon \tag{2}$$

and

$$q_{\max} = (C - \overline{\alpha})^{H/(H-1)} (K\sigma)^{1/(1-H)} H^{H/(1-H)} (1-H) \tag{3}$$

Where *C* can be interpreted as the resources (e.g., bandwidth) dedicated to this particular class, ε is the required QoS and $k = \sqrt{-2\ln\varepsilon}$.

Thus, since the queue length and expected delay are related, we have the following probabilistic delay bound:

$$\Pr(D(t) > D_{\max}) \approx \varepsilon \tag{4}$$

and

$$Dmax = \frac{(C-\overline{\alpha})^{H/(H-1)} (k\sigma)^{1/(1-H)} H^{H/(1-H)} (1-H)}{C} \tag{5}$$

This delay bound is used in the cost function.

B. Particle Swarm Optimization

Particle swarm optimization (PSO) was originally designed and introduced by Eberhart and Kennedy [19-21] in 1995. The PSO is a population based search algorithm based on the simulation of the social behavior of birds, bees or a school of fishes. This algorithm originally intends to graphically simulate the graceful and unpredictable choreography of a bird folk. A vector in multidimensional search space represents each individual within the swarm. This vector has also one assigned vector, which determines the next movement of the particle and is called the velocity vector. The PSO algorithm also determines how to update the velocity of a particle. Each particle updates its velocity based on current velocity and the best position it has explored so far; and based on the global best position explored by swarm [22-24]. The PSO process then is iterated a fixed number of times or until a minimum error based on desired performance index is achieved. It has been shown that this simple model can deal with difficult optimization problems efficiently.

A detailed description of PSO algorithm is presented in [19-21]. Here we will give a short description of the PSO algorithm proposed by Kennedy and Eberhart. Assume that our search space is d-dimensional, and *i*-th particle of the swarm can be represented by a d-dimensional position vector $X_i = (x^1{}_i, x^2{}_i, \ldots, x^d{}_i)$. The velocity of the particle is denoted by $V_i = (v^1{}_i, v^2{}_i, \ldots, v^d{}_i)$. Also consider best visited position for the particle is $P_{ibest} = (p^1{}_i, p^2{}_i, \ldots, p^d{}_i)$ and also the best position explored so far is $P_{gbest} = (p^1{}_g, p^2{}_g, \ldots, p^d{}_g)$. So the position of the particle and its velocity is being updated using following equations:

$$v_i(t+1) = w.v_i(t) + c_1\varphi_1(p_i(t) - x_i(t)) + c_2\varphi_2(p_g(t) - x_i(t)) \tag{6}$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \tag{7}$$

Where c_1 and c_2 are positive constants, and φ_1 and φ_2 are two uniformly distributed number between 0 and 1. In this equation, W is the inertia weight which shows the effect of previous velocity vector on the new vector. The inertia weight W plays the role of balancing the global and local searches and its value may vary during the optimization process. A large inertia weight encourages a global search while a small value pursues a local search. In [25] authors have proposed an Adaptive Weighted PSO (AWPSO) algorithm in which the velocity formula of PSO is modified as follows:

$$v_i(t+1) = w.v_i(t) + \alpha.[r_1(p_i - x_i(t)) + r_2(p_g - x_i(t))] \quad (8)$$

The second term in Equation (8) can be viewed as an acceleration term, which depends on the distances between the current position x_i, the personal best p_i and the global best p_g. The acceleration factor α is defined as follows:

$$\alpha = \alpha_0 + t/N_t \quad (9)$$

Where N_t denotes the number of iterations, t represents the current generation, and the suggested range for α is [0.5, 1]. As can be seen from Equation (9), the acceleration term will increase as the number of iterations increases, which will enhance the global search ability at the end of run and help the algorithm to jump out of the local optimum, especially in the case of multi-modal problems. Furthermore, instead of using a linearly decreasing inertia weight, they used a random number, which was proved by Zhang et al. [26] to improve the performance of the PSO in some benchmark functions as follows:

$$w = w_0 + r(1 - w_0) \quad (10)$$

Where $w_0 \in [0,1]$ is a positive constant, and r is a random number uniformly distributed in [0, 1]. The suggested range for w_0 is [0, 0.5], which makes the weight w randomly varying between 0 and 1. An upper bound is placed on the velocity in all dimensions. This limitation prevents the particle from moving too rapidly from one region in search space to another. This value is usually initialized as a function of the range of the problem. For example if the range of all x_{ij} is [-1, 1] then $V_{\max}$ is proportional to 1.

p_{ibest} For each particle is updated in each iteration when a better position for the particle or for the whole swarm is obtained. The feature that drives PSO is social interaction. Individuals (particles) within the swarm learn from each other, and based on the knowledge obtained then move to become similar to their "better" previously obtained position and also to their "better" neighbors. Individual within a neighborhood communicate with one other. Based on the communication of a particle within the swarm different neighborhood topologies are defined. One of these topologies which is considered here, is the star topology. In this topology each particle can communicate with every other individual, forming a fully connected social network. In this case each particle is attracted toward the best particle (best problem solution) found by any member of the entire swarm. Each particle therefore imitates the overall best particle. So the p_{gbest} is updated when a new best position within the whole swarm is found.

In order to evaluate the performance of individual particles, an appropriate evaluation function (or profit function) should be defined.

Now putting the revenue and cost components together, the provider's profit problem becomes:

$$max\{\sum_{i=1}^{k} p_i \phi_i C - \sum_{i=1}^{k} b_i D_i(\phi_i) \times exp[\beta(D_i(\phi_i) - d_i)]\} \quad (11)$$

Subject to the feasibility constraints previously described. In the above expression, $D_i(\phi_i)$ is given from Eq. 5 by substituting parameter C with $\phi_i.C$, since we are dealing with a network element with multiple input classes each of which is allocated a portion ϕ_i of the total C resources. Note also that $D_i(\phi_i)$ is actually $D_{\max,i}(\phi_i)$.

In the Eq.11 ϕ_i is generated as uniformly distributed random number within the interval [0,1]. In this way, we can obtain a uniformly distributed random ϕ_i combination, which is generated at every iteration. In the light of above considerations, the proposed algorithm can be summarized as follows:

1. Initialization**:** Set population number N and iteration number N_t. Initialize the position x_i and velocity v_i of the particles (ϕ_i) with random numbers within the pre-defined decision variable range. $V_{\max}$ is set to the upper bound of the decision variables. Set personal best position $p_i = x_i$, iteration counter *t*=0.
2. Evaluation**:** *t*=*t*+1. Evaluate each particle in the current population using equation (11). If $\pi_i(t) > \pi_i(t-1)$, then $p_i = x_i$. Find $\pi_{\max} = \max\{\pi_i\}$, and corresponding position $X_{\max}$. Select global best $p_g = X_{\max}$.
3. Calculate the new velocity and position**:** Calculate the new velocity NV_i and new position NX_i based on the current $x_i (i = 1,2,\ldots N)$, using equations (7) and (8), and the profit function values for all the new particles.

4. If all $v_i < 0.1V_{\max}$, execute the following steps, otherwise go to 5:
a) Randomly select 20% particles in current population and randomly change their positions by 10% of the $V_{\max}$.
b) Repeat a) *K* times (*K*=1~10), make sure the number of x dose not exceed *N*.
5. If $t < N_t$, go to 2.

Remark*:* For the under-provisioned case, the problem is not particularly interesting, since the QoS constraints would be surely violated. Hence, the service provider would allocate resources according to the average traffic intensities; further, it is easy to see that the operation would not be profitable. Hence, this regime is not studied in this paper.

4. RESULTS AND DISCUSSION

In this section, we evaluate our pricing model in the over provisioned case with a numerical case study. It is assumed that there are two types of service classes and the profit function becomes:

$$\pi(\phi_1,\phi_2) = p_1\phi_1 C + p_2\phi_2 C - b_1 D_1(\phi_1)\times$$
$$exp(\beta(D_1(\phi_1)-d_1)) - b_2 D_2(\phi_2)\times$$
$$exp(\beta(D_2(\phi_2)-d_2)) \quad (12)$$

Where

$$D_i(\phi_i) = \frac{(\phi_i - \bar{\alpha}_i)^{\frac{H_i}{H_i-1}} (k\sigma_i)^{\frac{1}{1-H_i}} H_i^{\frac{H_i}{1-H_i}} (1-H_i)}{\phi_i}, \quad (13)$$

$i = 1,2$

Hence, we have to solve the optimization problem:

$\max \pi(\phi_1,\phi_2)$ subject to $\phi_1 + \phi_2 = 1$

The parameters of the profit function used in the study are shown (Table 1). Optimal solution is shown when the arrival rate and the price coefficients are varied (Tables 2,3). It can be seen that with equal arrival rates and all the other parameters the same, the optimal solution allocates the resource equally amongst the two classes, as expected (Table 2). On the other hand, the class with the higher arrival rate is allocated a larger portion of the resources, especially if the system is not too stressed (see rows 2 and 3 in the Table 2). In that situation the profit does not also fluctuate much. Finally, when the system becomes stressed (last row in the Table 2) the class with higher arrival rate gets a higher proportion, but the overall profit for the provider decreases substantially, since violations of the SLA occur more often and therefore a large cost is incurred. In Table 3, the price coefficient varies, while all other parameters are held fixed (Table 1). Again, with equal prices we obtain equal allocations, while the allocation of resources exhibits a strong sensitivity to the price ratio P_1 / P_2 .

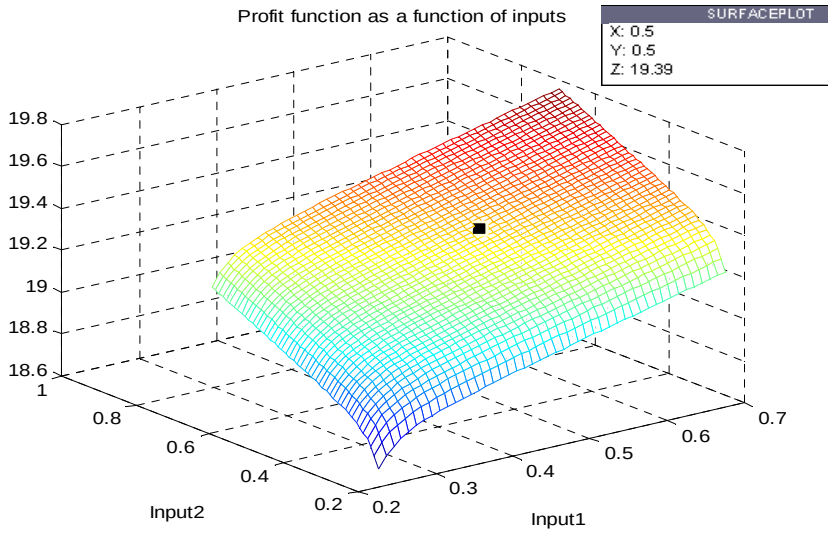

Fig. 4: Our Profit Function as a function of inputs (ϕ_1, ϕ_2)

Table1: Parameters for two different classes

	Class1	Class2
p (cents/Mbps)	1	1
b (cents/ms)	0.1	0.1
d (in delay units)	0.01	0.01
$QoS(=\varepsilon)$	10^{-6}	10^{-6}
$\overline{\alpha}$ (normalized to C)	0.2	0.2
σ (normalized to C)	0.01	0.01
H	0.70	0. 70

Table 2: Changing the arrival rates $\overline{\alpha}_i$

$(\overline{\alpha}_1, \overline{\alpha}_2)$	(ϕ_1^*, ϕ_2^*)	$\pi(\phi_1^*, \phi_2^*)$
(0.2,0.2)	(0.5,0.5)	19.3929
(0.3,0.2)	(0.5227,0.4773)	18.5157
(0.4,0.2)	(0.5609,0.4391)	17.5548
(0.4,0.5)	(0.4532,0.5468)	17.2027

Table3: Changing the pricing factor p_i

(p_1, p_2)	(ϕ_1^*, ϕ_2^*)	$\pi(\phi_1^*, \phi_2^*)$
(1,1)	(0.5,0.5)	19.3929
(2,1)	(0.6856,0.3144)	31.3450
(4,1)	(0.7566,0.2434)	39.6556
(4,4)	(0.5,0.5)	69.8043
(1,2)	(0.3144,0.6856)	31.3450
(1.5,6)	(0.1675,0.8325)	73.2586

5. CONCLUSION

In this paper, we have studied a pricing scheme for next generation multiservice networks. An optimization problem based on an intelligent searchable pricing model was formulated, whose solution yields the optimal resource allocation in a network/service node, given the QoS requirements of each service class that the network element serves. Our non-linear pricing model responds well to changes of the characteristics in the input traffic, pricing parameters and QoS requirements. Further, the resulting particle swarm optimization problem can easily and efficiently be solved using standard iterative methods and hence the proposed modeling framework approach is scalable to any number of service classes.

6. ACKNOWLEDGEMENT

The authors would like to express their gratitude to Iran Telecommunication Research Center for their support and consent.

7. REFERENCES

[1] G. Huston, ISP Survival Guide: Strategies for Running a Competitive ISP, John Wiley & Sons, 1999.

[2] Xiaohui Hu, Russell C. Eberhart, and Yuhui Shi , Engineering Optimization with Particle Swarm,2003, Indiana,USA.

[3] Leland W., Taqqu M., Willinger W., Wilson D., *On the Self-Similar Nature of Ethernet Traffic (Extended Version)*. IEEE/ACM Trans. Networking, pp 115, Feb, 1994.

[4] Xu P., Devetsikiotis M., Michailidis G., *Profit-oriented Resource Allocation Using Online Scheduling in Flexible Heterogeneous Networks*. Telecommunication Systems, pp 289-303, 2006.

[5] Kalyanasundaram S., Chong E. K., Shroff N. B. *Optimal Resource Allocation in Multiclass Networks with User Specified Utility Functions*. The International Journal of Computer and Telecommunications Networking, pp 613-630, April, 2002.

[6] Savagaonkar U., Chong E. K., Givan R. L., *Online pricing for bandwidth provisioning in multi-class networks*. Comput. Networks, pp 835–853, 2004.

[7] Chandra A., Gong W., Shenoy P. *Dynamic Resource Allocation for Shared Data Centers Using Online Measurements*. In Proceedings of ACM/IEEE Intl Workshop on Quality of Service, pp 381-400, 2003.

[8] Knightly E., Shroff N., *Admission Control for Statistical QoS: Theory and Practice*. IEEE Network 13(2), pp 20-29, 1999.

[9] Qiu J., Knightly E., *Measurement-based Admission Control with Aggregate Traffic Envelopes*. IEEE/ACM Transactions on Networking, pp 56-70, 1997.

[10] Xu P., *QoS Provisioning and Pricing in Multiservice Networks: Optimal and Adaptive Control over Measurement-based Scheduling*. PhD Dissertation, North Carolina State University, 2005.
[11] Oltsik J., SOA Appliance Case Studies - Web Services Meet the Network.
[12] Cisco Systems, *Cisco AON: A Network Embedded Intelligent Message Routing System,* Product Bulletin No. 2886, 2005.
[13] Georgiadis L., Guerin R., Peris V., Sivarajan K. N., Efficient Network QoS Provisioning based on Per Node Traffic Shaping. IEEE/ACM Transactions on Networking 4(4), pp 482-501, 1996.
[14] Breslau L., Jamin S., Shenker S., Comments on the Performance of Measurement-based Admission Control Algorithms. Proc. of IEEE INFOCOM, pp 1233-1242, 2000.
[15] Sun Q., Pleich R., Sauerwein R., *On-Line Measurement and Analysis of Fractional Brownian Traffic*. Proceedings of the IEEE Conference on High Performance Switching and Routing, pp 395-400, 2000.
[16] Courcoubetis C., Weber R., *Pricing Communication Networks*. John Wiley & Sons, 2003.
[17] Fonseca N., Mayor G. S., Neto C. A. V., *On the Equivalent Bandwidth of Self-Similar Sources*. ACM Transactions on Modeling and Computer Simulation, Vol. 10, No. 2, pp 104-124, April, 2000.
[18] Norros I., *The Management of Large Flows of Connectionless Traffic on the Basis of Self-similar Modeling*. In Proceedings of IEEE International Conference on Communications, pp 451-455.
[19] R. Eberhart, and J. Kennedy, "A New Optimizer Using Particles Swarm Theory", Proc. Sixth International Symposium on Micro Machine and Human Science, NJ, pp. 39-43, 1995.
[20] J. Kennedy, and R. Eberhart, "Particle Swarm Optimization", IEEE International Conference on Neural Networks , pp. 1942-1948, 1995.
[21] J. Kennedy and R. Eberhart. Swarm Intelligence. Morgan Kaufmann Publishers, Inc., San Francisco, CA, 2001.
[22] A. P. Engelbrecht. "Fundamentals of Computational Swarm Intelligence". Wiley, 2005.
[23] J. Sadri, and Ching Y. Suen, "A Genetic Binary Particle Swarm Optimization Model", IEEE Congress on Evolutionary Computation, Vancouver, BC, Canada, 2006.
[24] A. P. Engelbrecht, "computational Intelligence", John Wiley and Sons, 2002.
[25] M. Mahfouf, Min-You Chen, and D. A. Linkens, "Adaptive Weighted Particle Swarm Optimisation for Multiobjective Optimal Design of Alloy Steels", Lecture Notes in Computer Science vol 3242, pp. 762–771, 2004.
[26] Zhang and S. Hu, A New Approach to Improve Particle Swarm Optimization, Lecture Notes in Computer Science, vol. 2723, p. 134-139, 2003.

Hassan Yeganeh has BS degree in electronic engineering from Khajeh Nasir Toosi university of Tehran and Master degree In telecommunication-system from Amirkabir university of Tehran. Since 1997, he has been working for ITRC as a member of scientific board. His important experiences are NGN, IMS, communication systems and related protocols.

Maryam Shakiba has BS degree in electronic engineering from Shahid Chamran University of Ahvaz and Master degree in electrical engineering from Khajeh Nasir Toosi university of technology. Since 2008, she has been a faculty member in Jundi-Shapur university of technology. Her important experiences are Artificial Neural Networks, Global Optimization Algorithms and NGN.